

A Survey on the Implementation of Reinforcement Learning on Shared Taxi System

Madhurya T*, Karthik V

Department of Computer Science & Engineering, VVCE, Mysuru

DOI: <https://doi.org/10.21467/proceedings.1.41>

* Corresponding author email: madhuryat105@gmail.com

ABSTRACT

Reinforcement learning (RL) is an area of machine learning concerned with how software agents used to take actions in a nature so as to maximize some faith of increasing benefit. In machine learning, the nature is typically formulated as a Markov decision process (MDP), as many of the reinforcement learning algorithms for this context utilize active programming techniques. Taxis plays a important role in modern public transportation networks, especially in countries where public transportation services are still underdeveloped. Taxi drivers currently rely on a simple first come-first serve approach, with a high coefficient of luck controlling their profit Problems arising from the current system are low taxi utilization, long passenger waiting times, road safety issues, and traffic congestion. In this paper, we present a system that improves the shared taxi service by benefiting both the taxi drivers and the passengers. This study proposed a various reinforcement learning method for a shared-taxi system. The present system proposes a solution in the form of a good, efficient and economical transportation service.

Keywords— Reinforcement learning, Level of Service (LOS), Shared taxi system, Look ahead insertion policy, R-Sharing, Scheduling algorithm.

1 Introduction

Taxis are the heart of public transportation networks in countries where bus, train, and subway services are either unavailable or unreliable. In developing countries, sharing taxis by street hailing such as rain shower and many more is the most common method of public transportation. Although the service is relatively cheap, it is inefficient and has several major problems. Taxi drivers mainly suffer from congestion, long distance wasted mileage and a high risk of accidents on the high-way roads. Passengers often wait for extended amount of time before finding a taxi driver that is willing to take their request [2]. A shared taxi system is a new service for urban traffic. In such systems, passengers book their request for a transport service using a smart phone app and the system dynamically routes the vehicles to respond incoming request from the passengers. The system allows several people to share the capacity



© 2018 Copyright held by the author(s). Published by AIJR Publisher in Proceedings of the 3rd National Conference on Image Processing, Computing, Communication, Networking and Data Analytics (NCICCND 2018), April 28, 2018. This is an open access article under [Creative Commons Attribution-NonCommercial 4.0 International](https://creativecommons.org/licenses/by-nc/4.0/) (CC BY-NC 4.0) license, which permits any non-commercial use, distribution, adaptation, and reproduction in any medium, as long as the original work is properly cited. ISBN: 978-81-936820-0-5

of the same vehicle if their destination routes are on the same route. Therefore, the system can use less vehicles to satisfy the requests and dynamically respond to the travel demand [1]. The proposed design in this paper shows the daily operation of each individual taxi driver and makes the profit, distribution of profit among drivers is fair and equitable without imposing any additional cost on passengers. Previous studies have proven that the share-taxi system is a successful way to reduce urban traffic jam. With the development of smart phone commercial applications of shared-taxi systems such as UberPool, Ola, LiftLine, Split and DiDiXiaoba came into existence around the world. These systems benefit both urban traffic and travelers around the world [1].

Though today's taxi system is far from being efficient: a most common problem is that people often have difficulties in getting a taxi during rush hours and weekends where there will be more demand for taxis, though the occupied taxis might still have available seats which are taken by one or two passengers per taxi. This unsuccessful becomes much worse in countries and areas that are still in the process of urbanization, where the city population increases at a faster rate than the number of taxis. This problem has attracted some attention in recent years, resulting in a proposed solution called ride-sharing, in which people share a taxi with others who have similar destination and schedules [5].

Most existing powerful taxi sharing services usually respond to a passenger's sharing request by sending a taxi that is not fully occupied to pick up the passenger within a certain period of time, with the intention of minimizing the overall travel distance or travel time. However, they have two weaknesses.

- Passengers have different preferences on taxi-sharing. For example, some may prefer saving more money by sharing a taxi with more companions or passengers, while some may prefer less travel time by sharing a taxi with fewer companions. Existing systems fail to take personal preferences of passengers into consideration while planning the schedule for sharing a taxi.
- They also fail to provide the useful sharing schedule as they usually send a taxi that is not fully occupied to the same route sharing request. Given a new request, they can only provide a sub-optimal sharing schedule bounded by the current locations and traveling directions of taxis and the destinations of passengers already on board. Nevertheless, a sharing schedule could be changed or even degraded when a taxi needs to pick up a new passenger in the same route. For example, the so-called shortest travel time or route could be changed during a ride. Moreover, the already riding passengers may have extended and unpredictable travel distance and time [6].

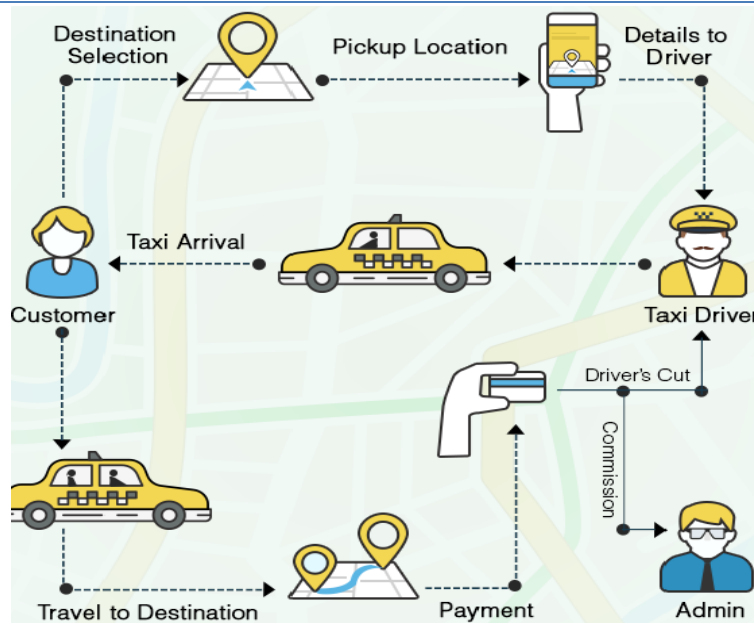


Figure 1: Flow for Online Cab Booking

When a new passenger request arrives, the taxi-sharing server will try to send a taxi to satisfy the new request while serving its existing requests. The above figure 1 depicts the flow of booking a cab through online where the customer select the destination and pickup location, when the request is accepted by the server customer gets the details of the taxi driver, later there will be arrival of the taxi. After traveling to the destination point customer can make the payment either through a mode of cash or card to the driver where a certain amount of payment will be transferred to the admin as a commission and rest of the amount will be paid to driver.

The shared-taxi problem is a multi-vehicle strong dial-a-ride problem (DARP) with unknown customers and without time-windows. In the DARP, the decision for inserting a new request could have an effect on the waiting time and the long distance of future customers. The global advantage of shared taxi system could be damaged if we insert requests inspite of these future unknown customers. We need to make use of a look-ahead policy to insert requests in order to avoid wrong decisions [1].

Previous studies have identified that a vehicle routing problem with unknown demand can be identified as a Markov decision process (MDP). Although the future demand for effective vehicle routing problems is not known, we can determine the daily request arrivals from previous known data. Several studies confirm that this kind of determining the request is critical information for the implementation of the look-ahead policy. Some studies also pay attention on the balance between travel demand and transport capacity of the whole system and implement the look-ahead routing policy through a general cost function. The cost

function is used as a local objective function for carrying out insertion of the customer's request [1].

In this study, we aim to propose a various reinforcement learning methods such as look ahead insertion policy, R-sharing, optimization-based approach and a scalable approach for the shared-taxi system.

2 LOOK AHEAD INSERTION POLICY

This study aims to improve the global level of service (LOS) of the shared-taxi system considering the future unknown customers. The LOS is a measure that reflects the server quality of a transport system. LOS is used to analyze highways by categorizing traffic flow and assigning quality levels of traffic based on performance measure like speed, density etc. A high LOS can enhance the customer loyalty and maintain the sustainability of a shared-taxi system; therefore, this measure is usually considered as an important factor of the valuation for the company that operates the shared-taxi system [1]. The highlights of the proposed method are described as follows:

1. Time-dependent arrival rate
2. System state representation
3. Operation in real-time
4. Off-line learning

When a new customer is added into the route of a taxi, the waiting time and tour distance of other customers served by this taxi can also be changed. In this study, they have defined the waiting time for a customer as the time between the moment a trip request is submitted and the moment the customer is picked up by a taxi [1].

If the request of a customer is accepted by the shared-taxi system, the customer needs to wait several minutes to be picked up. If the customer has to share the taxi with other customers during the trip, he/she may be toured around the shortest route. People always wish to experience the least waiting time and detour distance [1]. For modeling purposes, they have converted the waiting time and detour distance into waiting and detour cost. Therefore, the LOS for people who require a shared-taxi service can be described as:

$$LOS = WillingnessToPay - WaitingCost - DetourCost \quad (1).$$

They define LOS to be zero if the customer's request is rejected. Their goal is to find an insertion policy that can maximize the global LOS of the total M requests for a day as follows:

$$\sum WillingnessToPay_m - WaitingCost_m - DetourCost_m \quad (2)$$

However, it is difficult to use directly Eq. (2) as an objective function because of the unknown nature of the daily requests. Since they treat requests with a First-In-First-Service (FIFS) principle, they cannot reject a new request if there is a possible route for it.

Similarly, they insert a new request into the shared-taxi system through minimizing the following objective function:

$$\min \Delta WaitingCost + \Delta DetourCost \quad (3)$$

Where x is a possible insertion position for the new request. The insertion position contains the information about the carrier of the request as well as the pick-up/drop-off order. They use $\Delta\text{WaitingCost}$ and $\Delta\text{DetourCost}$ to denote the immediate increment of waiting and detour cost respectively, after inserting the new request. Inserting a new request to a taxi can change the waiting and detour cost of the active customers those who are currently travelling in the taxi. They account the increment cost for all affected customers to figure out $\Delta\text{WaitingCost}$ and $\Delta\text{DetourCost}$ respectively [1].

They have observed that the objective function of Eq. (3) ignores the system overhead issue completely. This leads the system to pursue an immediate benefit at the cost of the efficiency of transport. As a result, the total rejection rate of requests can be high and the global LOS will be damaged. The system overhead can be measured by the increment of the operation distance of the taxis. The increment of the operation distance is obtained as follows:

$$\Delta\text{OperatingDistance} = d1 - d0 \quad (4)$$

Where length of the route of active customers is $d0$ and after inserting a new request into the route of taxi, length of the route becomes $d1$. In order to achieve a better LOS, they have reformed the objective function by combining the incremental cost of waiting and detour with the incremental operating distance. In this new objective function, they have also converted $\Delta\text{OperatingDistance}$ into $\Delta\text{OperatingCost}$ through fuel price and obtained the following minimization problem:

$$\text{Min. } \mu \cdot (\Delta\text{WaitingCost} + \Delta\text{DetourCost}) + (1 - \mu) \cdot \Delta\text{OperatingCost} \quad (5)$$

Parameter μ is referred to as a look-ahead parameter and Eq. (5) is referred to as an insertion cost function. They have considered Eq. (3) as the benchmark method in their study [1]. This study does not discuss re-optimization issues, they have just focused on the insertion of new arriving requests and do not re-route active customers in the system. This study also inserts a request based on the balance between the operating cost and the penalty for waiting and detour. They have introduced an adaptive parameter to keep this balance. It determines that the proposed method does not impose any restrictions on the re-optimization process; therefore, one can straight forwardly use the proposed method and the re-optimization process.

3 R-SHARING

R-Sharing provides a taxi-sharing service from the perspective of passengers. Specifically given a taxi-sharing request, R-Sharing searches the most favorable set of nearby companions with compatible personal preferences and similar destination directions, recommends a meeting point which is easy to get a vacant taxi and requires short walking distance for them to meet, and plans the shortest sharing route [6]. R-Sharing significantly differs from existing dynamic taxi sharing systems by providing new sharing services with two major unique features.

1. R-Sharing enables passengers to set their preferences on four essential sharing experience, namely,
 - walking distance
 - waiting time for a taxi
 - travel fare and
 - extra travel time, which will be considered in searching companions and planning sharing schedules.
2. R-Sharing groups companions with compatible personal preferences to provide the optimal sharing schedule consisting of the shortest route before getting on a taxi.

They have proposed a novel dynamic taxi sharing system called R-Sharing with a meeting-sharing strategy. R-Sharing searches the favorable nearby companions for a sharing request, recommends the favorable meeting point, and plans their shortest sharing route based on their personal preferences, such that their probability of accepting the sharing schedule is maximized. A probabilistic model was proposed to estimate the sharing probability by considering potential passengers' personal preferences on sharing experience.

A companion candidate searching algorithm is proposed for finding candidates who are likely to satisfy a taxi-sharing request.

An exact taxi-sharing scheduling algorithm is developed to generate the favorable sharing schedule for a sharing request, such that the sharing probability is maximized. To improve the efficiency, a heuristic sharing scheduling algorithm is also proposed for generating the optimal sharing scheduling [6]. Dynamic ride sharing systems, such as Uber, Lyft and Didi Chuxing, have become popular transport alternatives in cities across the world. These systems offer real time platforms to both private car owners and the passengers for sharing rides in similar directions. They usually dispatch a taxi (or a private car) that is not fully occupied to a ride request, and schedule specified pick-up and drop-off locations and time sequences for the sharing ride, with the function of finding the best match between the drivers and passengers. The onboard passengers may encounter unexpected en-route sharing requests and thus could suffer unpleasant sharing experience with extended travel time during their trips [6].

Existing works usually formulate the dynamic taxi-sharing as the dynamic pick-up and delivery problem (DPDP) or dial-a-ride problem (DARP) with the function of minimizing overall travel distances, or maximizing the number of ride-matches or benefits for both passengers and drivers. Several factors, including waiting time, travel fare, and detour distance, have been considered as the constraints.

They have presented the three key concepts in R-Sharing, i.e., sharing request q , request queue Q and sharing schedule and then describe the framework and workflow of R-Sharing.

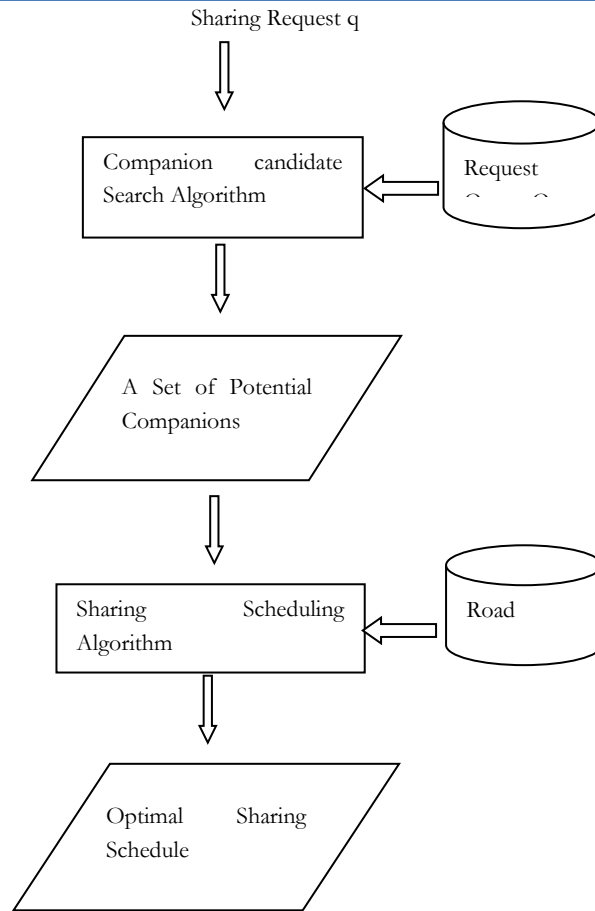


Figure 2: Framework of R-Sharing

The framework of R-Sharing is illustrated in Figure 2. R-Sharing takes a passenger’s sharing request q as an input and generates the optimal sharing schedule with a companion candidate searching algorithm and a sharing scheduling algorithm. Specifically, R-Sharing processes a sharing request by five steps:

1. A passenger submits a sharing request q to R-Sharing.
2. The companion candidate searching algorithm first searches companion candidates that are likely to satisfy q from the request queue Q .
3. The sharing scheduling algorithm generates the optimal schedule with the companions selected from the set of candidates.
4. The system informs the optimal sharing schedule to all the participants involved.
5. If all the participants accept the optimal sharing schedule, the system removes their requests from the request queue Q . Otherwise, the request q will be inserted to Q .

also removes the requests whose waiting time for companions exceeds their time window.

Their experimental results confirmed that R-Sharing is effective and efficient and provides excellent sharing experiences for its passengers.

4 OPTIMIZATION BASED APPROACH

The designed system is dynamic where new taxi drivers and passengers enter and leave the system.

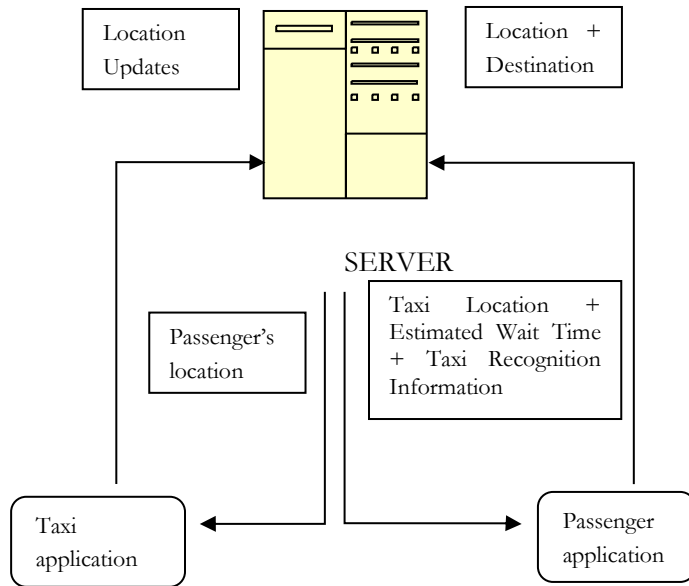


Figure 3: Taxi System Architecture

Furthermore, drivers and passengers periodically send their GPS location to the server. Optimization based approach have implemented a dynamic system that allows taxi drivers to pick up and drop off passengers without imposing any increase on the currently adopted low fares that taxi sharing makes possible. Their scheduling algorithm will take the costs associated with serving a certain passenger but will also maintain a certain level of fairness as to reduce variations in profit between competing taxi drivers. The overall profit of taxi drivers has to be maximized whereas the overall waiting time of passengers has to be minimized [2].

The system architecture of a taxi is illustrated in Figure 3. The system is composed of taxis, passengers, and a server. The system runs a highly dynamic algorithm that keeps track of the movement of all taxis in the system and has to efficiently allocate a newly arrived passenger to a taxi. The client applications of the taxis and passenger's applications utilize 3G/GPRS wireless networking to connect to the central server where it provides the necessary services

to the client and passengers applications. Their proposed system was built while taking some consideration and validated assumptions. They have assumed that all taxi drivers and passengers have Wi-Fi or 3G/4G and GPS enabled phones, where the passengers uses the phone to request taxis and the taxi drivers uses the phone to obtain the location of their assigned passengers. On the other hand, server receives information continuously from both taxis and passengers.

In other words, we can say that the taxi-application occasionally sends the GPS location of the taxi to the server. When a passenger uses his phone application to request a taxi, the passenger-application asks for the user's destination area and automatically extracts the user's current GPS location and sends it to the server. At the same time, the server has the location of all taxi drivers as well as the destination area of a passenger and the current whereabouts of that passenger. This is the area where the server runs the scheduling algorithm to determine which taxi should be allocated to this passenger. Once that taxi is determined, the server sends the location of the user that needs to be picked up by the taxi driver and sends the location of the taxi, the estimated waiting time, and taxi recognition information such as taxi number, driver name to the passenger. The core of their system is the scheduling algorithm that mediates between the server and the clients. [2].

An essential part of the design is the estimation of the incremental cost that is associated with serving a passenger. Every passenger is modelled by a pair $(S_i D_i)$, where S_i is the source and D_i is the destination. Cost is shared when passengers have same destination route.



Figure 4: Three possible scenarios for 2 passengers

For example, Figure 4(b) illustrates that passenger 2 requests a taxi after passenger 1, where the cost that is partially shared with passenger 1. Also, Figure 4(c) illustrates that passenger 2 does not force any additional cost on the taxi, since his/her cost is shared with passenger 1 only if when the second passenger's path is fully included in the first passenger's path [2].

Therefore, they have illustrated a fair and efficient taxi scheduling algorithm that tackles the common problem for inexpensive public transportation. The system utilizes smart phones and a server that runs the scheduling algorithm which benefits both taxi drivers and passengers. The passenger's waiting time is also significantly reduced due to the smart taxi allocation, scheduling, and routing algorithms. The overall system is expected to reduce traffic congestion levels, risks of accidents, wasted mileage and pollution.

5 SCALABLE APPROACH

Taxi ride sharing has been proposed as an effective approach to reduce traffic congestion, gasoline consumption, and pollution. Scalable approach proposed a data-driven simulation framework that enables the analysis of a wide range of ride-sharing scenarios. In this model, trips need not be known in advance which means the framework supports the simulation of real time ride sharing which serves unplanned trips, and fits the models using different vendors. Furthermore, the simulation model represent the taxis and trips as distinct entities, and provide variables that enable the study of realistic scenarios by taking into account the needs and constraints of multiple participants [7].

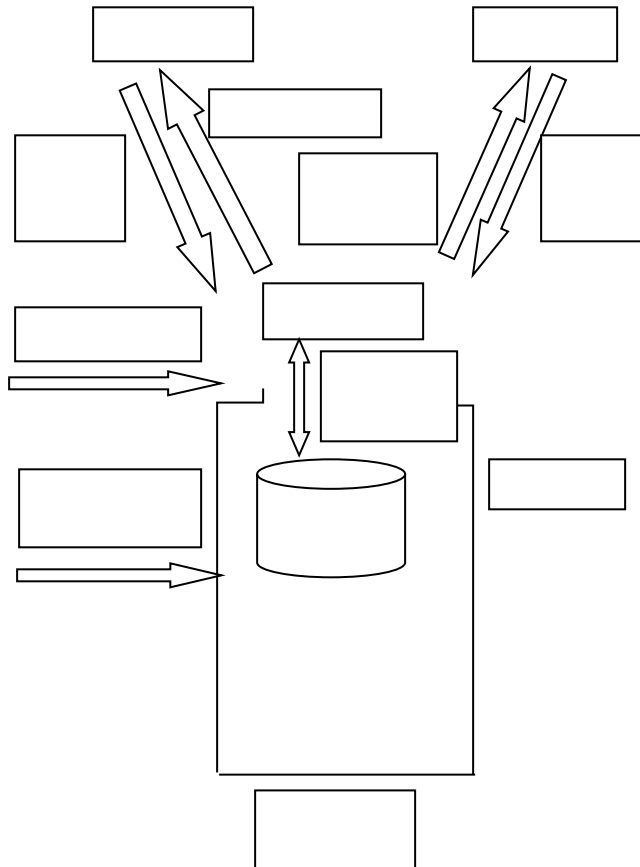


Figure 5: Taxi ride-sharing model

The flexibility of the model comes at a cost: assigning trips to taxis in real-time is highly expensive. They have described a new optimization algorithm that is arranged in the number of trips and makes use of an efficient indexing scheme, which is combined with parallelization, makes our approach scalable. They have evaluated the efficiency and effectiveness of their model using taxi data. Taxi ride-sharing model consists of three main entities: taxis, passengers and a scheduler. Given a trip request issued in real time, our model assigns this trip to a taxi in such a way to optimize a given cost function while at the same time respecting a set of predefined constraints [7]. In the taxi ride-sharing problem, the goal is to minimize the total cost or maximize the total utility of sharing while meeting a set of constraints.

The main components of the simulation model are illustrated in Figure. 5 which includes:

Taxi Fleet: The taxi fleet refers to the set of taxis that are involved in the simulation. They have also considered each taxi as a distinct object with its own ride-sharing specifications, which include: passenger capacity, maximum number of shared trips (often dictated by the vendors), maximum wait time for pick-up, and extra time for drop-off. In addition to these sharing constraints, each taxi also maintains information about its current speed, occupancy and the list of stops it has to make to serve its scheduled riders.

Passengers: They have assumed that the passengers ride in groups of size greater than or equal to one. Each group is associated with a drop-off location and a set of ride-sharing constraints.

Scheduler: The scheduler receives pick-up requests from passengers and finds the most appropriate taxi for each of them based on pre-defined metrics. To do so, the scheduler must be aware of all taxi locations along with their current states at all times.

Trip Data: The proposed simulation framework is data-driven, allowing historical trip data to be used to study the tradeoffs of different ride-sharing strategies. They have assumed that the availability of trip data which include: taxi ID, pick-up and drop-off time, latitude and longitude for both pick-up and drop-off, travel distance, and number of passengers.

Road Network: The underlying road network of a city is represented as a directed graph $G(V,E)$. All taxis travel along this road network. Each directed edge $e \in E$ represents a road segment, and each node $v \in V$ represents the intersection of two or more roads.

The simulation engine derives the best ride-sharing sequence based on a set of input parameters where pick-up requests are derived from historical data. It operates in an event-driven manner and updates its state when a pick-up request is issued. When a group of passengers request a taxi, the scheduler receives the information and requests all taxis to report their status i.e., position and sharing status.

The scheduler then calculates the additional cost for each taxi to orient this trip based on the cost, and selects the taxi with the minimal cost that satisfies all ride-sharing constraints. If no appropriate taxi is found, the request is rejected. In the taxi ride-sharing problem, the goal is to minimize the total cost or maximize the total utilization of sharing while meeting a set of

constraints [7]. Therefore, they have presented a new framework that is both scalable and flexible to support the simulation of a rich set of realistic taxi ride-sharing background. Also, at the same time, by providing a comprehensive set of parameters, they are able to study the taxi ride-sharing problem from different angles, considering different participants interest and constraints. To address the computational complexity of the model, they have described a new optimization algorithm that is sequential in the number of trips and makes use of an efficient indexing scheme, which is combined with parallelization and makes their approach scalable.

6 CONCLUSION

This study proposed a various reinforcement learning method for a shared-taxi system such as look ahead insertion policy, R-sharing, optimization approach and scalable approach that provides the solution for shared taxi problem. Look ahead insertion policy proposed a method to improve the global LOS for a shared-taxi system. They have defined the LOS as the total willingness-to-pay minus the total penalty for waiting and detour. Optimization approach illustrated a fair and efficient taxi scheduling algorithm that tackles the common problem for inexpensive public transportation. R-Sharing provides a personalized meeting point and sharing service. It enables passengers to set their preferences on four essential sharing experiences, i.e., walking distance, waiting time, travel fare, and extra travel time. Scalable approach provides a new framework that is both scalable and flexible to support the simulation of a rich set of realistic taxi ride-sharing outline.

REFERENCES

- [1]. Chong Wei, Yinhu Wang, Xuedong Yan, Chunfu Shao, "Look-ahead Insertion Policy for A Shared-taxi System Based on Reinforcement Learning", IEEE, 2017.
- [2]. Hadi El Hosni, Nourhan Farhat, Rakan Nimer, Nour Alawieh, Chadi El Masri, Mark Saroufim, Hassan Artail, Joe Naoum-Sawaya, "An Optimization-based Approach for Passenger to Shared Taxi Allocation", IEEE, 2017.
- [3]. Pedro M. d'Orey, Ricardo Fernandes and Michel Ferreira, "Empirical Evaluation of a Dynamic and Distributed Taxi-Sharing System", 15th International IEEE Conference on Intelligent Transportation Systems Anchorage, Alaska, USA, 2012.
- [4]. Mohammad MAGHROUR ZEFREH, Adam TOROK, Peter MANDOKI, Janos TOTH, "Maneuvers analysis of shared taxi and their effects on flow characteristics", 2015 Models and Technologies for Intelligent Transportation Systems (MT-ITS), June 2015.
- [5]. Shanfeng Zhang, Qiang Ma, Yanyong Zhang, Kebin Liu, Tong Zhu, Yunhao Liu, "QA-Share: Towards Efficient QoS-Aware Dispatching Approach for Urban Taxi-sharing", 12th Annual IEEE International Conference on Sensing Communication, and Networking (SECON), 2015.
- [6]. Yan Lyu, Victor C. S. Lee, Member, IEEE, Chi-Yin Chow, "R Sharing: Rendezvous for Personalized Taxi Sharing", IEEE, 2017.
- [7]. Masayo Ota¹, Huy Vo¹, Cláudio Silva¹, and Juliana Freire¹, "A Scalable Approach for Data-Driven Taxi Ride-Sharing Simulation", IEEE International Conference on Big Data (Big Data), 2015.