DRSP-Sim: A Simulator for Ride-Sharing with Pooling: Joint Matching, Pricing, Route Planning, and Dispatching

Marina Haliem[1], **Vaneet Aggarwal**[1], **and Bharat Bhargava**[1]

1 Purdue University, West Lafayette, IN

DOI: 10.21105/joss.03761

Software

- Review
- Repository
- Archive

Editor: Martin Fleischmann

Reviewers:

- @RafalKucharskiPK
- @1tylermitchell

Submitted: 10 September 2021
Published: 23 September 2021

Summary

Ridesharing is an emerging mode of transportation currently having a deep impact on the personal transportation industry. Although several ride-sharing algorithms have been developed, real-time evaluations remain the greatest challenge of such approaches. Also, algorithms of customer-vehicle matching, route planning, pricing and dispatching have been developed; however, these sub-problems tend to be studied separately and a complete integrated simulator that considers pooling is still lacking. In this paper, (1) we develop a real-time Dynamic RideSharing simulator with Pooling (DRSP-Sim) for evaluating ridesharing algorithms integrated into one simulator, and (2) we provide benchmarks for vehicle-customer matching, route planning, pricing and dispatching to test a wide range of scenarios encountered in the real world. Our work enables real-time evaluations and provides guidance for designing and evaluating future ridesharing algorithms.

Statement of need

Ride-Sharing (RS) has the potential of transforming urban mobility by providing personal convenience to each customer by accommodating their unique preferences. However, real-time evaluation of various ridesharing algorithms integrated together has been a real challenge due to the lack of a complete simulator that can support all the sub-problems at the same time. Vehicle-customer matching (Alonso-Mora et al., 2017; Tafreshian et al., 2020), vehicle route-planning (Molenbruch et al., 2017; Schönberger, 2017), ride pricing (Zhang et al., 2020) and vehicle dispatching (Al-Abbasi et al., 2019; Oda et al., 2018) are examples of the RS sub-problems that has been widely studied in literature; however, they have been studied separately or in combination of two. We present the first RS simulator that integrates all of these together, provides a baseline for each, and allows developers to plug-and-play algorithms to tackle any of the sub-problems while defaulting the rest to the provided benchmark. This, in turn, will greatly facilitate cross-study evaluations of any existing ridesharing algorithms, while providing guidance for future research.

Simulator Architecture

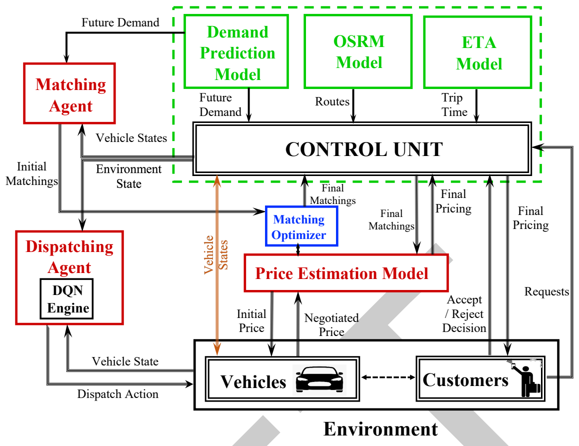

Figure 1: Overall architecture of the simulator.

Existing simulators (Segui-Gasco et al., 2019; Wu et al., 2017) either don't support pooling, lack the integration of the RS sub-problems, or generate artificial data and lack the support to real-world data and city maps. In contrast, DRSP-Sim combines the RS sub-problems and uses real-world data and maps. We construct a region graph of the New York Metropolitan area obtained from Open-StreetMap (*OpenStreetMaps*, 2018), along with a public dataset of taxi trips in NY (15 million trips) (NYC.gov, 2019). Figure 1 shows the main components of our simulator. We assume that the central control unit is responsible for: (1) maintaining the states such as current locations, current capacity, destinations, etc., for all vehicles. These states are updated in every time step based on the dispatching and matching decisions. (2) The control unit also has some internal components that help manage the ride-sharing environment such as: (a) the estimated time of arrival (ETA) model used to calculate and update the estimated arrival time. (b) The Open Source Routing Machine (OSRM) model used to generate the vehicle's optimal trajectory to reach a destination, and (c) the (Demand Prediction) model used to calculate the future anticipated demand. We adopt these three models from (Oda et al., 2018). We note that our simulator will support any algorithm for any sub-problem.

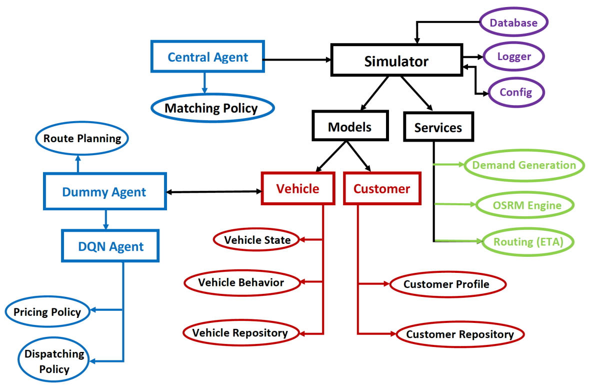

Figure 2: Code architecture of the simulator.

Here, we explain the work flow of DRSP-Sim using our benchmark algorithms provided for each sub-problem. For every time step, first, the ride requests are input to the system along with the heat map for supply and demand (which involves demand prediction in the near future). Then, based on the predicted demand, vehicles adopt a dispatching policy using DQN (Haliem et al., 2021), where they get dispatched to zones with anticipated high demand. This step not only takes place when vehicles first enter the market, but also when they experience large idle durations. Then, each vehicle receives the updated environment state from the control unit and performs a vehicle-passenger(s) matching approach, where one request (or more) gets assigned to each vehicle based on its maximum passenger capacity. Next, communicating with the Price Estimation model, each vehicle calculates the corresponding initial pricing associated with each request. Afterward, each vehicle executes its matching optimizer module to perform an insertion-based route planning. Using the demand-aware pricing provided by our simulator (Haliem, Mani, et al., 2020), vehicles weigh their utility based on the potential hotspot locations, and propose new pricing for the customer. Figure 2 shows the file organization of our code base which makes it easy to navigate through the simulator. We provide a *configuration* module which allows the user to specify which features to enable (e.g., pooling, pricing, dispatching, matching) as well as which algorithms to use for each. Besides, the simulator allows a *Logging* service that logs every event that takes place during the simulation runtime (e.g, customer pickup/dropoff).

Code Availability and Documentation

The code for this comprehensive simulator can be seen at https://github.itap.purdue.edu/Clan-labs/Dynamic-RideSharing-Pooling-Simulator, where the details on the data and the simulator setup are provided. We provide the pre-processed data of the NYC taxi trips at https://purr.purdue.edu/projects/ridesharing/files. In addition, instructions on how to generate this data from scratch is also available. Besides the integration of RS sub-problems and the ability to plug-in any such algorithm, DRSP-Sim provides a wide-range of flexibilites to enable conducting various RS scenarios. Some of these are: the ability to enable/disable the pooling feature, decide how many vehicles to populate in the simulation and how many of them adopt the DQN dispatch policy, store trained networks and replay memory, and control the map-related variables and DQN training hyper-parameters.

Benchmarks

DRSP-Sim supports pooling, which allows vehicles to pickup more than one customer at the same time. This adds more complexities to the ridesharing scenario where the route planning needs to be optimized to accommodate all customers. Matching, pricing, and dispatching algorithms need to be devised such that they take pooling into consideration.

Matching and Route Planning:

Non-dynamic Greedy Matching: In this algorithm, customer rides get assigned greedily to the nearest vehicle in its vicinity as long as the vehicle's capacity can accommodate. In this case, matching only happens at the beginning of every time-step for idle vehicles only. No dynamic matching takes place, meaning vehicles that are already assigned a route to accommodate one or more rides, aren't assigned any new rides until they become idle again.

Dynamic Insertion-based matching and route planning: In this algorithm, partially occupied vehicles are also considered while allocating new rides at the beginning of every time-step. In that case, when the partially occupied vehicle gets assigned new rides, it performs an insertion-based route planning mechanism to decide on the best route to take that would accommodate both the new customers and the on-board customers. During this route-planning, each vehicle identifies the potential rides in its vicinity, and then decides on which rides to accept according to their corresponding insertion cost to its current route (Haliem et al., 2021).

Dispatching:

Destination-driven dispatching: In this dispatching approach, vehicles only get dispatched according to the next stop in their route which is composed of a sequence of pickup and drop-off locations to serve all of its on-board customers. In that case, vehicles never get dispatched when they are idle and can't find new rides to serve.

Distributed DQN Dispatching: In this approach, in addition to the destination-driven dispatching of vehicles, a distributed DQN dispatch policy is utilized to re-balance idle vehicles to areas of predicted high demand and profits over the city, where they can better serve the demand and maximize their profits (Haliem et al., 2021).

Pricing:

Pooling Pricing: In our simulator, we provide two approaches to price the rides. One is to just accept the fare of the ride that is calculated according to these factors: (1) The total trip distance to serve this particular ride, (2) Number of customers who share travelling a trip distance, (3) The cost for fuel consumption associated with this trip, and (4) The waiting time experienced by the customer (Haliem, Mani, et al., 2020).

Demand-Aware Pricing: In this approach, each vehicle gains the necessary insight about how the supply-demand is distributed over the city through Q-network, and thus can make informed decisions on the pricing strategy that can yield him a higher profit. The implemented approach follows Haliem et al. (2021), where the vehicles propose a slightly higher price in case of having to drive to an area of predicted low demand.

Conclusion

In this work, we provide DRSP-Sim: a comprehensive simulator for the ride-sharing service with flexibilities of matching, dispatching, pricing, demand prediction, and routing, where the research on any component can be tested with the overall system. The simulator uses the NY city map, and the NYC taxi-cab data set retrieved from (NYC.gov, 2019). DRSP-Sim can run with different datasets, as well as help evaluating and testing of improved algorithms. The simulator has been used as a base for multiple analysis for ride-sharing evaluations (Haliem, Aggarwal, et al., 2020; Manchella et al., 2020, 2021; Singh et al., 2021), which exploit multi-hop passenger transportation, adaptivity in algorithms based on diurnal demand patterns, and combined goods and passenger transportation.

References

Al-Abbasi, A. O., Ghosh, A., & Aggarwal, V. (2019). Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, *20*(12), 4714–4727. https://doi.org/10.1109/TITS.2019.2931830

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, *114*. https://doi.org/10.1073/pnas.1611675114

Haliem, M., Aggarwal, V., & Bhargava, B. K. (2020). AdaPool: An adaptive model-free ride-sharing approach for dispatching using deep reinforcement learning. *BuildSys '20: The 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, Virtual Event, Japan, November 18-20, 2020*, 304–305. https://doi.org/10.1145/3408308.3431114

Haliem, M., Mani, G., Aggarwal, V., & Bhargava, B. (2020). A distributed model-free ride-sharing algorithm with pricing using deep reinforcement learning. *Computer Science in Cars Symposium*, 1–10. https://doi.org/10.1145/3385958.3430484

Haliem, M., Mani, G., Aggarwal, V., & Bhargava, B. (2021). A distributed model-free ride-sharing approach for joint matching, pricing, and dispatching using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 1–12. https://doi.org/10.1109/TITS.2021.3096537

Manchella, K., Haliem, M., Aggarwal, V., & Bhargava, B. K. (2020). PassGoodPool: Joint passengers and goods fleet management with reinforcement learning aided pricing, matching, and route planning. *CoRR*, *abs/2011.08999*. https://arxiv.org/abs/2011.08999

Manchella, K., Umrawal, A. K., & Aggarwal, V. (2021). Flexpool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/TITS.2020.3048361

Molenbruch, Y., Braekers, K., Caris, A., & Vanden Berghe, G. (2017). Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation. *Computers & Operations Research*, *77*, 58–71. https://doi.org/10.1016/j.cor.2016.07.020

NYC.gov. (2019). *NYC taxi and limousine commission-trip record data*. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

Oda, T., Joe-Wong, C., & M. (2018). MOVI: A model-free approach to dynamic fleet management. *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, 2708–2716. https://doi.org/10.1109/INFOCOM.2018.8485988

OpenStreetMaps. (2018). https://www.openstreetmap.org

Schönberger, J. (2017). Scheduling constraints in dial-a-ride problems with transfers: A metaheuristic approach incorporating a cross-route scheduling procedure with postponement opportunities. *Public Transport*, *9*(1), 243–272. https://doi.org/10.1007/s12469-016-0139-6

Segui-Gasco, P., Ballis, H., Parisi, V., Kelsall, D., North, R., & Busquets, D. (2019). Simulating a rich ride-share mobility service using agent-based models. *Transportation*, *46*. https://doi.org/10.1007/s11116-019-10012-y

Singh, A., Alabbasi, A., & Aggarwal, V. (2021). A distributed model-free algorithm for multi-hop ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*. https://doi.org/10.1109/tits.2021.3083740

Tafreshian, A., Masoud, N., & M. (2020). Trip-based graph partitioning in dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*, *114*, 532–553. https://doi.org/https://doi.org/10.1016/j.trc.2020.02.008

Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., & Bayen, A. M. (2017). Flow: Architecture and benchmarking for reinforcement learning in traffic control. *CoRR*, *abs/1710.05465*. http://arxiv.org/abs/1710.05465

Zhang, C., Xie, J., Wu, F., Gao, X., & Chen, G. (2020). Pricing and allocation algorithm designs in dynamic ridesharing system. *Theoretical Computer Science*, *803*, 94–104.