The Journal of Open Source Software

RidePy: A fast and modular framework for simulating ridepooling systems

Felix Jung [1] and **Debsankha Manik** [1]

1 Chair of Network Dynamics, Institute of Theoretical Physics and Center for Advancing Electronics Dresden (cfaed), TUD Dresden University of Technology, 01062 Dresden, Germany

DOI: 10.21105/joss.06241

Software

- Review
- Repository
- Archive

Editor: Patrick Diehl

Reviewers:

- @xoolive
- @abhishektiwari
- @ixjlyons

Submitted: 04 December 2023
Published: 06 May 2024

Summary

RidePy provides fast computer simulations of on-demand mobility modes such as ridehailing or ridepooling, enabling both theoretical research into on-demand transit systems and the estimation of their operational properties in realistic scenarios. It strongly focuses on modeling the mobility service itself, rather than its customers or the environment, enabling the user to use any mobility demand model in conjunction with RidePy. This makes it possible to use RidePy in a broad range of scenarios. Through a combination of Python, Cython, and C++, it offers ease of use at high performance. Its modular design makes customization easy, while the included modules allow for a quick start.

Statement of need

The facts that climate change is accelerating and that cities are congested call for an urgent change in the way we move (Winkler et al., 2023). To reduce carbon dioxide emissions as well as the number of vehicles on the road, digitally managed on-demand mobility services such as ridehailing and ridepooling are explored in research (Engelhardt et al., 2019; Santi et al., 2014) and on the road. Unfortunately, physically experimenting with such services for research purposes is extremely cost- and time-intensive. However, the operational properties of such systems are largely predefined in terms of the scheduling backends that manage them. This makes it possible to replace physical experiments with computer simulations, substituting virtual vehicles for actual ones and modeling the incoming mobility demand by sampling either historic requests or synthetic distributions. Another advantage of simulations is that the degree to which they represent reality may be freely adjusted. This makes it possible to both answer concrete operational questions (de Ruijter et al., 2023; Henao & Marshall, 2019; Lotze et al., 2022; Ruch et al., 2020; Wilkes et al., 2021; Zwick et al., 2021, 2022) and investigate idealized system behavior, gaining deeper insights into the general properties of on-demand mobility systems (Herminghaus, 2019; Manik & Molkenthin, 2020; Molkenthin et al., 2020; Tachet et al., 2017; Zech et al., 2022).

In this context, a simulation framework should appropriately allow for vastly different system sizes and degrees of realism. The system size incorporates the number of simulated vehicles as well as the extent of the space they operate on: A small system may consist of a single vehicle serving a network of just two nodes, while an example of a large system could be a fleet of several thousand vehicles operating on the street network of a large city. The degree of realism may be varied, for example, by sampling requests from either a uniform distribution or recorded mobility demand, or by operating on a continuous Euclidean plane versus a realistic city street network. Another option is to adjust the constraints imposed, such as the time windows assigned to stops, or the vehicles' seat capacities.

Finally, an on-demand mobility simulation framework should be fast, easy to use, and adaptable

The Journal of Open Source Software

to various applications.

Several open-source simulation software projects are already being used to investigate on-demand mobility services. Some of them focus on microscopic modeling in realistic settings, through which concrete predictions for service operation are enabled, guiding urban planning. Prominent examples are MATSim (ETH Zürich et al., 2016), which performs agent-based simulations of individual inhabitants, and Eclipse SUMO (Lopez et al., 2018), a microscopic traffic simulator. Both rely on additional packages to model on-demand mobility, such as AMODEUS (Ruch et al., 2018) for MATSim and Jade (Behrisch et al., 2014) for SUMO.

FleetPy (Engelhardt et al., 2022), a recently released on-demand mobility simulation, is primarily aimed at realistic modeling of the interactions between operators and users, specifically incorporating multiple operators. While its technical approach is similar to ours, integrating Python with fast Cython and C++ extensions, the project is predominantly focused on applied simulations, although its framework architecture promises to allow for adjustment of the model detail level.

A very different yet interesting route is taken by MaaSSim (Kucharski & Cats, 2022), which models on-demand mobility in the realm of two-sided mobility platforms such as Uber (Uber Technologies, 2023) and Lyft (Lyft, 2023).

RidePy extends this landscape by providing a universal and fast ridepooling simulation framework that is highly customizable while still being easy to use. It is focused on modeling the behavior of a vehicle fleet while covering a broad scope in terms of system size and degree of realism.

Philosophy and usage

RidePy simulates flexible mobility services based on *requests*, *dispatchers*, and *vehicles*. The vehicles continuously move along routes defined by scheduled *stops*. At each stop, passengers are picked up or dropped off, leading to a change in seat occupancy aboard the vehicle. A `RequestGenerator` supplies requests for mobility that are submitted to the simulated service, consisting of origin and destination locations and optional constraints. A `Dispatcher` processes these incoming requests. If a request cannot be fulfilled given the constraints (e.g., time windows, seat capacity), it is rejected upon submission. Otherwise, pick-up and drop-off stops are scheduled with a vehicle, respectively.

All individual components of the simulation framework may be customized or replaced. This includes `RequestGenerators`, `Dispatchers`, and the `TransportSpace` that the system operates on. Examples of `TransportSpaces` include the continuous Euclidean plane and arbitrary weighted graphs (e.g., street networks). Several components of RidePy are implemented in both pure Python and Cython/C++. While their pure Python versions are easier to understand, debug, and modify, the Cython/C++ versions make large-scale simulations tractable.

Running a RidePy simulation yields a sequence of `Events`. The included analytics code consumes these events and returns two extensive Pandas `DataFrames`: `stops` and `requests`. `stops` contains all stops that have been visited by each vehicle, along with additional information such as the vehicles' passenger occupancy. `requests` similarly contains all requests that have entered the system, enriched with secondary information such as the time riders have spent on the vehicle.

Additional included tooling allows for the set up, parallel execution, and analysis of simulations at different parameters (parameter scans). This includes the serialization of all simulation data in JSON format.

To ensure valid behavior, RidePy incorporates an extensive automated test suite.

Availability

RidePy is available from PyPI (Jung & Manik, 2023a). The source code is hosted on GitHub (Jung & Manik, 2020). Extensive documentation can be found on the project's webpage (Jung & Manik, 2023b).

Acknowledgements

We kindly thank Philip Marszal, Matthias Dahlmanns, Knut Heidemann, Malte Schröder, and Marc Timme for their input and advice.

This project was partially supported by the Bundesministerium für Bildung und Forschung (BMBF, German Federal Ministry of Education and Research) under grant No. 16ICR01 and by the Bundesministerium für Digitales und Verkehr (BMDV, German Federal Ministry for Digital and Transport) as part of the innovation initiative mFund under grant No. 19F1155A.

Competing interests

Debsankha Manik was employed at MOIA GmbH when this research was conducted. MOIA GmbH neither sponsored nor endorses his research.

References

Behrisch, M., Krajzewicz, D., & Weber, M. (Eds.). (2014). *Simulation of urban mobility: First international conference, SUMO 2013, Berlin, Germany, May 15-17, 2013. Revised Selected Papers* (1st ed. 2014). Springer Berlin Heidelberg: Imprint: Springer. https://doi.org/10.1007/978-3-662-45079-6

de Ruijter, A., Cats, O., Alonso-Mora, J., & Hoogendoorn, S. (2023). Ride-pooling adoption, efficiency and level of service under alternative demand, behavioural and pricing settings. *Transp. Plan. Technol.*, *46*(4), 407–436. https://doi.org/10.1080/03081060.2023.2194874

Engelhardt, R., Dandl, F., Bilali, A., & Bogenberger, K. (2019). Quantifying the benefits of autonomous on-demand ride-pooling: A simulation study for Munich, Germany. *2019 IEEE Intell. Transp. Syst. Conf. ITSC*, 2992–2997. https://doi.org/10.1109/ITSC.2019.8916955

Engelhardt, R., Dandl, F., Syed, A.-A., Zhang, Y., Fehn, F., Wolf, F., & Bogenberger, K. (2022). *FleetPy: A modular open-source simulation tool for mobility on-demand services*. arXiv. https://doi.org/10.48550/arXiv.2207.14246

ETH Zürich, Horni, A., Nagel, K., TU Berlin, & Axhausen, K. W. (Eds.). (2016). *The Multi-Agent Transport Simulation MATSim*. Ubiquity Press. ISBN: 978-1-909188-75-4

Henao, A., & Marshall, W. E. (2019). The impact of ride-hailing on vehicle miles traveled. *Transportation*, *46*(6), 2173–2194. https://doi.org/10.1007/s11116-018-9923-2

Herminghaus, S. (2019). Mean field theory of demand responsive ride pooling systems. *Transp Res Policy Pr.*, *119*, 15–28. https://doi.org/10.1016/j.tra.2018.10.028

Jung, F., & Manik, D. (2020). *PhysicsOfMobility/ridepy - github*. https://github.com/PhysicsOfMobility/ridepy

Jung, F., & Manik, D. (2023a). *Ridepy - PyPI*. https://pypi.org/project/ridepy/

Jung, F., & Manik, D. (2023b). *RidePy documentation*. https://ridepy.org/

Kucharski, R., & Cats, O. (2022). Simulating two-sided mobility platforms with MaaSSim. *PLOS ONE*, *17*(6), e0269682. https://doi.org/10.1371/journal.pone.0269682

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wiessner, E. (2018). Microscopic traffic simulation using SUMO. *2018 21st Int. Conf. Intell. Transp. Syst. ITSC*, 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

Lotze, C., Marszal, P., Schröder, M., & Timme, M. (2022). Dynamic stop pooling for flexible and sustainable ride sharing. *New J. Phys.*, *24*(2), 023034. https://doi.org/10.1088/1367-2630/ac47c9

Lyft, Inc. (2023). *Lyft*. https://www.lyft.com/

Manik, D., & Molkenthin, N. (2020). Topology dependence of on-demand ride-sharing. *Appl Netw Sci*, *5*(1), 49. https://doi.org/10.1007/s41109-020-00290-2

Molkenthin, N., Schröder, M., & Timme, M. (2020). Scaling laws of collective ride-sharing dynamics. *Phys. Rev. Lett.*, *125*(24), 248302. https://doi.org/10.1103/PhysRevLett.125.248302

Ruch, C., Horl, S., & Frazzoli, E. (2018). AMoDeus, a simulation-based testbed for autonomous mobility-on-demand systems. *2018 21st Int. Conf. Intell. Transp. Syst. ITSC*, 3639–3644. https://doi.org/10.1109/ITSC.2018.8569961

Ruch, C., Lu, C., Sieber, L., & Frazzoli, E. (2020). Quantifying the efficiency of ride sharing. *IEEE Trans. Intell. Transp. Syst.*, 1–6. https://doi.org/10.1109/TITS.2020.2990202

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., & Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *PNAS*, *111*(37), 13290–13294. https://doi.org/10.1073/pnas.1403657111

Tachet, R., Sagarra, O., Santi, P., Resta, G., Szell, M., Strogatz, S. H., & Ratti, C. (2017). Scaling law of urban ride sharing. *Sci Rep*, *7*(1), 42868. https://doi.org/10.1038/srep42868

Uber Technologies, Inc. (2023). *Uber*. https://www.uber.com/

Wilkes, G., Engelhardt, R., Briem, L., Dandl, F., Vortisch, P., Bogenberger, K., & Kagerbauer, M. (2021). Self-regulating demand and supply equilibrium in joint simulation of travel demand and a ride-pooling service. *Transp. Res. Rec.*, *2675*(8), 226–239. https://doi.org/10.1177/0361198121997140

Winkler, L., Pearce, D., Nelson, J., & Babacan, O. (2023). The effect of sustainable mobility transition policies on cumulative urban transport emissions and energy demand. *Nat Commun*, *14*(1), 2357. https://doi.org/10.1038/s41467-023-37728-x

Zech, R. M., Molkenthin, N., Timme, M., & Schröder, M. (2022). Collective dynamics of capacity-constrained ride-pooling fleets. *Sci Rep*, *12*(1), 10880. https://doi.org/10.1038/s41598-022-14960-x

Zwick, F., Kuehnel, N., & Hörl, S. (2022). Shifts in perspective: Operational aspects in (non-)autonomous ride-pooling simulations. *Transportation Research Part A: Policy and Practice*, *165*, 300–320. https://doi.org/10.1016/j.tra.2022.09.001

Zwick, F., Kuehnel, N., Moeckel, R., & Axhausen, K. W. (2021). Ride-pooling efficiency in large, medium-sized and small towns -simulation assessment in the Munich metropolitan region. *Procedia Computer Science*, *184*, 662–667. https://doi.org/10.1016/j.procs.2021.03.083