Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

The 14th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 15-17, 2023, Leuven, Belgium

Optimization of demand-responsive transport: The rolling horizon approach

Chengqi Lu[a,*], Michal Maciejewski[a], Hao Wu[b], Kai Nagel[a]

[a]*Technische Universität Berlin, Chair of Transport Systems Planning and Transport Telematics, Straße des 17. Juni 135, 10623 Berlin, Germany*
[b]*Technische Universität München, Chair of Transportation Systems Engineering, Arcisstraße 21, 80333 Munich, Germany*

Abstract

Demand-responsive Transport (DRT) is a popular topic in the field of transportation. The rolling horizon approach can be used to achieve an efficient operation of the DRT system in the context of trip pre-booking. In this study, a rolling horizon optimizer is implemented and used in an agent-based transport simulation framework. A series of experiments are performed to study the performance of the rolling horizon approach. The results of the experiments provide general guidelines on how to set up the rolling horizon optimizer for a DRT system. With a suitable setup, the rolling horizon approach demonstrates a clear advantage over the other approaches available in the simulation framework.

Keywords: Demand-responsive transport (DRT); Vehicle routing problem; Rolling horizon approach; Optimization; Agent-based Modeling

1. Introduction

Demand-responsive transport (DRT) is a popular topic in the field of transportation. Both private and public companies operate DRT services in various places around the world. Unsurprisingly, the operation of the DRT system is also a popular research topic. For example, Maciejewski et al. has implemented the DRT extension in an agent-based transport simulation framework, where city-scale DRT operation can be simulated and evaluated [15]. Multiple studies, such as [14], [2] and [20], have been conducted to explore the efficient operation of DRT services in different scenarios and how people will react to the new service. In that direction, Ruch et al. has gone one step further to develop a simulation platform that focuses on the operation policy of the fleet [17]. In study [9] and [18], it has been shown that the operational strategies play an important role in improving the efficiency of the DRT system. In addition

* Corresponding author. Tel.: +49-30-314-79821 ; fax: +49-30-314-23308
E-mail address: lu@vsp.tu-berlin.de

to good matching and dispatching strategies, [3] and [11] have shown that predicting travel demand based on past data is another effective way to achieve a high-quality and efficient DRT service.

While the focus of the above-mentioned studies is on the operational strategy side, we can also improve the efficiency of the DRT operation by enabling pre-booking. One recent study [12] has shown that it is possible to transport all the students in a rural area in Germany with a fleet of minivans with a similar cost as the conventional school bus service. And the pre-booking of the trips has helped to reduce the operational cost by 35% compared to the online optimization approach. In the case of school transport, as well as other use cases where travel demand are highly repetitive or predictable, such as daily commutes and trips to airports, train stations and events, we can actually compute the daily schedules of vehicles. There are several well-known studies on the Vehicle Routing Problem (VRP), such as [19] and [16], where effective heuristic approaches that are capable of creating a very good solution in a relatively short time are proposed.

Nevertheless, it is not always feasible to treat the DRT operation as a static (offline) optimization problem and solve it beforehand. The assumption that the travel demand is known in advance applies only to limited cases. Furthermore, as the problem size increases, which is not uncommon in real life, it is also challenging to acquire a good solution given the limited computational resources and time, even with the use of heuristic methods.

To tackle this issue of intractability of real problem instances, the rolling horizon approach may be applied. Instead of solving the problem for the whole day, we solve the problem piece by piece. By doing so, we usually lose the chance of finding an optimum solution, but instead we reduce the large, hard-to-solve optimization problem into a series of smaller sub-problems, for which we have a high chance of finding high-quality solutions in a much shorter time. By solving the VRP problem piece by piece, we do not need to know all the demand at the beginning of the day. Instead, we only need to know the demand within the horizon we are solving. For example, if the horizon length is 30 minutes, then we only need to know the request for trips departing in the next 30 minutes. That means this approach is more realistic and flexible for real-life DRT services as passengers only need to pre-book the rides shortly before their departures.

The rolling horizon approach is a popular method in the field of operations research. A more general description and formulation of this approach can be found in [1]. In article [5], several classical applications of the rolling horizon approach are summarized. In most of those applications, the rolling horizon approach has achieved satisfying performance. In study [8], it has been shown that with some additional constraints, the quality of the solution of the rolling horizon approach can be guaranteed. The idea of the rolling horizon approach can also be found in the studies on the (dynamic) VRP problems, such as [13] and [7], yet the focus of those studies is more on the optimization algorithms and there is not a systematic analysis on the characteristics of the rolling horizon approach.

In this paper, we will first briefly describe the implementation of the rolling horizon approach in the agent-based transport simulation framework, namely MATSim. Then we will systematically study the characteristics and the performance of the rolling horizon approach under that simulation framework. The performance will be compared to both the online optimization strategy and the fully offline optimization approach.

2. Methodology

In this study, we implement the rolling horizon approach[1] for DRT in an agent-based transportation simulation framework, MATSim [10]. MATSim is an open-soruce mesoscopic transport simulation framework that can simulate city-scale scenarios with a good level of detail within a relatively short time. With the DRT extension, MATSim is also capable of simulating the demand-responsive transport services [15]. Currently, the existing operational strategies in the DRT extension are all online strategies that optimize the DRT operation as new transport requests are submitted [4]. In addition, a new functionality that enables pre-calculating schedules for DRT vehicles has been recently introduced [12], which enables off-line optimization of DRT services assuming that we know the future demand. This functionality is a good starting point for the implemented rolling horizon approach.

Using a rolling horizon, we do not need to solve the complete optimization problem offline that spans over the simulation period. Instead, we split the original problem into a sequence of optimization problems that we solve grad-

[1] Available: `https://github.com/matsim-vsp/drt-rolling-horizon`

ually as the simulation progresses. For each planning horizon, we build a modified Dial-a-Ride Problem (DARP)[6] and solve it with a Rich-VRP (Rich Vehicle Routing Problem) solver. At the beginning of each planning period, we read all the DRT requests that will be submitted within a given time horizon. The data is then passed to the VRP solver along with the information about the current state of all the vehicles and already submitted but not yet handled requests. After a solution to this problem instance is computed by the VRP solver, we update accordingly the existing vehicle schedules. We repeat this planning procedure for each new planning period until the end of simulation.

Unlike the offline case, when we optimize the DRT operation with a rolling horizon, we need to consider the current state of the system (e.g. moving vehicles, travelling or waiting passengers, and already accepted requests). To model this problem as an offline DARP, we introduce dummy requests that represent picked up passengers. A dummy request starts at the current location of the vehicle and takes zero time to be picked up. The latest pickup time for such a request is set to the current time (i.e., the starting time of the horizon). A dummy request must be served, and it must be served by the same vehicle the actual passenger is on board. This can be done either by imposing customized hard constraints in the VRP solver or by setting up a prohibitively high penalty for rejecting the dummy requests or assigning them to other vehicles. The starting locations of vehicles are set to their current locations. Note that in the simulation, we can pause the simulation when computing vehicle plans for the next horizon, so the problem remains static while it is being solved. In the real world application, the computation of the vehicle plans can be performed shortly before the start of each planning horizon in order to accommodate the computational time. In that case, the optimizer can calculate the vehicle plans based on the projected locations of the vehicles at the beginning of the planning horizon.

When translating the computed solution, the pickup of the dummy requests will be ignored by the optimizer, as the passengers are already onboard. Only the dropoff of the dummy requests will be used to update the schedules of the vehicles. Besides the passengers that are already on board, we also need to distinguish the requests that are already accepted in the previous planning horizon and the new requests that will be submitted in the current one. As it is undesirable to reject requests after it has already been scheduled, a significantly higher cost is incurred when those requests are rejected.

In this study, we use the jsprit[2] as the Rich VRP solver, which is an open-source solver that uses the Ruin and Recreate meta-heuristics [19]. Offering a wide set of feature and being written in Java, the solver is a suitable choice for the first implementation of the rolling horizon approach for the DRT operation in MATSim. It needs to be pointed out that the implemented approach is designed to work with a general VRP solver, and the focus of this study is to explore the potential of the rolling horizon.

3. Experiments with Rolling Horizon Optimizer

In order to study the rolling horizon approach for DRT operation, we have created a test scenario for Mielec, a town in Poland. The network consists of 226 nodes (vertices) and 610 links (edges). Figure 1 illustrates the spatial and temporal distribution of the DRT demand. In Figure 1a, origin and destinations are indicated by red dots, while yellow lines represents the flows. There are 1637 DRT requests in total. Figure 1b shows their temporal distribution, which is aggregated to 5-minute time bins. Based on the results of preliminary simulations with the online DRT optimization strategy, 20 vehicles with 8 seats are enough to serve the whole demand without violating constraints of maximum waiting and travelling time. The maximum total travel time is calculated based on Equation 1, where t_{direct} is the time it takes to travel with a private car. In the test scenario, α is set to 2 and β is set to 900 (seconds). The maximum waiting time is set to 10 minutes. The same criteria will also be used to determine the pickup and delivery time window for the VRP solver, which will be used by the rolling horizon approach and offline optimization strategy.

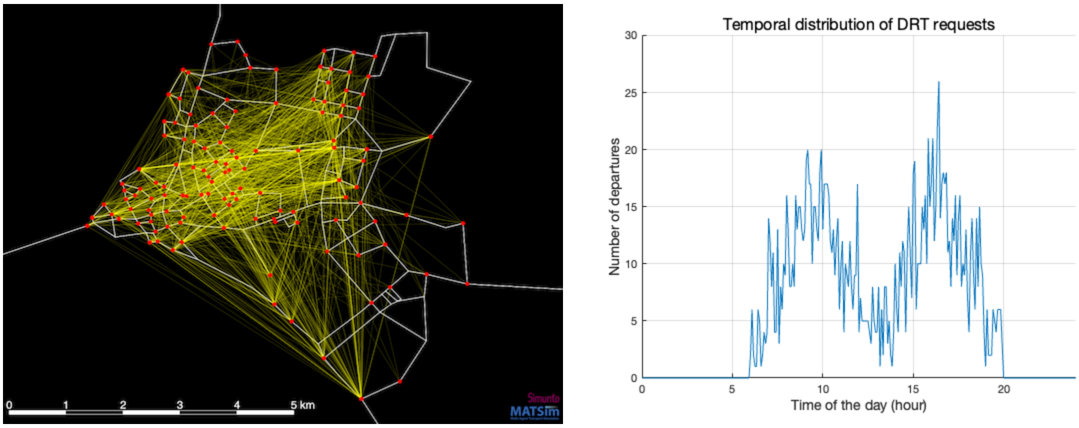
$$t_{max} = \alpha \cdot t_{direct} + \beta, \tag{1}$$

The default objective function in the existing online DRT optimization strategy in the MATSim DRT extension is to minimize the total driving time of the fleet. To achieve a fair comparison, the same objective function is used for the rolling horizon approach and the offline approach. Under this setup, the total fleet driving time can be used as the evaluation criteria of the performance. In addition to that, we keep track of the computational time in the experiments.

[2] https://jsprit.github.io/index.html

All the simulations in the experiments are performed on a scientific computation cluster node with two 10-core Intel Xeon E5-2630 processors running at 2.2 GHz.

As the rolling horizon and offline approaches use a heuristic VRP solver, each setup in our experiments is repeated five times with different random seeds and then the outputs (e.g., objective value and computation time) are averaged.



(a) spatial distribution

(b) temporal distribution

Fig. 1: Spatial and temporal distribution of the DRT demand

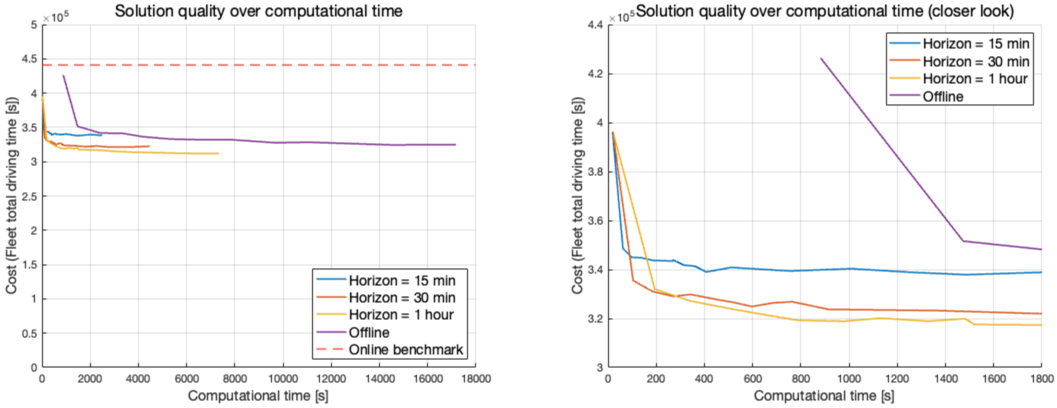
3.1. Experiment 1: Impact of Sub-problem Size

The first experiment focuses on the impact of the horizon length. In the rolling horizon approach, we break down a large (e.g. whole-day) problem into a sequence of smaller sub-problems. The size of sub-problems is an important parameter: the larger they are, the harder they are to solve. On the other hand, if we break a large problem into too small pieces, then the outcomes for each may be overly myopic. In both cases, the DRT performance may be far from optimal. The size of sub-problems is determined by the length of the horizon. In this experiment, we will explore the relation between the quality of the solution and the computational time under various horizon lengths.

Three different horizon lengths are analyzed: 15 minutes, 30 minutes and 1 hour. Since the test scenario is relatively small, we also use the standard offline approach without a rolling horizon. In fact, the offline approach can also be treated as a special case of the rolling horizon approach, where the planning horizon is infinite. In addition, we also solve the problem with the online optimization approach assuming no knowledge about the future demand, which serves as a benchmark.

The ruin and recreate meta-heuristics that we use for solving the DARP is an iterative method. By increasing the number of iterations, we increase the chances of finding better solutions at the cost of a longer computation time. In this experiment, we vary the number of iterations from 0 (i.e., the initial solution will be used) to 10 000 to observe the relation between the solution quality and computational time for each rolling horizon length (including the offline approach with its infinite planning horizon). The results are shown in Figure 2. The benchmark provided by the online approach is indicated by the horizontal dotted line in 2a. Since the performance of the rolling horizon approach saturates relatively quickly, Figure 2b provides a closer look at the left end of 2a.

Figure 2 shows that reducing the planning horizon gives higher quality results compared to the offline approach, which is caused by both using an approximate optimization and limiting the computation time. Furthermore, the shorter the horizon, the faster the computations, but the minimized total cost is higher. By extending the planning horizon, better results can be achieved when adequate computational resource is available, otherwise the objective value may actually increase. The offline approach (i.e., the purple line in Figure 2a) with its infinite planning horizon is significantly slower and gives results worse than those obtained with the rolling horizon. The results of this experiment suggest that the rolling horizon approach may improve the DRT performance given the limited amount of computational resources. In this specific test scenario, 30 minutes to 1 hour seems to be a good planning horizon. This



(a) Results (b) A closer look

Fig. 2: Results of Experiment 1: solution quality over computational time under different horizon length

means that only the knowledge of the near-future demand is required to improve the DRT performance, whereas the knowledge of the far-future demand is not crucial.

3.2. *Experiment 2: Impact of Re-planning Interval*

In addition to the planning horizon length, the re-planning interval length is another parameter in the rolling horizon approach. While the horizon length controls the size of the sub-problems (i.e., the number of requests), the replanning interval determines how often we solve the optimization problem. By design, the maximum replanning interval length (i.e., the lowest replanning frequency) is limited by the horizon length. For example, if the horizon length is 30 minutes, then we need to solve the optimization problem at least once every 30 minutes. This was the case in Experiment 1, where we assumed the replanning interval and the planning horizon are of equal lengths.

As DRT requests are constantly submitted throughout the day, it is very likely that there will be requests submitted shortly after the end of a horizon. Having both parameters equal may lead to boundary effects, as such requests will not be considered during the optimization. As a result, vehicle plans may seem optimal for a given replanning interval, but the state of vehicles at the beginning of the next replanning interval may not seem optimal anymore. By either shortening the replanning interval or extending the planning horizon, we may improve the overall performance of the system. The price for doing this is that we need to either run optimization more frequently or run optimization for bigger sub-problems. Either way, that means more computational workload. Figure 3 illustrates the impact of both parameters on the rolling horizon algorithm, where t_h is the planning horizon length and t_u is the replanning interval length. If $t_u < t_h$, part of the request assignment plan will be discarded and recalculated in the next planning horizon.

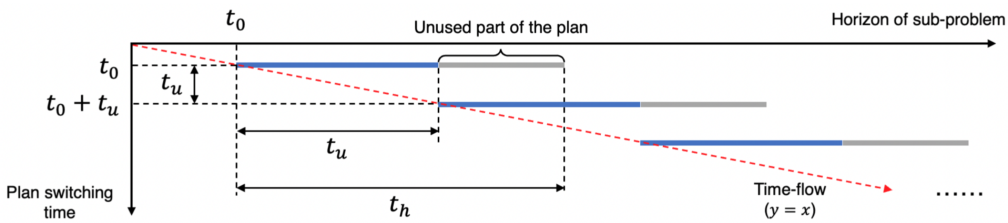
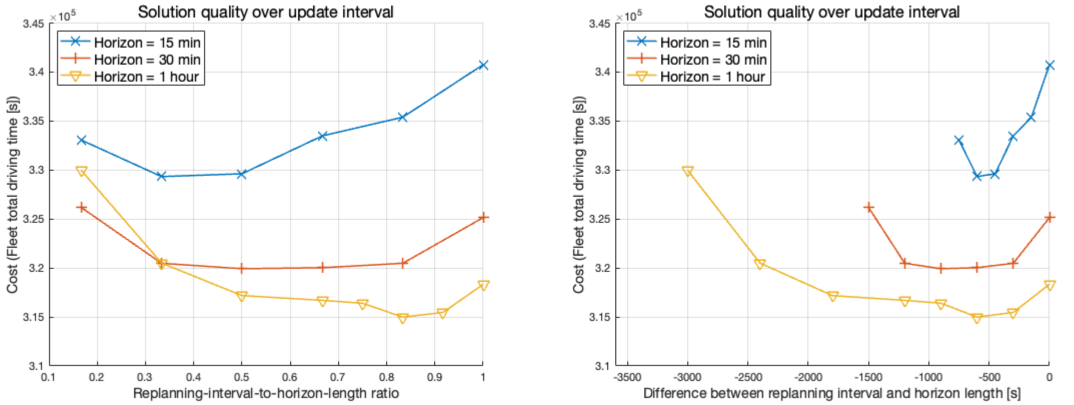


Fig. 3: Illustration of rolling horizon solver

In this experiment, we explore the impact of the replanning interval length given a fixed planning horizon length. We gradually increase the replanning interval from a small fraction of the planning horizon up to the full planning horizon. The experiment is carried out on the 3 planning horizon lengths used in the experiment 1 respectively. Figure

4 illustrates the results of this experiment. In Figure 4a, the performance of the DRT system is plotted as a function of the ratio between the replanning interval and the planning horizon. A lower ratio corresponds to a higher update frequency, and thus a larger portion of the solution being later updated. In Figure 4b, the performance of the DRT system is plotted as a function of the absolute difference between the replanning interval and the planning horizon.



(a) Relative difference between replanning interval and planning horizon (b) Absolute difference between replanning interval and planning horizon

Fig. 4: Results of Experiment 2: solution cost for different replanning intervals and planning horizons.

It can be seen from Figure 4 that all the curves are in the U-shape: the performance of the DRT system first improves as we reduce the replanning interval from the maximum value, and then it begins to deteriorate after the replanning interval reaches a certain value. In Figure 4b, the point at which the minimum cost is reached is similar for different horizon lengths. More specifically, a replanning interval that is around 10 to 15 minutes shorter than the planning horizon leads to good results in general. This means that discarding the last 10-15 minutes of the vehicle plans is enough to minimize the boundary effect that we discussed earlier.

In addition to changes in the DRT performance, the computation time also differs at different replanning intervals. The shorter the interval, the more optimization sub-problems we need to solve. Given a fixed horizon length, the sub-problem size is also fixed. Therefore, the total computation time depends on the frequency of how often the DARP is solved, which is the reciprocal of the replanning interval. Shortening the replanning interval by more than 15 minutes is not recommended for this specific test scenario.

3.3. *Experiment 3: Impact of Pre-booking Period*

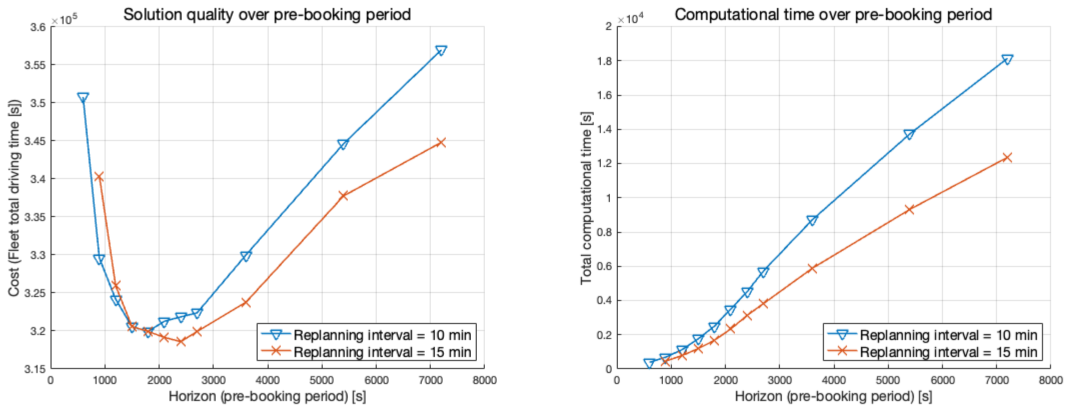
Up to now, we have explored the characteristics of the rolling horizon approach as an alternative for the static (offline) optimization approach to solve the optimization problem. As mentioned in the introduction section, unlike the static optimization approach, the rolling horizon approach can also be applied to the real-world DRT service with a relatively short minimum pre-booking time. In this experiment, we focus on a more application-oriented side and explore the impact of the minimum amount of time passengers need to book their trip in advance.

In the actual operation of the DRT system, the replanning interval needs to be relatively small. This is because of uncertainties in the system, such as delays or no-shows of passengers. A shorter replanning interval, meaning a higher update frequency, can prevent the pre-calculated vehicle schedules to diverge too far away from the actual state. While the replanning interval length is constrained, the planning horizon can still be varied, which corresponds to the minimum lead time for pre-booking. For example, a planning horizon of 30 minutes indicates that passengers need to pre-book the trip at least 30 minutes in advance.

Experiment 1 suggests that given the adequate computational resources, a longer planning horizon leads to better results. At the same time, Experiment 2 indicates that having the replanning interval significantly shorter than the planning horizon may have a negative impact on the service performance. Therefore, we want to analyze how the performance of the rolling horizon approach will be impacted by the planning horizon length given a fixed replanning interval. In this experiment, we set the replanning interval to 10 and 15 minutes and gradually increase the planning horizon length.

The results of this experiment are summarized in Figure 5, where we can see that the relation between the solution quality and the planning horizon (i.e., minimum lead time for pre-booking) is also in a U-shape. At the same time, the computational time increases monotonically as the horizon extends. While the curves for the computational time are intuitive, the relation between the solution quality and the planning horizon requires a deeper look. The quality improves (i.e., cost decreases) as we extend the horizon up to a certain point. Then, as we further increase the horizon, the solution quality becomes worse. This is because we use the same number of ruin and recreate iterations for each setup, as the planning horizon is the only variable in this experiment. As the planning horizon increases, the problem size becomes larger, and more iterations are needed to achieve a good solution. If we allocate adequate computational resource for the runs with longer planning horizons, then the U-shape should disappear. In the reality, however, the computational resource is usually limited, and therefore, a similar U-shape curve is likely to appear.

Combining both plots, we can conclude that it is not necessary to use a planning horizon that is considerably longer than the replanning interval. When the replanning interval is 10 minutes, it does not make sense to use a horizon of more than 30 minutes, or if the replanning interval is 15 minutes, then the horizon does not need to go beyond 40 minutes. This means that the minimum pre-booking time does not need to be very long. Given a short replanning interval, a minimum lead time of 30 to 40 minutes will be adequate to achieve good operational efficiency of the DRT system in this scenario, and requiring a longer time buffer for submitting requests is non-necessary. Still, customers may pre-book their trips long in advance, but the DRT system can simply store them in the system and only consider them once they fall into the planning horizon.



(a) Solution quality over pre-booking time period

(b) Computational time over pre-booking time period

Fig. 5: Results of Experiment 3: The impact of pre-booking time period.

4. Conclusion and Outlook

In this study, a rolling horizon DRT optimizer has been implemented in an agent-based transport simulation framework, MATSim. This enables the simulation of the DRT system with the rolling horizon approach. Compared to the online optimization strategy, which is currently the default DRT optimizer in the MATSim simulation framework, the rolling horizon optimizer provide a significantly better performance within a reasonable computational time when trip pre-booking is enabled. In comparison to the offline optimization approach, which solves the whole DRT problem at once, the rolling horizon approach can return similar or even better results in a considerably shorter time. This means the rolling horizon approach can also be applied to larger scenarios, where the offline optimization approach cannot provide a good solution within a limited computational time or computational resources. Furthermore, since the rolling horizon approach solves the DRT problem piece by piece, the travel demand departing within a horizon only need to be known by the time when the solver begins to solve the sub-problem of that horizon. In other words, the travel demand does not need to be fully known at the beginning of the DRT operation, which is required by the static optimization approach. This makes the rolling horizon approach more applicable for the real-world DRT services.

The carried out experiments provide us with some general guidelines for setting up the parameters of the rolling horizon optimizer. Here, we summarize some of the key findings from the experiments. A longer planning horizon leads to a better result in general, but it requires a longer time to compute. A shorter horizon results in shorter computation time, but may give a worse solution in general. A suitable horizon length needs to be chosen based on the available computational resources and the size of the DRT scenario. It is beneficial to have the replanning interval smaller than the planning horizon, which introduces an overlap between two consecutive optimization sub-problems and ensures that the state at the end of one replanning interval is more aligned with the DRT demand in the following one. However, having a too big overlap may unnecessarily increase the computational time without additional improvements in the solution quality. Finally, we also explored the impact of the minimum lead time for pre-booking in the real-life DRT operations, where the replanning interval is usually short. For a replanning interval of 10 to 15 minutes, planning beyond 30 to 40 minutes does not improve results, which means asking passengers to pre-book their trips 30 to 40 minutes ahead is adequate to achieve a better DRT service performance.

This study was carried out in the context of DRT pre-booking, which is a way to increase the efficiency of the DRT operation. The implementation of the rolling horizon approach adds a powerful tool to the MATSim framework, and it can be helpful for future research on the topic of DRT operation and pre-booking. For example, the result obtained with the rolling horizon approach can be used as a benchmark to evaluate the efficiency of the online operation strategy. At the same time, specific case studies can also be carried out based on the findings of this study. The impact of pre-booking and the incentive for passengers to pre-book trips can be explored in those specific case studies, and the outcomes can be utilized by the DRT operators. Furthermore, when it comes to DRT operators, the inclusion of immediate requests in the current pre-booking context is another direction to extend the current studies. Last but not the least, adapting the rolling horizon optimizer to various VRP solvers and thus improving the performance of the optimizer will remain an interesting long-term research topic.

Acknowledgements

This work was partly funded by the German Federal Ministry of Transport and Digital Infrastructure, grant number: 45KI04D041.

References

[1] Bean, J.C., Smith, R.L., 1984. Conditions for the existence of planning horizons. Mathematics of Operations Research 9, 391–401.
[2] Bischoff, J., Maciejewski, M., 2016. Simulation of city-wide replacement of private cars with autonomous taxis in berlin. Procedia computer science 83, 237–244.
[3] Bischoff, J., Maciejewski, M., 2020. Proactive empty vehicle rebalancing for Demand Responsive Transport services. Procedia Computer Science 170, 739–744. URL: https://www.sciencedirect.com/science/article/pii/S1877050920306220, doi:10.1016/j.procs.2020.03.162.
[4] Bischoff, J., Maciejewski, M., Nagel, K., 2017. City-wide shared taxis: A simulation study in berlin, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 275–280. doi:10.1109/ITSC.2017.8317926.
[5] Chand, S., Hsu, V.N., Sethi, S., 2002. Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. Manufacturing & Service Operations Management 4, 25–43.
[6] Cordeau, J.F., Laporte, G., 2007. The dial-a-ride problem: models and algorithms. Annals of Operations Research 153, 29–46. URL: https://doi.org/10.1007/s10479-007-0170-8, doi:10.1007/s10479-007-0170-8.
[7] Engelhardt, R., Dandl, F., Bogenberger, K., 2022. Simulating ride-pooling services with pre-booking and on-demand customers. ArXiv preprint arXiv:2210.06972.
[8] Glomb, L., Liers, F., Rösel, F., 2022. A rolling-horizon approach for multi-period optimization. European Journal of Operational Research 300, 189–206. URL: https://www.sciencedirect.com/science/article/pii/S0377221721006536, doi:https://doi.org/10.1016/j.ejor.2021.07.043.
[9] Hörl, S., Ruch, C., Becker, F., Frazzoli, E., Axhausen, K.W., 2019. Fleet operational policies for automated mobility: A simulation assessment for zurich. Transportation Research Part C: Emerging Technologies 102, 20–31.
[10] Horni, A., Nagel, K., Axhausen, K.W., 2016. The Multi-Agent Transport Simulation MATSim. Ubiquity Press. doi:10.5334/baw.
[11] Lu, C., Maciejewski, M., Nagel, K., 2020. Effective operation of demand-responsive transport (DRT): Implementation and evaluation of various rebalancing strategies. Working Paper, TU Berlin, Transport Systems Planning and Transport Telematics.

[12] Lu, C., Maciejewski, M., Wu, H., Nagel, K., 2022. Demand-responsive transport for students in rural areas: A case study in vulkaneifel, germany. Available at SSRN: https://ssrn.com/abstract=4181254 or http://dx.doi.org/10.2139/ssrn.4181254.

[13] Ma, Z., Koutsopoulos, H.N., 2022. Near-on-demand mobility. the benefits of user flexibility for ride-pooling services. Transportation Research Part C: Emerging Technologies 135, 103530. URL: https://www.sciencedirect.com/science/article/pii/S0968090X2100512X, doi:https://doi.org/10.1016/j.trc.2021.103530.

[14] Maciejewski, M., Bischoff, J., Nagel, K., 2016a. An assignment-based approach to efficient real-time city-scale taxi dispatching. IEEE Intelligent Systems 31, 68–77. doi:10.1109/MIS.2016.2.

[15] Maciejewski, M., Horni, A., Nagel, K., Axhausen, K.W., 2016b. Dynamic transport services. The multi-agent transport simulation MATSim 23, 145–152.

[16] Pisinger, D., Ropke, S., 2007. A general heuristic for vehicle routing problems. Computers and Operations Research 34, 2403–2435. URL: https://www.sciencedirect.com/science/article/pii/S0305054805003023, doi:https://doi.org/10.1016/j.cor.2005.09.012.

[17] Ruch, C., Hörl, S., Frazzoli, E., 2018. Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE. pp. 3639–3644.

[18] Ruch, C., Lu, C., Sieber, L., Frazzoli, E., 2021. Quantifying the Efficiency of Ride Sharing. IEEE Transactions on Intelligent Transportation Systems 22, 5811–5816. doi:10.1109/TITS.2020.2990202. conference Name: IEEE Transactions on Intelligent Transportation Systems.

[19] Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., Dueck, G., 2000. Record breaking optimization results using the ruin and recreate principle. Journal of Computational Physics 159, 139–171. URL: https://www.sciencedirect.com/science/article/pii/S0021999199964136, doi:https://doi.org/10.1006/jcph.1999.6413.

[20] Vosooghi, R., Puchinger, J., Jankovic, M., Vouillon, A., 2019. Shared autonomous vehicle simulation and service design. Transportation Research Part C: Emerging Technologies 107, 15–33.