



**HAL**  
open science

## Resource-constrained replanning in MATSim applied to the simulation of peer-to-peer car sharing services

Sebastian Hörl, Jeremy Sobieraj, Steffen Axer, Hannes Rewald

### ► To cite this version:

Sebastian Hörl, Jeremy Sobieraj, Steffen Axer, Hannes Rewald. Resource-constrained replanning in MATSim applied to the simulation of peer-to-peer car sharing services. 12th International Workshop on Agent-based Mobility, Traffic and Transportation Models (ABMTRANS 2023), Mar 2023, Leuven, Belgium. pp.698-703, 10.1016/j.procs.2023.03.091 . hal-04050638

**HAL Id: hal-04050638**

**<https://hal.science/hal-04050638>**

Submitted on 18 Apr 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



The 12th International Workshop on Agent-based Mobility, Traffic and Transportation Models,  
Methodologies and Applications (ABMTrans 2023)  
March 15-17, 2023, Leuven, Belgium

## Resource-constrained replanning in MATSim applied to the simulation of peer-to-peer car sharing services

Sebastian Hörnl<sup>a,\*</sup>, Jeremy Sobieraj<sup>a</sup>, Steffen Axer<sup>b</sup>, Hannes Rewald<sup>b</sup>

<sup>a</sup>*Institut de Recherche Technologique SystemX, Palaiseau, France*

<sup>b</sup>*Volkswagen AG, Wolfsburg, Germany*

---

### Abstract

Peer-to-peer car sharing is a service model in which car owners let others use their vehicle while it is not in use by themselves. To simulate such a service in the agent-based transport simulation framework MATSim, a novel synchronization concept is proposed that considers limited resources in the decision-making of the agents. The concept is introduced in detail and demonstrated for peer-to-peer car sharing on a conceptual use case and a large-scale simulation for Berlin. The concept is generalizable to other constrained resources, such as parking or micromobility.

© 2023 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0>)

Peer-review under responsibility of the scientific committee of the Conference Program Chairs

*Keywords:* p2p, peer to peer, car sharing, MATSim, agent-based, simulation

---

### 1. Introduction

Peer-to-peer (P2P) car sharing is a service in which individual car owners can interact with a platform on which they propose to share their vehicle with others at a predetermined place and time. Such a service has the potential to reduce the overall number of cars used in a region by allowing occasional drivers to make use of other people's cars. Several studies have been performed about user preferences towards such a service [6, 13, 3, 14] but simulation-based assessments that consider its dynamics and the synchronization between users are not available or very conceptual [7].

P2P car sharing is strongly affected by today's expectations towards autonomous driving and could evolve gradually from (1) the conventional form introduced above, to a service where (2) the vehicle autonomously drives between owners and users, or where it is (3) integrated into a shared on-demand fleet when not in use.

---

\* Corresponding author

*E-mail address:* [sebastian.horl@irt-systemx.fr](mailto:sebastian.horl@irt-systemx.fr)

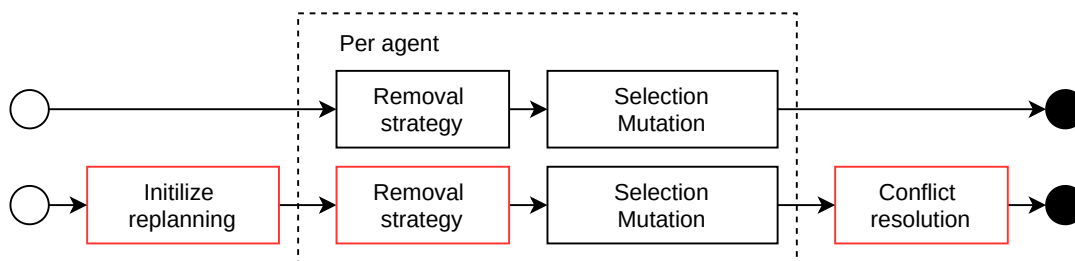


Fig. 1: Update of the replanning process

In order to understand the implications of varying user preferences, cost structures, and levels of automation on potential P2P car sharing business models, the goal of the present paper is to simulate such a service using the agent-based transport simulation framework MATSim [8]. Integrating such a service in MATSim means that it can be tested with a wide range of existing components, for instance, for the calculating of emissions [11], noise [16] and congestion [12], or in combination with other modes such as automated taxis [10] or micromobility [1].

Experiments on limited resources and synchronization have been performed in MATSim, for instance on the case of car-pooling [5], which is a complementary problem to P2P car sharing in which other travelers are picked up by the individual drivers. However, the proposed implementation modified many core components and concepts of MATSim and stayed merely conceptual [4].

In the following, a novel approach with minimal implications on the existing functioning and code base of the framework is introduced in a general manner, which is then specified for the P2P car sharing case.

## 2. Constrained replanning

MATSim simulations are performed in an iterative way: All agents, which represent the population of a city or region, have daily schedules with individual activities and transport legs in between. In the first phase of a MATSim iteration (the *mobility simulation*), all movements of the agents are jointly simulated, leading to phenomena such as congestion. During simulation, the agents' daily plans are scored, with activities performed at the right time for the expected duration giving a positive score and spending time in traffic accumulating negative scores. The obtained scores can be dependent on activity types, transport modes and even on a per-agent basis. After finishing the *mobility simulation*, a *replanning* process is performed for each agent, which is described further below. It leads to modifications of some plans (for instance, avoiding congestion) with the hope to reach a better score in the following iteration and the respective *mobility simulation*.

### 2.1. Standard approach

The standard approach for the *replanning* phase in MATSim is shown on top in Figure 1. Each agent has a *plan memory* which holds daily schedules that have been executed in previous MATSim iterations and have, hence, obtained a score. The memory is limited to  $M$  plans, and a *removal strategy* is applied in case too many plans are present. In the standard case, the plan with the worst score is removed. After, a *replanning strategy* is chosen for each agent dependent on a probability that is defined for each strategy. The strategy can be a *selection strategy* which selects (usually based on the scores) a plan from memory to be set as the *selected* one for the mobility simulation, or it can be an *innovation strategy* that usually randomly selects a plan from memory, applies a modification (departures times, transport modes) and notes it as the *selected* plan of this agent. The combination of removal and selection (biasing the memory towards higher scores) and innovation (finding better configurations) guides an agent towards higher scores over multiple iterations and resembles the utility maximization approach used in discrete choice modeling [9].

## 2.2. Conflict resolution

We propose a general modification to the replanning process, which is agnostic to the use case or the type of resource that is considered. The main idea revolves around the concept of potentially *conflicting* and surely *non-conflicting* plans. In the case of P2P car sharing, a plan that contains a leg using the P2P service is always *conflicting* as it may potentially clash with another traveler's plan (in case they want to use the same car at the same time) and *non-conflicting* are those plans that do not include the transport mode. In general, more complex definitions for other use cases can be constructed.

Given these definitions, the modified replanning strategy with resource constraints evolves as shown at the bottom of Figure 1. First, the initially selected plan before starting the replanning process is noted down for each agent in the *initialization* step. After, a modified removal strategy is executed per agent. We assume that, in the first iteration of a MATSim simulation, each agent has at least one *non-conflicting* plan defined in its input memory. In that case, the removal strategy must simply ensure that at least one *non-conflicting* plan remains in memory. For the standard worst-score removal strategy, this can be easily obtained by removing the plan with the worst score, *except* if it is the *only* one that is *non-conflicting*, then the second worst is removed.

After, replanning continues as in the standard case. No modifications need to be introduced to any of the existing selection and innovation strategies. The goal is to arrive at a new *selected* plan for each agent as before. However, a *conflict resolution* step is added to the replanning process. In this step, all selected plans are examined and a specific logic may decide whether they are *rejected* or *accepted*. In the rejection case, the replanning engine needs to decide which plan to select instead. In our proposed implementation, this is either the plan that was initially selected (before replanning) by the agent if it already was *non-conflicting* or a randomly selected *non-conflicting* one from memory, which surely must exist given the described procedure.

Note that the process described here is generic and needs to be specified individually for different services and resources. However, the only implementation that is necessary to add a new resource constraint is to define a logic that tags a plan to be potentially *conflicting* and a logic that defines which plans should be accepted and rejected.

## 3. Peer-to-peer car sharing

Using standard MATSim functionality, the P2P mode can be implemented as a *chain-based* mode such as the private car, which means that the innovation strategies will make sure that transport modes along legs are defined such that a vehicle is brought back to where it has been picked up. In an unconstrained case, this would let agents use this mode whenever a viable return trip is found.

However, a reservation logic can be implemented using the proposed resource-constrained replanning process. It needs to be defined such that the new p2p mode can only be used where and when another agent has moved its vehicle using the classic private car mode. Also, the vehicle must return (i.e., the user needs to finish the round-trip), before the owner is departing from that location again using the car mode. Likewise, care must be taken that not two agents want to use the same car at the same time.

In that setup, all plans that contain the p2p mode are potentially *conflicting*. The resolution logic then identifies all users with the p2p mode and sorts them in decreasing order by the total duration of their p2p legs. It identifies all *user slots* along those plans, being the shortest chain of activities and legs between departing with the p2p mode at a certain location and returning to it using the same mode. Analogously, all *owner slots* in the population can be identified by searching the shortest chains between arriving at a location using the car mode and departing again from the place using car. Note that the chains between departure/arrival in both cases can be arbitrarily complex, for instance, for an *owner* coming to work by car, *walking* to a restaurant for lunch and coming back, and then using the car again to go back home. Additional *owner slots* are generated in the morning and the evening.

Once all users are sorted and all *user slots* and *owner slots* are identified, the assignment logic is applied user by user. For each user, compatible owner slots are searched in the population. If no set of owner slots can be found that covers all user slots, the user plan is rejected. In case an assignment can be made, it is noted down. The next user is then treated the same way, but its slots are also checked against other users' slots that are already assigned to the compatible owner slots. In case a conflict is found, the algorithm needs to proceed by finding a different compatible set of owner slots, or reject the user plan if none is found. Finally, a set of plans that are accepted and rejected is

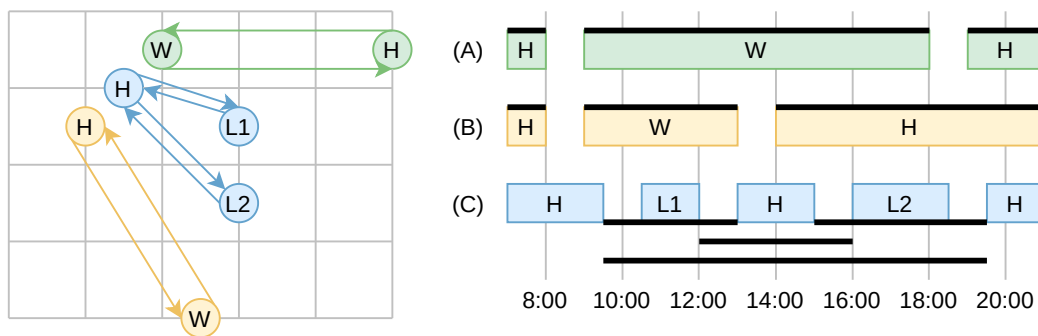


Fig. 2: Conceptual test simulation

obtained. Note that this procedure is a best-response assignment that favors sharing time, but more complex (but computationally heavy) approaches such as bipartite matching could be used to find system-optimal assignments.

To increase stability of the replanning process, one additional tweak is added to the process for the P2P use cases: First, all agents that initially had a p2p plan are assigned. Only after that, all others which have just switched to such a plan in the most recent replanning step are treated. This way, assignments are not constantly broken and remade, but they are kept consistent over multiple iterations.

#### 4. Results

Two experiments are presented in the following. The first is a conceptual set-up, which will be published as an easy-to-follow code implementation of the approach. The second example integrates the P2P mode into a large-scale simulation of Berlin.

##### 4.1. Conceptual experiments

Figure 2 describes the configuration of the conceptual use case. Three types of agents (A, B, C) exist. They have home (H), work (W), and leisure (L) activities. Each agent may choose between three modes (car, walk, p2p) to perform its travels. Whenever car is chosen, a score of  $-1$  is obtained, whenever walk is chosen,  $-2$  is obtained, and  $s \in \mathbb{R}$  for p2p. The example is constructed such that A and B will favor using their car, as they cannot make use of somebody else’s car. Their *owner slots* are shown in Figure 2 as black bars on top of their timed activities. In contrary, C may choose to use the car of A or B during the day as its home location is close enough to the work of A and the home of B. *Potential* user slots (which will depend on the outcome of the standard replanning) are indicated as black lines below C’s activities. Finally, C may simply choose to use the car for all or one of its slots.

In the test simulation, all modes are teleported with the same fixed speed, i.e. agents are simply moved on the grid using predefined time periods.

Figure 3 shows the evolution of the mode shares during the MATSim simulations, with values  $s = \{-2, 0.0, +2.0\}$  for scoring p2p and 100 agents of each type. One can see that a low score (a) leads to an initial accumulation of p2p plans, which are quickly discarded once better plans are found. On the contrary, p2p increases continuously when using a score that is substantially better than car (c), but reaches a plateau after around 150 iterations. Note that in the unconstrained case, p2p would become the dominant mode as it would simply replace all car trips since it is more favorable. One can, hence, clearly see the synchronization effect that limits the number of trips that can be performed using p2p.

##### 4.2. Large-scale experiments

Finally, the constrained replanning process is applied to an existing open and large-scale simulation for Berlin [15]. Technically, the general module for constrained replanning and the specific module needs to be added to the baseline

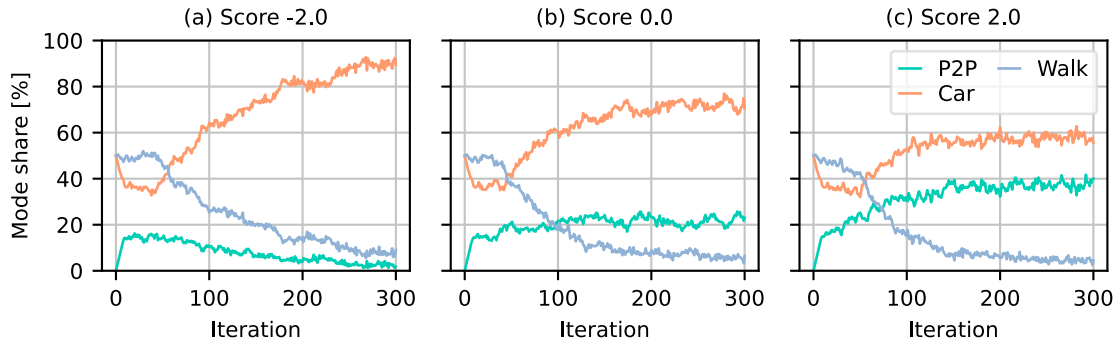


Fig. 3: Results of the conceptual simulations

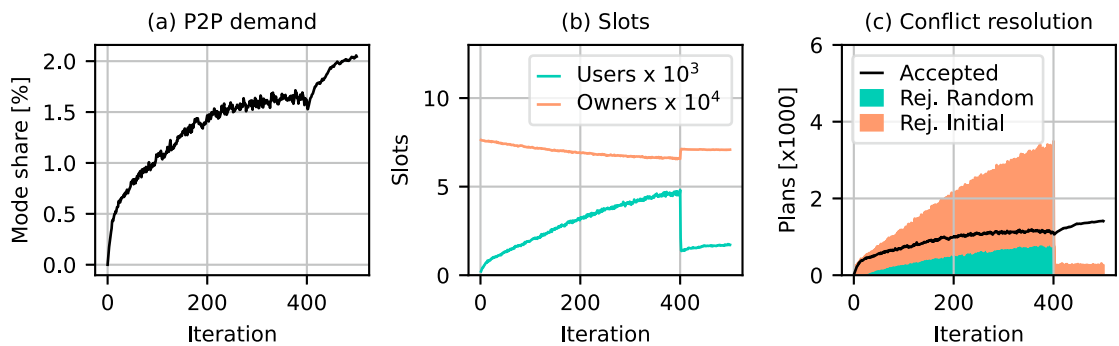


Fig. 4: Results of the Berlin simulations

simulation. Furthermore, the p2p mode is configured such that it is scored exactly as the car mode while removing daily monetary costs and increasing the per-distance cost by 50% to represent the service fees and owner profit. The p2p mode is routed according to car travel times in the network, but only teleported in the mobility simulation (no explicit car movements are considered).

Figure 4 shows the results of one simulation run in which user and owner slots are compatible if their departure/arrival locations are closer than 120 meters. In that configuration, Figure 4a shows that a mode share of about 1.5% of all trips is reached after 400 iterations. After, the Berlin simulation is configured such that innovation strategies are turned off in the replanning process. This means that only plans that are present in memory are selected from that point on. One can see that this allows to identify beneficial combinations of users and owners, leading to an increased share towards the final 500th iteration.

Similarly, Figure 4b shows the user and owner slots that are obtained in the assignment process. While the owner slots decrease (following the general mode share of car), the user slots increase strongly as they start from zero in the beginning of the simulation. At innovation turnoff, the user slots decrease drastically (as no new p2p are explored through innovation).

Finally, Figure 4c shows the number of accepted and rejected plans among those that were potentially conflicting. The rejections can be distinguished between those that lead to going back to the initially selected plan of an agent or those that lead to a random selection. One can see that after innovation turnoff the number of accepted plans largely dominates the number of rejected plans and that rejections almost always happen for agents which initially were following a non-conflicting plan but then tried again an existing conflicting one from memory.

## 5. Conclusion

The sections above show that the proposed approach for resource-constrained replanning is modular, integrates well with the existing code base of MATSim and can be used to simulate P2P car sharing services on the behavioral side of the framework.

However, currently, many dynamic aspects of such a service are not considered. Movements of p2p users are not linked to a vehicle (and especially not the vehicle of the owner) and remain “virtual”. A next step towards full integration (that is also compatible with emission and noise calculations, for instance) is to consider these movements. The projected approach is to use the dynamic vehicle simulation capabilities of MATSim to simulate relevant vehicles as individual agents that “follow” either the owner or the user. This may lead to complex situations in which an owner may arrive late, or a user may be delayed for the return. These cases should be covered naturally by the scoring approach of MATSim which will penalize such plans, and they allow exploring additional operational dimensions such as proposing different levels of time margins before and after rentals when performing the assignment.

The assignment itself can be improved by using more complex heuristics or solving a bipartite matching or mixed integer linear programming problem.

Finally, the paper started out describing the P2P car sharing service as a general resource synchronization problem. That means that the approach can be generalized to other use cases. Currently, parking is either modeled dynamically in MATSim with agents trying to approach their destination and then roaming to find a parking spot [2] or not at all. The resource-constrained replanning approach could be used by explicitly resolving parking competition in the replanning step of MATSim. Another use case is micromobility [1], where agents currently try to approach a station to pick up or return a shared bike or scooter and fall back to the next best station if their intended action is not possible. Managing the rentals upfront using resource-constrained replanning may also stabilize these simulations. Finally, it may be worthwhile to explore again the case of household or peer group synchronization with respect to the vehicles that are available to them.

## References

- [1] Balac, M., Horl, S., 2021. Simulation of intermodal shared mobility in the San Francisco Bay Area using MATSim, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA. pp. 3278–3283.
- [2] Bischoff, J., Nagel, K., 2017. Integrating explicit parking search into a transport simulation. *Procedia Computer Science* 109, 881–886.
- [3] Dill, J., McNeil, N., Howland, S., 2019. Effects of peer-to-peer carsharing on vehicle owners’ travel behavior. *Transportation Research Part C: Emerging Technologies* 101, 70–78.
- [4] Dubernet, T., Axhausen, K.W., 2013. Including joint decision mechanisms in a multiagent transport simulation. *Transportation Letters* 5, 175–183.
- [5] Dubernet, T.J.P., Rieser-Schüssler, N., Axhausen, K.W., 2013. Using a multi-agent simulation tool to estimate the car-pooling potential, in: TRB 92nd Annual Meeting Compendium of Papers. p. 13.
- [6] Golalikhani, M., Oliveira, B.B., Carravilla, M.A., Oliveira, J.F., Antunes, A.P., 2021. Carsharing: A review of academic literature and business practices toward an integrated decision-support framework. *Transportation Research Part E: Logistics and Transportation Review* 149, 102280.
- [7] Hampshire, R.C., Sinha, S., 2011. A simulation study of Peer-to-Peer carsharing, in: 2011 IEEE Forum on Integrated and Sustainable Transportation Systems, Vienna, Austria. pp. 159–163.
- [8] Horni, A., Nagel, K., Axhausen, K.W. (Eds.), 2016. *The Multi-Agent Transport Simulation MATSim*.
- [9] Hörl, S., 2021. Integrating discrete choice models with MATSim scoring. *Procedia Computer Science* 184, 704–711.
- [10] Hörl, S., Becker, F., Axhausen, K.W., 2021. Simulation of price, customer behaviour and system impact for a cost-covering automated taxi system in Zurich. *Transportation Research Part C: Emerging Technologies* 123, 102974.
- [11] Kaddoura, I., Bischoff, J., Nagel, K., 2020. Towards welfare optimal operation of innovative mobility concepts: External cost pricing in a world of shared autonomous vehicles. *Transportation Research Part A: Policy and Practice* 136, 48–63.
- [12] Kaddoura, I., Nagel, K., 2019. Congestion pricing in a real-world oriented agent-based simulation context. *Research in Transportation Economics* 74, 40–51.
- [13] Olaru, D., Greaves, S., Leighton, C., Smith, B., Arnold, T., 2021. Peer-to-Peer (P2P) carsharing and driverless vehicles: Attitudes and values of vehicle owners. *Transportation Research Part A: Policy and Practice* 151, 180–194.
- [14] Shaheen, S., Martin, E., Hoffman-Stapleton, M., 2021. Shared mobility and urban form impacts: a case study of peer-to-peer (P2P) carsharing in the US. *Journal of Urban Design* 26, 141–158.
- [15] Ziemke, D., Kaddoura, I., Nagel, K., 2019. The MATSim Open Berlin Scenario: A multimodal agent-based transport simulation scenario based on synthetic demand modeling and open data. *Procedia Computer Science* 151, 870–877.
- [16] Zwick, F., Kuehnel, N., Moeckel, R., Axhausen, K.W., 2021. Agent-based simulation of city-wide autonomous ride-pooling and the impact on traffic noise. *Transportation Research Part D: Transport and Environment* 90, 102673.