Decentralized Method in Ride-sourcing Reposition Decision-making Process

by

K. Cui

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	5512158	
Dissertation duration:	March 1, 2023 – Oct 30, 2023	
Thesis committee:	Prof. Dr. O. Cats,	TU Delft, supervisor
	Dr. S. S. Azadeh,	TU Delft
	A. de Ruijter,	TU Delft

Abstract

Efficient repositioning strategies for idle vehicles in ride-sourcing systems help reduce passengers' waiting time and drivers' operational costs, which help platforms attract more passengers and drivers. In this paper, we propose a decentralized repositioning strategy for drivers, in which drivers make individual decisions on where they reposition themselves, based on their own experiences. In comparison to an existing centralized repositioning strategy, in which drivers comply with reposition instructions provided by the platform, we examine the effects of the decentralized strategy on service rate, passengers' waiting time and drivers' net income. We compare the reposition strategies under different supply and demand levels and different demand spatial distribution dispersion rates. We also explore the influence of platform information on drivers' decision-making process.

We found that decentralized repositioning strategies have better performance in reducing waiting time, while the centralized strategy is better at increasing driver income and service rate. We also found that when platform information is accessible, the system has the best performance when 20% to 60% proportion of drivers utilize platform information when making decisions.

Keywords: Ride-sourcing; Repositioning; Agent Based Model; Transport Network Companies

Preface

This project is the result of graduation work for master of science civil engineering program. This project symbolizes an end of not only my master program, but also my study career temporarily. Lasting for more than 8 months, during the whole process, I spent great effort in coding and the simulation part, and I found that although I experienced many failures, without these attempts I could not have achieved such progress. Also, I am very proud that I kept trying my best, and overcame all the difficulties that made me nearly lost my hope for finishing my thesis.

I would like to express my gratitude to my thesis committee Oded, Shadi and Arjan. During every meeting they provided very helpful suggestions and detailed feedback. Their rigorous academic attitude impressed me deeply, and I learned a lot on how to conduct a research during the whole process. My daily supervisor Arjan not only put a lot of effort on helping me advancing my progress and providing feedback during every discussion, but also encouraged me throughout the whole process, especially when I was having a hard time working on my thesis while recovering from sadness.

Finally, I would like to thank my friends and my family. It takes a very long time to conduct a research, and it requires perseverance and determination to make it all through. My friends and my family kept supporting me whenever I was stuck on progress or whenever I was upset.

Kairui
Delft, October 2023

Contents

List of Figures

List of Tables

1
Introduction

1.1. Background

With the advanced development of digital technology, ride-sourcing systems have become a mainstream mode among traffic users. According to Uber annual report (2022), in spite of COVID situation, its gross booking grew 19% from 2021 to 2022, and its annual revenue expanded 49% year-over-year. Different from traditional taxi services, passengers are no longer required to wait and search for available taxis along the roadside. Instead, they utilize the app and connect with the nearest vehicles not occupied, saving passengers' time and energy. This would be efficient for passengers in remote areas, since they could get allocated even if no vacant taxis are searching nearby. Similarly, drivers of ride-sourcing vehicles have a better knowledge of nearby requests and head directly to the pick-up point, reducing the potential for accidents due to distraction from looking for passengers or sudden lane changes.

Ride-sourcing services connect drivers and passengers through a platform to enable intermediate interaction, thus could be defined as a two-sided market place. Different from traditional one-sided markets, the supplier side in two-side markets also possesses the flexibility to determine when and where they would like to work. They pay commission fee to the platform for using the platform service, yet have to undertake all the generated costs. This sometimes leads to inadequate income for ride-sourcing drivers, which could even be below the local minimum wage (de Ruijter et al., 2022), when insufficient orders have been completed. The same-side negative effect (Fan, 2022) suggests that competition on the suppliers' side results in a loss of drivers. To stimulate drivers' motivation to work for ride-sourcing platforms, it is essential for the platforms to come up with operational strategies to help drivers acquire more orders while saving up operational costs.

In contrast with traditional taxis, ride-sourcing nowadays enables drivers to connect with new customers from mobile devices. However, they still need to make their own decisions on whether remaining idle at the drop-off point before getting matched to the next consumer, or repositioning to a spot where they expect a higher probability of getting matched or a higher income. Sometimes when too few vehicles are available, drivers could sometimes get matched with distant orders. This leads to high operational costs for drivers and long waiting times. Based on a case study in Austin, it is found that for every 100 miles of delivering passengers, drivers have to travel additionally up to 126 miles empty (Tengilimoglu & Wadud, 2022). This phenomenon is the result of an unbalanced demand and supply (Qian et al., 2022), and should be restored by repositioning vehicles in under-served areas. Moreover, when passengers loses patience, the order could be cancelled, wasting drivers' time and cost, and adding congestion (Tengilimoglu & Wadud, 2022).

Due to ride-sourcing drivers' lack of knowledge and lack of incentive to go to remote areas, individual decisions of the drivers may not lead to a system optimum. Thus, it is essential to optimize the reposition strategy and improve its effectiveness. To begin with, it is necessary to examine how repositioning contributes to the system performance. Then, we search for a policy for the platform that better directs drivers to new pick-up points, saving costs and time for both demand and supplier side.

1.2. Literature Review

1.2.1. Repositioning Behaviors

Taxis usually reposition themselves based on personal preferences, which is highly correlated to the drivers' knowledge of passenger demand distribution. Usually expected journey time, cross-zonal cost, waiting time and toll (Sirisoma et al., 2010; Szeto et al., 2013) are significant external factors, while personal experience (Sirisoma et al., 2010) and information availability (Yang & Wong, 1998) could also affect behavior. Due to habitual zone effects, taxi drivers may also incline to search in familiar areas. Besides, drivers' repositioning behavior is never consistent and varies across hourly time periods (Szeto et al., 2013). For instance, at midnight drivers may travel to places in higher demand, while in rush hours they may consider other aspects such as congestion.

As ride-sourcing service becomes dominant, due to the easiness of getting matched and aversion to travel costs, more drivers may have a stronger bias to remain idle motionlessly (Urata et al., 2021). To help increase efficiency, platforms may provide incentives to stimulate drivers keep moving. In special cases, such as when there is an imbalance between demand and supply (Urata et al., 2021), drivers are likely to behave more actively, thus the platform could pay less effort on providing incentives.

Compared to former years, the higher information diffusion rate and information accessibility nowadays enables ride-sourcing drivers to speed up their learning process of demand distribution, and helps all drivers gain more accurate real-time information. Additional information such as the distribution of pre-booked rides, driving conditions in different areas (Ashkrof et al., 2022) and parking availability (Ashkrof et al., 2023) has significant effects on decision-making for drivers. Additionally, the information also enables platforms to process historical and real-time data, and come up with policies to help promote the system efficiency by guiding drivers to reposition themselves. Policies including surge pricing, extra bonus for heading for high-demand areas stimulate drivers to follow instructions from the platform (Ashkrof et al., 2023) when drivers highly trust it.

1.2.2. Centralized and Decentralized Repositioning Process

To improve the effectiveness of ride-sourcing drivers' repositioning process, many studies propose a **Centralized Method** to balance the demand and supply in a studied region. A centralized method here refers to a reposition strategy determined by the ride-sourcing platform. Instructions for reposition routes will be provided by the platform and drivers never make such decisions themselves. Currently, most studies leverage reinforcement learning and provide instructions to all drivers in the system (Jiao et al., 2021; Lin et al., 2018; Qian et al., 2022; Shou et al., 2020; Verma et al., 2017; Yu & Hu, 2022). A Markov reward-driven process is applied to simulate the whole searching process, and states are defined as arriving at a specific node on the map. The chain pointing to the highest driver revenue or highest driver equity (Y. Lin et al., 2020) is selected as the candidate route. The reinforcement learning method has been applied to observe various sizes of groups of drivers, ranging from single agents (Shou et al., 2020; Verma et al., 2017) and to groups of drivers (Jiao et al., 2021). However, this method relies on big historical data, and the high computation complexity could reduce its timeliness.

In spite of the agent-based strategy, Model Predictive Control (MPC) has also been an important method in vehicle relocation determination (Iglesias et al., 2018; Riley et al., 2020; Valadkhani & Ramezani, 2023). Based on the predicted zone-to-zone demand and observed state, an MPC model is leveraged to compute the vehicle rebalancing strategy. Usually the demand forecast is achieved by applying machine learning methods (Iglesias et al., 2018; Riley et al., 2020), and the rebalancing strategy is realized by optimization techniques for platform revenue. This approach assumes that, in order to reach a global optimization, all requests could be fulfilled, and all drivers will follow the command.

Yet, as a two-sided market, drivers are acting as agents making decisions individually, and many can not choose to comply with the instructions from the platform. Such behaviour can be related to their education level, employment status or road conditions (Ashkrof et al., 2023), current location or time (Urata et al., 2021). In addition, sometimes abrupt factors or personal preferences may also prompt them to opt out from the suggested route. This violates the optimization scheme, and fails to rebalance demand and supply. Usually, an acceptance rate is applied when simulating the centralized dispatch methods (Jiao et al., 2021; Lin et al., 2018; Qian et al., 2022; Shou et al., 2020). Dong et al. (2022) seeks mixed equilibrium with dual-sourcing strategy, including both contracted drivers and freelancing drivers. Some studies may also take policy scenarios into account, such as surge pricing or high-demand bonus (Ashkrof et al., 2022), to ensure a higher acceptance rate.

In a decentralized method, drivers make repositioning decisions based on their own preferences, whether which place they would reposition themselves to, and which route they would take. Choice modeling is a common approach to model the repositioning searching process based on revealed preference. Szeto et al. (2013) classified Hong Kong into 18 zones and calculated the rate of return (ROR) for each zone. Eventually drivers would select repositioning to the zone with the highest ROR. Knobbe (2022) and Qu et al. (2014) leverage potential profit value as an indicator to quantify the level of attraction of respective nodes, which is really similar to the definition of utility. The expected travel time and idle driving distance were key factors that were considered.

1.2.3. Ride-Sourcing Simulators

Insofar, several agent-based ride-sourcing simulators have been developed to help examine different policies in ride-sourcing platforms. Ruch et al. (2018) developed a simulator based on MatSIM, and tested different zonal rebalancing strategies on Automatic Vehicles. Nahmias-Biran et al. (2019) introduced a framework within SimMobility to model mobility on-demand, and behavioral models were utilized to represent drivers' decision-making process. Yet the two studies above are all implemented in a single day, and decisions such as order cancellation were not modelled in such simulators. Some simulators are modelled in a more microscopic level. Feng et al. (2023) introduced a multi-functional simulator that considers strategic interactions between drivers and passengers, and reinforcement learning is adopted to consider the current and future gain. Yao and Bekhor (2022) also implement complex interactions, yet the repositioning part is omitted. Kucharski and Cats (2022) constructed an agent-based simulator MaaSSim and supports researchers to reproduce novel phenomena in two-sided platforms. Other simulators such as those developed by Djavadian and Chow (2016),Lin et al. (2018), Ferreira and d'Orey (2014) also simulates the ride-sourcing or taxi system, testing policies in pricing strategies, repositioning methods and cruising behaviors, respectively.

Author Year	Model	Optimization Approach	Objective	Granularity Level
Lin et al. (2018)	Reinforcement Learning	Centralized	Max Driver Revenue	Hexagonal Grid
Qian et al. (2022)	Reinforcement Learning	Centralized	Max Reward for Reposition	Hexagonal Grid
Shou et al. (2020)	Reinforcement Learning	Centralized	Max Driver Revenue	Hexagonal Grid
Yu and Hu (2022)	Reinforcement Learning	Centralized	Max Driver Revenue	Grid/Vertices/Nodes
Lei et al. (2020)	Recurrent Neural Networks	Centralized	Min Pax Waiting Time	Grid
Dong et al. (2022)	Mixed-Equilibrium	Centralized	Max Driver Revenue	Hexagonal Grid
Iglesias et al. (2018)	MPC	Centralized	Min Reposition Costs	Regions
Riley et al. (2020)	MPC	Centralized	Min Reposition Costs	Regions
Valadkhani and Ramezani (2023)	MPC	Centralized	Min Driver Idle Time	Regions
Szeto et al. (2013)	Agent-based	Decentralized	Max Rate of Return	Regions
Qu et al. (2014)	Agent-based	Decentralized	Max Driver Revenue	Nodes
Knobbe (2022)	Agent-based	Decentralized	Increase Driver Revenue	Nodes
This study	Agent-based	Decentralized	Improve Pax Waiting Time Driver Income & Service Rate	Hexagonal Grid

Table 1.1: Relevant Studies regarding Ride-Sourcing Reposition Optimization

1.3. Problem Definition and Research Gap

Many studies have accomplished research on optimizing idle drivers' repositioning behaviors to maximize the sum of driver revenue (Dong et al., 2022; Lin et al., 2018; Qu et al., 2014; Shou et al., 2020; Yu & Hu, 2022), minimize overall driver reposition costs (Iglesias et al., 2018; Riley et al., 2020), minimize overall driver idle time (Valadkhani & Ramezani, 2023) or minimize overall passenger waiting time (Lei et al., 2020). Inefficient repositioning would lead to greater operational costs for drivers and longer pick-up distances after the driver and passenger get matched, resulting in drivers' income reduction and passengers' waiting time increase. Thus, it is important to study on the effects of repositioning behaviors to improve the two aspects mentioned above as well as increasing the service rate in busy situations.

In this study, we study the effect of a proposed reposition behavior in a ride-sourcing system on service rate, passengers' waiting time and drivers' revenue. As discussed in the previous section, most studies focus on providing repositioning suggestions for drivers based on historical data, in order to acquire a system optimal solutions. The most common method is to train a reinforcement learning model, yet training such a model in real ride-sourcing network would require great computation costs and instability to the system, since reinforcement learning algorithms demand a training process with thousands or millions of trail and errors in an environment (Feng et al., 2023). To provide better insights, latest conditions need to be updated frequently to the training set, resulting in very high efforts. It would be essential to find a more economical way to reduce the computation effort while not losing too much details.

Also, most previous studies requires a ride-sourcing platform to give repositioning commands to the drivers, and drivers do not make the repositioning decisions themselves. This is defined as **Centralized Repositioning** in this thesis. Yet this requires a high acceptance rate from drivers, which is only realistic within a system consisting of only self-driving vehicles. However, the vehicle fleet of ride-sourcing platforms is composed of conventional vehicles nowadays, and drivers who do not trust the platform or who are very experienced could make repositioning decisions by themselves. Moreover, typically the platform does not provide repositioning information, and drivers make decisions themselves. These phenomena are called **Decentralized Repositioning**. Although an acceptance rate is often considered in the centralized repositioning models, if the discrepancy of acceptance rate between expected value and realistic value is too high, the reposition recommendations could become ineffective. Several studies have also adopted decentralized repositioning approaches (Knobbe, 2022; Qu et al., 2014; Szeto et al., 2013), but on the one hand, many would assume that destinations drivers select from are very large territories but not specific points, and ignores the route and detour during the repositioning process, and it is more reasonable to model the repositioning process incrementally, which happens step-by-step and the choice for the next destination would be nearby. On the other hand, many studies focus on conventional taxis, and there is no matching algorithm between drivers' and passengers' digital devices considered, which is not applicable for current ride-sourcing vehicles.

Finally, as mentioned, while we model the repositioning process incrementally, drivers also need to avoid myopic decisions. Jiao et al. (2021) indicates that, when making decisions for repositioning destinations, where drivers head for is affected by the demand and road conditions along the way. Conversely, it is also important to consider the features of further areas while modelling drivers relocating to somewhere nearby. This is mostly considered in centralized repositioning strategies, since Reinforcement Learning and Markov Decision Process (MDP) allows to optimize repositioning behaviors considering the uncertainty in the near future. However, in decentralized repositioning models, most studies regard the repositioning destinations as isolated points.

Thus, in this study, we model drivers achieve the repositioning process based on their own decisions, i.e. a **Decentralized Repositioning** model. The entire studied area is divided into homogeneous hexagons. Drivers make decisions on whether moving to a neighboring hexagonal grid, or remain idle in the current hexagon. To make such decisions, they evaluate the **Properties** of each hexagon, including the total time they have waited and the probability of getting assigned there, which will be quantified as a comprehensive **Score**. Drivers are assumed to move to the hexagon with the highest score. Although the zonal granularity level is a hexagonal grid, the origins and destinations of requests will be specified as coordinates inside the hexagons, which we define as **Nodes**. Finally, a day-to-day process will be simulated, and a **Learning Set** regarding the properties will be established for every driver, to simulate drivers' learning process in the system.

The main contributions of this work include the following items:

- Examine the **System Performance**, including passengers' waiting time, drivers' net income and overall service rate, when drivers are assumed to make repositioning decisions based on a scoring method, compared with a Centralized Repositioning Model and Random Walk.
- Examine the impact of different supply and demand levels, and demand distribution patterns on the system performance .
- Examine the system performance of this decentralized repositioning strategy when platform provides necessary information directly, and test the impact of different acceptance rates.

1.4. Research Questions

To fill the research gap, we apply an agent-based decentralized repositioning model to simulate day-to-day learning among ride-sourcing drivers, and examine the evolution of system performance. We first divide the whole study area into a hexagonal grid, and while repositioning drivers can only move to a neighboring hexagon each time, which is defined as a **Step**. Total waiting time and requests achieved in each hexagon will be learned by drivers throughout the day-to-day process, forming up a learning set for each driver. Simultaneously, a scoring model based on the learning set is adopted, and a score is calculated for each step while repositioning through the hexagon, to model the decision-making process of drivers. In addition, numerous successive hexagons a driver intends to travel on while repositioning create a **Path**, and different paths will be evaluated to prevent myopic solutions. Drivers reposition until they get matched with a new request or when the end shift time everyday is exceeded. We then acquire the system performance after a defined number of days. **Driver Income** in system performance refers to the net income, which is the subtraction of collected fare and fuel cost and commission fee collected by the platform. Thus, the research question is formulated as following:

Compared to centralized repositioning strategies, how are service rate, passenger waiting time and driver income affected by decentralized repositioning strategies in a ride-sourcing system depending on demand and supply levels and spatial patterns, and to what extent can it be improved with platform providing necessary information?

Afterwards, different scenarios evaluate the effectiveness of the decentralized repositioning strategy. To begin with, we will study under which supply and demand level, the underlying behavior would affect most the system performance. Then, we are interested in the impact of this decentralized repositioning strategy on different origin or destination distribution patterns, namely different dispersion rates and whether requests scattered around a single center or multi-centers. Finally, we allow platforms to make properties of each hexagon available to drivers, including the average total waiting time and average number of times matched reported by every driver. The results provide implications for travelers, drivers, platforms and transport authorities. Several sub-questions will be answered to help accomplish this research:

1. In comparison with centralized repositioning methods, how does decentralized repositioning strategy affect passenger waiting time and driver income;
2. How does the decentralized repositioning strategies perform under different supply and demand levels;
3. How does the decentralized repositioning strategy perform under different origin and destination spatial distribution patterns;
4. To what extent could information on average waiting time and income of different hexagons provided by the platform improve system performance?

1.5. Thesis Structure

In this thesis, we first introduce the background and related works in section 1, and the research gap and corresponding research questions were developed. Then, in the next section, the methodology is described in details, providing a motivation for the reason why we applied agent-based simulator MaaSSim and a description over its details. Then, the Hexagonal grid-based Scoring Strategy and learning model are explained. In section 3, we provide insight into the centralized repositioning strategy we adopted, the experiment and scenario design, and expected KPIs. In the following section, results and figures are discussed. Finally, we answer the research questions, and implications and future work are presented in section 5.

2 Methodology

In a ride-sourcing operational system, we make the assumption that drivers autonomously determine their repositioning strategies, and the decision-making process is entirely based on the drivers' experience. Thus, we propose a day-to-day model with drivers continuously updating their knowledge throughout the whole process. Each driver is equipped with a learning set, primarily composed of information regarding waiting times and the number of completed ride requests in each zones. The learning set is continuously updated in real-time, ensuring that drivers have access to the most current information while making decisions. After the end of each day, the learning set will persist into the next day.

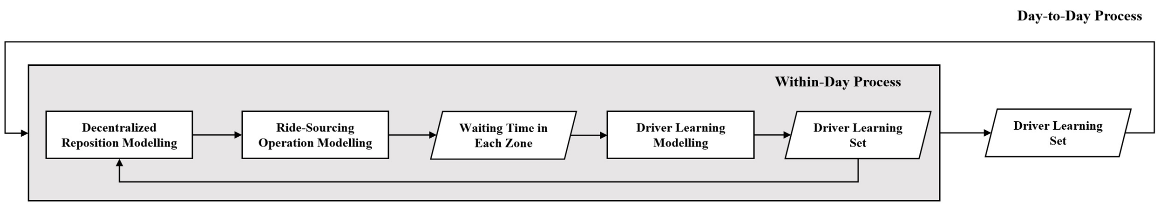


Figure 2.1: Conceptual Framework of Proposed Methodology

In this section, we first start with describing the choice of operation process simulator in 2.1. Then, the two models embed in within-day process, namely driver learning model and the decentralized reposition model, are discussed in sections 2.2.2 and 2.2.3, respectively.

2.1. Ride-Sourcing Operation Process Simulator

In this section, we first compare the candidate simulators and the reason we would select MaaSSim as our simulator. The overview of simulators applied in similar studies is already described in section 1.2.3. Then, MaaSSim will be introduced in details, including the hexagonal grid module we embed, and the operation process defined in the simulator.

Author Year	Reposition Decisions	Passenger Cancellation	Matching	No Historical Data
Feng et al. (2023)	√	√	√	×
Yao and Bekhor (2022)	×	√	√	√
Nahmias-Biran et al. (2019)	√	×	√	√
Djavadian and Chow (2016)	×	×	×	√
Ferreira and d'Orey (2014)	√	×	×	√
Lin et al. (2018)	√	×	√	×
Ruch et al. (2018)	√	×	√	√
Kucharski and Cats (2022)	√	√	√	√

Table 2.1: Comparison between Existing Ride-Sourcing Operation Simulators

2.1.1. Candidate Simulators

To begin with, the simulator should be able to embed a repositioning and self-learning model, and the actions intended by the drivers should be able to be implemented in the simulator. On passengers side, they may cancel the order if they lose patience, thus complex interactions between drivers and passengers must be able to be modelled. However, according to 1.2.3, many simulators failed to consider detailed within-day decision-making process of whether drivers or passengers (Ferreira & d'Orey, 2014; Lin et al., 2018; Ruch et al., 2018; Yao & Bekhor, 2022). Also, there should be a two-sided matching module in the simulator, which matches the passenger and platform side when a request is generated, and some simulators do not include this (Djavadian & Chow, 2016; Ferreira & d'Orey, 2014). The two factors above are essential in providing accurate results correspondent to real-world situations, or the effect of repositioning could be hardly persuasive. Finally, since we are implementing a decentralized repositioning method and drivers are expected to learn from their own experiences, we seek a simulator that does not require historical data input, thus simulators adopting Reinforcement Learning cannot be utilized (Feng et al., 2023; Lin et al., 2018). The comparison between the simulators of the factors above could be found in table 2.1.

Thus, we find MaaSSim to be the best simulator that fits all of our needs. MaaSSim is a Open-source python library, and is flexible enough for users to embed self-defined functions. Other open-source simulators such as MATSim, SimMobility might be more widely applied, but the focus of our study is the system performance of ride-sourcing system, and MaaSSim might be most convenient to reproduce the drivers' repositioning behaviors in a two-sided platform.

Moreover, MaaSSim simulates activities in street and node level, and the reposition and learning model we proposed requires information at the zonal level. Thus, MaaSSim is constructed on a more detailed network level than the models require. On the one hand, driver and passenger activities are simulated at a more precise level, and the waiting time, driving time would be more close to reality. On the other hand, we could simply reflect coordinates of nodes to specific hexagons, to fulfill the requirements of input of the reposition and learning model.

2.1.2. MaaSSim

MaaSSim is an agent-based simulator of on-demand two-sided mobility service, which models the behavior of and interaction between two types of agents, namely drivers and passengers (Kucharski & Cats, 2022). Passengers generate requests, and drivers supply service to transport passengers to their destination. Given a supply level and demand distribution, the platform acts as an intermediate agent and matches the two sides. The simulator is able to simulate the whole process of ride-sourcing operation throughout a day, including the decision-making process, matching, waiting, pick-up, riding and drop-off. As a two-sided platform, MaaSSim allows drivers and passengers both making individual decisions. Passengers opt out from the system when waiting time exceeds a threshold, while drivers determine themselves whether and where to relocate themselves. The heterogeneity among drivers' learning process can all be easily represented through different user-defined modules. Thus, we find this an appropriate agent-based simulator to examine the system performance of our defined scenarios.

MaaSSim has been applied multiple times in ride-sourcing studies so far. Based on the defined day-to-day operation process, self-defined modules could be embedded to the model, to test system performance from various perspectives. Currently many studies have been accomplished on drivers' side. Kucharski and Cats (2022) examined the system performance when enabling drivers' learning process, and conclude that learning process would result in system stabilisation. de Ruijter et al. (2022) studied on the evolution of labor supply. Knobbe (2022) also focused on testing the efficiency of a defined repositioning strategy by adding a repositioning model.

2.1.3. Model Description

Our application of MaaSSim mainly consists of four procedures, namely input, initialization, simulation and output. Contents within each procedure are listed in figure 2.2.

As an agent-based model, MaaSSim requires detailed information of the network, demand and supply. The network information in MaaSSim is generated by OpenSteetMap as default. By inputting the name of our studied area, a map in graphml format could be generated directly. Origins and destinations correspond to nodes in the generated map, and we utilize a skim matrix to define the distances between the nodes. Thus, the inputs of the model include the generated map and skim matrix, the location and time of supply and demand, and other necessary parameters (Table 3.4).

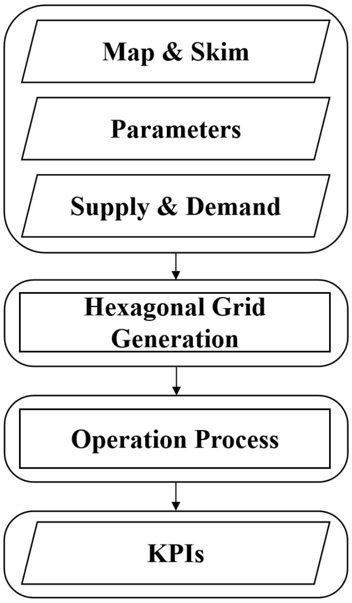


Figure 2.2: Model Workflow

2.1.4. Hexagonal Grid Generation

In MaaSSim, the map of a studied area is generated into graphml format, which is acquired from OpenStreetMap. The map is initialized with nodes and edges in the studied city area. Edges are highly consistent to the direction and location of roads in reality. A skim matrix presents the shortest distance between pairs of nodes. Since congestion is not considered in MaaSSim, the travel distance between two nodes always comply with the skim matrix under any circumstances.

In MaaSSim, although nodes and edges are already initialized based on the city map, the substantial number of nodes adds too much burden to the computation process. Thus, we would like to simplify it by grouping nodes in a hexagon grid system (Jiao et al., 2021; Lin et al., 2018; Riley et al., 2020; Shou et al., 2020). In a hexagon grid system, the euclidean distances between central points are identical, thus it is a great approximation of circles. Therefore, in our model, demand and supply all derive from cells, and drivers make decisions based on the attractiveness of each cell, in which the interior is regarded as homogeneous (Lin et al., 2018).

2.1.5. Within-Day Operation Process

We generate the hexagonal grid by applying package H3. H3 is an open source geospatial indexing system, and could be incorporated into graphml to generate hexagonal grid on a new layer. H3 provides 16 resolution levels, each resolution level representing a different level of granularity of the grid. At level 0, the whole earth is represented as a single grid, while as the level ascends, the resolution gets much smaller. The H3 indexing system assigns a unique 64-bit index to every hexagon in the grid, thus each node belongs to a corresponding index, and could be enquired by inputting the coordinates. To avoid adding too much burden to computation process, while ensuring homogeneity in a grid, we need to make a trade-off while determining the resolution level of hexagon. We hereby define the aperture size as 8. The hexagonal grid is presented as figure 3.1.

The operational process of drivers could be described as a loop (figure 2.3). At the start of a day or after dropping off a passenger at a destination, a driver would become idle. The vacant vehicle will be added to a queue waiting for new requests. Then, there would be three situations that drivers could encounter, whether getting matched with a new order immediately, remaining idle at the same place until getting assigned, or repositioning the vehicle to a new position and wait until getting matched. After getting assigned, the driver may head for the pick-up point, pick up the passenger, drive to the destination and then drop off. It could also occur that drivers get assigned during the repositioning process. Then the reposition process is terminated immediately. After dropping off, drivers determine themselves to end the day, or return back to the start of the loop. In MaaSSim, it is also possible that drivers decline a request when not satisfying. Then the driver may remain in the queue, waiting for the next chance of getting matched. However, this function is not adopted in our model, according to the assumption in chapter 2.2.1.

The whole process is realized by SimPy in MaaSSim , which is a process-based simulation framework. Discrete events are stored as *Events* with their simulation time, priority and id. An *Environment* consists of the discrete event list, and keeps track on the simulation time currently. When an *Event* is triggered, the status of an agent may move to the next discrete event on the list, and the defined event time will also be added to the simulation time. In MaaSSim, the whole ride-sourcing process is classified into 16 events, each allocated with an ID. The interactions between the 16 events is illustrated as figure 2.3. The precision of simulation in MaaSSim is 1 second.

In SimPy, it is also possible to wait for a new event after one is accomplished, or an event could also get interrupted when a *Timeout* is simultaneously reached. This highly fulfills the needs of the ride-sourcing system, since drivers may wait for requests, or may interrupt repositioning process and head for the pick-up location.

The time of starting time and ending time every day will be predefined before the simulation experiment, which are denoted as *START_TIME* and *END_SHIFT* in figure2.3. Time of requests will also be input or generated previously. As for the duration time of respective events, the length of drop-off and pick-up time are also previously defined. The travel time and repositioning time is calculated as the quotient of euclidean distance and travel distance, ignoring the effects of congestion. . In 2022, Amsterdam ranked as the 257th most congested city in the world, and suffers the least traffic jams among all European capital cities (tomtom traffic index, 2022), with a delay under 5 minutes per 10 kilometers. Thus we regard congestion has little impact on travel time in Amsterdam and ignore its effects.

Passengers' operation process only takes up a part in the loop in figure 2.3. They

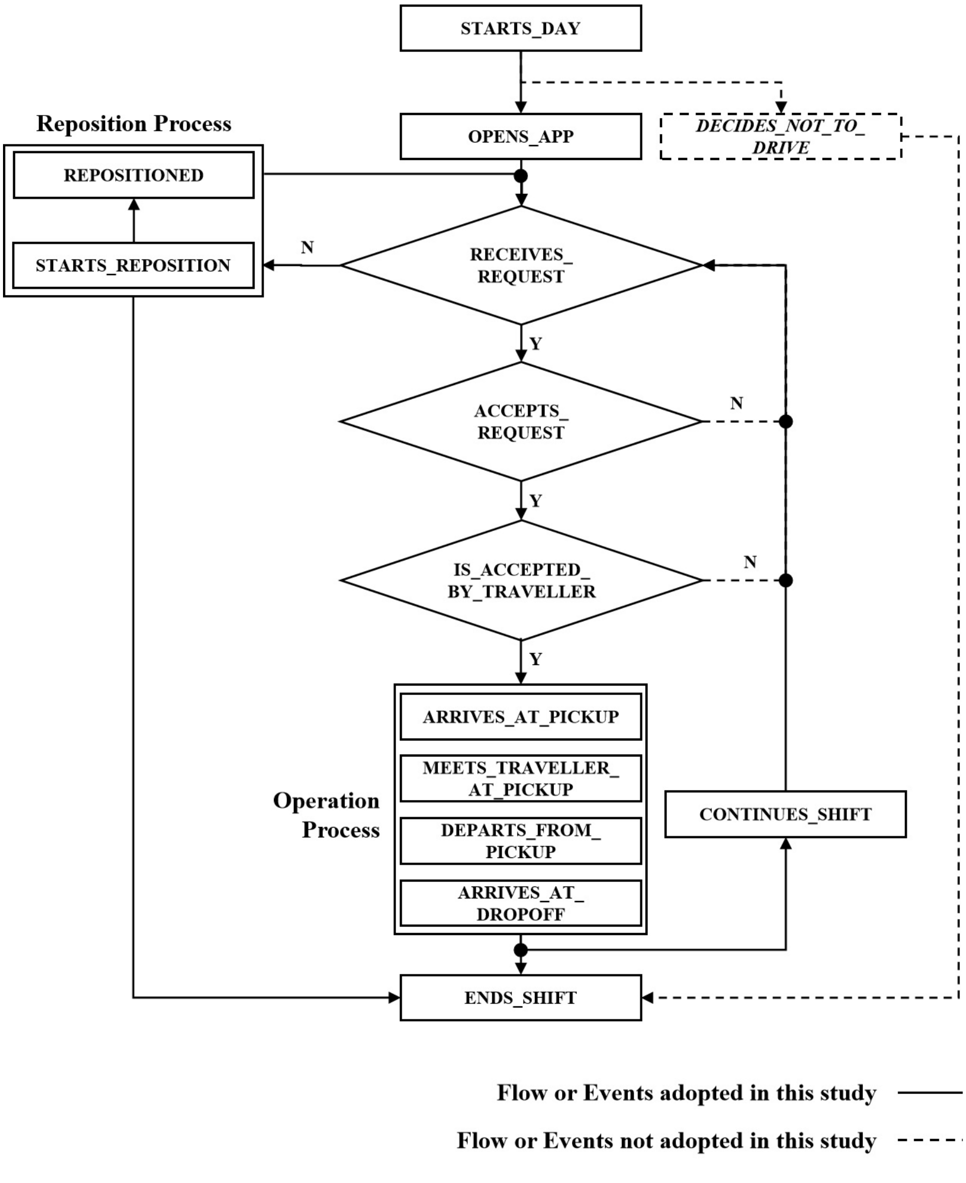


Figure 2.3: Interaction between Within-Day Driver Discrete Events in MaaSSim

emerge when requests are generated, and disappear after the drop off. After the passenger gets matched with a driver, passengers still have the right to repent if he or she loses patience. Otherwise, the passenger waits until the vehicle arrives, gets picked-up, travels in the vehicle and arrives at the destination. Thus, on drivers' and platforms' perspective, reducing the travel cost before arriving at pick-up point, and keeping passengers from losing patience are two main issues that they could optimize. We hereby focus on the repositioning behaviors, to help drivers get greater income, while enhancing passengers' travel experience.

2.2. Decentralized Repositioning Model

We hereby apply an agent-based model to simulate the repositioning process of drivers in a ride-sourcing system. The decision-making process is accomplished by each driver independently, which is defined as a decentralized process. The decentralized model includes learning process about the probability of getting assigned for the next $T^{\max}$ minutes (**PGA**$_{v,k,r}$) of vehicle v in a neighboring hexagon k during reposition process r, and a scoring process regarding the maximum expected revenue for choosing to move towards a hexagon.

The whole studied area is divided into a hexagonal grid, and each hexagon is assigned a **PGA**$_{v,k,r}$ based on a driver's own experience. In the beginning, drivers are assumed to possess no knowledge of the entire city, and we thus apply **Random Walk Strategy** to simulate the repositioning process. There could be seven alternatives for a driver to consider, including moving to hexagons in six directions plus staying in the current grid (Figure 2.4). Drivers randomly make a choice for the next time step.

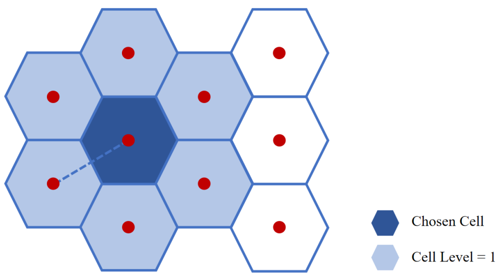


Figure 2.4: Seven Choices for the Next Step in Hexagonal Grid System

After 10 days, we assume that drivers have a better knowledge of the demand distribution in Amsterdam area, and start to make repositioning decisions based on their own knowledge, which denotes the start of **Decentralized Repositioning Process**. They still move through the hexagonal grid step-by-step while repositioning, and evaluate the adjacent and current hexagons every time before they move. The drivers are supposed to forward to the hexagon with the highest score, which is calculated based on their learning from previous day-to-day experiences, including the waiting time in different hexagons, and expected revenue per order.

To prevent drivers making myopic decisions, while calculating scores of adjacent cells, we also take further steps into account. In spite of the adjacent hexagon, further steps will be simultaneously examined, and the maximum number of steps is denoted by $s^{\max}$. The calculation method of the overall score will be further explained in 2.2.3. The $s^{\max}$ hexagons formulate a path, and the driver is expected to enter the path with the highest overall score. Thus, every time before the driver moves, he or she evaluates all feasible paths around, and moves to the hexagon that is via the optimal path. It is also possible that, the current hexagon appears $s^{\max}$ time in the path, then the driver would remain idle in the current hexagon, without repositioning.

In this subsection, we first list the notation variables (Table 2.2 and 2.3) that will be utilized in the decentralized model. Then, the assumptions are presented. Finally, we will give a detailed explanation of the scoring method.

2.2.1. Assumptions

1. Drivers make repositioning decisions after a fixed interval.
2. Origin and destination distribution of requests remain identical every day.
3. Drivers start every day at a random point in the road network, and there is no depot.
4. Drivers drive every day until the defined end shift time is reached.
5. Drivers do not decline requests, and passengers accept the first driver that is matched with them.
6. A fixed time limit for passenger waiting time will be set, and when waiting time exceeds this time limit passengers lose patience and cancel the order.

2.2.2. Learning Model

As described, the drivers learn an indicator name $\mathrm{PGA}_{v,k,r}$, which considers the total idle time spent in hexagon k for driver v in process r, and the count of passengers they picked up in this hexagon. The demand level and distribution varies throughout a day in different time periods, such as morning peak and evening peak, so drivers should learn the demand pattern of different time periods separately. Thus, the scores of same places in different time periods throughout the day could be different. We assume drivers adopt the same scoring method in different time periods, so to simplify the process, we focus on one time period every day.

After initializing a learning set for every driver consisting of all nodes in the studied area, to begin with, drivers adopt **Random Walk Strategy** in the first 10 days. When drivers decide to relocate themselves, they select a neighboring cell randomly for their next step. After relocating themselves to the central point in each hexagon, drivers wait inside the hexagon until the $T^{\max}$ minute time limit is exceeded, and then they leave immediately if no orders had been assigned. Thus, the waiting time is $T^{\max}$ minutes under this situation. If an event, such as the driver is assigned abruptly or the

Notation	**Description**
v	Index of Driver
d	Index of Request
k	Index of hexagon
r	Index of Reposition Process
k_s	Index of hexagon s steps from hexagon k
p	Path from hexagon k, A sequenced set of hexagon indexes
s	Count of steps from current hexagon
$s^{\max}$	Number of steps considered while drivers evaluating a path
β	Commission rate
μ	Global service rate
$T_{v,k,r}^{\text{wait}}$	Waiting time of driver v for repositioning process r in hexagon k (s)
$T^{\max}$	Maximum duration for every step while repositioning (s)
T_d^{duration}	Duration of trip d (s)
t^c	Current Time
t_v^{endshift}	Time of driver i ending shift
$t_{v,d}^{\text{req}}$	Time of driver v received a request from passenger d
Q_d^{income}	Net income for ride d (€)
Q_d^{fare}	Travel fare for ride d (€)
Q_v^{idle}	Total cost when idle driving by driver v (€)
$Q_{v,d}^{\text{cost}}$	Operational cost of driver v for ride d (€)
Q^{sum}	Total net income for drivers in the platform (€)
$m_{v,k,r}$	Orders that a driver was assigned in hexagon k at time r arriving at k
v_k^{served}	Served passengers in hexagon k
v_k^{all}	Requests generated in hexagon k
H_d	Distance of trip d
$PGAM_{v,k,r}$	Probability of Getting Assigned per Minute for Driver v at process r at hexagon k
$PGA_{v,k,r}$	Probability of Getting Assigned for Driver v at process r at hexagon k
$R_{v,k,r}$	Score of moving to hexagon k at time r visiting there
$R_{v,p,r}(s)$	Score of moving along path p from step s to s_{max}
$Z_{v,r}$	Maximum Score for vehicle v at repositioning process r

Table 2.2: Notations Variables in Decentralized Model

Notation	Description
V	All vehicles, index $v \in V$
D	All requests, index $d \in D$
D_v	All requests accomplished by vehicle v
R	All reposition process, index $r \in R$
P_k	All possible paths from hexagon k
K	All hexagons, index $k \in K$

Table 2.3: Notation Sets in Decentralized Model

driver ends shift of the day, occurs, then the waiting time is the subtraction of event time minus the starting time of this reposition event, presented as 2.1.

$$T^{\text{wait}}_{v,k,r} = \min\{T^{\max}, t^{\text{endshift}}_{v} - t^{c}, t^{\text{req}}_{v,d} - t^{c}\} \quad (v \in V, d \in D, k \in K, t^{\text{req}}_{v,d} > t^{c}) \tag{2.1}$$

Every time when a driver i accomplishes an order, the driver acquires information regarding the total time this driver has already waited in this hexagon, and the count of passengers the driver has picked up in this hexagon. This applies to both the random walk period and the decentralized repositioning period, and we model this process by establishing a learning set. During the learning process, we apply an indicator named Probability of Getting Assigned per Minute (**PGAM**) to define a driver's chance of getting matched with a new order in a hexagon, which is described in equation 2.2. It denotes that, for every minute spent in the hexagon, the probability a driver gets assigned to an order.

$$\text{PGAM}_{v,k,r} = \frac{m_{v,k,r}}{\sum_r T^{\text{wait}}_{v,k,r}} \tag{2.2}$$

where k is the index of the hexagon. $m_{v,k,r}$ represents the count that this driver has been assigned in this hexagon, hence every time the driver gets assigned, 1 is added to this value. $\sum T^{\text{wait}}_{v,k,r}$ represents the total waiting time the driver has spent in this hexagon, and it takes all previous experiences into account. Every time when the $T^{\text{wait}}_{v,k,r}$ time period exceeds, a driver could either get assigned or remain idle. If the driver is not assigned, then the $T^{\text{wait}}_{v,k,r}$ is added to the hexagon where the driver is staying, and the $m_{v,k,r}$ remains unchanged. If assigned to an order successfully, then the waiting time should contribute to the hexagon where the vehicle is currently staying, but the $m_{v,k,r}$ value of the hexagon at the pick-up point should add up 1.

According to the assumptions, the driver stays for a few minutes waiting or cruising until a time threshold for each step ($T^{\max}$) exceeds. Based on PGAM, the Probability of Getting Assigned (**PGA**) in that hexagon could be acquired easily based on probability formulas. This is described in equation 2.3, where $T^{\max}$ denotes the time length *(min)* a vehicle stays in a hexagonal grid.

$$\text{PGA}_{v,k,r} = 1 - (1 - \text{PGAM}_{v,k,r})^{T^{\max}} \tag{2.3}$$

2.2.3. Driver Scoring Model

As previously explained, to avoid myopic decisions, a driver may consider multiple ensuing steps during scoring process. The successive hexagons that a driver decides to travel on from the current location is called a **Path**. Hexagons that could be reached within the **n-th** step are defined as **Step-n Hexagons**. In this study, drivers always examine the next $s^{\max}$ steps before moving. Before making a decision, drivers calculate the scores of all possible **paths** (2.6). Then, they choose to move towards the **path** possessing the highest score, which is regarded as the most optimal **path** among all alternatives. However, this does not mean the driver would follow this path for the next few steps, since the driver only moves one step after making each decision. After each move, the driver evaluate all possible paths again, and sometimes other **paths** could become more attractive than the original path after the reconsideration. For instance, whenever moving to a new hexagon, there could always be some new hexagons included in the evaluation process, and very attractive hexagons could greatly influence the overall score and attract drivers to move along a new path.

To calculate the score of **paths**, we first need to acquire the score of **steps**, which is the expected revenue for this move. The score considers the potential revenue after moving to hexagon k, and the probability of getting this revenue, or the probability of getting assigned. The equation is presented as .

$$R_{v,k,r} = PGA_{v,k,r} \times Q_d^{\text{income}} \quad d \in D_v \tag{2.4}$$

where k is the index of a hexagon, and i is the index of driver. In Q_d^{income}, as equation 2.5 suggests, the potential revenue should be a net income, excluding the operational cost, thus it equals the subtraction of travel fare and operation cost of for trip d ($Q_{v,d}^{\text{cost}}$). β denotes the commission rate, which is subtracted from the travel fare and handed in to the platform.

$$Q_d^{\text{income}} = Q_d^{\text{fare}} \times \beta - Q_{v,d}^{\text{cost}} \quad d \in D_v \tag{2.5}$$

The calculation method of collected travel fare will be further introduced in experiment design part. The operational cost considers mainly the fuel cost while repositioning as well as serving a passenger, regardless of fixed operational costs.

Then, with the score of single steps, the overall score for a **path** could be calculated as following. (2.6). The iteration starts from the step-1 hexagons, i.e. the adjacent hexagons, and terminate at step-$s^{\max}$ hexagons, which are the furthest hexagons to be considered. It represents that in every step,in spite of the expected income in hexagon k, the probability of no matching there should be $1 - \text{PGA}_{v,k,r}$, and under this probability the driver may continue on further steps, thus the expected income for further steps is multiplied with this possibility.

$$R_{v,p,r}(s) = \begin{cases} R_{v,k_s,r} + (1 - PGA_{v,k_s,r}) \times R_{v,p,r}(s+1) & (1 \le s < s^{\max}) \\ R_{v,k_s,r} & (s = s^{\max}) \end{cases}$$
$$s.t.\ k_s \in p, p \subseteq P_k \tag{2.6}$$

To reduce computation efforts, we assume that drivers do not return to a hexagon after leaving it. This could significantly reduce the search space while searching for a maximum path. Scores of the paths originated from this adjacent hexagon will be calculated, and the path with the maximum score is selected as the best option. Then, the driver would reposition to the step-1 hexagon k_1 that is included in the path p. The maximum score $Z_{v,r}$ for vehicle v at repositioning process r is calculated with equation 2.7

$$Z_{v,r} = \max_{p \subseteq P_k} \{R_{v,p,r}(1)\} \tag{2.7}$$

3 Experiment and Scenario Design

3.1. Benchmark Model

In this study, we propose a repositioning strategy based on drivers' own learning experiences. Drivers move across a hexagonal grid, and each driver calculates the score of repositioning to an adjacent hexagon or staying idle in the current one. They always select the alternative with the highest score. Thus, the decision-making process totally depends on drivers' own thoughts, which is independent of platforms' advises.

To better assess the effectiveness of decentralized repositioning, we employ a centralized repositioning model as the benchmark for comparison. When centralized repositioning is applied, drivers fully comply with instructions given by the platform. The platform forecasts the potential time and location of request, and guides idle drivers to these spots before the requests pop up. The orders are assigned to drivers by the platform, and it is also assumed that drivers would never decline orders. The main objective is to minimize the global waiting time and repositioning distance of all vehicles.

3.1.1. Candidate Models

As mentioned in 1.2.2, Reinforcement Learning (RL) and Model Predictive Control (MPC) are two common real-time centralized methods employed in relevant studies. Ride-sourcing operation could be interpreted as a process with transition between discrete states, such as idle, pick-up, carrying, drop-off and repositioning. RL models employ Markov Decision Process (MDP) to simulate the state transition process. After training the model with large amount of historical data, the MDP model provides non-myopic solutions to drivers and guides them on routes with highest possibility of getting assigned. However, this strategy requires large amount of historical data to train the reward function, and the computation burden could be relatively high.

MPC models optimize the future behavior of the system (Valadkhani & Ramezani, 2023). With short-term demand forecast, the controller dispatch idle vehicles to zones with higher potential demand, and after a defined time length it repeats the whole process. Moreover, the forecast process does not require that large scale of historical

Method	Type	Input Data Scale	Computation Effort
A-RTRS	MPC	Low	Moderate
NMPC	MPC	Moderate	Moderate
DROP	RL	High	High
GNN	RL	High	High

Table 3.1: Comparison between Benchmark Model Candidates

Notation	Description
T	All time epochs, index $t \in T$
Z	All hexagonal zones, index $i, j \in Z$
V	All vehicles, index $v \in V$
N	Non-negative Integers

Table 3.2: Notation Sets in Centralized Model

data as RL methods. For instance, it could be fulfilled by auto-regression of time series data (Riley et al., 2020). To avoid providing myopic solutions, the input forecast demand could range over a longer time length, and the controller would come up with solutions for a longer time period in the future.

We hereby selected four models from other studies as our benchmark, including A-RTRS (Riley et al., 2020), NMPC (Valadkhani & Ramezani, 2023), DROP (Qian et al., 2022) and GNN (Yu & Hu, 2022). Table 3.1 suggests that, the MPC approach borrowed from Iglesias et al. (2018) and Riley et al. (2020) requires the lowest computation effort and input data scale among all candidates. Currently numerous MIP solvers have been developed to deal with complex MIP problems with high efficiency. Based on the superiority on these two aspects, we choose A-RTRS to benchmark our model. The notations utilized in the centralized model are presented in table 3.2 and 3.3.

3.1.2. Model Description

A-RTRS could be described as a novel end-to-end framework. This method divides time into time windows with fixed duration l^A, and in each epoch routes of idle vehicles are optimized to priorly arrive at the pick-up point of potential requests or unserved requests. The A-RTRS process includes three steps, which are demand forecast, optimization process and vehicle allocation process.

In MPC methods, the demand forecast is usually fulfilled with machine learning methods (Iglesias et al., 2018; Riley et al., 2020; Valadkhani & Ramezani, 2023). A great amount of data is required to train the learning model. Due to the reason that the spatial and temporal distribution of demand in our model maintains identical throughout the 30 days, we hereby skip the demand forecast process, and make an assumption that the platform has prior knowledge of the exact request data of this day.

The core of MPC methods is to rebalance idle vehicles in the entire area, to make

Notation	Description
i, j	Index of zones
t	Index of time epoch
t_1	Index of maximum time epoch
v	Index of vehicle
l^A	Duration of time epoch
x^p_{ijt}	Requests from zone i to zone j satisfied in epoch t
x^d_{ijt}	Requests from zone i to zone j unsatisfied in epoch t
x^r_{ijt}	Vehicles travelling idle from zone i to zone j in epoch t
y^p_{vjt}	Whether vehicle v is driving with passenger to zone j at time epoch t,boolean
y^r_{vjt}	Whether vehicle v is driving idle to zone j at time epoch t,boolean
τ_{ji}	Number of epochs that is required to travel from zone j to zone i
c^r_{ijt}	Operational cost from zone i to zone j at epoch t (€)
c^d_{ijt}	Penalty of not serving a passenger from zone i to zone j at epoch t (€)
c_{vj}	Operational cost for vehicle v travelling to zone j at time epoch t (€)

Table 3.3: Notation Variables in Centralized Model

the number of supply match with the number of potential request in different zone areas. The objective is to minimize the average waiting time as well as the reposition distance of drivers 3.1. Thus, an assignment optimization problem is formulated and solved throughout time epochs. To achieve this, the authors proposed a Mixed-Integer Programming (MIP) model over time windows $T = \{1, 2, 3...\}$, with a length of l^A respectively. Vehicles only move inside a defined hexagonal grid M.

$$\min \sum_{t=0}^{t_1-1} \sum_{i,j \in Z} c^r_{ijt} x^r_{ijt} + \sum_{t=0}^{t_1-1} \sum_{i,j \in Z} c^d_{ijt} x^d_{ijt} \tag{3.1}$$

subject to constraints

$$x^p_{ijt} + x^d_{ijt} - w_{ijt} = x^d_{ij(t-1)} \tag{3.2}$$

$$\sum_{j \in M} (x^p_{ijt} + x^r_{ijt} - x^p_{ji(t-\tau_{ji})} - x^r_{ji(t-\tau_{ji})}) = 0 \tag{3.3}$$

$$x^p_{ijt}, x^r_{ijt}, x^d_{ijt} \in N, \forall i, j \in M, t \in T \tag{3.4}$$

Equation 3.2 describes the continuity of passengers, which x^p_{ijt} denotes requests from zone i to zone j that were satisfied in epoch t. x^d_{ijt} denotes unfulfilled requests generated in epoch t, and w_{ijt} denotes the requests generated in epoch t. Equation 3.8 presents the continuity constraint of vehicles. Since the vehicles are non-shared vehicles,x^p_{ijt} also denotes the number of occupied vehicles travelling from zone i to zone j. x^r_{ijt} refers to vehicles that are idle, or repositioning. τ_{ji} denotes the number of l_A that is required to travel from zone j to zone i. Thus, $x^p_{ji(t-\tau_{ji})}$ and $x^r_{ji(t-\tau_{ji})}$ denotes the number of occupied and idle vehicles arriving at zone i in time epoch t.

This model provides information on how many orders could be accomplished with the existing vehicle fleet and given time length. However, we cannot know the specific departure or arrival time of each order, since this model only assigns the starting time and ending time of each order in time epochs. We also have no information about which driver is serving for which passenger. To gain more detailed information about waiting time, reposition time and cost of every single agent, we must assign the tasks to each vehicle. This could be fulfilled within the vehicle allocation process.

The vehicle allocation process is also accomplished with an optimization model. The main objective of this model is to minimize the global repositioning distance(Equation 3.5). We use binary variables y^r_{vjt} and y^p_{vjt} to represent if an idle or occupied vehicle v should be moved to zone j in time epoch t, respectively. When the value is 1, the vehicle would be travelling, while when the value is 0 then the vehicle should be going or staying elsewhere.

$$\min \sum_{v \in V} \sum_{j} \sum_{t} c_{vjt} y^r_{vjt} \tag{3.5}$$

subject to constraints

$$\sum_{v \in V} y^r_{vjt} = x^r_{ijt} \forall j \in Z, t \in T \tag{3.6}$$

$$\sum_{v \in V} y^p_{vjt} = x^p_{ijt} \forall j \in Z, t \in T \tag{3.7}$$

$$\sum_{j} y^p_{vjt} + y^r_{vjt} \leq 1 \forall v \in V, t \in T \tag{3.8}$$

$$y^p_{vjt}, y^r_{vjt} \in \{0, 1\} \forall v \in V, j \in Z, t \in T \tag{3.9}$$

constraints 3.6 and 3.7 ensure the number of idle and occupied vehicles travelling to zone j at time epoch t aligns with the number we calculated in the previous optimization model. constraint 3.8 ensures one vehicle is travelling to no more than one destination in each time epoch t. In this procedure, we acquire the trajectory of all vehicles, and orders could be easily assigned to drivers by matching the time epoch, origin and destination. The sequence of tasks of each driver provides explicit information of waiting time and cost of drivers and passengers, and reposition distance of respective vehicles.

3.2. Case Study

In this study, we investigate the case of Amsterdam. Amsterdam's population density will expand from 5,100 residents per square kilometer now to 5,700 by 2030. However, due to limited space, especially in the historic city center, insufficient room could be provided to accommodate bikes or cars. Thus, Amsterdam is converting to collective forms of mobility (Amsterdam.nl, 2018), and ride-sourcing is advantageous in

Figure 3.1: Hexagonal Grid in Amsterdam

increasing vehicle utilization while not occupying much parking space in metro areas. Currently numerous Transport Network Companies (TNC) are operating ride-sourcing services in Amsterdam such as Uber. The resolution level of hexagonal grid is set as level 8, which the distance between hexagon centers would be around 1.5 kilometers.

3.3. Scenario Design

A total of three scenarios are tested in our experiment. In the three scenarios, we assess the impact of demand and supply level, origin and destination distribution patterns, and the influence of platform providing supplementary information to drivers.

3.3.1. Experimental Parameters

The constant time parameters, except for the patience time, are the default values defined in MaaSSim. The values of experimental parameters are displayed in table 3.4.

The interval of repositioning to a new hexagon is set to 10 minutes in our study. This is based on the resolution level of hexagonal grid, since the average speed is 10 m/s and the distance between hexagon centers is around 1.5 kilometers, it takes 1.5 minutes for an idle vehicle to relocate itself. Drivers will stop or cruise inside a hexagon until the 10 minute limit exceeds, and then the driver would leave for the next hexagon.

The simulation time length is set to 4 hours. Distribution patterns can vary throughout the day in different time periods, such as the morning peak, evening peak and off-peak hours. Then, for each driver, the learning of demand distribution in different time periods should also be separated from each other. Thus, the simulation time for every time period should not last for long. We hereby study one time period every day, and

Notation	Name	Value	Unit
t^{req}	Request Time	15	s
$t^{\text{pick-up}}$	Pick-up Time	30	s
$t^{\text{drop-off}}$	Drop-off Time	10	s
t^{trans}	Transaction Time	20	s
t^{patience}	Patience Time	1200	s
t^{sim}	Simulation Time	4	h
t^{repos}	Reposition Interval	5	min
v	Average Speed	10	m/s
Q^{base}	Base Fare	4.25	€
Q^{dist}	Kilometer Fare	1.35	€/km
Q^{time}	Time Fare	0.31	€/min
C^{km}	Kilometer Cost	0.25	€/km
β	Commission Rate	70	%

Table 3.4: Constant Experimental Parameters

the time period endures for 4 hours.

The average driving speed v value is 10 m/s in our experiment, which is also the default value employed in MaaSSim (de Ruijter et al., 2022; Kucharski & Cats, 2022). This value applies both to occupied vehicles and repositioning vehicles.

The fare calculation method for every order is identical to that of Uber's standard in Amsterdam. The fare includes a start value, and is relevant to the travel distance and travel time. A minimum fare tariff is not considered in this model, since we already added a constraint that passengers must travel more than 3 kilometers.

$$Q_d^{\text{fare}} = 4.25 + H_d \times 1.1 + t_d^{\text{duration}} \times 0.31 \tag{3.10}$$

Finally, the average operational cost per kilometer is set as €0.25 /km. Since the fixed cost every month, such as the maintenance fee, cleaning fee, telephone bundles, is irrelevant to our study, we would not consider these fixed cost in net income calculation.

3.3.2. Scenario A: Demand and Supply Levels

We define the ratio of demand and supply as **DSR**. This part of experiment aims to find out the influence of different DSRs. We make the supply level fixed, and examine the influence of different supply levels to the system.

In 2019, according to data from the City of Amsterdam, daily taxi or ride-sourcing rides could count up to 20000 rides. Due to the reason that we are running the model for only 4 hours every day, which is approximately one-forth of the whole service time, and the simulation only takes account of a single TNC platform, we would take account

Label	Supply	Origin Dispersion	Centers	Acceptance
A1	50	-0.001	1	0
A2	63	-0.001	1	0
A3	100	-0.001	1	0
A4	150	-0.001	1	0
B1	100	-0.0003	1	0
B2	100	-0.001	1	0
B3	100	-0.01	1	0
B4	100	-0.001	4	0
B5	100	-0.01	4	0
C1	100	-0.001	1	0
C2	100	-0.001	1	0.2
C3	100	-0.001	1	0.4
C4	100	-0.001	1	0.6
C5	100	-0.001	1	0.8
C6	100	-0.001	1	1

Table 3.5: Input Varied Parameter Values in Scenario A, B and C

of half of the travellers. Then, we set the demand value as 2500. The total number of rides was served by 5,000 drivers in the whole year, according to data from the City of Amsterdam. Considering the mobility of drivers, the average working hours and that our simulation regards a single platform, and also the computation efforts, we set the supply values nV as 50, 63, 100 and 150. The values of input configurations could be found in table 3.5.

3.3.3. Scenario B: Demand Spatial Distribution Patterns

Under different demand distribution patterns, we expect drivers to possess different repositioning behaviors. In this part of experiment, we examine two different distribution patterns, in which passengers are scattered around a single center and multiple centers, which is inspired by the work of Jaime Soza-Parra and Cats (2022). The origin distribution dispersion value of demand could also be varied. Thus, in this scenario we test 5 groups of parameters, with 3 in the single center part and 2 in the multiple center part.

The location of centers is selected based on Amsterdam Density Map (Figure 3.2), and places with the highest density are chosen as centers. When passengers are scattered from only one center, we choose the geographical center of Amsterdam as our center, which is near the most dense part of the map. For scenario alternatives with multiple centers, we would locate the centers in the geographical center of Amsterdam, as well as in Sloterdijk, Zuid and Oost. Input parameters could also be found in table 3.5.

The spatial dispersion of distances from centers follows a negative exponential distribution, which is set as default in MaaSSim. For single-centered distribution pattern,

Figure 3.2: Amsterdam Density Map (Figure Reproduced from Amsterdam.nl, 2018)

we would test three dispersion values $\gamma^o = \{-0.0003, -0.001, -0.01\}$, which corresponds to normal conditions, concentrated conditions and extremely concentrated conditions, respectively. For multi-centered patterns, we would examine cases when $\gamma^o = \{-0.001, -0.01\}$. The value of demand and supply level are set to 2500 and 100 respectively.

3.3.4. Scenario C: Platform Information

The platform is fully aware of travel demand.Different KPIs directly indicate the system performance, and provide implications for strategies of better operation. We hereby assume that platforms make the average waiting time and average income for each hexagon available to all drivers. This is equivalent to enlarging the sample size of all drivers' learning set, which is expected to significantly speed up the learning process, while preventing the influences of outliers. We assume that a proportion of drivers would entirely trust the system, in which all drivers would make their decisions based on the given data. The proportion is defined as **Acceptance Rate**. When acceptance rate is 0, all drivers make decisions based on their own experiences, while when the acceptance rate is 1 all drivers make decisions based on the platform information. We examine 6 values of acceptance rate, which could be found in table 3.5.

3.4. Key Performance Indicators

3.4.1. Average Waiting Time

The output of decentralized repositioning model includes the waiting time of each passenger. Since we assume that the number of drivers and number of requests remain consistent every day, we simply evaluate the performance by comparing the average waiting time of passengers every day.

3.4.2. Net Income

Higher income stimulates more drivers to participate in driving for ride-sourcing platforms, which helps improve the service and attract more consumers. The net income could be calculated by subtracting the gross income minus the operational cost. The gross income depends on the population of travellers and the travel distance of each traveller. According to our assumption, the requests are generated identically every day, and the gross income is supposed to be identical every day if all requests are served. Nevertheless, under different operation behaviors, the number of passengers that loses patience could also make a difference every day, which changes the gross income correspondingly. Therefore, the trend of income could help us examine whether more orders are fulfilled, and whether drivers are spending less on operational costs. In this case, we adopt the sum of net income (3.11) for all drivers as our KPI, to evaluate the overall system performance.

$$Q^{\mathrm{sum}} = \sum_{v,d}(Q_d^{\mathrm{fare}} \times \beta - Q_{v,d}^{\mathrm{cost}}) - \sum_{v} Q_v^{\mathrm{idle}} \quad v \in V, d \in D_v \tag{3.11}$$

3.4.3. Service Rate

Service rate is defined as the ratio of number of served requests and number of all requests (3.12). The service rate is an indicator of the efficiency of a ride-sourcing system. By comparing the overall service rate between different repositioning strategies, we gain knowledge about the ability each strategy is able to handle with.

$$\mu = \frac{\sum_k v_k^{\mathrm{served}}}{\sum_k v_k^{\mathrm{all}}} \quad k \in K \tag{3.12}$$

3.5. Replications

The simulations in every scenario should be replicated several times due to the stochasticity of several simulation components. In the first 10 days of random walk repositioning strategy, drivers choose from 7 alternatives randomly, which is a uniform distribution. The starting position of vehicles and the temporal demand distribution also complies with a uniform distribution. Finally, the spatial demand distribution of pick-up and drop-off points comply with a negative exponential distribution. Thus, each time we simulate the same parameters, we acquire different outcomes. To reduce variation and ensure the consistency, we repeat the simulations several times, and the number of replications is determined by equation 3.13 (Burghout, 2004).

$$R(m) = \left(\frac{S(m) \times t_{m-1,\frac{1-\alpha}{2}}}{X(m) \times \epsilon}\right)^2 \tag{3.13}$$

which $R(m)$ denotes the replication number given the m replications already executed. $S(m)$ and $X(m)$ are the standard deviation and mean of a key performance indicator. ϵ is the allowable percentage error of the mean X(m). $t_{m-1,\frac{1-\alpha}{2}}$ denotes the critical value of t-distribution with $m-1$ degrees of freedom and α level significance.

4
Results

4.1. Reference Scenario

In this section, a reference scenario, which considers decentralized repositioning, is compared with results of centralized model and random walk model. We assume that no platform information is provided to drivers, and they reposition fully based on their own knowledge. The results are shown in table 4.1.

Examining the results of reference scenario, the decentralized model increases the overall service rate in comparison to the random walk state, from 91.6% to 93.9%, which is slightly inferior to that of centralized model, which is 94.6%, and the increase of service rate from random walk is 2.3 pp and 3.0 pp, respectively. Since the objective of centralized model is to reduce the number of passengers that lost patience, and the centralized model assumes the platform know where and when a demand exactly pops up during the whole simulation process, it is not surprising that the centralized model has better performance.

The service rate indicates the number of passengers who don't receive service within the defined time constraint minutes after requesting it. When supply is insufficient, sometimes distant idle drivers may be assigned to requests, causing the pick-up time to exceed what passengers can tolerate. By repositioning efficiently, the average pick-up time for each order is shortened, and drivers are able to complete their orders much sooner. Thus, drivers become idle earlier and are able to start preparing for the next order, possibly resulting in earlier pick-up time for the next order. Thus, in our model, we conclude that the decentralized increases the service rate due to drivers' more efficient reposition behaviors, yet the reposition behavior of decentralized model is less efficient than that of the centralized model.

As for the waiting time, According to table 4.1, the decentralized model performs the best, and centralized state rates at the second, while the random walk performs least optimal. The decentralized and centralized method reduces the average waiting time by 78 and 45 seconds respectively, which is 19.6% and 11.3%. Figure 4.1 is the waiting time evolution plot of one of the replications. It indicates that there is a significant fall of waiting time after the random walk state expires, from averagely 390 seconds

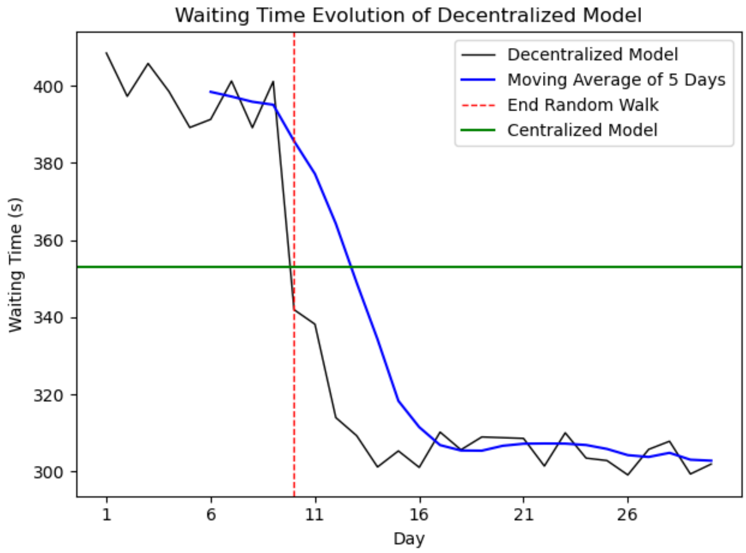
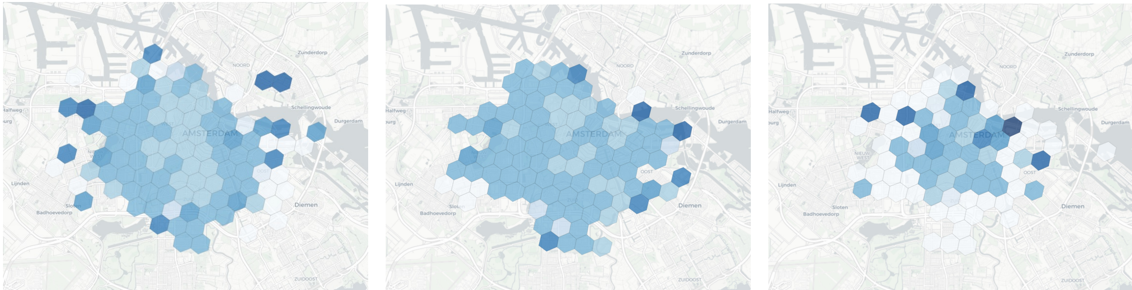


Figure 4.1: Waiting Time Evolution of Decentralized Model in Reference Scenario

to eventually around 310 seconds. After learning about the demand distribution patterns in the studied area, drivers are more aware of the locations of high demand areas. Thus, before generating requests, drivers head for these areas previously, resulting in shorter waiting time. It is also explicit that, although the moving average eventually converges, the waiting time still fluctuates. When too much drivers are biased towards a high demand area, the over-supplied situation results in longer waiting time for drivers, and the PGAM for the corresponding area would descend. Then passengers would seek for other hexagons while repositioning in the following days, resulting in shorter waiting time for drivers in high demand areas, yet longer waiting time for passengers. Thus, there would always be a fluctuation for passengers' waiting time.

It is surprising that centralized repositioning performs less optimal than decentralized repositioning when considering waiting time. Since the optimization of reposition in centralized model emphasizes most on minimizing the repositioning distance and maximizing the service rate, and does not consider the waiting time, it is possible that sometimes the model would recommend drivers to remain idle and wait for the next order nearby, instead of repositioning to somewhere distant. Yet in the decentralized model, vehicles always head for high-demand areas, regardless of reposition cost. Also, it is possible that the centralized method would allow passengers to wait for a vehicle that is currently occupied, but whose destination falls very near to the passenger. The decentralized model, however, would assign requests to the nearest idle driver, thus passengers could experience less waiting time.

The spatial pattern of average waiting time in each hexagon could be found in figure 4.2. The decentralized method exhibits a divergent feature explicitly, which the waiting time is shorter in the center and gets longer as the distance from the center gets higher. As described, the demand is most dense in the center and gets sparser



(a) Decentralized Repositioning (b) Random Walk (c) Centralized Repositioning

Figure 4.2: Spatial Distribution of Average Waiting Time in Amsterdam Area when Utilizing Decentralized, Random Walk and Centralized Repositioning Methods

Indicators	Decentralized	Random Walk	Centralized
Service Rate	93.9%	91.6%	94.6%
Avg. Waiting Time	320	398	353
Avg. Income	70.6	68.7	72.8

Table 4.1: Results of Decentralized Model , Random Walk Model and Centralized Model when Supply = 100, Demand = 2500, γ^o = -0.001, γ^d = -0.001

as distance from the center grows larger. In comparison, the average waiting time of random walk is distributed more uniformly than decentralized and centralized repositioning. Thus, the decentralized method is efficient in reducing the average waiting time in high demand areas. Also, the served area of decentralized method is greater than that of random walk method.

As for the centralized method, surprisingly, the periphery areas presents a very short waiting time which is nearly 0, while in central areas the waiting time is relevantly higher. This is totally adverse from what we found in decentralized method. We can conclude that the centralized model we applied benefits low density areas more, to avoid passengers losing patience, yet in busy areas passengers could experience longer waiting time.

Finally, centralized repositioning helps driver make greater average income than decentralized repositioning. The objective of centralized model is to minimize the number of passengers unsatisfied as well as repositioning distance. Serving more passengers ensures a higher fare income, and less reposition distance results in lower operational cost. Since the fare contributes positively to the driver income, and the operational cost contributes negatively, The two objectives would both make the drivers' income ascend. In addition, the centralized model searches for optimal solution with exact demand, thus it is not surprising that the centralized model makes greater driver income than the other two. The income of decentralized model is greater than that of random walk model, also because of serving more passengers while decreasing detours.

4.2. Scenario A: Supply and Demand Level

In this scenario, different combinations of supply and demand levels will be tested in the decentralized model, random walk and centralized (benchmark) model. The supply and demand level combinations examined in this section includes (50,2500), (63,2500),(100,2500),(150,2500). If we define **Demand-Supply Ratio (DSR)** as the proportion between demand level and supply level, then the corresponding DSR of the combinations above are 50, 40, 25, 17 respectively. We still evaluate the performance of models from service rate, passenger waiting time and driver income.

In line with what we found in reference scenario, we still expect that the centralized model outperforms than decentralized model on increasing the service rate from the random walk state. According to figure 4.3, this is true when the DSR is 40 and 25. When the DSR is 17, the ride-sourcing system has high performance when adopting either any repositioning strategy, with 100% service rate. However, when DSR value is 50, the centralized model is surprisingly inferior to even the random walk model. This could be due to limitation of mathematical model when dealing with extreme busy situations (Iglesias et al., 2018).

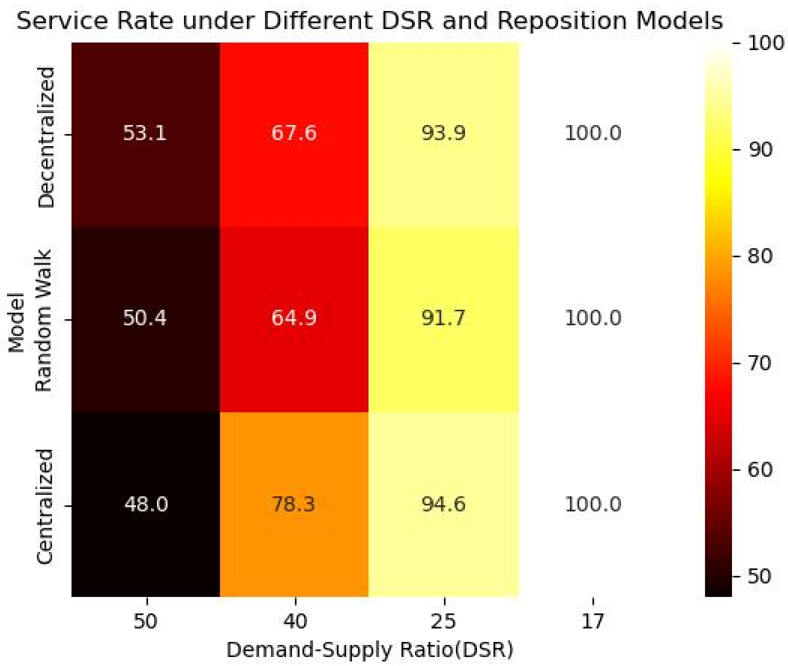


Figure 4.3: Service Rate of Decentralized and Random Walk Model and Centralized under Different DSR and Reposition Models

Except for the situation when DSR = 17, the decentralized model always increases the service rate from random walk state by 2 pp to 3 pp, and there is a slight increase as DSR gets higher. This denotes that, although in extreme busy situations the number of chances for a vehicle to reposition descends, the decentralized model is still able to increase the efficiency of reposition process.

Centralized repositioning has the best performance when DSR value is 40, which is 10.7 pp and 13.4 pp higher than decentralized repositioning and random walk situation respectively. This is even better than what we discovered in the reference scenario, which suggests that the centralized repositioning is supposed to have greater effect in over-demand situations. We additionally tested the situation when DSR = 30, and the result is still consistent with the above situations. Yet when the DSR value is too high, the limitation of mathematical model itself could cause system collapse.

In reference scenario, which DSR = 25, the passenger waiting time when adopting decentralized repositioning strategy has the best performance in comparison to the other two strategies. The centralized repositioning strategy is less optimal in shortening waiting time than decentralized strategy, yet still better than that of random walk. In other DSR levels (Figure 4.4), this phenomena is still consistent despite the situation that DSR = 50. This further proves that there is a system collapse occurring when applying centralized model. The consistency in other DSR values further proves that the decentralized model is better at reducing average waiting time.

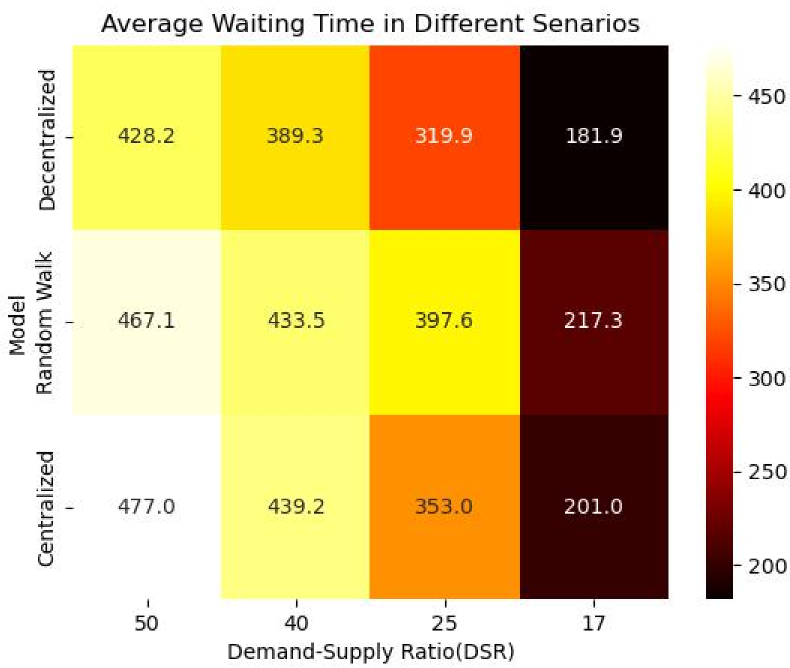


Figure 4.4: Heat Map of Waiting Time under Different Demand-Supply Ratios and Reposition Models

The descend rate of waiting time from random walk state to decentralized state is 16.1%, 10.2%, 10.1% and 8.3%. When DSR level is high, the demand is too high and can no longer be balanced by the given supply level. The backlog requests provide drivers greater chance to get assigned immediately after dropping off, and repositioning is less important in this situation. When DSR is too low, there would be many idle drivers in the system waiting for orders, and passengers could have greater chance to get matched to a near driver. Thus, the decentralized model outperforms most at reducing waiting time when the DSR level is moderate. If we compare with the service

Figure 4.5: Waiting Time Distribution under Centralized Model and Decentralized Model

rate figure, this is when the supply level and demand level is nearly balanced.

In comparison to centralized repositioning, decentralized repositioning is also most advantageous when DSR is 40, which is approximately 50 seconds shorter. When the balance between supply and demand gets deteriorated, the waiting time gets closer.

Figure 4.5 presents the distribution of passenger waiting time from one of the replications of decentralized and centralized repositioning when DSR is 17, 25 and 50 respectively. Obviously, centralized repositioning is able to provide a number of services without waiting time or short waiting time. This proportion becomes lower as DSR becomes higher, from 0.6 to 0.4 to 0.3. However, the existence of very long waiting time holds back the performance of centralized repositioning, making the average waiting time even larger. In comparison, although the passenger waiting time distribution under decentralized repositioning follows a bell curve and most passengers have to wait for some time, there is less passengers that are suffering from very long waiting times. Thus, decentralized repositioning benefits more people and increases the equity among all platform users, yet centralized repositioning could seem more attractive to a proportion of passengers.

The driver income is not only relevant to number of passengers served, but also the total operational cost. In this study, the total travel distance is considered. With the

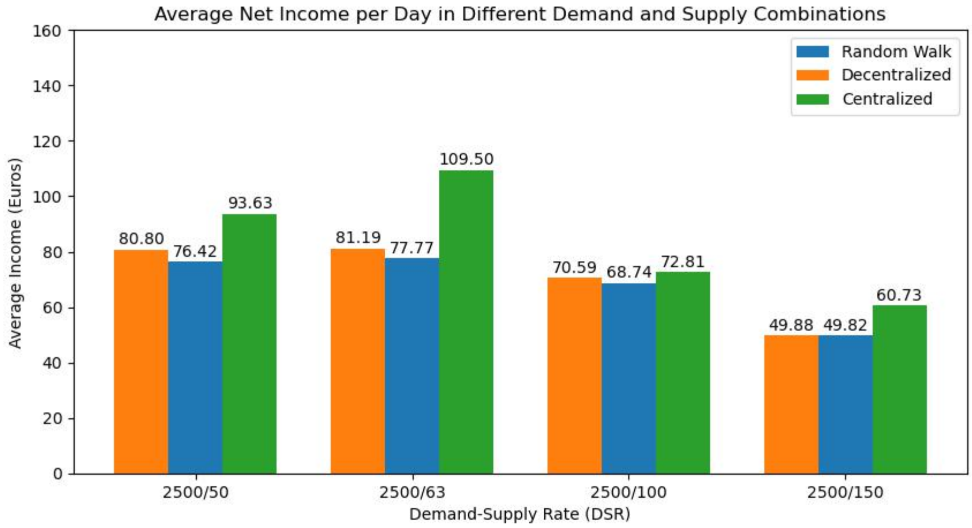


Figure 4.6: Average Net Income under Different Supply and Demand Level of Random Walk, Decentralized and Centralized Model

replications, we aim to reduce the discrepancy of passengers' total travel distance in different experiments. Thus, the most influential factors to the driver income should be the idle travel distance plus the service rate.

The results in figure 4.6 clearly show that the centralized model significantly boosts the average net income across different DSR levels. Specifically, when DSR is set to 50, indicating high passenger demand, it is remarkable to observe that the driver income in the centralized model far surpasses that in the other two models, based on the fact that the centralized model can only handle limited number of orders. In this case, the idle travel distance is expected to be minimal. This is corroborated by the statistic indicating that the repositioning distance when utilizing centralized repositioning strategy is nearly zero. This also applies to the situation that DSR is 17 . Thus, when the service rate is all 100%, the collected travel fare is identical when utilizing different repositioning strategies, yet centralized repositioning reduces the operational cost by minimizing reposition distance, resulting in higher profit for drivers. In addition, the income for drivers of centralized repositioning is significantly high when DSR is 40, when the system is able to handle much more requests than decentralized repositioning.

In extremely busy situations, when the DSR level is higher, the centralized model would dispatch vehicles to the nearest passenger, but not the passenger that requested a ride earliest. Thus, the centralized model is able to sharply reduce idle travel distance. This also accounts for the phenomenon when DSR is 17, since the dispatch is calculated based on exact demand data. However, when DSR is higher but not extremely high, the performance of repositioning in centralized and decentralized model becomes similar, resulting in closer driver income. Thus, the decentralized model is better at dealing with slightly over-demand situations regarding driver income.

4.3. Scenario B: Demand Spatial Distribution Dispersion

The request generation is a random process, and the distance from origin or destination points to the central point(s) follows a negative exponential distribution. Thus, a dispersion value is defined to determine how concentrated the origin or destination points are. As the dispersion value gets closer to zero, the pick-up and drop-off points are more scattered. To simplify, in this section we use origin (ODV) to represent the dispersion values mentioned. As discussed in experimental design part, we would also test on multi-centered situations, which would be denoted as '**M**' in this section, while '**S**' represents the single centered situations.

In this section, we would test on three different ODVs in single centered conditions, and two ODVs in multi-centered conditions. The combinations of ODV and center conditions would be (-0.0003,S), (-0.001,S), (-0.01,S), (-0.001, M) and (-0.01, M) respectively.

According to figure 4.7, decentralized repositioning significantly increases the service rate under all combinations from random walk state, yet is less efficient than the centralized method under all categories. When ODV equals -0.0003, the distribution of requests is nearly uniform in the whole area, and it is clear that the decentralized method has limited effect on increasing the service rate, while the centralized method increases the service rate by 8.1%. However, when ODV is -0.001 and -0.01, the service rate of decentralized repositioning increases by 2.2 pp and 4.3 pp from random walk state, which is more significant than the -0.0003 case. Centralized repositioning increases the service rate by 2.9 pp and 12.1 pp, respectively. In multi-centered situations, centralized repositioning also exhibits a strong ability to increase the overall service rate, which is around 10 pp from random walk state. In comparison, decentralized repositioning could only increase 2 pp to 3 pp. Thus, the centralized method is much more efficient in increasing the service rate than decentralized repositioning under different demand distribution patterns.

This phenomenon further corroborates our previous argument. Centralized repositioning penalizes every passenger that loses patience, thus despite extreme situations in which the mathematical model could hardly deal with, centralized repositioning possess advantage in increasing the system service rate in comparison to decentralized repositioning.

As figure 4.8 suggests, in single-centered situations, it is clear that as the request origins become more concentrated, centralized repositioning sharply reduces the average waiting time, yet on the other hand, as the requests get more scattered, the centralized model begins to become less optimal than the decentralized model, and even the random walk model. A similar pattern is also found in multi-centered situations, which when the requests are concentrated on the centers, centralized repositioning has a similar waiting time as decentralized repositioning, yet when the dispersion level gets more closer to zero, the waiting time becomes less optimal again. Thus, decentralized repositioning is better at handling scattered patterns, while centralized repositioning has advantage in concentrated situations.

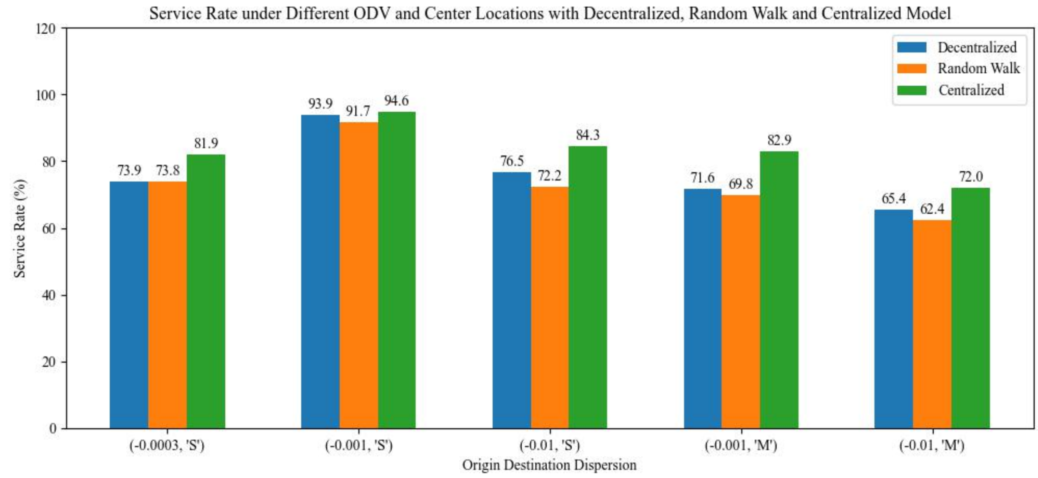


Figure 4.7: Service Rate in Different Models under Different Demand Dispersion Patterns

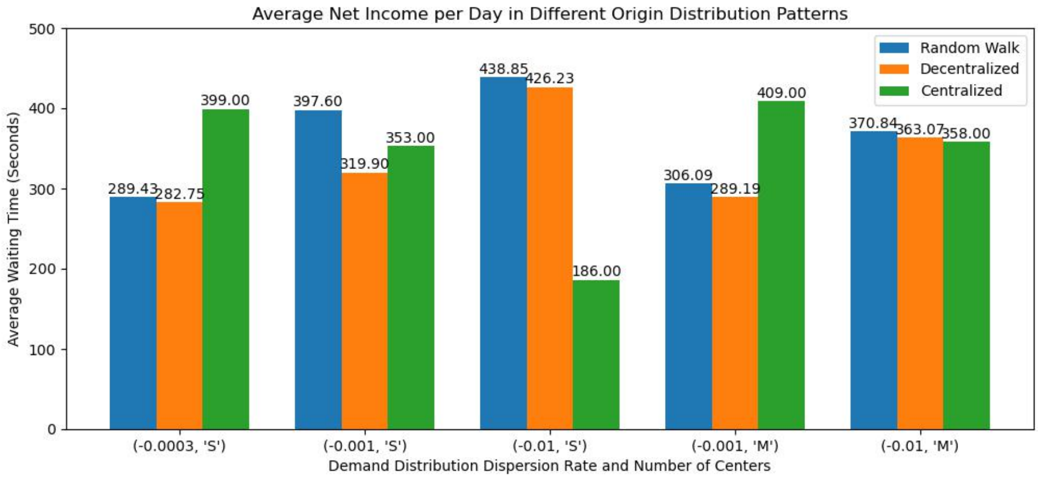


Figure 4.8: Heat Map of Average Waiting Time of Decentralized, Random Walk and Centralized Method under Different Origin Distribution Patterns

Consistent with the founding in scenario A, centralized repositioning always has the greatest net income among the three repositioning strategies. Quite different from what we discovered in the waiting time, Figure 4.9 indicates that centralized repositioning performs much better than the other two strategies in very scattered situations, while as the requests get more concentrated, the discrepancy becomes smaller. This applies to both single-centered and multi-centered situations. When the ODV is -0.0003 in single-centered condition, and the ODV is -0.001 in multi-centered condition, the average income is higher than the decentralized method for 21.22 and 20.99 euros, which is quite considerable for drivers. Yet in very ODV is -0.01, the increase is only 7.15 and 12.35 euros respectively.

Based on the analysis presented in the previous scenario, it can be observed that

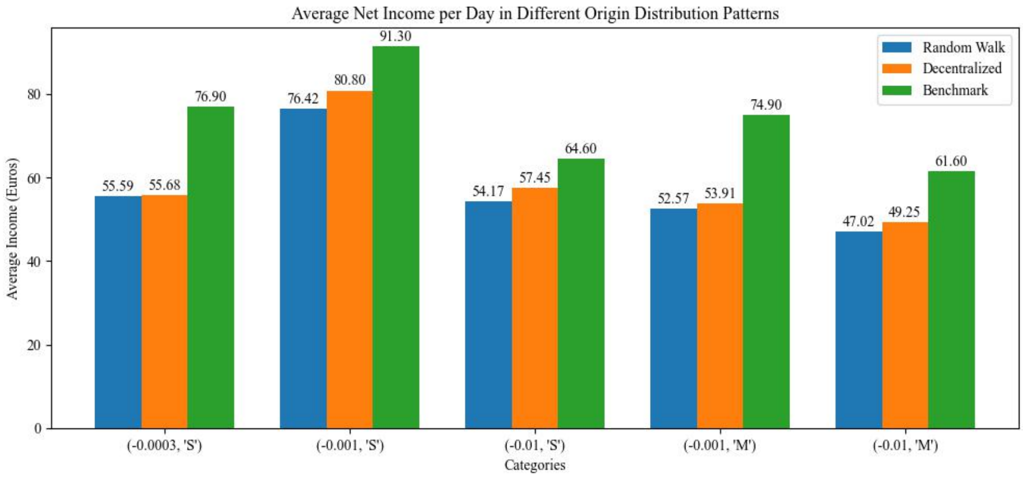


Figure 4.9: Distribution of Average Gross Income among All Drivers under Different Origin and Destination Patterns

in dispersed scenarios, centralized repositioning excels in minimizing repositioning distance but compromises passenger waiting times to some extent. In cases when demand is distributed nearly uniform, which ODV is close to 0, both decentralized repositioning and random walk exhibit nearly identical repositioning levels. However, in concentrated conditions, decentralized repositioning outperforms in reducing repositioning distance, and the advantage of centralized repositioning diminishes.

4.4. Scenario C: Platform Information

In previous scenarios, we made an assumption that drivers would learn the PGAM of every hexagon separately, and each drivers make decisions individually based on their own learning set. In this scenario, we assume that the platform itself would have a learning set derived from the collective experiences of all drivers on the platform, and this information would be accessible to all drivers In this case, drivers would have the option to base their repositioning decisions either on their own learning set or the information provided by the platform. The **Acceptance Rate** mentioned in this section refers to the proportion of drivers that would comply with platform information, while other drivers would still make decisions from their own experiences. We would continue on evaluating the performance of different repositioning strategies based on the service rate, passenger waiting time and driver income.

Figure 4.10 illustrates the service rate every day across decentralized repositioning, random walk, and centralized repositioning. The analysis reveals that when the acceptance rate is 0.2, 0.4 and 0.6, this new decentralized repositioning strategy performs better than drivers' repositioning without platform information, which increases the service rate around 0.5 pp to 1 pp, and the service rate could even grow higher than that of centralized repositioning strategy. However, as the acceptance rate gets higher, the service rate begins to descend, and when the acceptance rate is 0.8 the service rate becomes lower than when that is 0. Moreover, when the value is 1.0,

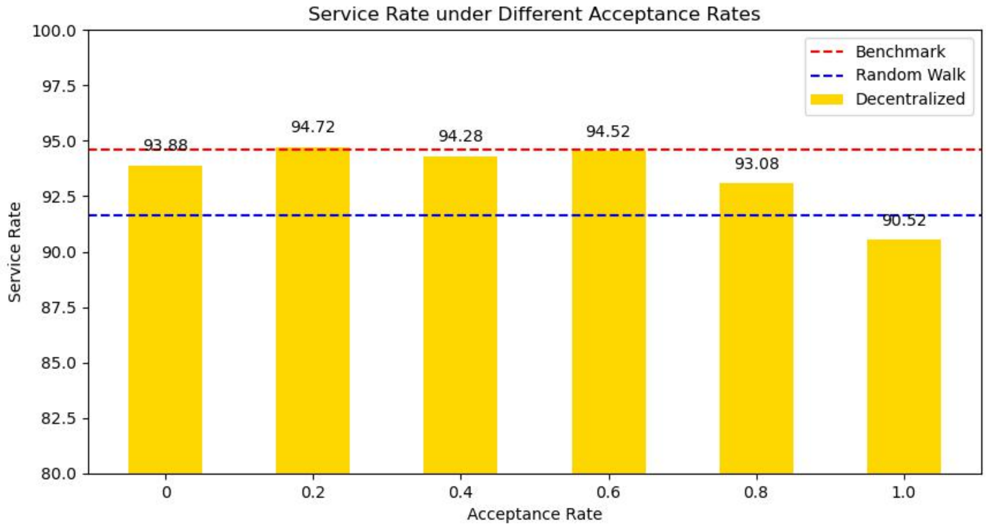


Figure 4.10: Proportion of Travellers Served under Different Driver Acceptance Rates

which is when all drivers make decisions based on platform information, the performance is even inferior to the random walk model.

Having access to platform information equips drivers with a more comprehensive understanding of passenger demand patterns. When drivers lack of experience, their learning set is constructed based on very limited information. In this case, occasional events may lead drivers to form inaccurate impressions about the actual demand patterns within hexagons. Forming up a collective learning set contributed by all drivers helps rectify this limitation. Moreover, for remote areas, new drivers could discover from the collective learning set that some orders have been successfully completed here by other drivers before, making these areas viable alternatives when drivers making repositioning decisions. This, in turn, contributes to an improvement in the service rate for remote areas. This is why, when platform information is accessible, more drivers make better choices while repositioning.

When too much drivers adhere to the information provided by the platform, the service rate could once again experience a decline. In such scenarios, all vehicles tend to converge on hexagons with higher PGAM during repositioning. Simultaneously, hexagons with lower PGAM are often disregarded by all drivers. Thus, when a request is generated in these remote areas, passengers have to spend more time waiting for pick-up, since no available driver is nearby. After a while, the hexagons with high PGAM, which seems more attractive to drivers, would be over-supplied and drivers would experience longer waiting time here. Gradually, drivers have lower interest visiting this high-demand hexagon, causing longer waiting times in high demand areas.

Thus, we draw on a conclusion that moderate acceptance rate of platform information induces increase of service rate, yet overflow of information could pose even greater negative effects on the service rate.

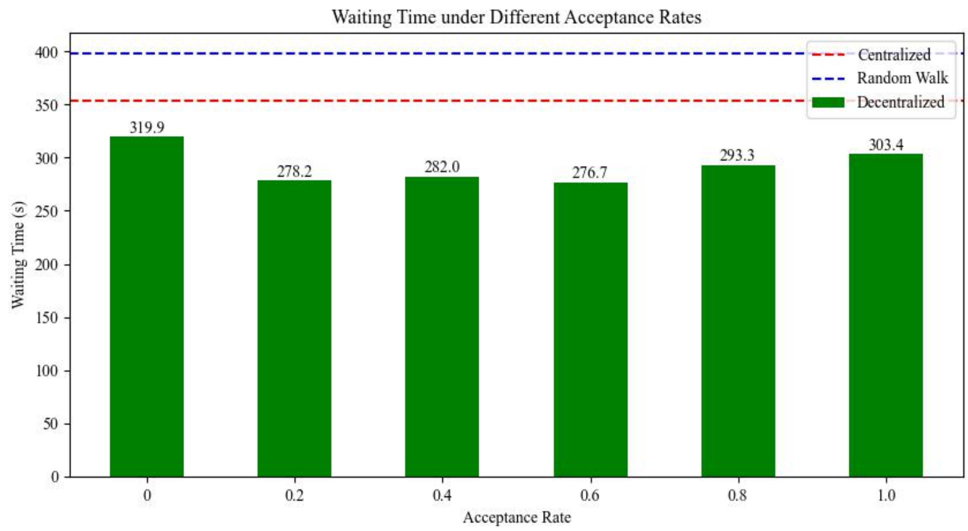


Figure 4.11: Waiting Time under Different Acceptance Rates

Correspondent to the founding on service rate, the waiting time also outperforms when the acceptance rate is 0.2, 0.4 and 0.6. Figure 4.11 suggests that the waiting time with a relevant low acceptance rate is shorter than situations of high acceptance rate and 0 acceptance rate. However, what is different from the phenomenon in service rate is that the waiting time of full acceptance rate is even lower than zero acceptance rate. Although more passengers cannot be served when the acceptance rate is very high, the average waiting time of the served passengers is surprisingly lower.

Figure 4.12 suggests the rationale behind the superior performance of a full acceptance rate compared to a zero acceptance rate. While both acceptance rate scenarios result in a waiting time distribution resembling a bell curve, the peak value is notably higher when the acceptance rate is set to zero. Therefore, a higher acceptance rate still has the potential to provide shorter waiting times for most individuals, but it comes at the expense of reducing the service rate.

Consistent to the previous scenarios, decentralized repositioning is better at shortening the average waiting time than centralized repositioning under all examined service rates. As presented in figure 4.12, it appears that with platform information, the central of bell curve moves towards y-axis. This is also the case in 4.5, which the majority passengers in centralized model is distributed very close to the y-axis. Thus, the platform information model somehow indicates similarity with the case that repositioning is centralized.

Driver Income is quite consistent over different acceptance rates. The bar chart 4.13 indicates that even when the acceptance rates are 0.2, 0.4 and 0.6, which are regarded as optimal values for service rate and waiting time, there is only a slight increase in average daily income. Although the service rate is significantly higher in

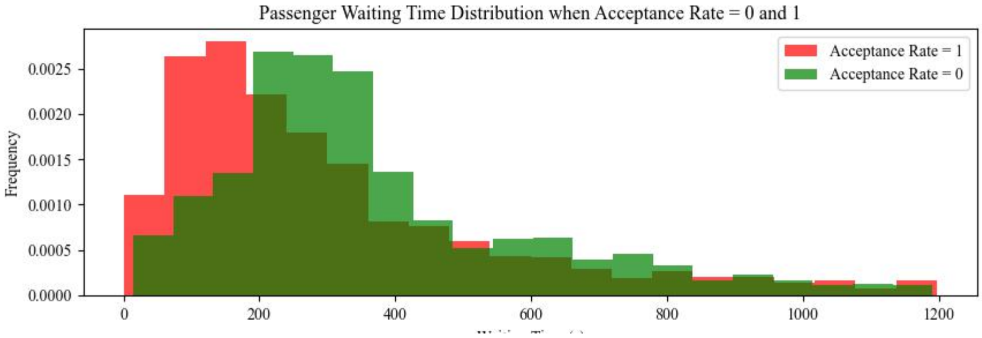


Figure 4.12: Passenger Waiting Time Distribution under Acceptance Rate 0 and 1

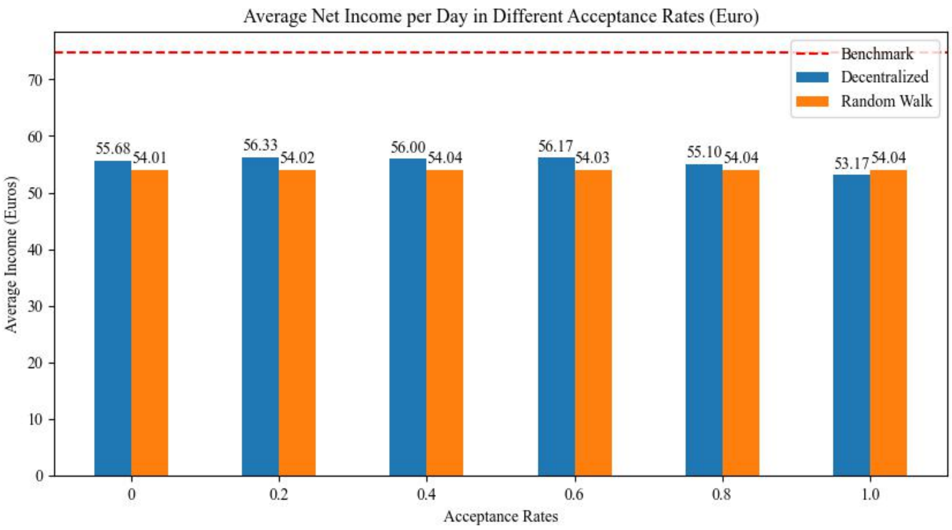


Figure 4.13: Average Income per Day per Driver under Different Acceptance Rates

these conditions, which should lead to higher collected fare, the insignificant increase in driver profit suggests that the cost could be higher under these acceptance rates. This is further proved by the phenomenon that, when acceptance rate is 1.0, the driver profit of decentralized repositioning is even 0.87 euros less than random walk. Thus, platform information benefits more on passengers side, while has limited advantage for drivers. Instead, if too many drivers make decisions based on platform information, there could be negative effects on their earnings.

In comparison to that of centralized repositioning, decentralized repositioning with platform information is consistently less advantageous. This still is in line with our expectation, since the objective of centralized repositioning here is to reduce the repositioning distance, resulting in lower operational cost.

4.5. Computation Complexity

Simulations were run on AMD Ryzen 7 5800H with Radeon Graphics□3201 Mhz CPU core. The random walk model and centralized model were simulated with python 3.8.8, and the simulation of centralized model is performed with python 3.8.8 using CPLEX. Due to time limit, each scenario is simulated 5 times, and the average running time of each simulation day of scenario A, B and C are displayed in table 4.2, 4.3 and 4.4.

Supply/Demand	150/2500	100/2500	63/2500	50/2500
Random Walk	55	68	62	55
Decentralized	405	329	237	191
Centralized	45	45	782	678

Table 4.2: Average Running Time per Day for Scenario A: Supply and Demand (seconds)

Centers	Single	Single	Single	Multiple	Multiple
Origin Dispersion	-0.0003	-0.001	-0.01	-0.001	-0.01
Random Walk	64	68	48	56	50
Decentralized	265	329	258	251	258
Centralized	274	45	65	207	44

Table 4.3: Average Running Time per Day for Scenario B: Demand Spatial Dispersion (seconds)

Acceptance Rate	0	0.2	0.4	0.6	0.8	1
Random Walk	68	68	64	66	64	60
Decentralized	329	330	331	341	319	291

Table 4.4: Average Running Time per Day for Scenario C: Platform Information (seconds)

The average running time of decentralized model in table 4.2, 4.3 and 4.4 suggests that the demand dispersion and platform information acceptance rate does not have significant impact on running time, while an increase in supply level could lead to an increase in average running time. The running time of centralized model is polarized in comparison to the decentralized model. On the one hand, when the Demand Supply Ratio (DSR) is small, or when the demand is not very scattered, the computation time is much smaller than either random walk or centralized model, while on the other hand when DSR is large or when the demand is very scattered, the solver takes much longer time to solve such a large optimization problem.

5
Conclusion

5.1. Key Findings and Discussion

Based on the results discussed in the previous chapter, we answer the main research question.

- *In comparison with centralized repositioning methods, how does decentralized repositioning strategy affect passenger waiting time and driver income.*

Decentralized repositioning is less efficient than centralized repositioning in increasing the service rate and driver income from random walk state. On the one hand, centralized repositioning base on pre-knowledge of demand data. On the other hand, the objective of centralized repositioning here is to minimize the total cost of repositioning as well as penalizing passengers losing patience. The two objectives ensure centralized repositioning to have a better performance on reducing number of passengers losing patience and operational costs, which corresponds to the outstanding performance of service rate and driver income.

Nonetheless, achieving this comes at the cost of passengers having to wait longer when utilizing centralized strategies. In the pursuit of minimizing repositioning distance, when the repositioning is centralized, the platform might exhibit a preference for keeping drivers wait for requests within the same grid instead of repositioning them to get assigned sooner. This tendency becomes particularly noticeable in extremely busy scenarios, or when demand is almost evenly distributed. The findings from corresponding experiments validate this assertion, highlighting that in such conditions, decentralized repositioning tends to benefit passengers more compared to repositioning by the platform.

- *In comparison with centralized methods, how does the decentralized repositioning strategy perform under different supply and demand levels?*

Decentralized repositioning is more efficient in reducing waiting time in comparison to centralized repositioning, especially in over-demand situations. Yet, centralized repositioning is much more efficient in increasing drivers' net income and the service rate, and has best performance in situations when demand is insufficient to serve the supply.

- *How does the decentralized repositioning strategy perform under different origin and destination distribution patterns?*

When the origin points become more clustered around a central location or multiple central locations, decentralized repositioning gains a greater advantage over the random walk state in terms of service rate and income. However, in scenarios where requests are more widely scattered, repositioning in a centralized manner results in significantly more advantageous driver income compared to the other two repositioning methods. In widely scattered conditions, decentralized repositioning does not necessarily help reduce the total reposition distance, yet in more concentrated situations the performance becomes better.

Decentralized repositioning, however, is better at reducing passenger waiting time when the requests are scattered widely, compared to centralized repositioning. Yet in extremely clustered situations, centralized repositioning would benefit passengers more, by ensuring a very short waiting time, averagely 4 minutes less than that of decentralized repositioning.

Thus, decentralized repositioning benefits more in situations where demand is distributed more evenly, which ensures passengers experiencing less waiting time. Yet for drivers, centralized repositioning would always be more attractive, since it ensures higher income.

- *To what extent could information on average waiting time and historical request numbers of different hexagons provided by the platform improve system performance?*

According to the results, decentralized repositioning has the highest service rate and least waiting time when the acceptance rate is 0.2, 0.4 or 0.6. At high acceptance rates, the waiting time would increase again but not exceed the condition that no platform information is provided. This is mainly because with platform information most passengers would experience shorter waiting time, yet overflow of information would cause over-supply in high demand areas, and insufficient idle vehicles in remote areas. This further leads to very low service rates, which falls even below the condition of drivers repositioning randomly. The acceptance rate does not necessarily affect the driver income, since although the service rate increases, the average reposition distance also increases.

Thus, providing platform information seems less attractive to drivers, since the income is lower. Yet it is an efficient way to improve passengers' ride experience. The accessibility of the platform information has to be controlled by the platform, to avoid overflow of information, leading to even negative effects.

5.2. Implications

Decentralized repositioning has the highest efficiency when demand is scattered around a single center, when the dispersion rate is moderate, and when the DSR value is moderate. Whether selecting centralized repositioning or not highly depends on the objective one wants to achieve. In such situations, allowing decentralized repositioning could lead to higher passenger satisfaction and attract more passengers, since the waiting time is significantly shorter, and in extremely busy situations this approach ensures a higher service rate. Yet centralized repositioning has advantage in increasing driver income, and attracts more new drivers to enter the system. As a two sided platform, both situations could always happen due to competition in the market. Thus, it is reasonable for platforms to apply centralized repositioning when the platform has too much demand, while allowing decentralized repositioning when the platform lacks of passengers.

For platforms, it is also more efficient to make necessary information available to drivers, such as average waiting time, or number of rides accomplished in every hexagon. However, high acceptance rates could have negative effects, so the visibility of information should depend on real-time acceptance rate. When acceptance rate is too high, measures such as not providing information or providing incorrect information could diminish drivers' trust on the platform, lowering the acceptance rate.

Finally, for self-driving vehicles, it is also feasible to implement decentralized decision-making process, where computers in vehicles process calculation individually, and drive autonomously according to their own results. Platforms can also take over the decision-making process whenever they consider centralized repositioning necessary.

5.3. Limitations and Future Research

In this study, we made several assumptions to simplify the modelling part and reduce computation complexity, yet some simplifications could be unrealistic. The request cannot be identical every day, and in future work the model could be implemented on different generated requests in day-to-day process. Also, congestion is not considered in this study, which could have a great impact in drivers' route choice.

As for the learning model and score-based repositioning model itself, more aspects should be included, since decision-making is a very comprehensive and complex process. In future work, based on sensitivity analysis on survey results or historical data, more relevant aspects could be adopted to better model drivers' behaviors.

The current benchmark model minimizes simultaneously the reposition distance and travellers that lose patience. In future work, different objective functions for centralized repositioning should be considered, such as minimizing waiting time. More importantly, the demand forecast part in benchmark model is omitted, and real demand is applied as the model input. To better compare the model performances, in future work it would be necessary to embed a demand forecast algorithm that does not require large scale of historical data.

The algorithm complexity could be reduced in further study to speed up the simulation process. Currently the scoring process of paths is written in a recursive function, which requires high computation efforts. Also, due to the long simulation times, the number of repetitions is insufficient to reach statistical significance. In future work more repetitions could be executed to gain better results.

Finally, due to limitation of hardware, the number of steps, number of days simulated, scale of demand and supply is very limited. In further work, larger data scale experiments could be implemented to gain more insights.

Bibliography

Amsterdam.nl. (2018). *Policy: Traffic and transport*. https://www.amsterdam.nl/en/policy/policy-traffic/

Ashkrof, P., de A Correia, G. H., Cats, O., & van Arem, B. (2023). On the relocation behavior of ride-sourcing drivers. *Transportation Letters*. https://doi.org/10.1080/19427867.2023.2192581

Ashkrof, P., de Almeida Correia, G. H., Cats, O., & van Arem, B. (2022). Ride acceptance behaviour of ride-sourcing drivers. *Transportation Research Part C: Emerging Technologies*, *142*, 103783. https://doi.org/10.1016/j.trc.2022.103783

Burghout, W. (2004). A note on the number of replication runs in stochastic traffic simulation models.

de Ruijter, A., Cats, O., Kucharski, R., & van Lint, H. (2022). Evolution of labour supply in ridesourcing. *The European Physical Journal B Transportmetrica B: Transport Dynamics*, *10*, 599–626. https://doi.org/10.1080/21680566.2021.2024917

Djavadian, S., & Chow, J. Y. J. (2016). Agent-based day-to-day adjustment process to evaluate dynamic flexible transport service policies. *Transportmetrica B*. https://doi.org/10.1080/21680566.2016.1190674

Dong, T., Luo, Q., Xu, Z., Yin, Y., & Wang, J. (2022). Strategic driver repositioning in ride-hailing networks with dual sourcing. *SSRN Scholarly Paper*. https://doi.org/10.2139/ssrn.4040164

Fan, R. (2022). Commuting service platform: Concept and analysis. *Transportation Research Part B: Methodological*, *158*, 18–51. https://doi.org/10.1016/j.trb.2022.01.0066

Feng, S., Chen, T., Zhang, Y., Ke, J., Zheng, Z., & Yang, H. (2023). A multi-functional simulation platform for on-demand ride service operations.

Ferreira, M., & d'Orey, P. M. (2014). Can ride-sharing become attractive? a case study of taxi-sharing employing a simulation modelling approach. *IET Intelligent Transport Systems*, *9*. https://doi.org/10.1049/iet-its.2013.0156

Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., & Pavone, M. (2018). Data-driven model predictive control of autonomous mobility-on-demand systems. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–7. https://doi.org/10.1109/ICRA.2018.8460966

Jaime Soza-Parra, R. K., & Cats, O. (2022). The shareability potential of ride-pooling under alternative spatial demand patterns. *Transportmetrica A: Transport Science*, *0*(0), 1–23. https://doi.org/10.1080/23249935.2022.2140022

Jiao, Y., Tang, X., Qin, Z., Li, S., Zhang, F., Zhu, H., & Ye, J. (2021). Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research: Part C*, *130*(103289). https://doi.org/10.48550/arXiv.2103.04555

Knobbe, D. (2022). *Impact of empty vehicle repositioning strategies on ride-hailing system performance*.

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

Lei, Z., Qian, X., & Ukkusuri, S. V. (2020). Efficient proactive vehicle relocation for on-demand mobility service with recurrent neural networks. *Transportation Research Part C: Emerging Technologies*, *117*, 102678. https://doi.org/10.1016/j.trc.2020.102678

Lin, Y., Wang, X., Wang, L., & Wan, P. (2020). Dynamics modeling and optimal control for multi-information diffusion in social internet of things. *Digital Communications and Networks*, *S2352864823000500*. https://doi.org/10.1016/j.dcan.2023.02.014

Lin, Zhao, R., Xu, Z., & Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 1774–1783. https://doi.org/10.1145/3219819.3219993

Nahmias-Biran, Oke, J. B., Kumar, N., Basak, K., Araldo, A., Seshadri, R., Akkinepally, A., Azevedo, C. L., & Ben-Akiva, M. (2019). From traditional to automated mobility on demand: A comprehensive framework for modeling on-demand services in simmobility. transportation research record. *Transportation Research Record*, *2673*, 15–29. https://doi.org/10.1177/0361198119853553

Qian, X., Guo, S., & Aggarwal, V. (2022). Drop: Deep relocating option policy for optimal ride-hailing vehicle repositioning. *Transportation Research Part C: Emerging Technologies*, *145*, 103923. https://doi.org/10.1016/j.trc.2022.103923

Qu, M., Zhu, H., Liu, J., Liu, G., & Xiong, H. (2014). A cost-effective recommender system for taxi drivers. *KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 45–54. https://doi.org/10.1145/2623330.2623668

Riley, C., van Hentenryck, P., & Yuan, E. (2020). Real-time dispatching of large-scale ride-sharing systems: Integrating optimization, machine learning, and model predictive control. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 4417–4423. https://doi.org/10.24963/ijcai.2020/609

Ruch, C., Horl, S., & Frazzoli, E. (2018). Amodeus, a simulation-based testbed for autonomous mobility-on-demand systems. *2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA*, 3639–3644. https://doi.org/10.1109/ITSC.2018.8569961

Shou, Z., Di, X., Ye, J., Zhu, H., Zhang, H., & Hampshire, R. (2020). Optimal passenger-seeking policies on e-hailing platforms using markov decision process and imitation learning. *Transportation Research Part C: Emerging Technologies*, *111*, 91–113. https://doi.org/10.1016/j.trc.2019.12.005

Sirisoma, R. M. N. T., Wang, D., Lam, W. H. K., & Zhang, P. (2010). Empirical evidence for taxi customer-search model. *Proceedings of the Institution of Civil Engineers - Transport*, *163*, 203–210. https://doi.org/10.1680/tran.2010.163.4.203

Szeto, W. Y., Wong, R. C. P., Wong, S. C., & Yang, H. (2013). A time-dependent logit-based taxi customer-search model. *International Journal of Urban Sciences*, *17*, 184–198. https://doi.org/10.1080/12265934.2013.776292

Tengilimoglu, O., & Wadud, Z. (2022). Evaluating the mileage and time efficiency of ridesourcing services: Austin, texas case. *Transportation Letters*, *5*, 478–491. https://doi.org/10.1080/19427867.2021.1892936

tomtom traffic index. (2022). *Traffic index ranking*. https://www.tomtom.com/traffic-index/ranking/

Urata, J., Xu, Z., Ke, J., Yin, Y., Yang, H., & Ye, J. (2021). Learning ride-sourcing drivers' customer-searching behavior: A dynamic discrete choice approach. *Transportation Research Part C: Emerging Technologies*, *130*. https://doi.org/10.1016/j.trc.2021.103293

Valadkhani, A. H., & Ramezani, M. (2023). Dynamic ride-sourcing systems for city-scale networks, part ii: Proactive vehicle repositioning. *Transportation Research Part C: Emerging Technologies*, *152*(104159). https://doi.org/10.1016/j.trc.2023.104159

Verma, T., Varakantham, P., Kraus, S., & Lau, H. C. (2017). Augmenting decisions of taxi drivers through reinforcement learning for improving revenues. *Vol. 27 (2017): Twenty-Seventh International Conference on Automated Planning and Scheduling*, *27*, 409–417. https://doi.org/10.1609/icaps.v27i1.13846

Yang, H., & Wong, S. C. (1998). A network model of urban taxi services. *Transportation Research Part B: Methodological*, *32*, 235–246. https://doi.org/10.1016/S0191-2615(97)00042-8

Yao, R., & Bekhor, S. (2022). A ridesharing simulation model that considers dynamic supply-demand interactions. *JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS*. https://doi.org/10.1080/15472450.2022.2098730

Yu, Z., & Hu, M. (2022). Deep reinforcement learning with graph representation for vehicle repositioning. *IEEE Transactions on Intelligent Transportation Systems*, *23*, 13094–13107. https://doi.org/10.1109/TITS.2021.3119662

Decentralized Method in Ride-sourcing Reposition Decision-making Process

Kairui Cui

October 23, 2023

Abstract

Efficient repositioning strategies for idle vehicles in ride-sourcing systems help reduce passengers' waiting time and drivers' operational costs, which help platforms attract more passengers and drivers. In this paper, we propose a decentralized repositioning strategy for drivers, in which drivers make individual decisions on where they reposition themselves, based on their own experiences. In comparison to an existing centralized repositioning strategy, in which drivers comply with reposition instructions provided by the platform, we examine the effects of the decentralized strategy on service rate, passengers' waiting time and drivers' net income. We compare the reposition strategies under different supply and demand levels and different demand spatial distribution dispersion rates. We also explore the influence of platform information on drivers' decision-making process.

We found that decentralized repositioning strategies have better performance in reducing waiting time, while the centralized strategy is better at increasing driver income and service rate. We also found that when platform information is accessible, the system has the best performance when 20% to 60% proportion of drivers utilize platform information when making decisions.

Keywords: Ride-sourcing; Repositioning; Agent Based Model; Transport Network Companies

I Introduction

The rise of digital technology has made ride-sourcing systems, like Uber, a prevalent mode of transportation, with significant growth reported in spite of the COVID-19 pandemic. Unlike traditional taxis, these platforms provide passengers with efficient access to nearby vehicles via mobile apps, eliminating the need to wait for taxis on the street. This convenience is especially advantageous in remote areas, where passengers can secure rides even when taxis are scarce. For drivers, the system offers improved visibility of nearby ride requests, reducing distractions and enhancing safety.

However, challenges persist in the ride-sourcing market. Drivers often face income instability, sometimes falling below minimum wage standards due to low demand (de Ruijter et al., 2022). Additionally, drivers may accept distant orders in response to limited availability, resulting in high operational costs, long waiting times, and potential cancellations. Addressing these issues necessitates optimizing repositioning strategies to minimize costs and waiting times, ultimately improving the overall efficiency of the system.

Literature Review

Reposition Behaviors: Historically, taxi drivers relied on personal preferences and knowledge of passenger demand distribution, considering factors like journey time, cross-zonal costs, waiting times, and tolls (Sirisoma et al., 2010; Szeto et al., 2013). Repositioning behaviors were influenced by various external and personal factors (Yang & Wong, 1998), leading to inconsistent decisions across different time periods. However, with the rise of ride-sourcing services, drivers are now more inclined to remain idle due to the ease of getting matched (Urata et al., 2021), which can result in inefficiencies. Platforms may incentivize active behavior, particularly when there's an imbalance between supply and demand. The availability of real-time information, including pre-booked ride distribution, driving conditions, and parking availability, plays a crucial role in drivers' decision-making (Ashkrof et al., 2023; Ashkrof et al., 2022) , enabling platforms to develop effective policies.

Decentralized Repositioning: The repositioning process in ride-sourcing systems can be optimized using either centralized repositioning strategies, in which passengers comply with platform instructions, or decentralized repositioning strategies, in which drivers make decisions on repositioning routes themselves. Centralized approaches involve platform-determined repositioning instructions provided to all drivers, often relying on reinforcement learning (Jiao et al., 2021; Lei et al., 2020; Lin et al., 2018; Qian et al., 2022; Shou et al., 2020) or Model Predictive Control (Iglesias et al., 2018; Riley et al., 2020; Valadkhani & Ramezani, 2023). However, drivers may not always comply with these instructions, resulting in deviation from optimal results. Decentralized repositioning, however, studies on passengers discretely, allowing drivers perform different behaviors while examinging the overall system performance (Knobbe, 2022; Qu et al., 2014; Szeto et al., 2013).

Research Gap

This study aims to assess the impact of a proposed repositioning behavior in a ride-sourcing system on service rates, passengers' waiting times, and drivers' revenue. Most existing studies focus on providing repositioning recommendations to drivers based on historical data, often using reinforcement learning models. However, training such models in a real ride-sourcing network can be computationally expensive and may destabilize the system due to the large number of trial-and-error iterations required.

Additionally, many previous studies assume a centralized repositioning approach, where the platform gives repositioning commands to drivers, and drivers do not make repositioning decisions themselves. However, this approach relies on a high acceptance rate from drivers, which may not be realistic in systems primarily composed of conventional vehicles. In such cases, decentralized repositioning, where drivers make their own repositioning decisions, becomes essential. However, existing decentralized models often overlook factors like route choice and detours during repositioning, making them less applicable to current ride-sourcing vehicles.

In this study, the repositioning process is modeled incrementally, considering both near and distant repositioning destinations. The decentralized repositioning model divides the study area into small hexagons, with drivers deciding whether to move to a neighboring hexagon or remain idle based on a comprehensive score that considers factors like wait time and ride assignments. The granularity of decision-making is at the hexagonal grid level, while the origins and destinations of ride requests are specified as coordinates within hexagons. The study also simulates a day-to-day learning process for drivers to capture their adaptive behavior within the system.

II Methodology

MaaSSim

MaaSSim, an agent-based simulator for two-sided mobility services, models interactions between drivers and passengers in ride-sourcing systems (Kucharski & Cats, 2022). Passengers generate ride requests, drivers provide transportation, and the platform acts as an intermediary to match them. This simulator replicates the day-to-day ride-sourcing operation.

a. Initialization and Hexagonal Grid Generation: MaaSSim relies on detailed network, demand, and supply information. To simplify computation due to the substantial scale of nodes, a hexagon grid system is employed, grouping nodes into cells where demand and supply are derived. Drivers evaluate each neighboring cell with a score, and the cell with the highest score is chosen for the next step. The interior of hexagon is considered as homogeneous. The hexagons approximate circular areas and streamline the model's complexity.

b. Within-day Operation Process: The hexagonal grid serves as a foundational structure for modeling the ride-sourcing network. In MaaSSim, drivers' operational processes follow a loop, starting with them being idle, joining a queue, and encountering situations such as immediate matching, remaining idle until assignment, or repositioning. After assignment, drivers navigate the pick-up and drop-off process, potentially receiving new assignments during repositioning. Passengers' operations are integrated into this loop, emerging when ride requests are generated and disappearing upon reaching their destinations. Passengers have the option to cancel requests if they lose patience. Travel times are calculated based on euclidean distances and travel distances.

Decentralized Repositioning Model

In our agent-based model, we simulate the decentralized repositioning process of drivers within a ride-sourcing system. Each driver independently makes decisions based on their knowledge, which includes a learning process about the probability of receiving an assignment in a neighboring hexagon and a scoring process related to the expected revenue for choosing to move towards a hexagon. The study area is divided into a hexagonal grid, and initially, drivers have no knowledge of the entire city, so we employ a Random Walk Strategy for their repositioning. After 10 days, drivers gain better knowledge of the demand distribution, and we introduce the Decentralized Repositioning Process. During repositioning, drivers evaluate adjacent and current hexagons based on factors such as waiting time and expected revenue, selecting the hexagon with the highest score. To avoid myopic decisions, drivers also consider further steps by examining multiple paths and choosing the one with the highest overall score, which may include staying idle in the current hexagon if it offers the best option.

a. Assumptions:

1. Drivers make repositioning decisions after a fixed interval.
2. Origin and destination distribution of requests remain identical every day.
3. Drivers start every day at a random point in the road network, and there is no depot.
4. Drivers drive every day until the defined end shift time is reached.
5. Drivers do not decline requests, and passengers accept the first driver that is matched with them.
6. A fixed time limit for passenger waiting time will be set, and when waiting time exceeds this time limit passengers lose patience and cancel the order.

b. PGAM Update: In this modeling approach, drivers continuously learn and update an indicator called PGAM (*Probability of Getting Assigned per Minute*). The learning process distinguishes between different time periods throughout the day, such as morning and evening peaks, and assumes that drivers employ the same scoring method across time periods.

As drivers complete orders, they gather information about their waiting time and passenger pickups in each hexagon. This information is used to update PGAM (Equation 1), which represents the probability of driver v getting assigned to a new order in hexagon k per minute after reposition process r. PGAM is calculated based on the number of requests $m_{v,k,r}$ assigned in hexagon k and previous waiting times $T^{\text{wait}}_{v,k,r}$. When a driver's waiting time exceeds a time threshold, they can either be assigned a new order or remain idle. If not assigned, the waiting time contributes to the current hexagon's PGAM, and the driver's assignment count remains unchanged. If assigned, the waiting time contributes to the current hexagon's PGAM, and the assignment count of the hexagon at the pick-up point increases by 1.

$$\text{PGAM}_{v,k,r} = \frac{m_{v,k,r}}{\sum_r T^{\text{wait}}_{v,k,r}} \tag{1}$$

This approach allows drivers to adapt their repositioning decisions based on their experience and the likelihood of getting assigned to an order in different hexagons, considering their waiting times and assignment history.

c. Score-based Process: In the decision-making process for drivers, to ensure they make informed choices and avoid myopic decisions, they evaluate multiple preceding steps to form a path of potential movements. These successive steps create **Paths**, and hexagons that can be reached within the n-th step are termed **Step-n Hexagons**. In this study, drivers examine the next $s^{\max}$ steps before making a decision. They calculate the scores for all possible paths and choose the path with the highest score as the optimal choice. However, drivers do not commit to following this path for multiple steps; instead, they make one-step decisions. After each move, they may reconsider all possible paths, and sometimes new attractive hexagons can influence their choices.

The scoring of paths involves calculating the expected revenue for a move, considering potential revenue (Equation 2) and the probability of assignment (Equation 3), which could be calculated with equation 4. The overall score for a path is computed by considering the scores of individual steps within that path, iteratively from the adjacent hexagons (step-1) to the furthest ones (step-$s^{\max}$). The iteration considers both the expected income in each hexagon and the probability of not matching there, presented as equation 5.

$$Q^{\text{income}}_d = Q^{\text{fare}}_d \times \beta - Q^{\text{cost}}_{v,d} \quad d \in D_v \tag{2}$$

$$\text{PGA}_{v,k,r} = 1 - (1 - \text{PGAM}_{v,k,r})^{T^{\max}} \tag{3}$$

$$R_{v,k,r} = PGA_{v,k,r} \times Q^{\text{income}}_d \quad d \in D_v \tag{4}$$

$$R_{v,p,r}(s) = \begin{cases} R_{v,k_s,r} + (1 - PGA_{v,k_s,r}) \times R_{v,p,r}(s+1) \\ (1 \leq s < s^{\max}) \\ R_{v,k_s,r} \quad (s = s^{\max}) \end{cases}$$

$$s.t.\ k_s \in p, p \subseteq P_k \tag{5}$$

in which Q^{income}_d, Q^{fare}_d and $Q^{\text{cost}}_{v,d}$ represents the net income, collected fare for order d and operation cost for order d by vehicle v. $T^{\max}$ is the time of a step of repositioning. $R_{v,k,r}$ denotes the score for entering hexagon k for vehicle v, and $R_{v,p,r}(s)$ represents the total score of path p from step s to step $s^{\max}$. D_v is the set of requests accomplished by vehicle v, and P_k is the set of feasible paths originated from hexagon k.

Scores of paths ($R_{v,p,r}$) originating from adjacent hexagons are calculated as equation 6, and the path with the highest score is selected. Drivers then reposition to the adjacent hexagon included in the selected path.

$$Z_{v,r} = \max_{p \subseteq P_k} \{R_{v,p,r}(1)\} \tag{6}$$

III Experiment Design

Benchmark Centralized Method

A-RTRS is a model predictive control method that rebalances idle vehicles in the entire area, to make the number of supply match with the number of potential request in different zone areas. This method divides time into time windows with fixed duration l^A, and in each epoch routes of idle vehicles are optimized to priorly arrive at the pick-up point of potential requests or unserved requests. The A-RTRS process includes three steps, which are demand forecast, optimization process and vehicle allocation process. In this model, we assume that we possess pre-knowledge of exact demand of the whole day, thus we skip the demand forecast step. The objective of A-RTRS is to minimize the passengers that lost patience, and minimize the operational cost. Further information of the centralized model could be found in papers written by Iglesias et al. (2018) and Riley et al. (2020).

Case Study

This study is applied to the city of Amsterdam. Several Transport Network Companies (TNCs) like Uber currently provide ride-sourcing services in Amsterdam.

Scenario Design

A total of three scenarios will be tested in our experiment. We would make a comparison of the service rate, waiting time and driver income between Random Walk State, Benchmark Centralized Model and our Decentralized Model.

Scenario A: Demand and Supply Levels In this experimental phase, an indicator Demand-Supply Rate (DSR) is introduced, which denotes the ratio between demand number and supply number. DSR are set at nP $= 50, 40, 25, 17$, which the combination of demand and supply are (2500,50), (2500,63), (2500,100),(2500,150).

Scenario B: Demand Spatial Distribution Patterns In this experiment, we investigate the impact of different demand distribution patterns on ride-sourcing system performance. Two distribution patterns, scattering passengers around a single center (**S**) and multiple centers (**M**), are considered for the pick-up points. Center locations are chosen based on Amsterdam's density map, with centers located at the geographical center of Amsterdam, Sloterdijk, Zuid, and Oost for scenarios with multiple centers. We test scattered and clustered conditions in both single- and multi-center situations. Then, The combinations of origin dispersion value (ODV) and number of centers are namely (-0.0003,S), (-0.001,S), (-0.01,S), (-0.001, M), (-0.01, M).

Scenario C: Platform Information The platform collects and analyzes comprehensive travel data from both passengers and drivers, facilitating continuous data aggregation and visualization. In this context, platforms are assumed to provide information regarding average waiting times for each hexagon to all drivers. An acceptance rate is applied to determine the proportion of drivers who would base their calculations on the collective set of waiting time. The acceptance rates include 0, 0.2, 0.4, 0.6, 0.8 and 1.0.

IV Results

Decentralized repositioning is less efficient than the centralized repositioning in increasing the service rate and driver income from random walk state. On the one hand, the centralized model is based on pre-knowledge of demand data. On the other hand, the objective of centralized model is to minimize the total cost of repositioning as well as penalizing passengers losing patience. The two objectives ensure centralized repositioning to have a better performance on reducing number of passengers losing patience and operational costs, which corresponds to the outstanding performance of service rate and driver income.

Indicators	Dec	RW	Cen
Service Rate	93.9%	91.6%	94.6%
Avg. Waiting Time	320	398	353
Avg. Income	70.6	68.7	72.8

Table 1: Results of Decentralized Model , Random Walk Model and Centralized Model when Supply = 100, Demand = 2500, γ^o = -0.001, γ^d = -0.001

Nonetheless, achieving this comes at the cost of passengers having to wait longer in centralized approaches. In the pursuit of minimizing repositioning distance, when the repositioning is centralized, the platform might exhibit a preference for having drivers wait for requests within the same grid instead of repositioning them to get assigned sooner. This tendency becomes particularly noticeable in extremely busy scenarios, or when demand is almost evenly distributed. The findings from corresponding experiments validate this assertion, highlighting that in such conditions, decentralized repositioning tends to benefit passengers more compared to repositioning by the platform.

Scenario A: Demand and Supply Levels

Centralized repositioning is much more efficient in increasing drivers' net income and the service rate, and has best performance in situations when DSR is very high. Yet, as figure 1 suggests, decentralized repositioning is more efficient in reducing waiting time in comparison to centralized repositioning, especially in over-demand situations.

This aligns with the founding in reference scenario. Due to the objective of centralized repositioning, the service rate and driver net income performs more optimal than that of decentralized repositioning, yet the centralized repositioning strategy sometimes hinder vehicles moving around due to its objective function, resulting in longer waiting time.

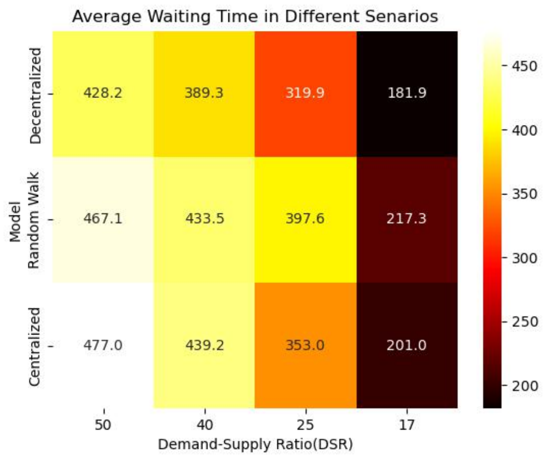


Figure 1: Waiting Time Heat Map

Scenario B: Demand Spatial Distribution Patterns

When the origin points become more clustered around a central location or multiple central locations, decentralized repositioning performs much better than the random walk state in terms of service rate and income. However, in scenarios where requests are more widely scattered, repositioning in a centralized manner results in significantly more advantageous driver income compared to the other two repositioning methods, which is presented in figure 2. In widely scattered conditions, decentralized repositioning does not necessarily help reduce the total reposition distance, yet in more concentrated situations the performance becomes better.

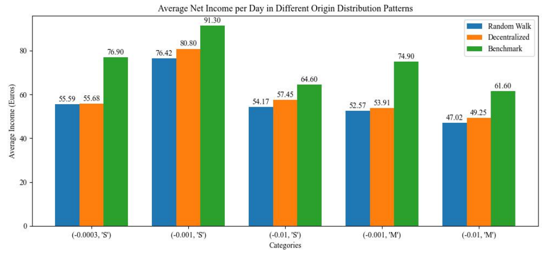


Figure 2: Average Driver Net Income under Different Demand Spatial Distribution Patterns

Decentralized repositioning, however, is better at reducing passenger waiting time when the requests are scattered widely, compared to centralized repositioning. Yet in extremely clustered situations, selecting the centralized method would benefit passengers more, reducing waiting time significantly, averagely 4 minutes less than that of decentralized repositioning. Thus, the decentralized model is better at dealing with situations where demand is distributed more evenly, which ensures passengers experiencing less waiting time. Yet for drivers, the centralized method would always be more attractive.

Scenario C: Platform Information

According to figure 3 and 4, decentralized repositioning has the highest service rate and least waiting time when the acceptance rate is 0.2, 0.4 or 0.6. At high acceptance rates, the waiting time would increase again but not exceed the condition that no platform information is provided. This is mainly because with platform information most passengers would experience shorter waiting time, yet overflow of information would cause over-supply in high demand areas, and insufficient idle vehicles in remote areas. This further leads to very low service rates, which falls even below the condition of drivers repositioning randomly. The acceptance rate does not necessarily affect the driver income, since although the service rate increases, the average reposition distance also increases.

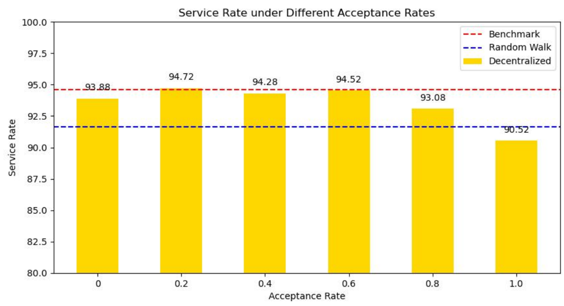


Figure 3: Service Rate under Different Acceptance Rate

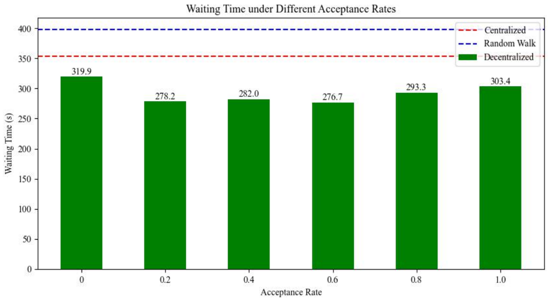


Figure 4: Waiting Time under Different Acceptance Rate

Thus, providing platform information seems less attractive to drivers, since the income is lower. Yet it is an efficient way to improve passengers' ride experience. The accessibility of the platform information has to be controlled by the platform, to avoid overflow of information, leading to even negative effects.

V Discussion and Conclusions

Decentralized repositioning has the highest efficiency when demand is scattered around a single center, when the dispersion rate is moderate, and when the DSR value is moderate. It is also better at reducing average waiting time compared with centralized model. Whether selecting a centralized or decentralized repositioning strategy highly depends on the objective one wants to achieve. It would also be wise to apply the methods interchangeably, which attract more passengers by allowing decentralized repositioning, while applying centralized repositioning strategies when driver population is insufficient. For platforms, it is also more efficient to make necessary information available to drivers, such as average waiting time, or number of rides accomplished in every hexagon. Yet the acceptance rate should not be too high, or there could be negative effects. By controlling the penetration rate of information to drivers could this value be adjusted.

For further research, more aspects could be included in drivers' evaluation process and learning process, to better simulate the actual driver behaviors. Future research could also consider making the supply and demand level dynamic throughout the simulation process. Unsatisfied drivers and passengers may quit, while new users may enter, which could better indicate the two-sided characteristics of ride-sourcing systems. Finally, the coding could be optimized to reduce the computation effort of decentralized method, which more steps of hexagons could be taken into account during the decision-making process.

References

Ashkrof, P., de A Correia, G. H., Cats, O., & van Arem, B. (2023). On the relocation behavior of ride-sourcing drivers. *Transportation Letters*. https://doi.org/10.1080/19427867.2023.2192581

Ashkrof, P., de Almeida Correia, G. H., Cats, O., & van Arem, B. (2022). Ride acceptance behaviour of ride-sourcing drivers. *Transportation Research Part C: Emerging Technologies*, *142*, 103783. https://doi.org/10.1016/j.trc.2022.103783

de Ruijter, A., Cats, O., Kucharski, R., & van Lint, H. (2022). Evolution of labour supply in ridesourcing. *The European Physical Journal B Transportmetrica B: Transport Dynamics*, *10*, 599–626. https://doi.org/10.1080/21680566.2021.2024917

Iglesias, R., Rossi, F., Wang, K., Hallac, D., Leskovec, J., & Pavone, M. (2018). Data-driven model predictive control of autonomous mobility-on-demand systems. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 1–7. https://doi.org/10.1109/ICRA.2018.8460966

Jiao, Y., Tang, X., Qin, Z., Li, S., Zhang, F., Zhu, H., & Ye, J. (2021). Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research: Part C*, *130*(103289). https://doi.org/10.48550/arXiv.2103.04555

Knobbe, D. (2022). *Impact of empty vehicle repositioning strategies on ride-hailing system performance.*

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

Lei, Z., Qian, X., & Ukkusuri, S. V. (2020). Efficient proactive vehicle relocation for on-demand mobility service with recurrent neural networks. *Transportation Research Part C: Emerging Technologies*, *117*, 102678. https://doi.org/10.1016/j.trc.2020.102678

Lin, K., Zhao, R., Xu, Z., & Zhou, J. (2018). Efficient large-scale fleet management via multi-agent deep reinforcement learning. *KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 1774–1783. https://doi.org/10.1145/3219819.3219993

Qian, X., Guo, S., & Aggarwal, V. (2022). Drop: Deep relocating option policy for optimal ride-hailing vehicle repositioning. *Transportation Research Part C: Emerging Technologies*, *145*, 103923. https://doi.org/10.1016/j.trc.2022.103923

Qu, M., Zhu, H., Liu, J., Liu, G., & Xiong, H. (2014). A cost-effective recommender system for taxi drivers. *KDD '14: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 45–54. https://doi.org/10.1145/2623330.2623668

Riley, C., van Hentenryck, P., & Yuan, E. (2020). Real-time dispatching of large-scale ride-sharing systems: Integrating optimization, machine learning, and model predictive control. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 4417–4423. https://doi.org/10.24963/ijcai.2020/609

Shou, Z., Di, X., Ye, J., Zhu, H., Zhang, H., & Hampshire, R. (2020). Optimal passenger-seeking policies on e-hailing platforms using markov decision process and imitation learning. *Transportation Research Part C: Emerging Technologies*, *111*, 91–113. https://doi.org/10.1016/j.trc.2019.12.005

Sirisoma, R. M. N. T., Wang, D., Lam, W. H. K., & Zhang, P. (2010). Empirical evidence for taxi customer-search model. *Proceedings of the Institution of Civil Engineers - Transport*, *163*, 203–210. https://doi.org/10.1680/tran.2010.163.4.203

Szeto, W. Y., Wong, R. C. P., Wong, S. C., & Yang, H. (2013). A time-dependent logit-based taxi customer-search model. *International Journal of Urban Sciences*, *17*, 184–198. https://doi.org/10.1080/12265934.2013.776292

Urata, J., Xu, Z., Ke, J., Yin, Y., Yang, H., & Ye, J. (2021). Learning ride-sourcing drivers' customer-searching behavior: A dynamic discrete choice approach. *Transportation Research Part C: Emerging Technologies*, *130*. https://doi.org/10.1016/j.trc.2021.103293

Valadkhani, A. H., & Ramezani, M. (2023). Dynamic ride-sourcing systems for city-scale networks, part ii: Proactive vehicle repositioning. *Transportation Research Part C: Emerging Technologies*, *152*(104159). https://doi.org/10.1016/j.trc.2023.104159

Yang, H., & Wong, S. C. (1998). A network model of urban taxi services. *Transportation Research Part B: Methodological*, *32*, 235–246. https://doi.org/10.1016/S0191-2615(97)00042-8