A Hierarchical Control Framework for Vehicle Repositioning in Ride-hailing Systems

Caio Vitor Beojone\textsuperscript{a}, Pengbo Zhu\textsuperscript{a}, Isik Ilber Sirmatel\textsuperscript{b}, and Nikolas Geroliminis\textsuperscript{a,*}

\textsuperscript{a}Urban Transport Systems Laboratory (LUTS), École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland
\textsuperscript{b}Control Section, Department of Electrical and Electronics Engineering, Faculty of Engineering, Trakya University, 22030 Edirne, Türkiye

\textbf{Keywords:} Vehicle Repositioning, Hierarchical Control, Model Predictive Control (MPC), Coverage Control, Macroscopic Fundamental Diagram (MFD)

This paper introduces a multi-layer control strategy for efficiently repositioning empty ride-hailing vehicles, aiming to bridge the gap between proactive repositioning strategies and micro-management. The proposed framework consists of three layers: an upper-layer employing an aggregated model based on the Macroscopic Fundamental Diagram (MFD) and model predictive control (MPC) to determine optimal vehicle repositioning flows between each pair of regions, a middle-layer converting macroscopic decisions into dispatching commands for individual vehicles, and a lower-layer utilizing a coverage control algorithm for demand-aligned positioning guidance within regions. The upper-layer contributes to the proposed framework by providing a global (macroscopic) view and predictive capabilities including traffic and congestion features. The middle-layer contributes by ensuring and optimal assignment of repositioning vehicles, considering the decision from the upper- and lower- layers. Finally, the lower-layer contributes with operational details at the intersection or node level providing the precision required for microscopic vehicle guidance. Experimental validation using an agent-based simulator on a real network in Shenzhen confirms the effectiveness and efficiency of the framework in improving empty vehicle repositioning strategies for ride-hailing services in terms of average passenger waiting time and abandonment rates.

1 Introduction

As an emerging transportation mode, ride-hailing service attracts more and more drivers and passengers, which occupies a significant portion of daily urban trips (Erhardt et al., 2019). However, spatiotemporal...
**Nomenclature**

**Sets**
- $\mathcal{R}$ and $\mathcal{R}_o$: set of all regions and set of adjacent regions to region $o$
- $Q$ and $Q^R$: set of all nodes in the graph and set of all nodes in region $R$
- $E$: set of edges in the undirected graph of the city map

**Functions**
- $q_{od}^K(t)$: inflow demand for class $K_{od}$ at time $t$
- $\lambda_{od}^K(t)$: exogenous demand input for class $K_{od}$ at time $t$
- $F(\cdot)$: compact form of the dynamics used in the upper-layer
- $F_P(\cdot)$ and $F_R(\cdot)$: compact form of the dynamics of private vehicles and ride-hailing vehicles
- $l(\cdot)$: objective function of the upper-layer, stage cost
- $l_e(\cdot)$: stage cost related to economic optimization
- $l_u(\cdot)$: regularization term for obtaining smoothly varying control inputs
- $D_R$: overall travel distance for inter- and intra-regional repositioning
- $G$: undirected graph of the city map
- $n$: Number of regions
- $\phi(\cdot)$: probability that a request starts at a node
- $V_i(p_i)$: Voronoi tessellation for current position $p_i$
- $H(P,V)$: coverage objective function
- $C(V_i)$: centroid of a graph Voronoi cell
- $d(\cdot, \cdot)$: shortest distance between two nodes in a graph
- $T_l$ and $T_u$: update period for the lower-layer and for the upper-layer

**Parameters**
- $L_{od}^K$: average trip length for class $K_{od}$
- $\theta_{ohd}$: transfer flow ratio (from region $o$ to $d$ via $h$)
- $\gamma$: loss probability parameter $i$
- $\omega$: waiting time tolerance
- $\rho$: weight on the regularization term $l_u(\cdot)$
- $u_0$: initial vector of inter-regional control inputs
- $t_{ir}^{out}$ and $t_{il}^{in}$: estimated traveling time for vehicle $i$ to reach region $r$ or the position $l$

**Variables**
- $n_{od}^K(t)$: accumulation state for class $K_{od}$ at time $t$
- $n_o(t)$: total accumulation in region $o$ at time $t$
- $v_o(t)$: speed as modeled by the speed-MFD of region $o$ at time $t$
- $O_{od}^K(t)$: total trip completion term at time $t$
- $O_{ohd}^K(t)$: transfer flow term at time $t$
- $u(t)$: vector of inter-regional control inputs at time $t$
- $u_{od}(t)$: inter-regional control input from region $o$ to $d$ at time $t$
- $c_{od}(t)$: number of vehicles that must reposition from region $o$ to $d$ at time $t$
- $\psi_{ir}$ and $\eta_{il}$: whether or not vehicle $i$ is assigned to region $r$ or to position $l$
- $y$: node in the graph of the city map
variations in demand can create supply imbalances between drivers and passengers, manifesting as deterioration of system efficiency and service quality (Mo et al., 2022). For instance, such demand variations may result in discrepancies between origins of new trips and destinations of past trips. Therefore, it is expected that an efficient fleet management strategy, that keeps drivers well-distributed in space and time over the served area, is of critical importance to provide satisfying mobility service.

The majority of the existing literature on repositioning of ride-hailing vehicles (RHVs) focus on the regional level, and consequently are only capable of providing decisions regarding the macroscopic traffic flows between different neighborhoods. For example, Ramezani and Nourinejad (2018) uses Model Predictive Control (MPC) to reposition idle taxis in a macroscopic set of regions. Differently, from a microscopic perspective, Zhu et al. (2022a) propose coverage control to proactively position idle drivers in areas more likely to originate new requests. While both methods use control techniques, Zhu et al. (2022a) focuses on microscopic movements and instructions, and Ramezani and Nourinejad (2018) focuses on the regional and temporal imbalances of demand. However, the development of a comprehensive traffic management framework that allows for the development of a repositioning strategy, which links macroscopic decision-making processes to detailed repositioning instructions for individual vehicles, remains as an area for improvement in ride-hailing fleet management.

In vehicle repositioning applications, a forecast of future conditions is crucial for deploying the fleet proactively towards improved performance. The complexity of the operations due to large fleet size, demand uncertainty and spatiotemporal heterogeneity in the distribution of congestion make the control problem challenging. The first challenge is to coordinate the movements of all vehicles. For instance, instructions causing coverage overlap decrease system capacity for serving incoming requests and should be avoided. In large-scale environments with fleets of hundreds or thousands of vehicles, the complexity of any proposed solution can quickly turn the problem unsolvable or, at least, infeasible for real-time implementation (Zhang et al., 2016). The idea of breaking the repositioning problem into different layers, with different scales and scopes each, appears as a plausible option. However, such an approach can raise concerns about the possibility of balancing and designing the different layers so as to ensure both computational efficiency and good system performance. Addressing such large-scale management problem for complex traffic systems appear to require appropriate integration of traffic modeling and control methods.

We propose a hierarchical control strategy for the repositioning of idle RHVs, for addressing the gap in the literature related to the integration of proactive macro-repositioning strategies and micro-management of vehicles partaking in such activities. The upper-layer utilizes an aggregated model, obtained via macroscopic traffic modeling methods (using macroscopic fundamental diagram – MFD). A model predictive control (MPC) framework is employed to determine the number of idle vehicles to be repositioned between each pair of regions. Unlike MPC methods with perimeter control actuation for improving general traffic congestion, MPC for fleet management requires more sophisticated MFD-based models describing mixed dynamics of private vehicles and taxis. In the lower-layer, given the demand density over the current region, a coverage control scheme is employed to distribute the vehicles within the region to achieve a demand-aligned configuration, which provides each vehicle with relatively detailed (i.e., intersection/node-level) position guidance. To bridge the two layers, a middle-layer mechanism is developed for converting the upper-layer decisions into dispatching commands for individual vehicles by solving an assignment problem, which minimizes the distance required to achieve the optimal coverage and repositioning decisions.

Hierarchical structures for fleet management in ride-hailing services usually associate vehicle repositioning with the matching and dispatching processes, which on one hand can bring efficiency. On another hand, however, it loses flexibility by constraining service processes to a single matching strategy in an environment with new service options appearing regularly. Hence, the novelty of the contributions can be
summarized as follows:

1. Development of a unified RHV fleet repositioning framework via integrating vehicle-level (i.e., coverage control) and network-level (i.e., MPC) methods. Although versions such methods can be found separately in the literature (e.g., coverage control for RHVs in Zhu et al. (2022a), MPC for RHVs in Ramezani and Nourinejad (2018), and prediction model in Beojone and Geroliminis (2023a)), integration of the methods towards a real-world and real-time applicable RHV repositioning system remains unexplored. A purely vehicle-level RHV fleet control system might suffer from either excessive computational effort (if centralized) or lack of spatial coordination (if decentralized), while a purely network-level control system is susceptible to intra-regional uncertainties and is not directly field-applicable without mechanisms providing field interface. These points emphasize the significance of the development of a unified RHV fleet control framework, which is the main contribution of the paper.

2. Design of a region-level vehicle assignment mechanism that enables realizing network-level commands at the physical (i.e., vehicle) level. This optimization-based mechanism, operating at the middle-layer between the coverage control and MPC, enables realization of MPC commands by selecting and dispatching individual vehicles to be repositioned. The optimization not only ensures the realization of the upper-layer decisions but also minimizes the associated costs for inter- and intra-regional repositioning.

3. Adaptation of a coverage controller to operate only within the boundaries of a region in the urban-area, instead of the whole network to ensure spatial coordination.

4. The design of a repositioning-exclusive framework (not linked to the matching process), allows for an easier integration into the entire service framework as a module, alongside other modules, such as the matching algorithms. For instance, this modular aspect does not constrain the matching algorithm to work with only single-passenger rides, and allows for elaborate and highly optimized matching strategies with multiple passengers assigned to each vehicle.

The remainder of the paper is structured as follows. Firstly, Section 2 provides a brief review of the literature on ride-hailing fleet management with a focus on the repositioning process. Then, we provide an overview of the proposed hierarchical control framework for vehicle repositioning. More specifically, in Section 3.1, we present an MFD-based dynamical model, which is leveraged for designing an economic nonlinear MPC scheme as the upper-layer controller. To bridge the macroscopic commands and microscopic implementations, the middle-layer is introduced in Section 3.2 as a mechanism involving solutions to an optimal assignment problem. Then, in Section 3.3, we describe the coverage control algorithm, which is capable of guiding individual idle vehicles within each region towards spatial coverage-optimal configurations, as the lower-layer controller. The performance of the presented framework is validated in Section 4 with an agent-based simulator utilizing a computer model of the real traffic network of Shenzhen. Finally, in Section 5, we provide concluding remarks, discussions, and suggestions for future work.

2 Related literature

The literature explored ride-hailing fleet management in several directions (Yu et al., 2023). One direction extensively explored involves improving passenger-driver matching algorithms to enhance service quality
and capacity (Alonso-Mora et al., 2017; Simonetto et al., 2019; Liu and Samaranayake, 2020; Ramezani and Valadkhani, 2023). Other directions are not explored as extensively, such as the cruising behavior of these fleets (Chen et al., 2021). One can also include the study of regulatory schemes (Ke et al., 2021; Bimpikis et al., 2019; Ke et al., 2020a; Zha et al., 2016) and the study of the societal externalities associated with the ride-hailing service (Beojone and Geroliminis, 2021). For instance, Alisoltani et al. (2021) and Ke et al. (2020b) evaluate the potential congestion alleviation by dynamic ride-pooling. In Wei et al. (2020) the authors focused on understanding the labor supply and multi-modal traveler decision making with aggregated traffic dynamics.

### 2.1 Repositioning of ride-hailing vehicles

Repositioning of vehicles in ride-hailing services has been investigated through a diverse number of strategies and mechanisms. It is worth mentioning the implementation of incentives for drivers (Powell et al., 2011; Lu et al., 2018; Sadeghi and Smith, 2019) in the form of mechanisms such as surge pricing, high-demand area flags (Castillo et al., 2018; Zha et al., 2018), or repositioning incentives with revenue forecasting (Beojone and Geroliminis, 2023b).

Differently, others studied methods involving dispatching empty RHVs to the locations of recently unserved customers (Alonso-Mora et al., 2017; Wallar et al., 2018; Simonetto et al., 2019; Liu and Samaranayake, 2020) to avoid losing future requests. However, a limitation of these approaches is their reactive nature, as they primarily rely on past events or the current situation. Consequently, customers may opt for other more reliable transportation modes instead of ride-hailing services if they consistently experience service losses in a particular area.

#### 2.1.1 Inter-regional repositioning

Since it is hard to predict the exact location of future requests in ride-hailing services, repositioning strategies usually aggregate demand information into zones or regions, where vehicles are dispatched to search their next assignments.

One form of aggregating demand information is to import the concept of stations present in carsharing problems. Zhang and Pavone (2016) proposed to solve the repositioning of RHVs using a queueing network model. An MILP computes the repositioning decisions in the form of virtual demand flows in the queueing network, such that vehicles are not forced to reposition. The study also presents a recurrent characteristic of repositioning problems in the literature, that is the objective to minimize the number of repositioning vehicles to achieve vehicle availability balance in the network. This objective comes as a response to avoid creating additional traveling in a likely already congested network. As an extension of the previous work, Zhang et al. (2016) reposition vehicles using a MPC controller in a station-based system. It also addresses electric vehicle needs, such as considering battery charging constraints. Another difference is the objective function proposed, which primarily tries to serve all of the waiting customers as quickly as possible, and only secondarily to avoiding congestion.

The most popular method to aggregate demand in repositioning problems, however, is to separate the service area in zones. Among the benefits of such approach is to incorporate into the problem the notion that the service area is a continuous space, not discrete (Wen et al., 2017). In Guo et al. (2021) matching is incorporated into the repositioning problem, such that the matching component is modeled in a zone level too. The idea is to minimize the deadheading, summing the repositioning distance and the later distance to
pick-up passengers into the total cost function.

Given the concern with the potential additional traveling associated to repositioning and the idea to aggregate demand in zones, MFD-based modeling and control of ride-hailing systems have attracted increased interest, branching from the initial efforts in traffic control (Daganzo, 2007) to recent efforts in route guidance using deep reinforcement learning (Jiang et al., 2024). Building from many methods and considering various actuation approaches, such as perimeter control, congestion pricing, route guidance, or a combination of these; recent advances in MFD-based modeling have made it possible to capture the coupled dynamics of private vehicles and ride-hailing services, and thus predict their future behavior. As an example of use in repositioning, in Ramezani and Nourinejad (2018), an accumulation-based model was introduced with the aim of predicting taxi movements in an effective taxi repositioning strategy. The method was extended in Nourinejad and Ramezani (2020) to accommodate pricing structures within the ride-hailing services, allowing both drivers and passengers to make decisions of either remaining with or discontinuing the service.

Apart from the previous methods, which rely on prediction models for the demand, or graph modeling and optimization, the advance of machine learning and other data-based approaches emerge in the problem of repositioning RHVs among different zones as a solution to large-scale problems that require real-time solutions. For instance, a data-enabled predictive control algorithm for empty vehicle repositioning was proposed in Zhu et al. (2023), which utilizes historical data to guide repositioning policies and employs a non-parametric representation to predict future actions, thereby requiring no system modeling effort. Wen et al. (2017) used reinforcement learning to solve a repositioning problem in order to achieve large-scale real-time applications, while also accounting for shared rides (multiple passengers per vehicle). Lei et al. (2020) repositioned vehicles using an optimal and stable matching problem to train a machine learning algorithm, which is later used to predict future repositioning policies. Guo et al. (2023) used a data-driven method to perform vehicle repositioning optimization directly from the contextual data leveraging the prediction problem to integrate it with optimization models.

2.1.2 Intra-regional repositioning

Despite the uncertainty in future demand positions, some approaches delve into the local movements of RHVs while searching for new assignments, generally referred as their cruising behavior. For instance, the use of Markov Decision Processes (MDP) is explored to understand drivers and their decision-making in the search for next passengers, providing cruising directions, and allowing drivers to maximize their profits (Shou et al., 2020; Zhou et al., 2020; Yu et al., 2019). Other control applications provided improved service quality. Zhu et al. (2022b) proposed coverage control to proactively position idle drivers in areas more likely to originate new requests. In a different direction, Chen et al. (2021) proposed a decentralized cooperative cruising method for autonomous fleets as a contingency during a full communication shutdown with an objective of maximizing the number of served passengers.

Given the street-hail nature of taxi services, the cruising strategy has a direct impact on their matching opportunities. Therefore, Wong et al. (2014) proposed a cell-based model to predict cruising movements of taxi drivers, incorporating principles of logit-based search and intervening opportunity. Wong et al. (2015) extended the previous cell-based model to predict taxi cruising behavior in two stages, including the zone choices and the circulation time and distance in the chosen zone. Duan et al. (2020) designed autonomous dispatchers to plan short- and long-term routes for taxis in real time.
2.2 Hierarchical fleet management strategies

Particularly to the fleet management operations, both matching and repositioning processes are optimized to improve service quality, in terms of number of passengers served and waiting times. In a continuously growing and denser service, the combination of these problems become challenging to solve in real-time.

Therefore, hierarchical structures for fleet management in ride-hailing services usually combine the vehicle repositioning and the matching and dispatching processes. For instance, Chen et al. (2017) proposed a hierarchical framework, where a higher level optimizes the idle mileage by repositioning vehicles, while a lower level scheduled the passenger-vehicle assignments in each region. The already mentioned works of Ramezani and Valadkhani (2023), and Valadkhani and Ramezani (2023) had a similar hierarchical structure, where Nonlinear Model Predictive Control is used to reposition vehicles at a higher level, while in the lower level, an adaptive spatio-temporal matching method computes the assignments constrained by the decisions of the higher level decisions (e.g. some vehicles cannot be used for matching because they are repositioning). Guériaud and Dusparic (2018) proposed a reinforcement learning-based decentralized approach to vehicle repositioning as well as ride request assignment in shared mobility-on-demand systems. Qian et al. (2022) used a hierarchical deep reinforcement learning algorithm with pseudo rewards to improve the effectiveness of a vehicle’s repositioning. The hierarchy is in the training of the reinforcement learning algorithm, where one generates directions and the other one refines them to lower dithering. Tuncel et al. (2023) simultaneously optimized matching and repositioning in a service with shared rides. The idea is built on shareability and trip-vehicle graphs (Santi et al., 2014; Alonso-Mora et al., 2017). Interestingly, it considers the supply contribution of the repositioning route (not just the target region). In summary, it integrates the matching process from Alonso-Mora et al. (2017) with the addition of the repositioning cost in the optimization function, which tries to minimize the deviation from the desired supply level in a region.

Typically, the repositioning process in these hierarchical structures is solved solely with an inter- or intra-regional perspective, facing the limitations of each of them. Ding et al. (2022) presents a hierarchical structure with macroscopic and microscopic decisions. The macroscopic decisions are solely related to repositioning, while the microscopic decisions combine the vehicle’s cruising (local repositioning) with the matching and dispatching policy. The limitation here becoming the combination of the cruising and matching processes, which in one hand can bring efficiency, but loses the flexibility of allowing elaborate matching processes or highly specialized cruising strategies.

3 Hierarchical control framework for vehicle repositioning

In implementing a controller for a large-scale system, one may face problems such as high computational effort due to complex models and high dimensions required for accurate network modeling, especially if the model and controller are developed to compute control actions for every individual RHV over the whole network. One way to solve this problem is to build a hierarchical control structure. Such structures decompose the control problem into a hierarchy of decision-making levels, and operate via coordinating between the actions of an upper-layer controller (operating at the aggregated traffic level) and a lower-layer controller (managing individual RHVs). The control structure is shown in Figure 1.

The upper-layer controller collects aggregated information, such as how many empty RHVs are in each region, from all urban regions at a relatively large update period $T_u$. The control action generated from the upper-layer determines how many RHVs should stay in current regions and how many RHVs should reposition to other regions (any regions in the system), in order to improve availability and thus minimize the
Figure 1: A hierarchical control framework for RHV repositioning.

total waiting time of passengers. Furthermore, the middle-layer transfers the obtained upper-layer guidance to the lower-layer and specifies which RHV should stay or move, considering the travel costs caused by repositioning. It is operated within each region and requires relatively more detailed information, such as the coordinates of each RHV and whether it is occupied or not. Note that the middle-layer can only be activated when the upper-layer is active. The lower-layer is operated in a distributed manner so that each RHV can obtain its own control action, which facilitates its implementation at a fast update period $T_l$. The empty RHVs that are commanded to stay in the current region (i.e., idle RHVs, see the left part of lower-layer in Figure 1) communicate and cooperate with each other to achieve better RHV position configuration, while the rest of the RHVs (i.e., repositioning RHVs, see the right part of the lower-layer in Figure 1) are be guided to other desired regions as per the repositioning commands. All these decisions are summarized in Figure 2 for each layer.

3.1 Upper-layer: Prediction model and MPC controller

The designed framework incorporates urban traffic dynamics, tracking private vehicle activities, which formed the majority of background traffic (Ramezani and Nourinejad, 2018). In such a case, we can track vehicles in the system based on their ongoing activities and regional movements, as summarized in Table 1.

Note that vehicles in class $B$ are assumed to be completely busy and cannot receive new assignments, while vehicles in class $V$ are considered available. For simplicity, we assume that when private vehicles are not driving, they park outside the street and do not interfere with traffic congestion. Therefore, their only activity considered in the model is driving, while any other one (e.g., cruising for parking) is not considered.

The formulation of the prediction model follows an accumulation-based model, allowing for the formu-
Table 1: List of vehicle classes based on their activities.

<table>
<thead>
<tr>
<th>Description</th>
<th>Notation</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vacant (Idle or Repositioning)</td>
<td>V</td>
<td>V_{od}</td>
</tr>
<tr>
<td>Busy</td>
<td>B</td>
<td>B_{od}</td>
</tr>
<tr>
<td>Private vehicle</td>
<td>P</td>
<td>P_{od}</td>
</tr>
</tbody>
</table>

Hence, total trip completion terms $O_{od}^K(t)$ (from region $o$ to $d$) and transfer flow terms $\theta_{ohd}$ (from region $o$ to $d$ via $h$) can be written as follows:

$$O_{od}^K(t) = n_{od}^K(t) \frac{v_o(n_o(t))}{L_{od}^K}$$ (1)

$$O_{ohd}^K(t) = \theta_{ohd} O_{od}^K$$ (2)
where $K$ refers to the activity, as depicted in Table 1 and $od$ stands for the regional Origin-Destination pair of the vehicle, such that their combination $K_{od}$ represents their class. Therefore, $n_{od}^K$ and $L_{od}^K$ become the accumulation state and the average trip length for class $K_{od}$, respectively, whereas $v_o(n_o(t))$ is the speed as modeled by a speed-MFD of region $o$, expressing the space-mean speed at accumulation $n_o(t)$, which expresses the total accumulation in region $o$. Note that the trip length $L_{od}^K$, refers to all the distance traveled by the vehicle in its respective class. Therefore, for busy RHVs, it comprehends the distance traveled from the moment of the assignment until dropping the passenger at the destination, or reaching the border of the next region on its path to deliver the passenger. Similarly, in the case of the background traffic, the trip length refers to the distance traveled by the vehicle from its origin (when it enters the system) until its the destination, or the regional border (therefore leaving the system or the region). Since idle RHVs are assumed to park, they do not have a trip to complete, nor a trip length, by consequence.

Boundary capacity between the sending and receiving regions can have adverse effects on transfer flows, as high accumulations in the receiving region might restrict inflow reception at the boundary between the flow-exchanging regions. However, the constraints representing the boundary capacity effect can be omitted from the formulations for computational advantage. This omission can be justified based on the following points: 1) The boundary capacity is expected to decrease only for accumulations much larger than the critical accumulation, and 2) a network-level congestion controller (which can, in principle, be operated together with, or combined with, the RHV fleet controller) will not allow the systems to approach such high accumulations (see Haddad et al. (2013)). With this reasoning, we omit the boundary capacity effects in the formulations as the focus of the study is on RHV fleet operations instead of congestion control (see Sirmatel and Yildirimoglu (2023)) for a study where boundary capacity is considered for congestion control design).

Using the definitions of transfer flows given in eq. (1) and eq. (2), we can formulate the accumulation dynamics for the three classes of vehicles. Accumulation dynamics of private vehicles can be formulated as follows (mainly based on Ramezani et al. (2015)):

\[ \dot{n}_{oo}^P(t) = q_{oo}^P(t) + \sum_{h \in \mathcal{R}_o} O_{ho0}^P(t) - O_{oo}^P(t) \quad (3a) \]
\[ \dot{n}_{od}^P(t) = q_{od}^P(t) + \sum_{h \in \mathcal{R}_o, h \neq o} O_{hod}^P(t) - \sum_{h \in \mathcal{R}_o} O_{ohd}^P(t) \quad o \neq d, \quad (3b) \]

where $n_{od}^P(t)$ (veh) and $q_{od}^P(t)$ (veh/h) are the accumulation state and inflow demand, respectively, for private vehicles in region $o$ destined to $d$, with $\mathcal{R}_o$ denoting the regions adjacent to $o$ ($\mathcal{R}$ is the set of all regions in the network).

Accumulation dynamics of busy RHVs can be formulated as follows:

\[ \dot{n}_{oo}^B(t) = q_{oo}^B(t) + \sum_{h \in \mathcal{R}_o} O_{ho0}^B(t) - O_{oo}^B(t) \quad (4a) \]
\[ \dot{n}_{od}^B(t) = q_{od}^B(t) + \sum_{h \in \mathcal{R}_o, h \neq o} O_{hod}^B(t) - O_{od}^B(t) \quad o \neq d, \quad (4b) \]

where $n_{od}^B(t)$ (veh) and $q_{od}^B(t)$ (veh/h) are the accumulation state and trip demand, respectively, for busy RHVs in region $o$ destined to $d$. For busy RHVs, the trip demands $q_{od}^B(t)$ depending on the availability of vacant RHVs, as modeled in Equation (5a). For private vehicles, an exogenous demand input is added to
The unserved passengers, to keep the number of generated trips, as in Equation (5b), based on Beojone and Geroliminis (2023a) using loss probabilities:

\[
q^B_{od}(t) = \lambda^B_{od}(t)\left(1 - \exp\left(-\gamma_0 n^V_o(t)\kappa_v(t)\omega^\gamma\right)\right)
\]

\[
q^P_{od}(t) = \lambda^P_{od} + \lambda^B_{od}(t)\exp\left(-\gamma_0 n^V_o(t)\kappa_v(t)\omega^\gamma\right)
\]

where \(\lambda^P_{od}(t)\) and \(\lambda^B_{od}(t)\) are exogenous demand inputs for private and busy RHVs, respectively. The expression \(\exp\left(-\gamma_0 n^V_o(t)\kappa_v(t)\omega^\gamma\right)\) is the representation of the probabilities of losing an incoming request in the matching algorithm (operated in a microscopic scale) into convenient functional form for a macroscopic model, only used within the prediction module of the upper-layer. Therefore, the parameters \(\gamma_i, i \in \{0, 1, 2, 3\}\) translate the sensitivity of the matching algorithm to the number of vacant RHVs in region \(o\) (\(n^V_o(t)\)), traveling speeds (\(v_o(t)\)) and passenger waiting time tolerance (\(\omega\)), in terms of probability of losing an arriving assignment. Note that different operations can employ different matching algorithms, such as assigning the closest RHV immediately at the arrival (Hanna et al., 2016), or performing batch assignments to optimize the service performance (Qin et al., 2021). Similar approaches for translating the matching process into a functional form in macroscopic models can be seen in Li et al. (2021) and Beojone and Geroliminis (2023a) and the references therein.

To compute the loss probability, \(\lambda^B_{od}(k)\) is an exogenous demand input for busy RHVs; the remaining \(\gamma_i, i \in \{0, 1, 2, 3\}\) are parameters expressing the sensitivity of the matching algorithm to the number of vacant RHVs (\(n^V_{o,k}\)), traveling speeds (\(v_{o,k}\)) and passenger waiting time tolerance (\(w\)). Beojone and Geroliminis (2023a) suggested the use of a Monte Carlo simulation followed by a linear regression model to estimate parameters \(\gamma_i, i \in \{0, 1, 2, 3\}\), and reported elevated goodness of fit, reaching a \(R^2 = 0.96\) for a single region case.

Accumulation dynamics of vacant RHVs can be formulated as follows:

\[
\dot{n}^V_{oo}(t) = O^B_{oo}(t) - \sum_{d \in \mathbb{H}} \left(\frac{n^V_{od}(t)}{n^V_o(t)} q^B_{od}(t) + u_{od}(t)\right) + \sum_{h \in \mathbb{H}_o} O^V_{ho}(t)
\]

\[
\dot{n}^V_{od}(t) = u_{od}(t) + \sum_{h \in \mathbb{H}_o} O^V_{ho}(t) - O^V_{od}(t) - \frac{n^V_{od}(t)}{n^V_o(t)} \sum_{h \in \mathbb{H}} q^B_{oh}(t) \quad o \neq d
\]

where \(u_{od}(t)\) (veh/h) is the repositioning control input, expressing the number of vacant RHVs commanded to start their repositioning trips from region \(o\) to \(d\). Note that the accumulation \(n^V_{od}(t)\) denotes the number of vacant RHVs that are present in region \(o\) and ready to respond to trip requests (i.e., these RHVs stay in region \(o\)), while the accumulation \(n^V_{od}(t)\) denotes the number of vacant RHVs that are currently repositioning from \(o\) to \(d\), which are also available for trip request response, sharing the incoming requests with the other vacant RHVs in region \(o\).

The dynamic equations given in Equations (3) to (6) can be discretized in time for facilitating predictive control design. Writing them in compact form, we obtain the following vector nonlinear difference equation:

\[
x(k + 1) = F(x(k), q(k), u(k)),
\]

where \(k \in \mathbb{N}_0\) is the time step of sampled real time (i.e., \(t(k) = T \cdot k\)) (with sampling time \(T\) (h)), \(x(k) \in \mathbb{R}^{nx}\) (state) is a vector containing all state variables (i.e., accumulation states \(n^V_{od}(k), n^P_{od}(k), \text{ and } n^V_{od}(k)\), \(q(k) \in \mathbb{R}^{nq}\) (measured disturbance) is a vector containing all exogenous demands (i.e., \(q^B_{od}(k)\) and \(q^P_{od}(k)\)), whereas \(u(k) \in \mathbb{R}^{nu}\) (control input) a vector containing all repositioning control input terms \(u_{od}(k)\).
For real road traffic networks where the total number of vehicles in the network is usually in the tens of thousands, the RHV fleet is expected to be only a relatively small part of the whole network (e.g., up to about 10% of the total). Dynamics of RHVs thus have a roughly negligible effect on the overall traffic. Observing this aspect, for numerical advantage in control design we can split the prediction model Equation (7) into two decoupled parts, namely that of private vehicles and RHVs, as follows:

\[
x_{p}(k+1) = F_{p}(x_{p}(k), q_{p}(k)) \quad \text{(8a)}
\]
\[
x_{R}(k+1) = F_{R}(x_{R}(k), q_{R}(k), u(k), v(k)) \quad \text{(8b)}
\]

where \( F_{p}(\cdot) \) is the function expressing dynamics of private vehicle accumulations (i.e., dynamics of \( n_{p}^{V} \)) and \( F_{R}(\cdot) \) is that of the RHVs (i.e., dynamics of \( n_{od}^{V} \) and \( n_{od}^{B} \)). Similarly, the vectors \( q_{p}(k) \) and \( q_{R}(k) \) contain the inflow demands for private and ride-hailing trips, respectively, the vectors \( x_{p}(k) \) and \( x_{R}(k) \) contain the accumulation states for the same, \( u(k) \) is the vector of repositioning control inputs, while \( v(k) \) is the vector of regional space-mean speeds. With such a reformulation of the dynamics in two parts, we can create a forward simulator that predicts future traffic variables of interest, namely the regional speeds, for a prediction horizon of \( N \) (to be chosen equal to MPC prediction horizon) to be used as input to the MPC scheme. The reasoning here is that, the MPC decisions related to repositioning have a very small effect (due to the aforementioned point about RHV fleet being a small portion of the total number of vehicles) on the evolution of private vehicle dynamics, thus these can be taken outside of the MPC problem (and the space-mean speeds can be made an input to the problem) with negligible effect to prediction (and thus, control) performance.

We can formulate the problem of finding the optimal repositioning control input values that maximize the number of served passengers as the following discrete-time economic nonlinear MPC problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{\kappa=0}^{N-1} I(x_{R,\kappa}, u_{\kappa}, v(k+\kappa)) \\
\text{subject to} & \quad x_{R,0} = x_{R}(k) \\
& \quad \text{for } \kappa = 0, \ldots, N - 1: \\
& \quad x_{R,\kappa+1} = F_{R}(x_{R,\kappa}, q_{R}(k+\kappa), u_{\kappa}, v(k+\kappa)) \\
& \quad n_{oo}^{V} \leq n_{oo,\kappa}^{V} \text{ for } o \in \mathcal{R}, \\
& \quad 0 \leq Tu_{od,\kappa}^{V} \leq n_{od,\kappa}^{V} \text{ for } o, d \in \mathcal{R}, \\
& \quad 0 \leq u_{od,\kappa} \leq \bar{u}_{od,\kappa} \text{ for } o, d \in \mathcal{R} \\
\end{align*}
\]

where \( \kappa \) is the MPC time interval counter (i.e., internal time step of the MPC), \( k \) is the current time step (of the real, i.e., wall-clock, time), \( N \) is the prediction horizon, \( x_{R}(k) \) is the RHV accumulation state at time step \( k \), \( F_{R}(\cdot) \) is the function expressing the prediction model for RHV accumulation dynamics given in Equation (8b), \( x_{R,\kappa} \) and \( u_{\kappa} \) are the predicted RHV accumulation state and repositioning control input vectors, respectively, \( q_{R}(k) \) is the ride-hailing demand estimate, \( v(k) \) is the space-mean speeds vector predicted by the forward simulator using Equation (8a), the constraint in Equation (9e) reflects that the number of available vacant RHVs in a region should not fall below a fixed threshold \( n_{oo}^{V} \) (veh), the constraint in Equation (9f) expresses that the total repositioning for a time step cannot exceed available RHVs, while the constraint in Equation (9g) specifies the limits of repositioning inputs.

The MPC objective function \( \sum_{\kappa=0}^{N-1} I(x_{R,\kappa}, u_{\kappa}, v(k+\kappa)) \) reflects the control task of the network-level controller, which is minimizing the weighted sum of two terms, the main one related to maximizing RHV

\footnote{1We present an argument for this assumption in the Appendix A.}
system performance and the secondary one related to avoiding severe variations (due to operational concerns) in the repositioning inputs, for a finite time horizon. Due to the objective function not being positive definite with respect to a fixed point, this is an economic MPC formulation, where the concern is to control a dynamical system by minimizing an arbitrary objective function, in contrast to regulation/tracking MPC involving a standard control-oriented objective function which is positive definite with respect to a point (see Sirmatel and Geroliminis (2021) for a discussion on economic and standard control objectives for MFD-based predictive congestion control).

The objective function is simply the stage cost $l(x_t, u_t, v(k + \kappa))$ summed over the finite time horizon, while the stage cost consists of the two terms:

$$l(x_t, u_t, v(k + \kappa)) = l_e(x_t, u_t, v(k + \kappa)) + \rho l_u(u_t), \quad (10)$$

with the first term $l_e(x_t, u_t, v(k + \kappa))$ expressing the economic optimization objective of MPC as maximizing number of served passengers, while the second term $l_u(u_t)$ is a regularization term for yielding control inputs varying smoothly in time, where $\rho$ is a scalar expressing the weight on the regularization term.

The stage cost term $l_e(x_t, u_t, v(k + \kappa))$ related to economic optimization can be detailed as follows: From eq. (1) we can see that the dropping-off flows for busy RHVs can be written as follows (using MPC predicted states; note that the speed term $v(k + \kappa)$ is an input, i.e., fixed parameter, of the MPC problem, since it is obtained via forward simulation using eq. (8a)):}

$$l_e(x_t, u_t, v(k + \kappa)) = \sum_{o \in \mathcal{R}^B} n^n_{oo,\kappa} \frac{v_o(k + \kappa)}{L^n_{oo}}, \quad (11)$$

which is equal to the flow of passengers reaching their destination. Thus, maximizing this term (summed over time) would be equivalent to optimizing RHV system performance, as this would lead to maximizing total number of served passengers over the operation time. Note that MPC is a real-time repeated optimal control method, thus the idea is to optimize the objective function formulated for a finite horizon repeatedly (by shifting the horizon forward) and applying the obtained optimal control inputs to the system, and thus obtaining a feedback control system, with the goal of minimizing the objective function for the whole operation period in time, i.e., the infinite horizon version of the objective function, although the infinite horizon objective does not appear in the formulation.

The second term $l_u(u_t)$ is a regularization term for obtaining control inputs varying smoothly in time:

$$l_u(u_t) = (u_0 - u(k - 1))^T(u_0 - u(k - 1)). \quad (12)$$

This second term expresses the secondary objective of smooth variations of the inputs in time, which can be important for operational reasons where the RHV fleet operator may not want apply severe changes in number of repositioned RHVs from the previous solution $u_0$.

The problem given in Equation (9) is a nonconvex (due to the nonlinear prediction model) nonlinear optimization problem, which can be solved efficiently via optimization solvers based on sequential quadratic programming or interior point methods.

### 3.2 Middle-layer: Selecting and dispatching vehicles

Once the upper-layer provides the number of RHVs transferring between regions, a further step is to select which RHVs should move to other desired regions (i.e., repositioning RHVs). Simultaneously, vacant
RHVs staying in the current region (i.e., idle RHVs) are operated to maintain a good spatial configuration to uphold service quality. In summary, for each region $R$, the middle-layer is responsible for bridging the results from the upper- and lower- layers, such that inter- and intra-regional instructions are optimized.

The middle-layer of our control framework plays a role in coordinating actions by integrating information from both the upper- and lower- layers. Firstly, it follows the upper-layer commands: the number of RHVs that should remain in region $R$ and the number of RHVs to be relocated to each other region. And from the lower-layer, it needs the set of target intersections that the RHVs remaining in region $R$ are expected to locate at.

In the middle-layer, the number of RHVs to be assigned to region $d$ from their current region $R$ is denoted $c_{Rd}$, which is derived from the upper-layer control input $u_{Rd}$, following the formula $c_{Rd} = \lfloor u_{Rd}(k) \cdot T_u \rceil$, $\forall d \in R$ (i.e., rounding the product $u_{Rd}(k) \cdot T_u$ to the nearest integer). Particularly, $c_{RR}$ is the number of RHVs that should remain in the current region $R$.

Note that the middle-layer actuates before the actions of the coverage controller in the lower-layer. Therefore, this precedence requires an estimated optimal configuration for RHVs that remain in region $R$. To achieve this, we randomly select $c_{RR}$ RHVs in region $R$, whose positions are then used as initial positions in computing the centroids that maximize coverage, a process elaborated in Section 3.3.

Then, with this information, the middle-layer is able to perform the assignment of RHVs by solving the optimization problem in eq. (13), complying with the upper-layer decision, while also minimizing the costs for achieving optimal coverage at the lower-layer afterwards. Specifically, for objective $D_R$ of minimizing the overall travel time, the first term considers the reposition time between regions and the second term takes the intra-regional traveling time into account. In other words, the inter-regional times refer to the costs associated with the actuation of the upper-layer; and the intra-regional times refer to the costs associated with the actuation of the lower-layer.

\[
\begin{align*}
\text{minimize} & \quad D_R = \sum_{i \in I} \sum_{r \in R} \psi_{ir} t_{ir}^{\text{out}} + \sum_{i \in I} \sum_{l=1}^{c_{RR}} \eta_{il} t_{il}^{\text{in}} \\
\text{subject to} & \quad \sum_{r \in R} \psi_{ir} + \sum_{l=1}^{c_{RR}} \eta_{il} = 1 \quad \forall i \in I \\
& \quad \sum_{i \in I} \psi_{ir} = c_{Rr} \quad \forall r \in R \setminus \{R\} \\
& \quad \sum_{i \in I} \eta_{il} = 1 \quad \forall l \in \{1, \ldots, c_{RR}\} \\
& \quad \sum_{i \in I} \sum_{l=1}^{c_{RR}} \eta_{il} = c_{RR} \\
\psi_{ir} & \in \mathbb{B} \quad \forall i \in I \text{ and } \forall r \in R \\
\eta_{il} & \in \mathbb{B} \quad \forall i \in I \text{ and } \forall l \in \{1, \ldots, c_{RR}\}
\end{align*}
\]  

where $\psi_{ir}$ and $\eta_{il}$ are binary decision variables expressing whether a RHV $i$ is assigned to region $r$ or to position $l$. $t_{ir}^{\text{out}}$ and $t_{il}^{\text{in}}$ are the estimated traveling times for a RHV $i$ to reach region $r$ or the position $l$ in the current region, respectively.\(^2\) $I$ is the set of vacant RHVs, $c_{Rr}$ is the control action obtained in the upper-

\(^2\)For simplicity, the estimated traveling times $t_{ir}^{\text{out}}$ and $t_{il}^{\text{in}}$ can be computed based on the relationship between the estimated traveling speed (as per the Speed-MFD relation for each region) and the distance of such a trip.
layer, indicating how many RHVs should move from region $R$ to region $r$, while $c_{RR}$ is the number of idle RHVs that should stay in the current region.

Note that the constraint in Equation (13b) limits a RHV either to stay in the current region or move to another region. Equations (13c) and (13e) ensure compliance with the upper-layer decision. This problem was inspired by classic assignment problems, such as those in Pentico (2007).

Assignment problems are fundamental combinatorial optimization problems. Note that the number of tasks (destinations inside and outside the evaluated region) is $|\mathcal{R}| - 1 + c_{RR}$, while number of agents (idle RHVs in the evaluated region) is $\sum_{r \in \mathcal{R}} c_{Rr}$. One could see the previous as an unbalanced assignment problem for having different number of tasks and agents. However, differently from the classical assignment problem, the proposed problem assigns multiple ($c_{Rr}$) RHVs to inter-regional tasks. Therefore, if one repeats $c_{Rr}$ times each of the columns of the cost matrix associated with inter-regional assignments, the problem becomes balanced (the number of agents and tasks become equal). Finally, as typical to assignment problems, the solution can be obtained using the Hungarian method (Kuhn, 2005) in strong polynomial times (Munkres, 1957), without the use of dummy agents nor tasks.

A possible solution to the problem in a two-region setting is illustrated in Figure 3. It presents a situation where some RHVs from Region 2 are assigned new positions in their current region and the remaining ones are sent to Region 1 to the lower-layer controller to perform the local repositioning instructions. We must highlight that RHVs are dispatched only to the border between Regions 1 and 2 by the shortest path. Therefore, the lower-layer will be able to instruct these RHVs as quickly as possible. Moreover, note that the RHVs in the current region are chosen such that the cost (total traveled distance) to reach the desired covered is minimized, enabling quick intra-regional response to unbalanced demand coverage, as well.

![Figure 3: Illustration of the solution for the middle-layer in the form of an assignment problem.](image)

### 3.3 Lower-layer: Coverage control method

In the lower-layer, the coverage control algorithm is operated for the RHVs that are commanded to stay in the current region (i.e., idle RHVs). The coverage controller steers these RHVs towards an optimal spatial configuration (indirectly, towards maximizing availability for service) by operating at a fast time scale and with detailed position guidance.

For each region, we formulate this task as a coverage control problem for the coordination and deployment of multiple mobile agents on the city network (Zhu et al., 2022a). Such coordination provides benefits...
to the system by dynamically allocating the RHVs according to the different demand densities of various city districts.

The city map can be presented as an undirected graph $G = (Q, E)$. If origin-destination pairs for trips are recorded in historical taxi data, we can compute the probability that a request starts at a node as $\phi(y)$. With a slight abuse of notation, $y$ in this section denotes a node on the graph (with $\sum_{y \in Q} \phi(y) = 1$).

The city area is clustered into $R = R_1 \cup R_2 \cup \ldots \cup R_n$ regions, with the set of all nodes in region $R$ denoted as $Q_R$. For idle RHV $i$ in region $R$, whose current position is $p_i$, the Voronoi tessellation can be defined as follows (Erwig, 2000):

$$V_i(p_i) = \{y \in Q^R : d(p_i, y) \leq d(p_j, y), \forall i \neq j\},$$

(14)

where $d(p_i, p_j)$ stands for the shortest distance between node $p_i$ and node $p_j$ on graph, computed by the Floyd-Warshall algorithm Floyd (1962) and $P = \{p_1, p_2, \ldots, p_{n_{idle}}\}$, where $n_{idle}$ is the number of current idle RHVs in region $R$. Figure 4 shows a two-node tessellation case on a graph.

![Voronoi tessellation on a graph. We show a two-node case where Node 5 and Node 10 are seed nodes. For example, Node 5’s Voronoi tessellation includes Node 1, 2, 3, 4, 5, 6, and 12, due to they are closer to Node 5 than to Node 10.](image)

Figure 4: Voronoi tessellation on a graph. We show a two-node case where Node 5 and Node 10 are seed nodes. For example, Node 5’s Voronoi tessellation includes Node 1, 2, 3, 4, 5, 6, and 12, due to they are closer to Node 5 than to Node 10.

The coverage objective function can be formulated as:

$$H(P, V) = \sum_{i=1}^{n_{idle}} \sum_{y \in V_i} d(p_i, y)^2 \phi(y).$$

(15)

According to Durham et al. (2012), the optimal position configuration of all RHVs is attained when each RHV is at the centroid of its respective Voronoi cell. The centroid of a graph Voronoi cell can be computed as an integer optimization problem as

$$C(V_i) = \arg\min_y \sum_{y \in V_i} d(p_i, y)^2 \phi(y).$$

(16)

At the beginning of each fast-loop time interval $T_l$, the idle RHV $i$ is commanded to move towards its current Voronoi centroid $C(V_i)$ after solving Equation 16. As it only requires local information for each RHV to calculate the Voronoi tessellation, this control algorithm is able to provide each RHV with an intersection/node-level repositioning command in a distributed manner.
4 Case Studies

4.1 Experimental Settings

In order to evaluate the proposed strategy, we employ an agent-based simulator, adapted from Beojone and Geroliminis (2021). The simulator replicates the urban road network of the Futian and Luohu districts of Shenzhen, China, consisting of 1,858 intersections interlinked by 2,013 road segments. We clustered the intersections in three urban regions using the snake clustering algorithm from Saeedmanesh and Geroliminis (2016) adapted to cope with demand similarities, as shown in Figure 5.

![Urban road network and its partition as three regions.](image)

Figure 5: Urban road network and its partition as three regions.

We test the proposed method under a scenario considering the asymmetry between trip origin and destination as shown in Figure 6. In this scenario, fewer trips originate from Region 3, while a larger number of trips terminate in Region 3. Without fleet monitoring, idle RHVs tend to accumulate in Region 3, far from the high-demand density areas of Region 1 and 2, which significantly hampers the overall service level.

In total, we tested 3 demand scenarios with a fleet of 250 RHVs. We consider a three-hour simulation, where, in the base demand scenario, 600 requests per hour are issued in the first and third hours and 1200 requests per hour are issued in the second hour, emulating a peak demand hour. In each of the subsequent scenarios, the base demand is increased by 50%, reaching a 100% increase in the last scenario. The passenger-driver matching process assigns the nearest available RHV to the incoming request, respecting waiting time tolerances and regional constraints. The upper-layer is operated with an update period of \( T_u = 5 \text{ min} \), while the coverage control in the lower-layer works with a relatively faster time interval with an update period of \( T_l = 10 \text{ s} \).

Table 3, in the Appendix B, depicts other parameters including values of \( \gamma_i \) used in the loss probability function and average trip lengths (with their respective standard deviation), and the share of demand for each regional Origin-Destination pair. These values were explicitly used in the prediction model of the upper-layer. Therefore, these values are not used in the simulation framework, which computed the microscopic

\[ ^3 \text{This choice is arbitrary, but in our experiments, no dominant setting was found about the update period of the upper-layer, with similar performance in terms of service level.} \]
movement of RHVs in a network and directly evaluated matching constraints (recalling that the loss probability function is only a depiction of this process in a convenient functional form). In any case, they help provide an overview of the traffic characteristics and conditions. Note that the demand share is a depiction of Figure 6(left) and, with the hourly number of requests from the previous paragraph, provides an overview of the general demand pattern for ride-hailing services. For simplicity, the simulation generated background traffic trips following the same demand pattern, but at larger numbers, such that the total demand for background traffic corresponds to 85% of the trips in the system (the remaining 15% are ride-hailing requests). Regarding the trip lengths, note that the displayed numbers correspond to the delivery portion of a trip, thus not accounting for the pick-up distances, which were assumed negligible. Since background traffic had the same demand pattern, these average trip lengths correspond to the entirety of their trips.

We ran the experiments in a 64-bit Windows PC with an Intel(R) Core(TM) i7-8700 CPU at 3.19GHz and a single-channel 16GB of DDR4 RAM memory at 2666MHz. We coded and ran the simulations using MATLAB R2018b using the parallel toolbox to compute the lower-layer control inputs (6 simultaneous computations). The MPC scheme is formulated using direct multiple shooting (Bock and Plitt, 1984), with dynamics discretized via Runge-Kutta 4 method, while the implementation is done using CasADi (Andersson et al., 2018), with IPOPT (Wächter and Biegler, 2006) as a solver. MPC prediction horizon is chosen as 6 steps (corresponding to a horizon of 50 minutes), while the input smoothing weight $\rho$ is chosen as $10^{-2}$ (further tests, which are omitted for brevity, reveal that the results are roughly insensitive to $\rho$ values between 0, $10^{-3}$ and $10^2$).

4.2 Results and Analysis

To evaluate the performance of the proposed methods, we use the ratio of ride requests that are not successfully fulfilled (i.e., a driver is not assigned and arrives to pick up the passenger) to the total number of ride requests, termed as abandonment rate; and average waiting time of fulfilled requests, computing the time each passenger spends waiting from the moment the request is issued until the passenger is picked up by the RHV, as the performance metrics. The following 7 methods are tested in this section:

- **full framework**: the proposed hierarchical control method;
• **upper only**: uses the upper-layer controller (MPC) and the middle-layer (as described in Sections 3.1 and 3.2, respectively). The control inputs include the inter-regional assignments to the border of the destination region. Once a vehicle crosses the regional border and reaches its designated target region, it starts to cruise randomly inside the region;

• **lower only**: using the low-level controller (coverage control) only, where the only control inputs are the intra-regional position guidance commands (as depicted in Zhu et al. (2022b));

• **no control**: a case with no controllers, and RHVs cruise randomly inside their regions;

• **no repositioning**: the case with no active controllers, involving holding an empty RHV at its position except when serving a passenger;

• **DBR-inspired**: RHVs are dispatched to hot-spots to replenish the number of RHVs in areas lacking RHVs and prioritized by the ratio between the number of RHVs near the hot-spot and the hot-spot capacity (as depicted in Beojone and Geroliminis (2021));

• **ASWFR**: dispatches RHVs to the location of lost requests in the last pre-determined time interval (as depicted in Alonso-Mora et al. (2017), the acronym ‘ASWFR’ refers to the first letters of the authors of this paper).

Table 2: Average waiting times (AWT), abandonment rates (AR), and CPU times for the lower- (L), middle- (M), and upper- (U) layer controllers.

<table>
<thead>
<tr>
<th>control scheme</th>
<th>performance vs. demand scenario</th>
<th>average CPU time (s)</th>
<th>maximum CPU time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original</td>
<td>+50%</td>
<td>+100%</td>
</tr>
<tr>
<td>full framework</td>
<td>73 s</td>
<td>3%</td>
<td>99 s</td>
</tr>
<tr>
<td>upper only</td>
<td>97 s</td>
<td>7%</td>
<td>116 s</td>
</tr>
<tr>
<td>lower only</td>
<td>81 s</td>
<td>5%</td>
<td>103 s</td>
</tr>
<tr>
<td>no control (fleet = 250)</td>
<td>102 s</td>
<td>9%</td>
<td>125 s</td>
</tr>
<tr>
<td>no repositioning</td>
<td>116 s</td>
<td>25%</td>
<td>122 s</td>
</tr>
<tr>
<td>DBR-inspired (6 hot-spots)</td>
<td>107 s</td>
<td>7%</td>
<td>116 s</td>
</tr>
<tr>
<td>ASWFR (instantaneous)</td>
<td>111 s</td>
<td>21%</td>
<td>122 s</td>
</tr>
<tr>
<td>ASWFR (30 seconds)</td>
<td>116 s</td>
<td>16%</td>
<td>130 s</td>
</tr>
<tr>
<td>no control (fleet = 300)</td>
<td>96 s</td>
<td>6%</td>
<td>117 s</td>
</tr>
<tr>
<td>no control (fleet = 350)</td>
<td>88 s</td>
<td>3%</td>
<td>112 s</td>
</tr>
</tbody>
</table>

*: These strategies run on a event-based case (one request or vehicle at a time) and the computational times were negligible.
We compare the performance of the different methods according to two main performance metrics: AR, i.e., Abandonment rate; and AWT, i.e., average waiting time, as described in the previous paragraph. Table 2 presents a summary of the performances and CPU times for the four cases. These results show that under all scenarios, the proposed hierarchical control method (full framework) has the best performance in terms of both abandonment rate and average waiting time. It achieves 28% lower waiting times than the no control case and an 8 s (10%) reduction compared to the coverage control only. At the same time, each individual layer applied alone manages to improve the waiting times and abandonment rates. The microscopic coverage control provides, alone most of the improvements in waiting times, while the upper-layer provides improvements in the abandonment rates, mainly. Besides, we test the no control policy with additional fleets. A problem associated with a larger fleet size is that it may result in higher costs for TNC operations and maintenance. Additionally, it can contribute to additional congestion in the network. It can be seen that full framework even outperforms (or achieved similar performance) the no control case with 50 or 100 more RHVs (fleets of size 300 and 350, respectively), in terms of average waiting times and abandonment rate. This aspect is a critical point that highlights the strong potential of our proposed method for practical impact. When compared to other benchmarks, the only one achieving performance levels near to the individual layers of the proposed framework (upper only and lower only cases) was the DBR-inspired strategy, in terms of abandonment rates. It achieved abandonment rates close to the upper only case in the original demand scenario; close to the lower only case in the 50% increased scenario. However, in terms of average waiting times, it presented little improvement compared to the no control case. The other benchmark, ASWFR only surpassed the no repositioning case, in terms of abandonment rates. Finally, it is important to note that allowing RHVs to cruise freely (no control case) was only outperformed in all tests by the DBR-inspired, upper only, lower only and full framework strategies.

These results highlight the added value of combining the different layers in the framework. On the one hand, the upper-layer utilizes an MPC framework, which takes an aggregated view of the traffic system and uses predictions about future system states to adjust fleet distribution among regions proactively. However, MPC lacks the precision required for microscopic RHV guidance within each region. On the other hand, the lower-layer employs a coverage control-based mechanism to steer idle RHVs toward high-demand areas. It provides each individual RHV with operational details at the intersection or node level, however, it lacks the global view and predictive capabilities of the upper-layer. Thus, even though implementing either layer can improve system performance, the combination of the layers in a hierarchical control structure offers the best of both (macroscopic and microscopic) worlds.

From the CPU time results in Table 2 (about 2 s for the middle-layer and 0.5 s for the upper-layer), we can see that the CPU time (a total of about 2.5 s) is negligible compared to the (middle and upper-layer) sampling time of 5 minutes. For the lower-layer (unlike the integrated middle-layer mechanism and upper-layer controller, which have to be operated on a single centralized computer), the control algorithm runs on the individual RHVs (one coverage controller for each RHV). Please note that, in the presented simulation experiments, the lower-layer control problem is solved by one individual computer and the total CPU time over all RHVs is about 0.5 s, which is still roughly negligible compared to the (lower-layer) sampling time of 10 seconds. As it should spend much less (virtually zero) time in real implementation in a distributed manner, these CPU time results suggest that the proposed framework is real-time feasible, indicating the potential for field application from the computational effort perspective.

In Figure 7, the cumulative number of abandonments is depicted for each strategy in a scenario with a 100% increase in demand. It can be seen that the curve for no control shows a steady increase in abandonments for the whole simulation. However, with the full framework, most abandonments occur during the peak demand hour, and it curtails the rise in abandonments post-peak hours, which verifies the effectiveness
of the proposed method. At the regional level, separating demand based on region of origin and destination, the full framework also outperforms other strategies. However, some points deserve attention. For instance, regional ODs with low demand were the least affected by the control strategies (in both relative and absolute terms). It shows Region 3 is the least affected by the control layers, while Regions 1 and 2 had the most significant reductions for applying the full framework. Note that even removing RHVs, mainly from Region 3 (as we discuss later in Figure 8), there is no prejudice to the service quality there, but these movements are responsible for the improvements in the other regions highlighting the effect of the imbalance in the demand.

Figure 7: Cumulative number of abandonments over time and OD-wise total abandonments (scenario with 100% increase in demand).

Figure 8 shows trajectories of vacant RHV accumulation states $n_{vo}^V$, together with the repositioning control inputs $u_{od}^*(k)$ (the first element of the optimal control input trajectory computed by the MPC at time step $k$) and the ratio of the idle RHVs in the region being dispatched to another region. Firstly, the great majority of the control inputs (translated to the number of vehicles to reposition) are leaving Region 3, as $c_{31}(t)$ and $c_{32}(t)$ are the largest control inputs. This is in accordance with the regional results from Figure 7(Right) and the demand pattern depicted in Figure 6. Moreover, during the peak periods, the upper-layer controller continuously moves more than 70% of the idle RHVs in that region to Regions 1 and 2. Only during a brief period in the first hour, MPC decides to move a few RHVs from Region 2 to Region 1, but never reposition RHVs in Region 1 (the area with the highest demand) to other regions.

Figure 9 shows a sequence of snapshots of the simulator, illustrating the repositioning command which requires a subset of empty RHVs in Region 3 to reposition to Regions 1 and 2 in a scenario with 100% demand increase. Due to the asymmetry between trip origin and destination distribution, Region 3 continuously has more idle RHVs than the other regions as more trips end in this region. However, the upper-layer MPC is able to capture the demand imbalance to counterpoise the number of RHVs available in regions with higher demand. Additionally, it identifies short periods requiring non-simultaneous flows between Regions 1 and 2. We have studied a scenario where there exists asymmetry between trip origin and destinations, as RHVs originating from high-demand areas became available in regions with lower demand, leading to the undesired concentration of idle and repositioning RHVs in the low-demand region i.e., Region 3. It can be seen from video link to Demo video without repositioning. In contrast, although more empty RHVs (consisting of blue and white in Figure 9) are found in Region 3 due to the imbalance nature, our proposed method assigns RHVs from Region 3 (as shown in Figure 5) to Regions 1 and 2, which benefits the system performance.
Figure 8: Trajectories of vacant RHVs accumulation states $n_{od}^V(t)$, repositioning inputs $c_{od}(t)$ and the ratio of the current idle RHVs instructed to reposition (scenario with 100% increase in demand).

Particularly, we compare the evolution of vacant RHVs in each region under the full framework and the no control strategies, both in scenarios with 50% and 100% increase in demand, as shown in Figure 10. It can be seen that the full framework has fewer vacant RHVs than no control in both cases, indicating that the majority of the fleet is actively engaged in passenger service. During the first simulation hour, most accumulation of idle RHVs is observed in Region 3 under no control. Most of them remain idle while the abandonment keeps increasing as illustrated in Figure 7, due to their disadvantageous position configuration. In contrast, our proposed hierarchical framework addresses this issue by repositioning RHVs from Region 3, which improves the spatiotemporal balance between passenger demand and RHV supply, thus enhancing the overall system efficiency. The number of repositioning RHVs shows that many RHVs fulfill the repositioning instructions before the upcoming prediction step (5 minutes). That indicates that the middle-layer managed to identify repositioning RHVs near regional borders while maintaining intra-regional coverage. Finally, even though the full framework requested mostly drivers from Region 3 to reposition, it is worth mentioning the difference between the number of repositioning RHVs between these two demand scenarios. In a lower demand scenario, fewer RHVs are required to move due to the MPC’s prediction that moving more RHVs would not be worthwhile, as it could compromise the short-term availability of RHVs. Conversely, in a higher demand scenario, the MPC predicts that it is worthwhile to temporarily sacrifice capacity in order to have these RHVs available where they are most needed, given the scarcity of idle RHVs.
Figure 9: Sequence of snapshots of the network with ride-hailing fleet for a scenario with 100% demand increase: Non-repositioning idle RHVs (white), repositioning idle RHVs (blue), busy RHVs traveling to pick up (green) and drop off passengers (red). At the time step shown, 28 RHVs are being repositioned mainly from Region 3 to Region 2. A demo video is available on Google Drive: link to Demo video of the full framework and link to Demo video without repositioning.

As pointed out earlier, the objective of the middle-layer $D_R$ (Equation (13a)) is to minimize both the total distance needed to comply with the upper-layer control inputs and the costs for achieving optimal coverage after the instructions. Figure 11 demonstrates the computed value of the middle-layer objective, $D_R$, compared to random inter-regional assignment and assignment of the nearest RHVs to the regional borders. Although the optimization problem in Equation (13) is deterministic, the intra-regional centroids are randomly sampled and then optimized for coverage, resulting in slight variations in the final objective function. Initially, the proposed assignment problem does not exhibit significant performance improvement compared to the alternatives. However, after the first hour of operation, when the upper-layer issues more repositioning orders (refer to Figure 8), the performance of each method becomes distinguishable. The random assignment performs the least favorably, while the MILP demonstrates the best performance, resulting in the lowest repositioning costs. On average, the costs are approximately half of those incurred by the random assignment and 25% lower than when dispatching the nearest RHVs to the borders.

Besides the service quality measurements, such as waiting times and abandonment rates, it is worth investigating whether the proposed framework could have additional externalities to the traffic system. Figure 12 evaluates the Vehicle-Kilometers-Traveled (VKT), a performance measure associated with emissions and traffic accidents (Beojone and Geroliminis, 2021), of RHVs. It is important to note that we do not compare the VKT of intra-regional repositioning movements due to their conceptual similarity to cruising. The proposed framework successfully minimizes empty VKT while maximizing busy VKT, with RHVs being occupied for 70% of the total traveled distances. This contrasts with the 55% busy-idle ratio observed in the No control case. On the other hand, scenarios only with the upper or lower-layer active have comparable performances during the observed period, closely following the full framework.

5 Discussions and Conclusions

In this paper, we have introduced a hierarchical control framework for repositioning idle RHVs. The method incorporates MPC at the upper-layer, optimal assignment at the middle-layer, and coverage control at the
lower-layer. MPC optimizes fleet availability among urban regions, leveraging near-future predictions of system behavior and demand. The coverage controller of each region provides each RHV with detailed position guidance. In between, the assignment optimization bridges the MPC decisions and the area coverage, minimizing the cost for both layers. The framework specifies a proactive strategy for dynamically deploying the fleet, thereby facilitating configurations advantageous for spatiotemporal RHV availability.

To put the proposed hierarchical repositioning framework under perspective, the inter-regional repositioning decisions performed by the upper-layer have parallels in the literature. Ramezani and Nourinejad (2018) utilizes the MFD-based models to represent dynamic traffic conditions and a MPC controller to perform inter-regional repositioning of RHVs. However, our approach differs significantly in the objective functions; while Ramezani and Nourinejad (2018) focus on minimizing total network delay, our upper-level controller aims to optimize the number of serving passengers and avoid severe variations in control input. Another difference lies in the representation of the dynamics, which, to support the new objective function, is based in the dynamics of Beojone and Geroliminis (2023a) and the probability of losing incoming requests. Valadkhani and Ramezani (2023) also employs a MFD-based models in a MPC controller with an objective of minimizing the total unassigned time of the RHVs, which is more in line with the objective of our upper-layer. The main difference is that, depending on the characteristics of service requests, minimizing unassigned period might prioritize long trips which make RHVs busier for longer periods, while our objective tries to serve as many requests as possible, which might give emphasis to shorter trips that might be unprofitable. The idea of virtual passengers used in Zhang and Pavone (2016) echoes with the control inputs of the MPC, which, in a sense, creates a virtual demand for ride-hailing to reposition RHVs. A couple of key differences must be highlighted, however, where the demand in Zhang and Pavone (2016) is fixed, and the control inputs in our proposed upper-layer enforces movements and they are bounded to ensure their execution. In the extended study of Zhang et al. (2016), besides the obvious similarity for using MPC controllers, the authors use other objectives to avoid unnecessary repositioning decisions, similarly to our smoothing factors. However, the proposed method has a scalability problem (only 30 RHVs analyzed in the
Figure 11: Compared middle-layer objective results (scenario with 100% increase in demand).

Figure 12: Vehicle-Kilometers-Traveled (VKT) measurements for RHVs separated into empty (not carrying passengers), repositioning (inter-regional movements only) and busy activities.

problem), which our framework handles, as shown by the low CPU times in Table 2. Other inter-regional repositioning strategies, such as Chen et al. (2017) and Miao et al. (2021) did not employ an explicit system model, relying in data-driven methods, while our methodology incorporates a comprehensive system model to enhance the accuracy. Finally, in terms of formulation, the assignment problem in Tuncel et al. (2023) shares similarities with the middle-layer in our approach (noting that the middle-layer does not assign requests, i.e., it does not perform matching), where the assignment can be in a microscopic level (match assignment vs local centroid assignment) or a macroscopic assignment (inter-regional repositioning).

Extending the discussion under the perspective of the local repositioning used in the lower-layer, the cruising strategies in Zhou et al. (2020) and Yu et al. (2019) are intended to maximize their profits, and the optimal policy does not necessarily guide RHVs to high-demand areas but areas with a higher matching probability (Shou et al., 2020). Similarly, the applied coverage control used in the lower-layer does concentrate all RHVs around the areas of higher demand, it distributes them also maximizing the matching probability, with the difference that it coordinates all the RHVs in the region simultaneously. The decentralized cooperative cruising behavior in Chen et al. (2021) is analogous to the lower-layer in the proposed framework, which provides a decentralized repositioning strategy, requiring only RHVs to communicate among themselves, not a central dispatcher. Differently, Ding et al. (2022) uses an attraction model to calculate the cruising direction to specific intersections where passengers are waiting, establishing a dynamic
matching relationship. The demand density distribution used in the coverage controller that we use at the lower-layer can be understood as a form of attraction model too. However, it does not rely in the number of passengers already waiting at an intersection. Instead, the attraction comes from the expected areas where passengers will arrive. Therefore, acting proactively to serve incoming requests. Another difference between these approaches is the matching relationship, which is assumed to be static (a passenger-vehicle assignment is definitive), avoiding possible recalculations of cruising directions and competition between RHVs in the case other RHVs pick-up the passengers first.

Individually, these perspectives (inter- and intra-regional repositioning) have limitations, which the proposed hierarchical framework tackles by exploring their strengths. In this work, results have shown an improved service quality by combining the approaches into a hierarchical framework. It has managed to reduce waiting times and serve more requests compared to one not applying any repositioning. Furthermore, it has managed to increase RHV utilization and provide timely instructions in an instance for a large number of RHVs. The combination of perspectives showed benefits also in Valadkhani and Ramezani (2023), where, besides the inter-regional repositioning decisions, it implements a link-level allocation strategy that assigns a link in one region to each selected RHV for inter-regional repositioning based on the location of the waiting passengers in this region, a strategy that echoes with the proposed middle-layer in our framework. The contrast is that our optimization problem in the middle-layer addresses how to strategically select the subset of idle RHVs, which does not exist in existing literature to the best of our knowledge. Additionally, our approach also employs a coverage control algorithm that considers both demand distribution and RHV coordination. As mentioned before, Ding et al. (2022) also presents a hierarchy for repositioning RHVs, uniting the inter-regional repositioning to a local cruising strategy combined with the matching process. Therefore, constraining the service design to a single matching strategy, while possibly missing other strategies such as ridesplitting (Beojone and Geroliminis, 2023a; Vignon et al., 2021) or larger capacity vehicles (e.g. seen in Alonso-Mora et al. (2017)).

Besides the improved performance, this study has presented a general and flexible structured strategy that enables modifications in each layer for testing other algorithms and solutions for repositioning idle RHVs. We must point out that these solutions have great importance to ride-hailing service operators, which aim to maximize service efficiency, benefiting the operator and its passengers. The decrease in the empty kilometers traveled is one indicator of improvements in traffic conditions but further studies are required to assess other consequences to society’s welfare.

It is important to highlight that the adoption of such an optimized strategy providing instructions every few seconds would be feasible in cases with an autonomous fleet of RHVs. Challenges worth investigating include the definition of the regions, which can have a significant impact on the overall performance of the framework, particularly in the upper-layer. Therefore, future research will explore cases with more regions, examine various spatiotemporal demand scenarios in the robustness of the framework, assess the impacts of varying fleet sizes. Using such repositioning frameworks in emerging shared modes, including ride-pooling (also called ride-splitting) is worth investigating, possibly linking with routing to maximize matching opportunities. A starting effort in this direction can be seen in Beojone and Geroliminis (2023a). Another direction for further research is to explore theoretical properties with analytical derivations of such hierarchical frameworks in more comprehensible scenarios of the problem.
6 Acknowledgements

This paper is partially funded by both Dit4Tram “Distributed Intelligence & Technology for Traffic & Mobility Management” project from the European Union’s Horizon 2020 research and innovation programme under grant agreement 953783, and Swiss National Science Foundation (SNSF) under National Centre of Competence in Research, Dependable Ubiquitous Automation (NCCR Automation) under grant agreement 51NF40_180545.

A Argument on the decoupled dynamics in the upper-layer

The objective of including private vehicles in the prediction model is to predict average traveling speeds (congestion levels) in the near future. Additionally, the idea for decoupling the dynamics raises from the assumption that the number of RHVs will remain constant during, at least, the prediction horizon. As a consequence, the impact of RHVs on the congestion levels is approximately constant. Furthermore, besides the direct impact of RHVs on congestion levels, we assume, as a penalty to the system, that unserved ride-hailing requests (abandonments) will increase the demand for private vehicles (see eq. (5b)). However, using MPC (where the prediction model is used) only, or the full framework, considerably dropped the number of abandonments, such that the addition of these abandoned trips to the private vehicles demand has a secondary effect on their total demand.

Therefore, in predictions of the NMPC, the decoupled dynamics for private vehicles will provide the predictions of the function \( v(t) \). Without loss of generality, assume a single-region case, such that the dynamics for private vehicles become the following:

\[
\dot{n}_P(t) = q_P(t) - O_P(t) = q_P(t) - \frac{n_P(t)v(t)}{L_P}.
\]

In steady-state conditions, \( \dot{n}_P(t) = 0 \), it is possible to isolate the average speed term \( v(t) \), as follows:

\[
v(t) = \frac{q_P(t)L_P}{n_P(t)}.
\]

Therefore, the relative error in the estimated speed is equivalent to the relative error in the private vehicle demand \( q_P(t) \):

\[
\varepsilon = \frac{v_{est}(t) - v_{real}(t)}{v_{real}(t)} = \frac{q_{est}(t) - q_{real}(t)}{q_{real}(t)}.
\]

However, since \( q_P(t) = \lambda_P(t) + \lambda_B(t) \cdot pl(t) \), where \( pl(t) \) is the loss probability, the penetration rate of ride-hailing \( \lambda_B(t)/(\lambda_P(t) + \lambda_B(t)) \) does not affect the estimation of \( q_{est}(t) \) directly. Actually, the source of errors is the error in the estimation of loss probabilities, which can be amplified by higher penetration rates.

In Figure 13, we summarize the relative errors in the traveling speeds and total accumulation as a result of the process of decoupling the dynamics of private vehicles from RHVS. It compares three scenarios for an initial estimation of the loss probabilities. Note that, errors normally remain below 3.5% for penetration.

\[^4\text{This assumes a fixed value for } n_P(t). \text{ If we invert the assumption and fixes } q_P(t), \text{ instead, the result will clearly become the relative error of the inverse of } n_P(t).\]
rates lower than 15% (assumed penetration rate in the studied case). This number is only surpassed in situations where the penetration rate of ride-hailing is a lot higher and for cases with high initial estimates for loss probabilities.

Figure 13: Average errors in the traveling speeds and total accumulation for decoupling the dynamics for private vehicles and ride-hailing vehicles.

Therefore, given the elevated goodness of fit for loss probabilities, reaching $R^2 = 0.96$ in Beojone and Geroliminis (2023a), penetration rates usually lower than 15% (Erhardt et al., 2019), and the improved performance of the ride-hailing service (in terms of abandonments), the errors associated with the decoupled dynamics can be assumed secondary to the problem.

### B Parameters used in the prediction model

Table 3: Summary of parameters for the forecasting model in the upper-layer.

<table>
<thead>
<tr>
<th>Regional OD</th>
<th>Demand share</th>
<th>Avg. trip length (STD)</th>
<th>Loss probability parameters $\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$\gamma_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>0.2564</td>
<td>3.339 ($\pm$2.497)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.5535</td>
<td>0.9224</td>
</tr>
<tr>
<td>1-2</td>
<td>0.0493</td>
<td>6.830 ($\pm$1.883)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.3049</td>
<td>0.5681</td>
</tr>
<tr>
<td>1-3</td>
<td>0.2071</td>
<td>6.893 ($\pm$5.022)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.5092</td>
<td>0.7992</td>
</tr>
<tr>
<td>2-1</td>
<td>0.0472</td>
<td>4.678 ($\pm$2.519)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.4983</td>
<td>0.5896</td>
</tr>
<tr>
<td>2-2</td>
<td>0.2550</td>
<td>1.533 ($\pm$1.538)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.5690</td>
<td>0.9229</td>
</tr>
<tr>
<td>2-3</td>
<td>0.0018</td>
<td>6.935 ($\pm$0.403)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.4094</td>
<td>0.6520</td>
</tr>
<tr>
<td>3-1</td>
<td>0.0417</td>
<td>5.467 ($\pm$1.159)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.3683</td>
<td>0.4149</td>
</tr>
<tr>
<td>3-2</td>
<td>0.0003</td>
<td>3.987 ($\pm$0.109)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.3752</td>
<td>0.5950</td>
</tr>
<tr>
<td>3-3</td>
<td>0.1412</td>
<td>4.765 ($\pm$6.363)</td>
<td>$1 \times 10^{-2}$</td>
<td>0.5939</td>
<td>0.8145</td>
</tr>
</tbody>
</table>
References


