2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Time-invariant Network Flow Model for Two-person Ride-pooling Mobility-on-Demand

Fabio Paparella, Leonardo Pedroso, Theo Hofman, Mauro Salazar

Abstract— **This paper presents a time-invariant network flow model capturing two-person ride-pooling that can be integrated within design and planning frameworks for Mobility-on-Demand systems. In these type of models, the arrival process of travel requests is described by a Poisson process, meaning that there is only statistical insight into request times, including the probability that two requests may be pooled together. Taking advantage of this feature, we devise a method to capture ride-pooling from a stochastic mesoscopic perspective. This way, we are able to transform the original set of requests into an equivalent set including pooled ones which can be integrated within standard network flow problems, which in turn can be efficiently solved with off-the-shelf LP solvers for a given ride-pooling request assignment. Thereby, to compute such an assignment, we devise a polynomial-time algorithm that is optimal w.r.t. an approximated version of the problem. Finally, we perform a case study of Sioux Falls, USA, where we quantify the effects that waiting time and experienced delay have on the vehicle-hours traveled. Our results suggest that the higher the demands per unit time, the lower the waiting time and delay experienced by users. In addition, for a sufficiently large number of demands per unit time, with a maximum waiting time and experienced delay of 5 minutes, more than 90% of the requests can be pooled.**

I. INTRODUCTION

Ride-sharing is a service that is revolutionizing urban transportation. Within this service, ride-pooling is the concept of having multiple users traveling at the same time on a single vehicle at lower costs, e.g., emissions, energy consumption, fleet size, and also the cost of the ride charged to the user. Nevertheless, these improvements come at the expense of additional waiting time and delays caused by detours. Ride-pooling is a difficult problem to deal with due to its combinatorial nature. For this reason, the microscopic nature of ride-pooling is, at first sight, incompatible with an approach on a different scale. However, sometimes it is enough to study a ride-sharing system from a macroscopic point of view, especially when dealing with mobility planning or design [1], [2]. In this paper, we propose a framework to deal with ride-pooling from a mesoscopic point of view by moving from a deterministic to a stochastic approach. We devise a framework to easily incorporate ride-pooling into a linear time-invariant multi-commodity network flow model, also known as traffic flow model, that is a mesoscopic modeling framework usually used for mobility planning and design.

Related Literature: This paper pertains to the research streams of traffic flow models and ride-pooling, that we review in the following. One of the approaches to characterize and control ride-sharing systems is the multi-commodity network flow model, that is suited for easy implementation of many constraints of different nature and can be efficiently solved with commercial solvers. This model has been used for multiple design purposes such as minimizing fleet size [3], [4], minimizing electricity costs [5], and joint optimization with public transport [1] and the power grid [6]–[8]. For example, in [2], [9] the authors proposed a joint optimization framework for the siting and sizing of the charging infrastructure for an electric ride-sharing system. Yet in all these models the assumption of one person per vehicle is made.

Ride-pooling has been extensively studied. Alonso-Mora et al. [10] conceived the vehicle group assignment algorithm, which optimally solves the ride-sharing problem with high capacity vehicles in a microscopic setting. In [11], [12] the benefits of vehicle pooling and the pricing and equilibrium in on-demand ride-pooling markets were analyzed, respectively. Fieldbaum et al. [13] studied ride-pooling considering that users can be picked-up and dropped-off within a walkable distance, while in [14] they examine how to split costs between users that share the same ride. In [15] a time-expanded network flow model is leveraged to compute the optimal routes of a mobility system that allows for ride-pooling. However, in all of these papers the ride-pooling problem has been studied from a microscopic perspective, whereby each request is considered individually. Recently, in [16] an interesting step towards a mesoscopic stance has been carried out from a stochastic matching perspective.

Statement of Contributions: The main contributions of this paper are twofold. First, we propose a framework to capture ride-pooling, a microscopic combinatorial phenomenon, in a time-invariant network flow model, whereby the arrival process is stochastic and the complexity of the problem is independent from the number of requests. Second, within the proposed framework, we devise a method to compute a ride-pooling request assignment that is optimal w.r.t. a relaxed version of the minimum overall travel time problem.

Organization: The remainder of this paper is structured as follows: Section II introduces the multi-commodity traffic flow problem and the framework to capture ride-pooling. Section III details the case study of Sioux Falls. Last, in Section IV, we draw the conclusions from our findings and provide an outlook on future research endeavors.

Notation: We denote the vector of ones, of appropriate dimensions, by $\mathbb{1}$. The ith component of a vector v is denoted by v_i and the entry (i, j) of a matrix A is denoted by A_{ij}.

Mechanical Engineering, Control Systems Techonology, Eindhoven University of Technology, Eindhoven, The Netherlands {f.paparella, l.pedroso,t.hofman,m.r.u.salazar}@tue.nl

The cardinality of set $\mathcal{S}$ is denoted by $|\mathcal{S}|$.

II. Ride-pooling Network Flow Model

In this section, we introduce the network traffic flow model [3]. Then, we extend it to take into account ride-pooling, and finally present a brief discussion on the model.

A. Time-invariant Network Flow Model

We model the mobility system as a multi-commodity network flow model, similar to the approaches of [1], [2], [5], [9], [17]. The transportation network is a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$. It consists of a set of vertices $\mathcal{V} := \{1, 2, ..., |\mathcal{V}|\}$, representing the location of intersections on the road network, and a set of arcs $\mathcal{A} \subseteq \mathcal{V}\times\mathcal{V}$, representing the road links between intersections. We indicate $B \in \{-1, 0, 1\}^{|\mathcal{V}|\times|\mathcal{A}|}$ as the incidence matrix [18] of the road network $\mathcal{G}$. Consider an arbitrary arc indexing of natural numbers $\{1, \dots, |\mathcal{A}|\}$, then $B_{ip} = -1$ if the arc indexed by p is directed towards vertex i, $B_{ip} = 1$ if the arc indexed by p leaves vertex i, and $B_{ip} = 0$ otherwise. We denote t as the vector whose entries are the travel time t_a required to traverse each arc $a \in \mathcal{A}$, ordered in accordance with the arc ordering of B, which we assume to be constant. Similarly to [3], we define travel requests as follows:

Definition II.1 (Requests)**.** *A travel request is defined as the tuple $r = (o, d, \alpha) \in \mathcal{V}\times\mathcal{V}\times\mathbb{R}_{>0}$, in which α is the number of users traveling from the origin o to the destination $d \neq o$ per unit time. Define the set of requests as $\mathcal{R} := \{r_m\}_{m\in\mathcal{M}}$, where $\mathcal{M} = \{1, \dots, M\}$.*

We assume, without any loss of generality, that the origin-destination pairs of the requests $r_m \in \mathcal{R}$ are distinct. In this paper, we distinguish between active vehicle flows, which correspond to the flows of vehicles serving users whether they are ride-pooling or not, and rebalancing flows which correspond to the flows of empty vehicles between the drop-off and pick-up vertices of consecutive requests. We define the active vehicle flow induced by all the requests that share the same origin $i \in \mathcal{V}$ as vector x^i, where element x^i_a is the flow on arc $a \in \mathcal{A}$, ordered in accordance with the arc ordering of B. The overall active vehicle flow is a matrix $X \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{V}|}$ defined as $X := \begin{bmatrix} x^1 & x^2 & \dots & x^{|\mathcal{V}|}\end{bmatrix}$. The rebalancing flow across the arcs is denoted by $x^{\mathrm{r}} \in \mathbb{R}^{|\mathcal{A}|}$. In the following, we define the network flow problem.

Problem 1 (Multi-commodity Network Flow Problem)**.** *Given a road graph $\mathcal{G}$ and a demand matrix D, the active vehicle flows X and rebalancing flow x^{r} that minimize the cost in terms of overall travel time result from*

$$\begin{aligned} \min_{X,x^{\mathrm{r}}}\ & J(X, x^{\mathrm{r}}) = t^\top(X\mathbb{1} + x^{\mathrm{r}}) \\ \text{s.t. } & BX = D \\ & B(X\mathbb{1} + x^{\mathrm{r}}) = 0 \\ & X, x^{\mathrm{r}} \geq 0, \end{aligned}$$

where the demand matrix $D \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ represents the requests between every pair of vertices, whose entries are

$$D_{ij} = \begin{cases} \alpha_m, & \exists m \in \mathcal{M} : o_m = j \wedge d_m = i \\ -\sum_{k\neq j} D_{kj}, & i = j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Since Problem 1 is totally unimodular, X and x^{r} can be decoupled and computed separately [4]. The objective function can also be interpreted as the minimum fleet size required to implement the flows [3], [4].

B. Ride-pooling Time-invariant Network Flow Model

In this paper, we propose a formulation to take into account ride-pooling without the need to change the original structure of the problem. We transform the original set of requests, portrayed by D, into an equivalent set of requests accounting for ride-pooling, portrayed by D^{rp}. We define the ride-pooling network flow problem as follows:

Problem 2 (Ride-pooling Network Flow Problem)**.** *Given a road graph $\mathcal{G}$ and a demand matrix D^{rp}, the active vehicle flows X and rebalancing flow x^{r} that minimize the cost in terms of overall travel time result from*

$$\begin{aligned} \min_{X,x^{\mathrm{r}}}\ & J(X, x^{\mathrm{r}}) = t^\top(X\mathbb{1} + x^{\mathrm{r}}) \\ \text{s.t. } & BX = D^{\mathrm{rp}} \\ & B(X\mathbb{1} + x^{\mathrm{r}}) = 0 \\ & X, x^{\mathrm{r}} \geq 0. \end{aligned}$$

The ride-pooling demand matrix D^{rp} in Problem 2, which describes the pooling pattern, has to be determined according to four key conditions. First, the individual requests, described by D, must be served. Second, ride-pooling two requests is only spatially feasible if the detour travel time is not greater than a threshold $\bar{\delta} \in \mathbb{R}_{\geq 0}$. Third, ride-pooling two requests is only temporally feasible if the waiting time for a request to start being served does not exceed a threshold $\bar{t} \in \mathbb{R}_{>0}$. Fourth, the requests are pooled to minimize the cost function of Problem 2 at its solution. Due to the combinatorial nature of such an endeavor, we relax the problem in order to attain a computationally tractable algorithm, according to the following approximation.

Approximation II.1. *For the purpose of computing the demand matrix D^{rp}, the cost function of Problem 2 is approximated by $\tilde{J}(X) := t^\top X\mathbb{1}$.*

It is crucial to remark that this approximation is only employed in a first step to compute D^{rp} and, afterwards, in a second stage, the passenger and rebalancing flows are optimized jointly according to Problem 2. This is in order for an Autonomous Mobility on Demand fleet, for example, which is centrally operated and whose vehicles do not compete for rides. This is a very common approximation used in the literature [10], which we leverage to devise a polynomial-time algorithm to compute D^{rp} that is optimal w.r.t. the approximated version of the problem.

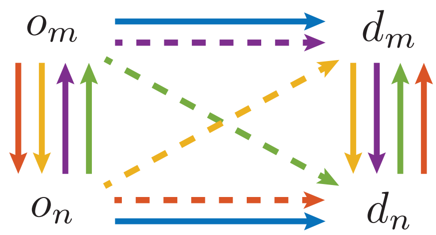

Fig. 1. Distinct configurations for serving two requests $r_m, r_n \in \mathcal{R}$. Each arrow represents a flow of $\alpha = 1$ vehicles. The dashed arrows represent a flow with two users, whilst the solid ones represent a flow with one user.

C. Approximate Computation of the Demand Matrix

In this section, we present a framework to compute the demand matrix D^{rp} under Approximation II.1.

1) Spatial Analysis of Ride-pooling: In this section, we analyze the feasibility and optimal configuration of ride-pooling two requests from a spatial perspective. First, we define δ as the delay experienced by each user, representing the time required to travel the additional detour distance w.r.t. the scenario without ride-pooling. If the delay experienced by any of the two users is higher than the threshold $\bar{\delta}$, then pooling the two requests is unfeasible. Second, given the feasible pooling itineraries, we analyze which one is the optimal, i.e., the best itinerary to serve the requests, and whether pooling is advantageous w.r.t. no pooling. Consider two requests $r_m, r_n \in \mathcal{R}$. To restrict this analysis to the spatial dimension, we temporarily make two key considerations, that we lift in Section II-C.3: i) the requests r_m and r_n are made at the same time; and ii) both requests have the same demand, which we set, without any loss of generality, to $\alpha = 1$. There are five different ways of serving two requests $r_m, r_n \in \mathcal{R}$, as depicted in Fig. 1. The goal is to assess whether it is feasible to ride-pool r_m and r_n and which is the best configurations among the five. Index each configuration with number $c \in \{0, \dots, 4\}$, with $c = 0$ corresponding to no pooling. Each configuration can be split into either two or three equivalent travel requests, as shown in Fig. 1, each corresponding to an arrow. Denote the set of such equivalent requests for configuration c as $\mathcal{R}^c_{mn}$ ($\mathcal{R}^c_{nm} = \mathcal{R}^c_{mn}$) and we define $\Pi(\mathcal{R}^c_{mn})$ as the order of visited nodes. For each configuration c, one can now solve Problem 2, under Approximation II.1, with a simplified demand matrix $D^{\mathrm{rp}} = D^{mn,c}$ obtained from the set of requests $\mathcal{R}^c_{mn}$ with (1), obtaining a flow $X^{mn,c} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. The delay $\delta^{m,c}$ of request r_m for a configuration c, is

$$\delta^{m,c} = \sum_{p \in \pi^c_{mn}} [t^\top X^{mn,c}]_p - [t^\top X^{mn,0}]_{o_m},$$

where $\pi^c_{mn} \subseteq \Pi(\mathcal{R}^c_{mn})$ is the ordered set of nodes $\Pi(\mathcal{R}^c_{mn})$ from o_m to the node before d_m. The feasible configurations are those whose delay of both users is below the threshold $\bar{\delta}$. Then, among the feasible ones, comprehending also the no pooling option, the optimal configuration is the one whose flow $X^{mn,c}$ achieves the lowest cost $\tilde{J}(X^{mn,\star})$. Henceforth, the simplified demand matrix of the optimal configuration for ride-pooling r_m and r_n is denoted by $D^{mn,\star}$.

Remark II.1. *The demand matrix $D^{mn,c}$ contains either two or three equivalent travel requests between the origin and destination nodes of requests r_m and r_n. To reduce the computational load, a graph search technique can be employed instead, to compute the shortest path between every pair of nodes. Since each instance has a worst-case computational complexity of $\mathcal{O}(|\mathcal{V}|^2)$, the overall computational complexity is $\mathcal{O}(|\mathcal{V}|^4)$. The procedure depends on the graph $\mathcal{G}$, meaning that the computations have to be performed only once.*

2) Temporal Analysis of Ride-pooling: In this section, we analyze the temporal alignment of two requests for ride-pooling. We derive the probability of two requests taking place within the maximum waiting time, $\bar{t}$. As common in traffic flow models [3], we consider that the arrival rate of a request $r_m \in \mathcal{R}$ follows a Poisson process with parameter α_m. Consider two requests $r_m, r_n \in \mathcal{R}$. In the following lemma, we indicate the probability of the two events occurring within a maximum time window $\bar{t}$.

Lemma II.1. *Let $r_m, r_n \in \mathcal{R}$ be two requests whose arrival rate follow a Poisson process with parameters α_m and α_n, respectively. The probability of each having an occurrence within a maximum time interval $\bar{t}$ is*

$$P(\alpha_m, \alpha_n) := 1 - \frac{\alpha_m e^{-\alpha_n \bar{t}} + \alpha_n e^{-\alpha_m \bar{t}}}{\alpha_m + \alpha_n}. \tag{2}$$

Proof. The proof can be found in Appendix I. □

3) Expected Number of Pooled Rides: In Section II-C.1, we analyzed the spatial dimension of the ride-pooling problem, whereby we computed the best feasible pooling path given two requests. In Section II-C.2, we analyzed the temporal dimension of the ride-polling problem, whereby we derived the probability of two requests happening within a time window. By lifting the temporary assumptions made in Section II-C.1, we formulate the ride-pooling demand matrix given a certain pooling assignment, defined in what follows.

A fraction of the demand of every request $r_m \in \mathcal{R}$ can be assigned to be pooled with a request $r_n \in \mathcal{R}$. Let $\beta \in \mathbb{R}^{|\mathcal{R}| \times |\mathcal{R}|}_{\geq 0}$ denote the assignment matrix, whose entry (m, n) is the demand of r_m that is assigned to be pooled with r_n. For the remainder of this subsection we assume that β is given. In Section II-C.4, we propose an algorithm to compute the optimal value of β under Approximation II.1.

From the analysis in Section II-C.2, it is noticeable that only a fraction of the allocated ride-pooling demand β_{mn} can actually be pooled due to the aforementioned temporal constraints. Specifically, the probability of pooling is given by $P(\beta_{mn}, \beta_{nm})$ according to Lemma II.1. Moreover, given that we only consider pooling between two requests, at most, the maximum pooled demand between $r_m, r_n \in \mathcal{R}$ is $\min(\beta_{mn}, \beta_{nm})$. Therefore, the effective expected pooled demand between two requests $r_m, r_n \in \mathcal{R}$ is given by $\gamma_{nm} = \gamma_{mn} := \min(\beta_{mn}, \beta_{nm}) P(\beta_{mn}, \beta_{nm})$. As a result, according to the spatial analysis in Section II-C.1, this pooled demand is portrayed by the demand matrix $\gamma_{mn} D^{mn,\star}$. Note that the effective expected pooling demand follows

$\sum_{n\in\mathcal{M}}\gamma_{mn} \leq \alpha_m, \forall r_m \in \mathcal{R}$ with equality if the full demand of r_m is pooled. The full ride-pooling demand matrix $D^{\rm rp}$ is made up of two contributions: i) the sum of the expected pooled active vehicle flows of the form $\gamma_{mn}D^{mn,\star}$ for $r_m, r_n \in \mathcal{R}$; and ii) the requested demands that were not ride-pooled. Thus, the entry (i,j) of $D^{\rm rp}$ can be written as

$$D^{\rm rp}_{ij} = \begin{cases} \sum\limits_{\substack{p,q\in\mathcal{M}\\ p\geq q}} \gamma_{pq} D^{pq,\star}_{ij} + \left(D_{ij} - \sum\limits_{p\in\mathcal{M}} \gamma_{mp} D^{mp,\star}_{ij} \right), & \\ & \exists m \in \mathcal{M} : d_m = i \wedge o_m = j \\ -\sum_{k\neq j} D^{\rm rp}_{kj}, & i = j \\ \sum\limits_{\substack{p,q\in\mathcal{M}\\ p\geq q}} \gamma_{pq} D^{pq,\star}_{ij}, & \text{otherwise.} \end{cases}$$

Finally, one can input $D^{\rm rp}$ to Problem 2, which yields an LP, given a pooling assignment β.

4) Optimal Ride-pooling Assignment: In this section, we will compute the optimal ride-pooling assignment matrices $\beta^\star$ and $\gamma^\star$, under Approximation II.1, leveraging an iterative approach, which is described in what follows. For every pair of requests $r_m, r_n \in \mathcal{R}$, we can compute the unitary improvement of the objective function of Problem 2, denoted by $\Delta\tilde{J}_{mn}$, w.r.t. the no-pooling scenario. Specifically, it amounts to the difference between $\tilde{J}_{nm}$, which denotes the cost with $D^{\rm rp} = D^{mn,\star}$, and $\tilde{J}_n + \tilde{J}_m$, which again denotes the cost with $D^{\rm rp} = D^{mn,0}$. Let $\alpha'_m, m \in \mathcal{M}$ stand for an auxiliary variable throughout the iterations and represent the demand of request r_m that has not yet been assigned, and which is initialized as $\alpha'_m = \alpha_m$. Further, the pair of requests with the highest improvement is prioritized with the highest possible pooling demand assignment. That is, in each iteration, if $r_m, r_n \in \mathcal{R}$ is the pair of requests with the highest $\Delta\tilde{J}_{mn}$, we set $\beta_{mn} = \alpha'_m$ and $\beta_{nm} = \alpha'_n$. Moreover, the rides that have been assigned but not pooled, are added back to the original requests, i.e., we set $\alpha'_m = \beta_{mn} - \gamma_{mn}$ and $\alpha'_n = \beta_{nm} - \gamma_{nm}$. Let $\Delta\tilde{J}'_{mn}, m,n \in \mathcal{M}$ denote another auxiliary variable throughout the iterations, initialized as $\Delta\tilde{J}'_{mn} = \Delta\tilde{J}_{mn}$. At the end of every iteration, $\Delta\tilde{J}'_{mn}$ is set to 0. This procedure is repeated until convergence is achieved, i.e., $\max_{m,n}(\Delta\tilde{J}'_{mn}) \leq 0$. The pseudocode of this procedure is presented in Algorithm 1. In the following theorem, we establish the convergence and optimality of Algorithm 1.

Theorem II.1. *Let $X^\star_\gamma$ denote the optimal solution of Problem 2, under Approximation II.1, for the effective ride-pooling demand matrix γ. Then, in $|\mathcal{M}|(|\mathcal{M}|+1)/2$ iterations at most, Algorithm 1 converges to $\beta = \beta^\star$ and $\gamma = \gamma^\star$, which is a minimizer of $\tilde{J}(X^\star_\gamma)$ among all valid effective ride-pooling matrices.*

Proof. The proof can be found in Appendix II. □

D. Discussion

A few comments are in order. First, the mobility system is analyzed at steady-state in a time-invariant framework, which is unsuitable for an online implementation, but it has been used for planning and design purposes by several works in the literature as seen in Section 1. This assumption is reasonable if the travel requests vary slowly w.r.t. the average time of serving each request. This is the case especially in highly populated metropolitan areas [19]. Second, our framework does not take into account the stochastic nature of the exogenous congestion that determines that travel time in each road arc. However, this deterministic approach is suitable for our purposes as it provides an average representation of these stochastic phenomena in a mesoscopic scale [20]. Third, Problems 1 and 2 allow for fractional flows, which is acceptable because of the mesoscopic perspective of the work [1], [2], [9]. Finally, $D^{\rm rp}$ is not optimal w.r.t. the objective function of Problem 2, but it is w.r.t. its relaxed version, enabling a polynomial-time computation.

Algorithm 1 Compute optimal assignment matrices $\beta^\star, \gamma^\star$.

$\tilde{J}_{mn} \leftarrow$ input $D^{mn,\star}$ to Problem 2, $\forall m,n \in \mathcal{M}$
$\tilde{J}_m + \tilde{J}_n \leftarrow$ input $D^{mn,0}$ to Problem 2, $\forall m,n \in \mathcal{M}$
$\Delta\tilde{J}_{mn} \leftarrow \tilde{J}_m + \tilde{J}_n - \tilde{J}_{mn}$
$\Delta\tilde{J}'_{mn} \leftarrow \Delta\tilde{J}_{mn}, \forall m,n \in \mathcal{M}$
$\alpha'_m \leftarrow \alpha_m, \forall m \in \mathcal{M}$
while $\max_{m,n}(\Delta\tilde{J}'_{mn}) > 0$ **do**
  $(m,n) \in \operatorname{argmax}_{m,n}(\Delta\tilde{J}'_{mn})$
  if $o_n = o_m$ and $d_n = d_m$ **then**
    $\beta_{mn} \leftarrow \alpha'_m, \beta_{nm} \leftarrow \beta_{mn}$
    $\gamma_{mn} \leftarrow \beta_{mn}P(\beta_{mn}, \beta_{nm})/2, \gamma_{nm} \leftarrow \gamma_{mn}$
  else
    $\beta_{nm} \leftarrow \alpha'_n, \beta_{mn} \leftarrow \alpha'_m$
    $\gamma_{mn} \leftarrow \min(\beta_{nm}, \beta_{mn})P(\beta_{mn}, \beta_{nm})$
    $\gamma_{nm} \leftarrow \gamma_{mn}$
  end if
  $\alpha'_m \leftarrow \alpha'_m - \gamma_{mn}, \alpha'_n \leftarrow \alpha'_n - \gamma_{nm}$
  $\Delta\tilde{J}'_{mn} \leftarrow 0, \Delta\tilde{J}'_{nm} \leftarrow \Delta\tilde{J}'_{mn}$
end while

III. Case Study

This section showcases our modeling and optimization framework in a real-world case study of Sioux Falls, USA, with data obtained from the Transportation Networks for Research repository [21]. Problems 2 was parsed with YALMIP [22] and solved with Gurobi 9.5 [23]. We compute it leveraging the optimal ride-pooling assignment, obtained as described in Section II-C.4, for a varying amount of hourly demands, obtained by uniformly scaling the demand of the historical requests, and for various waiting times and maximum delays. In Fig. 2, we compare the relative improvement in objective of Problem 2 w.r.t. Problem 1, i.e., the improvement of the overall travel time. Fig. 2 shows that ride-pooling always contributes to lowering the overall travel time. In particular, the larger the number of hourly demands, the larger the relative improvement. The reason is that the probability function in (2) is monotonically increasing w.r.t. β_m and β_n that, in turn, are monotonically increasing with the number of demands. In Fig. 3 we note that the percentage of rides that are pooled is strongly influenced by the number

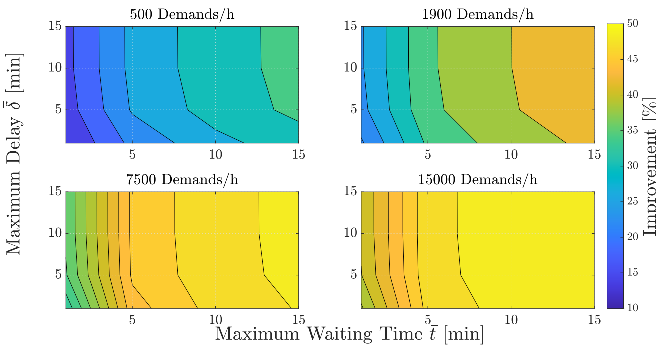

Fig. 2. Improvement of the objective function of Problem 2, i.e., overall travel time, w.r.t. no ride-pooling, as a function of maximum waiting time, delay, and demand intensity.

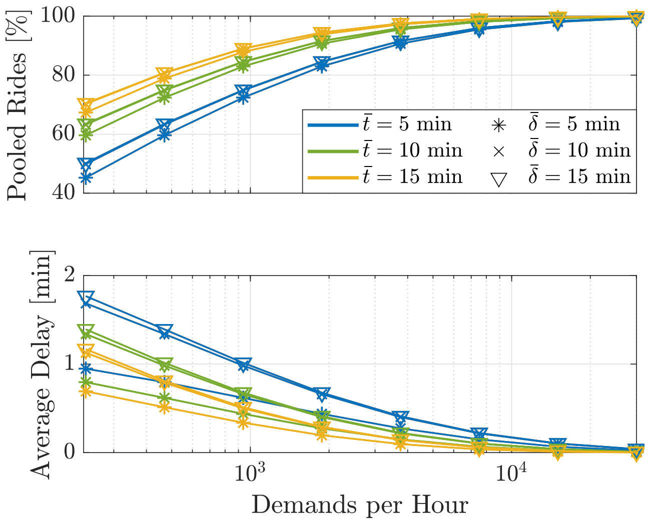

Fig. 3. Percentage of pooled rides and average experienced delay as a function of the overall number of hourly demands, waiting time, and maximum delay.

of demands, represented in logarithmic scale, to a lower extent by the maximum waiting time, and marginally by the maximum delay. In addition, for large demands, both the waiting time and the delay have a minor impact on the percentage of rides being pooled and on the relative improvement. Fig. 3 also depicts the average experienced delay by the users, which is significantly lower than the bounds imposed on Section II-C.1. Moreover, we highlight that the experienced delay decreases significantly for an increasing number of hourly demands. This phenomenon resembles the Mohring Effect [24], stating that the more people use a mobility service, the shorter the waiting time they experience. Conversely, the fewer people use a mobility service, the higher the waiting time, reflecting in a lower percentage of requests that can be effectively ride-pooled. Moreover, for the simulations performed, $t^\top x^\mathrm{r}$, i.e., the rebalancing time, accounts for less than 5% of the overall travel time for every scenario studied. The original rebalancing time for no ride-pooling is roughly the same. Thus, not only does ride-pooling decrease the overall rebalancing time due to the lower number of trips, but it also does not lead to a relative increase w.r.t. the ride-pooling overall travel time. This supports the hypothesis of Approximation II.1. Last, we notice that for a sufficiently large number of requests, by setting both a maximum delay and waiting time of 5 minute, it is possible to ride-pool more than 90% of the requests.

A MATLAB implementation of the methods presented is available in an open-source repository at `https://github.com/fabiopaparella/ride-pooling-MoD`.

IV. Conclusions

This paper presented a framework to capture ride-pooling in a time-invariant network flow model. Specifically, we proposed a framework wherein we devise an equivalent set of requests w.r.t. the original set so that the structure of the traffic flow problem remains unchanged. This allows to still obtain an LP problem that can be efficiently solved with off-the-shelf solvers in polynomial-time. Additionally, we proposed a method to compute a ride-pooling request assignment, that is optimal w.r.t. a relaxed version of the minimum travel time problem. Our case study of Sioux Falls quantitatively showed that the overall number of requests per unit time is a crucial factor to assess the benefit of ride-pooling in mobility-on-demand systems. In fact, for a sufficiently large number of travel requests, we achieved average improvements in the overall travel time of up to 45%. We also showed that, for a large number of requests, more than 90% of them could be pooled with a relatively short waiting and delay time.

This work opens up the research into multi-commodity traffic flow model planning, taking into account an ideal ride-pooling scenario. In the future, we would like to analyze the results with respect to the granularity of the road graph and build on this research by including endogenous traffic congestion.

Acknowledgments

We thank Dr. I. New, F. Vehlhaber, and J. Kampen for proofreading the paper. This publication is part of the project NEON with number 17628 of the research program Crossover, partly financed by the Dutch Research Council.

References

[1] M. Salazar, N. Lanzetti, F. Rossi, M. Schiffer, and M. Pavone, "Intermodal autonomous mobility-on-demand," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3946–3960, 2020.

[2] J. Luke, M. Salazar, R. Rajagopal, and M. Pavone, "Joint optimization of electric vehicle fleet operations and charging station siting," in *Proc. IEEE Int. Conf. on Intelligent Transportation Systems*, 2021.

[3] M. Pavone, S. L. Smith, E. Frazzoli, and D. Rus, "Robotic load balancing for Mobility-on-Demand systems," *Proc. of the Inst. of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 31, no. 7, pp. 839–854, 2012.

[4] F. Rossi, "On the interaction between Autonomous Mobility-on-Demand systems and the built environment: Models and large scale coordination algorithms," Ph.D. dissertation, Stanford University, Dept. of Aeronautics and Astronautics, 2018.

[5] F. Rossi, R. Iglesias, M. Alizadeh, and M. Pavone, "On the interaction between Autonomous Mobility-on-Demand systems and the power network: Models and coordination algorithms," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 384–397, 2020.

[6] F. Rossi, R. Zhang, Y. Hindy, and M. Pavone, "Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms," *Autonomous Robots*, vol. 42, no. 7, pp. 1427–1442, 2018.
[7] K. Spieser, K. Treleaven, R. Zhang, E. Frazzoli, D. Morton, and M. Pavone, "Toward a systematic approach to the design and evaluation of Autonomous Mobility-on-Demand systems: A case study in Singapore," in *Road Vehicle Automation*. Springer, 2014.
[8] R. Iglesias, F. Rossi, K. Wang, D. Hallac, J. Leskovec, and M. Pavone, "Data-driven model predictive control of autonomous mobility-on-demand systems," in *Proc. IEEE Conf. on Robotics and Automation*, 2018.
[9] F. Paparella, K. Chauhan, T. Hofman, and M. Salazar, "Electric autonomous mobility-on-demand: Joint optimization of routing and charging infrastructure siting," in *IFAC World Congress*, 2023, in Press.
[10] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, jan 2017.
[11] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 3, pp. 13 290–13 294, 2014.
[12] K. Jintao, Y. Hai, L. Xinwei, W. Hai, and Y. Jieping, "Pricing and equilibrium in on-demand ride-pooling markets," *Transportation Research Part B: Methodological*, vol. 139, no. C, pp. 411–431, 2020.
[13] A. Fielbaum, X. Bai, and J. Alonso-Mora, "On-demand ridesharing with optimized pick-up and drop-off walking locations," *Transportation Research Part C: Emerging Technologies*, vol. 126, p. 103061, 2021.
[14] A. Fielbaum, R. Kucharski, O. Cats, and J. Alonso-Mora, "How to split the costs and charge the travellers sharing a ride? aligning system's optimum with users' equilibrium," *European Journal of Operational Research*, vol. 301, no. 3, pp. 956–973, 2022.
[15] M. Tsao, D. Milojevic, C. Ruch, M. Salazar, E. Frazzoli, and M. Pavone, "Model predictive control of ride-sharing autonomous mobility on demand systems," in *Proc. IEEE Conf. on Robotics and Automation*, 2019.
[16] A. Aouad and O. Saritac, "Dynamic stochastic matching under limited time," *Operations Research*, vol. 70, no. 4, pp. 2349–2383, 2022.
[17] F. Paparella, B. Sripanha, T. Hofman, and M. Salazar, "Optimization-based comparison of rebalanced docked and dockless micromobility systems," in *Proc. of the Conf. on Sustainable Urban Mobility*. Cham: Springer Nature Switzerland, 2022, pp. 633–644.
[18] F. Bullo, *Lectures on Network Systems*, 1st ed. Kindle Direct Publishing, 2020, with contributions by J. Cortes, F. Dorfler, and S. Martinez. [Online]. Available: http://motion.me.ucsb.edu/book-lns
[19] R. Iglesias, F. Rossi, R. Zhang, and M. Pavone, "A BCMP network approach to modeling and controlling autonomous mobility-on-demand systems," *Proc. of the Inst. of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 38, no. 2–3, pp. 357–374, 2019.
[20] H. Neuburger, "The economics of heavily congested roads," *Transportation Research*, vol. 5, no. 4, pp. 283–293, 1971.
[21] T. N. for Research Core Team. Transportation networks for research. https://github.com/bstabler/transportationnetworks. Accessed January, 05, 2023.
[22] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *IEEE Int. Symp. on Computer Aided Control Systems Design*, 2004.
[23] Gurobi Optimization, LLC. (2021) Gurobi optimizer reference manual. Available at http://www.gurobi.com.
[24] A. Fielbaum, A. Tirachini, and J. Alonso-Mora, "Economies and diseconomies of scale in on-demand ridepooling systems," *Economics of Transportation*, vol. 34, p. 100313, 2023.
[25] G. B. Dantzig, "Discrete-variable extremum problems," *Operations Research*, vol. 5, no. 2, pp. 266–288, 1957.

APPENDIX I
PROOF OF LEMMA II.1

Recall that the exponential distribution, whose probability density function is given by $f(x) = \alpha e^{-\alpha x}$, models the time between events in a Poisson process of parameter α. Since the two Poisson processes are independent,

$$P(\alpha_m, \alpha_n) = \int_0^{\bar{t}} \alpha_n e^{-\alpha_n t_n} \left(\int_0^{t_n+\bar{t}} \alpha_m e^{-\alpha_m t_m} \mathrm{d}t_m \right) \mathrm{d}t_n + \int_{\bar{t}}^{\infty} \alpha_n e^{-\alpha_n t_n} \left(\int_{t_n-\bar{t}}^{t_n+\bar{t}} \alpha_m e^{-\alpha_m t_m} \mathrm{d}t_m \right) \mathrm{d}t_n,$$

where the presence of two terms arises from the fact that the time interval $[0, +\infty)$ is considered. Making use of standard integral calculus techniques, it can be rewritten as (2).

APPENDIX II
PROOF OF THEOREM II.1

The convergence of Algorithm 1 in at most $|\mathcal{M}|(|\mathcal{M}|+1)/2$ iterations is immediate. In fact, since for each pair (m, n) chosen in each iteration we set $\Delta\tilde{J}'_{mn} = \Delta\tilde{J}'_{nm} = 0$, neither (n, m) nor (m, n) will be chosen again. The optimality of the solution $\beta^\star$ and associated $\gamma^\star$ is carried out making use of an analogy with the continuous Knapsack problem, which can be solved by a well-known polynomial-time greedy algorithm [25]. Recall that such algorithm consists in, every iteration, allocating the maximum amount of the resource with the highest improvement in the objective function per unit of the resource, which is intuitively evident. Similarly to the continuous Knapsack problem, the goal is to minimize $\tilde{J}(X^\star_\gamma)$ by allocating $\gamma_{mn} \geq 0$ with $m, n \in \mathcal{M}$. First, borrowing the notation from Section II-C.1, if γ_{mn} is assigned, then the corresponding decrease in the cost function amounts to $\tilde{J}(\gamma_{nm} X^{mn,0}) - \tilde{J}(\gamma_{mn} X^{mn,\star}) = \gamma_{mn}(\tilde{J}(X^{mn,0}) - \tilde{J}(X^{mn,\star})) = \gamma_{mn}\Delta\tilde{J}_{mn}$, where the linearity of $\tilde{J}$ played a key role. Thus, the allocation of γ_{mn} leads to a relative improvement on the cost that amounts to $\Delta\tilde{J}_{mn}$. Second, as pointed out in Section II-C.4, throughout the algorithm, α'_m corresponds to the demand of r_m which has not yet been ride-pooled with another request. Thus, the value of γ_{mn} that can be allocated has an upper bound given by $\gamma_{mn} \leq \min(\alpha'_m, \alpha'_n) P(\alpha'_m, \alpha'_n)$. Note that Algorithm 1 corresponds to allocating the maximum amount of γ_{mn}, where m and n are such that, at each iteration, the highest positive relative improvement in the objective function is achieved, i.e., $(m, n) \in \operatorname{argmax}_{m,n}(\Delta\tilde{J}'_{mn})$, which shows its optimality.