ANTICIPATORY ROUTING METHODS FOR AN ON-DEMAND RIDEPOOLING MOBILITY SYSTEM

Andres Fielbaum, TU Delft – a.s.fielbaumschnitzler@tudelft.nl
Maximilian Kronmuller, TU Delft – m.kronmuller-1@tudelft.nl
Javier Alonso-Mora TU Delft– j.alonsomora@tudelft.nl

Palabras clave: Ridepooling , On-demand , Anticipatory , Mobility , Ridesharing

ABSTRACT

On-demand mobility systems in which passengers use the same vehicle simultaneously are a promising transport mode, yet difficult to control. One of the most relevant challenges relates to the spatial imbalances of the demand, which induce a mismatch between the position of the vehicles and the origins of the emerging requests. This paper introduces two types of techniques for anticipatory routing that affect how vehicles are assigned to users and how to route vehicles to serve such users. Firstly, we introduce rewards that reduce the cost of an assignment between a vehicle and a group of passengers if the vehicle gets routed towards a high-demand zone. Secondly, we include a small set of artificial requests, whose request times are in the near future and whose origins are sampled from a probability distribution that mimics observed generation rates. These artificial requests are to be assigned together with the real requests. We test these techniques in combination with a state-of-the-art trip-vehicle assignment method, using a set of real rides from Manhattan. Introducing rewards can diminish the rejection rate to about nine-tenths of its original value. On the other hand, including future requests can reduce users' traveling times by about one-fifth, but increasing rejections.

1. INTRODUCTION

Centrally controlled on-demand ridepooling systems, in which different users can ride the same vehicle at the same time if their paths are compatible, are a promising mobility system for the future of cities, because they can exhibit many of the advantages of popular (non-shared) on-demand systems more sustainably without increasing congestion.

Massive on-demand systems (apps) have become popular due to a number of virtues: short waiting times, door-to-door service, ease of payment, an increase of comfort, and no need for parking nor driving (Tirachini & del Río (2019); Tang et al. (2019)). All these positive features can be kept when rides are shared (pooled) as well.

Moreover, sharing can effectively fight congestion and emissions if an adequate fleet is selected (Tirachini et al. (2019); Li et al. (2021)). Empirical studies have shown that carsharing systems in which rides are not shared have increased congestion, as they attract many users from public transport (Henao & Marshall (2019); Tirachini & Gomez-Lobo (2020); Diao et al. (2021)). When rides are shared, vehicles make more efficient use of the scarce vial space.

One of the main difficulties of massive on-demand systems is related to their dynamics. The system needs to decide the assignments as the requests appear. Even if these assignments are decided optimally according to the current conditions, they might leave the system in a state that is inefficient to serve the demand that will emerge afterward. Let us consider an extreme and simplified example that helps to visualize the situation: a circular city, in which the users are located at the border and are all traveling to the center. Figure 1 shows such a scenario, simulating an exaggerated version of a morning peak situation. This demand unbalanced demand pattern will make all the cars converge rapidly at the center; if there are bounds on the maximum waiting time (a usual assumption in these models) and the time required to go back to the border exceeds this bound, then vehicles will not be assigned to any new request, and the system would collapse.

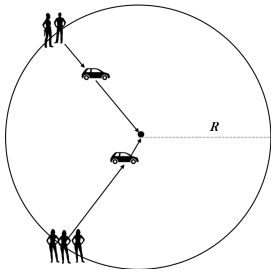

Figure 1: A ridepooling system in an extreme version of a morning peak situation. If the time required to drive R exceeds the maximum waiting time, all vehicles will get stuck in the center.

To prevent such situations, most of the algorithms include a *rebalancing* step, in which cars *that are not being used* are sent to some zone in which they are expected to be required. Note that underlying this idea, there has to be some way to determine which zones of a city will require vehicles in the future, i.e., there are some *anticipatory* decisions. However,rebalancing does not interact directly with assignments, and it only deals with idle vehicles. These aspects can have substantial drawbacks. A relevant limit to rebalancing strategies is given by the number of vehicles that can be controlled, which might be much lower than the total number of vehicles in the system when only idle vehicles are considered.

Furthermore, sometimes the assignment process may produce some inefficient matches, which cannot be corrected through rebalancing. In particular, some vehicles might be directed towards low-demand areas to give a better quality of service at a specific time, but being barely shared. Such situations worsen future service, as vehicles move to zones in which they are not likely to be required or used to their full potential (if the remain being low-demand).

These problems are inherent to the system's combination between sharing and having flexible routes. When cars are shared but rides are not (i.e., on-demand taxis), finding a single passen-

ger is as good as possible, so there is no need to go to the most demanded zones if there is some demand in the rest of the network. In public transport, lines are designed a priori to serve better the most demanded zones through higher frequencies, shorter distances to bus stops, and more direct services (Fielbaum et al. (2020)), i.e., the mismatch between supply and demand is prevented thanks to having fixed routes.

To prevent such situations, techniques that are beyond rebalancing are needed. In this paper, we study methods that introduce anticipatory decisions at an earlier stage, namely when deciding the assignments (i.e., which vehicle is carrying which passengers) and the routes (i.e., in which order are they served), in order to have the whole system better prepared for future service. The techniques we propose do not require any exogenous assumptions about future requests. They take as input only the endogenous information that is generated while operating the system, namely the origins of current and recent requests (we also study the use of historical data as a benchmark, and we show it performs worse).

We study two types of techniques: First, by modifying the cost of each possible assignment between vehicles and set of requests, favoring those assignments that conduct the vehicle towards the most demanded zones; second, we introduce artificial future requests to the pool of requests that have to be assigned at each time, with origins in those same high-demanded zones so some vehicles might be sent towards them. Several specific implementations for each technique are analyzed and compared.

2. RELATED WORKS

In the context of ridepooling, few papers deal with anticipatory routing or assignments, as we do here. In a simplified context, in which users travel between specific stations among the area covered by the system, Barth et al. (2004) propose a method that could be used in more general schemes: splitting groups of passengers into many vehicles if their destinations are located in a zone that is expected to require more vehicles in the near future. van Engelen et al. (2018) works with an event-based model, in which each time a new request emerges, it is assigned to a vehicle that is chosen using demand forecast; similar ideas are proposed by Wang et al. (2020), but assigning several requests at a time.

Two papers propose predictive routing and assignment ideas building upon the same model that we use for simulations (Alonso-Mora, Samaranayake, et al. (2017)), and using related techniques: Alonso-Mora, Wallar, & Rus (2017)) estimates historical demand for each zone, and includes some artificial requests according to this estimated demand in order to push the whole system towards a better preparation for the future; Huang & Peng (2018) modifies the cost function with an additive term that depends on the spatial distribution of the vehicles and its distance to the optimal one. These papers have some drawbacks: the former requires historical data that is not always available, it increases the computational time heavily, and it does not have a meaningful impact on reducing the number of rejected requests of the system (it does reduce average waiting times and delay); while the latter requires a perfect knowledge of the demand distribution, and it does not affect the routes of the vehicles for a given set of pick-ups and drop-offs.

3. TWO ANTICIPATORY METHODS

To explain our methods in detail, we first introduce some terminology and the formal statement of the assignment problem. We aim to match requests together and assign them to vehicles efficiently, during some predefined period of operation that lasts PO. However, the requests that are going to be served are not known beforehand but emerge throughout the operation.

Let us first introduce relevant terminology and notation:

- The problem takes place over a directed graph $G = (V, E)$ representing the road network used by the vehicles.
- A *request* r is a single call from a user (or a number of them traveling together) that emerges at time tr_r, and needs to be transported from an origin o_r to a destination d_r, both located on the nodes of the graph.
- A *vehicle* v is characterized at each time t by its capacity μ_v, its current position $Pos_v(t)$, a set of requests that it is currently serving $Req_v(t)$, and its planned route $\pi_v(t)$ (i.e., a path over the graph). The set of vehicles at time t is denoted $\mathcal{V}(t)$.
- A *trip* $T = (r_1, ..., r_k)$ is a set of requests. Such a trip is feasible to be transported by a vehicle v at time t, if there exists a route π that serves the requests in T and in $Req_v(t)$, fulfilling a set of constraints like vehicles' capacities or maximum waiting times.
- Consider a feasible matching between a vehicle v and a trip T at time t, that instructs v to follow a new route π. We define the cost $c(v, T, \pi, t)$ induced to the system, that might include the costs for requests in T, extra costs for requests in $Req_v(t)$ (because the updated route might induce longer traveling times for such requests), and operator's costs c_O.

As this is a dynamic problem with partial information, the system needs to decide how to group the requests and how to assign them to vehicles several consecutive times during the whole period of operation. The methods we introduce now affect each of these consecutive assignments.

3.1. Assignment introducing rewards

The first assignment method we propose consists in modifying the cost functions of feasible assignments to favor those that move the vehicles towards high-demand zones. Without any anticipatory technique, the system does not account for where the vehicle will be situated when new requests emerge, so we aim to face this issue by affecting the optimization procedure by reducing the costs of those routes and assignments that instruct the vehicles to move toward more convenient locations for the future.

Recall that $c(v, T, \pi)$ is the original cost (without anticipatory methods) of inserting trip T into vehicle v if the updated route including T is π (that is required to serve all requests in T and all the

previous requests being served by v). The route before inserting the new trip is π_v. The anticipatory routing and assignment are achieved by modifying this cost function, adding a *reward* R, which is a (negative) additive term:

$$c_A(v, T, \pi) = c(v, T, \pi) - \Theta R(v, T, \pi) \tag{1}$$

The impact of Eq. (1) on the system can be twofold: on the one hand, if v is assigned to serve T, there might be more than one feasible route that fulfills all the constraints $\mathcal{C}$, so usually (and we assume this is the case for the analyses that follow) the route is chosen minimizing the cost function; thus, different routes might be selected when using c_A instead of c. On the other hand, the decision of which vehicles assign to which trips is taken minimizing the sum of the costs of the selected assignments, a procedure that yields different results when c_A is used instead of c.

The crucial question is how to define the reward R. We propose several specifications (explained in detail in section 4); for the sake of simplicity, all of them depend on some characteristics of a particular node of the induced route π. Two questions naturally arise: which node to look at, and what to observe from that node? First, the rewards will be a function of the last node of π. Second, we will consider either the number of requests that departs there $Gen(u, t)$ (a *generation rate*, as in Vosooghi et al. (2019); Lioris et al. (2016)), or the number of requests that are rejected there $Rej(u, t)$ (a *rejection rate*, as in Alonso-Mora, Samaranayake, et al. (2017)).

There might be several ways to define the generation and rejection rate of a particular node. We propose three methods, explained in section 4. None of these definitions shall require any knowledge about the future. We do consider a fourth method as a benchmark that is based on historical requests, which does not imply assumptions regarding the future but require exogenous data.

3.2. Assignment inserting future artificial requests

The method explained above has the virtue of affecting the system's decisions at every level (routing and assignments). However, it lacks a global insight into the system's performance: all the rewards are evaluated equally if evaluated in the same nodes, so all the vehicles are pointed towards the same zones. There is no mechanism to achieve a balance at a whole-network level. With this in mind, we implement a method based on Alonso-Mora, Wallar, & Rus (2017) but able to work with different generation rates (other than requiring historical data) and assignment procedures (other than the one from Alonso-Mora, Samaranayake, et al. (2017)).

As the introducing rewards method, this method can also be used with different assignment procedures, as long as they can handle requests with future request times, and that there is a fixed penalty per rejected request p_{KO}.

Recall that we are deciding how to assign the vehicles to a set of requests during a single stage of the whole operation. At a high-level description, this anticipatory method consists in adding to that set of requests some future artificial ones, whose origins are located in high-demand zones, so that

some vehicles might get assigned to them and move towards their origins:

- First, define a generation rate per node. The same generation rates used to define the rewards will be considered, which are explained in section 4.
- Second, generate m artificial random requests whose:
  - Origins are selected randomly, following a distribution given by the generation rates.
  - Destinations are selected such that the length of the artificial requests is similar to the average length of the real ones. This is done to yield operator's costs that are close to the ones of the real requests. By this means, the operator's costs do not play a too determinant role when deciding whether to serve the artificial requests.
  - Times at which they emerge are $\tau_i + k \cdot \phi$ with $k = 1, ..., m$, where τ_i is the current time and ϕ is some parameter (that has time units). Artificial requests are equidistant in time to prevent them from being too close, which could make them either impossible to group (if they are close in space), or feasibly to be matched with any vehicle (in the opposite case). Therefore, ϕ should be large enough so that the artificial requests are not too close to each other, but not too large so that some future requests could be matched with the current ones.
- The assignments are decided following the same original assignment procedure, considering both the real and the artificial requests. Artificial requests, however, present a lower rejection penalty $p'_{KO} = \Gamma \cdot p_{KO}$, where p_{KO} is the rejection penalty for a real request and $\Gamma \in (0, 1)$.
- The artificial requests are erased after deciding the assignment and updating vehicles positions. That is, they are not kept in vehicles' lists. Therefore, they are never served, and they do not affect subsequent assignments. They only impact the immediate routes followed by the vehicles.

4. DIFFERENT DEFINITIONS FOR GENERATION AND REJECTION RATES

4.1. Basic rates

The simplest way to define the generation (rejection) rate of a node u is to look at the number of requests that have just been generated (rejected) at u. Note that the generation rate depends only on the set of requests, and the rejection rate also depends on how they are assigned. Denoting $Rej(\tau_i, u)$ as the number of requests emerging from u that were rejected by the system at the corresponding assignment, the basic rates are defined by:

$$Gen_B(u, \tau_i) = |\{r \in \mathcal{R}_{e,\tau_i} : o_r = u\}|, Rej_B(u, \tau_i) = Rej(\tau_{i-1}, u) \tag{2}$$

For $i = 1$, the rejection rates are defined as zero everywhere.

4.2. Smooth rates

The aim of anticipatory routing is that vehicles remain closer to where demand is expected. From that point of view, the rates of a node could also consider the information of its neighboring nodes: for instance, it might be better for a vehicle to be in a node that does not generate requests if all its neighbors do. With this in mind, the Gen_S and Rej_S rates are defined considering also the requests that depart from close nodes:

$$Gen_S(u,\tau_i) = \sum_{w \in V} \frac{Gen_B(w,\tau_i)}{\psi + t_V(u,w)}, Rej_S(u,\tau_i) = \sum_{w \in V} \frac{Rej_B(w,\tau_i)}{\psi + t_V(u,w)}. \quad (3)$$

Where $t_V(x,y)$ is the time-length of the fastest path between x and y, and ψ is a tuning parameter (the higher this parameter, the more uniform the resulting rates). This method is called "smooth rates" because rates become more stable in space. Note that all nodes are included in Eq. (3), but distant nodes do not affect much.

4.3. Particle filters

This method applies the ideas from Wallar et al. (2018) (based on the particle filter methods proposed by Arulampalam et al. (2002)) to calculate the generation/rejection rates of a zone, although Wallar et al. (2018) used it for rebalancing purpose. It requires first dividing the nodes into clusters $C_1, ..., C_M$, which are obtained by minimizing the number of required "centers", such that each node in the network can be reached from at least one center in a time lower than a parameter t_M. This problem is solved through an ILP that is fully described in Wallar et al. (2018), and each node is then assigned to its closest center. All the nodes assign to the same center comprise a zone. Note that by this method all the zones have a similar area. This is crucial to have rates that are comparable (otherwise, larger zones could present higher absolute rates but having a lower density of requests, so that it would be unclear whether to prioritize them). The particle filter method updates the rates of each zone by randomly perturbing previous rates, according to the recent basic rates. For details, the reader is referred to Wallar et al. (2018).

4.4. An additional definition

As explained above, we only use information that is endogenously generated by the system when assigning. The only exception is the rates we explain now, which are based on exogenous historical data so that they might be used as a benchmark to analyze how useful current requests are to approximate what will happen in the near future.

This method adapts the ideas from Alonso-Mora, Wallar, & Rus (2017). They also divide the nodes into clusters, so we keep here the technique used for the particle filter method (previous subsection). Then, they estimate the number of requests emerging from a zone using a historical dataset. Which dataset to use is not a trivial issue, as transport demand can be heavily affected by weather, traffic events, among others (Böcker et al. (2013); Liu et al. (2021)), which makes data-based demand

prediction a quite complex challenge, beyond the scope of this paper. Nevertheless, we do take this discussion into account: instead of considering a whole year of data (as Alonso-Mora, Wallar, & Rus (2017)), we use only some weekdays in the past, which are expected to have more similar weather (same season) and fewer differences in the private and public transport networks. Denoting $\mathcal{D}$ the set of days in the dataset, and $G(u, d, t_1, t_2)$ the number of requests emerging from node u during (t_1, t_2) on day d:

$$Gen_H(z, \tau_i) = \sum_{u \in z} \sum_{d \in \mathcal{D}} \frac{G(u, d, \tau_{i-1}, \tau_i)}{|\mathcal{D}|} \tag{4}$$

With all the zones beginning with a nil generation rate. And

$$Gen_H(u, \tau_i) = Gen_H(z, \tau_i) \forall u \in z \tag{5}$$

We only use this method for generation rates, as there is no such thing as "historical rejection rates".

5. NUMERICAL SIMULATIONS

5.1. The real-life study case

The proposed methods are tested over a publicly available dataset of real trips performed by taxis in Manhattan, New York, that started between 1-2 p.m. on January 15th, 2013. The total number of requests is 7,748, while 4,091 nodes and 9,452 edges form the city network. We take as a base the assignment procedure by Alonso-Mora, Samaranayake, et al. (2017).

A fleet of 1,000 vehicles of capacity 3 was considered. This is a small fleet, unable to serve all the demand. Having a significant rejection rate enables us to analyze the impact of the methods over rejections in a crisp way, as well as how anticipatory techniques affect the trade-off between the number of served requests and the quality of service for those who are served.

5.2. Global performance of the anticipatory methods

5.2.1 Assignment introducing rewards: Comparison of the different rates

In section 4, four definitions for the rates were provided: basic (B), smooth (S), calculated through particle filters (PF), and through historical data (H). The first three ones can be applied to generation or rejection rates of each node (that, as just explained, is the final node of the route), whereas historical data only gives generation rates. All this together makes seven methods to define the different rates.

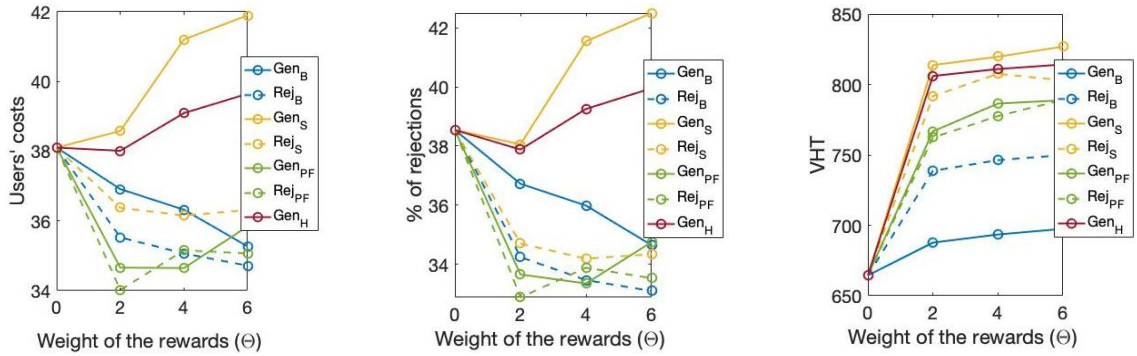

Figure 2: Average users' costs (left), percentage of rejected requests (center), and vehicle-hours-traveled (right), as a function of the parameter Θ, for each of the generation and rejection rates when introducing rewards. Solid lines represent results based generation rates, whereas dotted lines represent results based on rejection rates. $\Theta = 0$ represents no anticipatory methods.

The results obtained with each of these seven rates are shown in Figure 2. Some of the rates are much more effective in reducing the number of rejections and users' costs than others. Two rates highlight as the best ones: Rej_{PF} and Gen_B. While the former achieves the minimum values for the rejection rates and the users' costs, the latter achieves slightly worse results in those measures, requiring a lower increase in VHT. It is worth saying that if Θ is further increased, Gen_B does not longer improve its results. Moreover, the results of the best methods show that these rewards can be very fruitful. For instance, the number of rejected requests with Rej_{PF} drops from 3,025 to 2,631. Of course, whether this is good news depends on how opposing objectives are evaluated, as increasing VHT is unavoidable.

The tuning parameter Θ emerges as a crucial issue for these models. Which is the best Θ depends on what rate is being used. The good news is that for all of them, the range of values of Θ that yields good results is wide, meaning that these methods are robust even if the optimal Θ is not known. Rates that are based on rejections have, in general, better results concerning both lower users' costs and VHT. The only exception is Gen_B, which increases VHT much less than Rej_B, with similar (although worse) results regarding users' costs. Regarding smoothing rates, they can improve the system if based on rejections. The rates based on historical data perform worse than the ones based on recent information. That is to say, our results highlight the potential of utilizing the information that is directly generated by the system.

As a synthesis, **the introduction of rewards reduces the percentage of rejections of the system if the right generation or rejection rates are selected, at the cost of providing worse service for the users that are transported**.

5.2.2 Assignment inserting future artificial requests: Comparison of the different rates

The same four generation rates Gen_B, Gen_S, Gen_{PF} and Gen_H were used to determine the origin of m artificial requests. We take $\phi = \delta$ (1 minute). Inserting future requests makes the algorithm

much slower, as they can be combined with most of the current requests without violating the constraints regarding maximum waiting times and delay, which increases the number of feasible groups. To overcome this issue, we use a heuristic that consists of discarding a larger amount of feasible-but-costly vehicles when analyzing trips of size one.

Figures 3 and 4 show the results when inserting the artificial trips with each of the generation rates. The baseline (in black) corresponds to the results shown in the previous subsection for $\Theta = 0$ (i.e., without tightening the heuristic that discards vehicles), whereas the results for $\Gamma = 0$ include the change on the heuristic and are equivalent to having no artificial requests. The comparison of the results against $\Gamma = 0$ shows the direct effect of introducing these artificial requests. However, the most relevant comparison is against the baseline because it is achieved if no artificial requests are added. Note that $\Gamma = 0$ (i.e., when the heuristic is tightened) includes more rejections than the baseline, but better results in the other indices (much lower waiting times and detours just a bit larger), which is a natural consequence of removing the most costly vehicles for each request: each request has fewer options to be served, but the options that remain are less costly, i.e., provide a better quality of service (recall that the cost is defined as the sum of users' costs and operator's costs).

In general, all the rates are able to reduce waiting times and detours significantly. The percentage of rejections, on the other hand, is always higher than in the baseline: when compared with $\Gamma = 0$, rejections are sometimes larger and sometimes fewer, but changes are always minor (these results are similar to the ones obtained by Alonso-Mora, Wallar, & Rus (2017)).

These results can be synthesized by stating that **inserting artificial requests by itself improves the quality of service for those users that are transported, but the induced increase to the computational time requires using heuristics that might increase the number of rejections.** The interpretation is as follows: Artificial requests effectively push the vehicles towards the origins of future requests. However, when they are inserted, they compete with current requests for the same vehicles so that sometimes the system will prioritize serving the artificial ones despite their lower rejection penalty. VHT always increases.

The method that achieves the best results is Gen_B: it presents the largest reduction in the rejection rate with $\Gamma = \frac{1}{80}$, and in detours for $\Gamma = \frac{1}{60}$. Reductions in waiting times are similar for all the generation rates, except for Gen_H when $\Gamma = \frac{1}{40}$. Results obtained by Gen_S and Gen_{PF} are almost identical. The fact that Gen_H might yield the worst results if Γ is not properly selected (i.e., it is a less robust method) reinforces the conclusion that the direct use of past information can be an unfruitful idea for these transport systems.

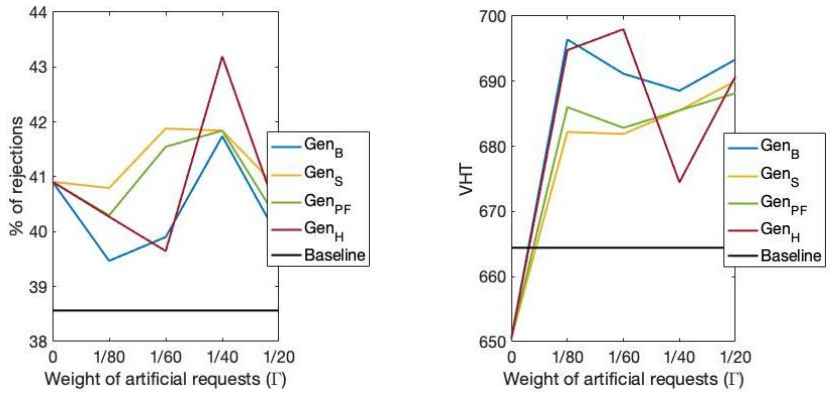

Figure 3: Rejection rates (left), and vehicle-hours-traveled (right), when inserting artificial requests as a function of the parameter Γ, for each of the four generation rates. The baseline results, i.e. with no anticipatory methods and without modifying the heuristics of the assignment algorithm, are shown in black.

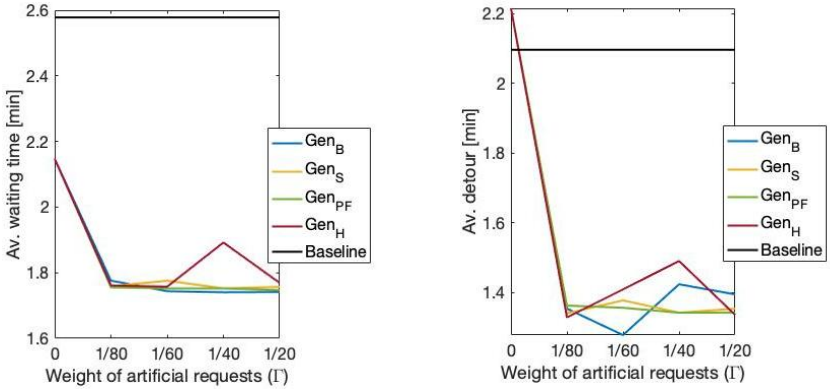

Figure 4: Average users' waiting time (left), and detour (right) when inserting artificial requests, as a function of the parameter Γ, for each of the four generation rates. The baseline results, i.e. with no anticipatory methods and without modifying the heuristics of the assignment algorithm, are shown in black.

In all, the insertion of future requests achieves reductions precisely where rewards fail: waiting times and detours. This happens because both methods move the vehicles towards high-demand zones, but inserting future requests might yield to rejecting some real current ones, so that the gains in efficiency translate into waiting and delay for the requests that will emerge afterward.

5.3. Detailed analysis of the impact over the system

So far, we have analyzed the methods in terms of the most relevant indices of the system. However, when we introduced the need for this type of method, we justified it by analyzing the spatial heterogeneity of the results, and the influence of deciding with partial current information. Therefore,

we now turn to analyze how the operation of the system is being modified. We focus on the method that introduces rewards, as it yields the best results, considering the rates Gen_B and Rej_{PF} that proved stronger.

5.3.1 Impact over the temporal evolution of the system

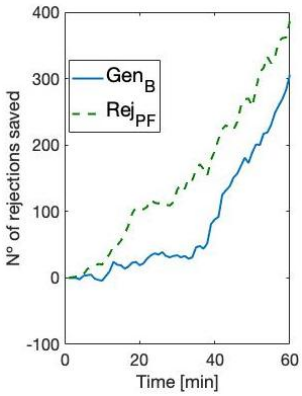

Figure 5: Difference in the accumulated rejections with or without anticipatory methods. The y-axis shows the difference between the number of accumulated rejections if using no anticipatory methods at all, or introducing rewards and using Gen_B (solid blue curve) or Rej_{PF} (dotted green curve); the x-axis represents the number of iterations (evolution in time), where we run one iteration per minute.

We first study how the number of rejections evolves in time, compared to the case with no anticipatory methods ($\Theta = 0$). We know that Gen_B and Rej_{PF} have less rejection in total, but when does this happen? Figure 5 shows the difference between the accumulated rejections with no rewards, and the accumulated rejections with rewards for both rates: as before, the solid blue curve represents Gen_B, and the dotted green curve represents Rej_{PF}.

Both curves begin quite flat and even take negative values, meaning that in the first iterations, rewards worsen the system's quality. The Rej_{PF}'s curve rapidly starts to increase (i.e., to have fewer rejections than the method with no anticipation), whereas Gen_B requires almost half an hour to do so, which verifies that the central motivation of these methods is achieved: modifying its current decisions to be better prepared for future requests. Note that, eventually, both methods reach an almost-linear increase, i.e., they keep saving rejections at a rate that keeps somewhat constant.

5.3.2 Impact over the spatial mismatch between vehicles and requests

We now analyze how the operation changes in space. We have noticed before that the most demanded zones were receiving a worse quality of service, so that more vehicles seemed to be required there. To analyze the changes, Figure 6 shows, at the end of the hour that was modeled, the differences in the vehicles' positions between having no anticipatory methods and Gen_B (left) or

Rej_{PF} (right). We partition the whole map into the same zones used for the methods with particle filters and historical data, and each vehicle is assigned to the zone corresponding to its closest node. A red sector means that there were more vehicles assigned with rewards, whereas blue means the opposite: the more intense the color, the higher the difference.

Both Figures have almost only blue zones at the north of the network. In the center, intense red zones clearly dominate for Rej_{PF}, and not so clearly for Gen_B. That is to say, rewards are indeed moving some vehicles from the north of the network (a low demand area) to the center. It is worth saying that Gen_B makes no noticeable difference in the south, while Rej_{PF} increases the number of vehicles there.

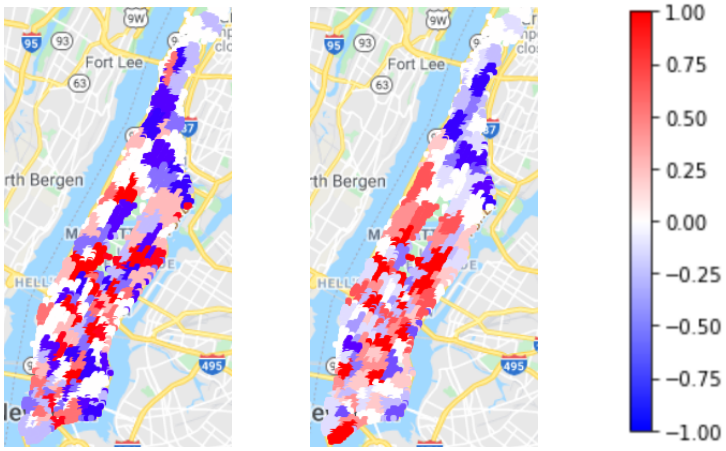

Figure 6: Differences in the location of the vehicles when using no anticipatory methods at all, or introducing rewards and using Gen_B (left) or Rej_{PF} (center). Each vehicle is assigned to the zone corresponding to its closest node after sixty minutes of operation of the system. A red zone means that more vehicles are assigned there in the anticipatory scheme, whereas a blue zone means the opposite; the more intense the color, the higher the difference, as shown in the colormap (right). Figures are normalized with respect to the maximum values.

6. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we propose techniques to reduce the mismatch between vehicles' positions and users' origins that occur in an on-demand ridepooling system, when the process that optimizes the assignments between vehicles and users does not take future requests into account. Such techniques only require the information that is generated when operating the system, by looking at the zones where most requests are emerging and being rejected.

We propose two anticipatory techniques: in the first one, we modify the objective cost function by introducing a reward (a negative additive term) to each feasible matching between a vehicle and a group of requests, such that the reward is greater when the vehicle's route finishes in a high-demand area; the second method includes artificial future requests emerging from the high-demand zones, to be assigned together with the real current ones, so that if a vehicle is assigned to a future request,

it will be moved towards its origin.

Numerical simulations reveal that introducing rewards effectively reduces the number of rejections, at the cost of increasing total delay for the served users. In contrast, the inclusion of artificial future requests reduces total delay but increasing the number of rejections unless the number of requests is small enough to have zero real requests during some minutes. Both methods require vehicles to move more. Results depend on the rates being used, and applying both methods together does not yield good results.

As on-demand ridepooling is an emerging mobility system, there is plenty of room for future research. A relevant direction identified through the paper is the optimal selection of the parameters Θ and Γ, i.e., the relative weight of the anticipatory components. Different methods and scenarios require adapting Θ and Γ, so using a non-constant value would be ideal. How to tune this value according to the external and internal conditions is a challenging and relevant question that could be addressed using learning procedures. Additionally, converting the rewards into monetary incentives, as well as comparing the methods we propose with others that use reinforcement learning, are other relevant topics for future research.

REFERENCES

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., & Rus, D. (2017, January). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. **Proceedings of the National Academy of Sciences**, 114 (3), 462-467. doi: 10.1073/pnas.1611675114

Alonso-Mora, J., Wallar, A., & Rus, D. (2017, September). Predictive routing for autonomous mobility-on-demand systems with ride-sharing. In **2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)** (p. 3583-3590). doi: 10.1109/IROS.2017.8206203

Arulampalam, M. S., Maskell, S., Gordon, N., & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. **IEEE Transactions on signal processing**, 50 (2), 174–188.

Barth, M., Todd, M., & Xue, L. (2004). User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. **Transportation Research Record**, 1887, 137-144.

Böcker, L., Dijst, M., & Prillwitz, J. (2013). Impact of everyday weather on individual daily travel behaviours in perspective: a literature review. **Transport reviews**, 33 (1), 71–91.

Diao, M., Kong, H., & Zhao, J. (2021). Impacts of transportation network companies on urban mobility. **Nature Sustainability**, 1–7.

Fielbaum, A., Jara-Diaz, S., & Gschwender, A. (2020). Beyond the Mohring effect: Scale economies induced by transit lines structures design. **Economics of Transportation**, 22, 100163.

Henao, A., & Marshall, W. E. (2019). The impact of ride-hailing on vehicle miles traveled. **Transportation**, 46 (6), 2173–2194.

Huang, X., & Peng, H. (2018). Efficient mobility-on-demand system with ride-sharing. In **2018 21st international conference on intelligent transportation systems (itsc)** (pp. 3633–3638).

Li, W., Pu, Z., Li, Y., & Tu, M. (2021). How does ridesplitting reduce emissions from ridesourcing? a spatiotemporal analysis in chengdu, china. **Transportation Research Part D: Transport and Environment**, 95, 102885.

Lioris, J., Cohen, G., Seidowsky, R., & Salem, H. H. (2016). Dynamic evolution and optimisation of an urban collective taxis systems by discrete-event simulation. In **Its world congress 2016, melbourne, australia.**

Liu, S., Jiang, H., & Chen, Z. (2021). Quantifying the impact of weather on ride-hailing ridership: Evidence from haikou, china. **Travel Behaviour and Society**, 24, 257–269.

Tang, B.-J., Li, X.-Y., Yu, B., & Wei, Y.-M. (2019). How app-based ride-hailing services influence travel behavior: An empirical study from China. **International Journal of Sustainable Transportation**, 1–15.

Tirachini, A., Chaniotakis, E., Abouelela, M., & Antoniou, C. (2019). The sustainability of shared mobility: Can a platform for shared rides reduce motorized traffic in cities? **Transportation Research Part C: Emerging Technologies**, 117, 102707.

Tirachini, A., & del Río, M. (2019). Ride-hailing in Santiago de Chile: Users' characterisation and effects on travel behaviour. **Transport Policy**, 82, 46–57.

Tirachini, A., & Gomez-Lobo, A. (2020). Does ride-hailing increase or decrease vehicle kilometers traveled (VKT)? a simulation approach for Santiago de Chile. **International Journal of Sustainable Transportation**, 14 (3), 187–204.

van Engelen, M., Cats, O., Post, H., & Aardal, K. (2018). Enhancing flexible transport services with demand-anticipatory insertion heuristics. **Transportation Research Part E: Logistics and Transportation Review**, 110, 110–121.

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

Wallar, A., Van Der Zee, M., Alonso-Mora, J., & Rus, D. (2018). Vehicle rebalancing for mobility-on-demand systems with ride-sharing. In **2018 ieee/rsj international conference on intelligent robots and systems (iros)** (pp. 4539–4546).

Wang, J., Cheng, P., Zheng, L., Feng, C., Chen, L., Lin, X., & Wang, Z. (2020). Demand-aware route planning for shared mobility services. **Proceedings of the VLDB Endowment**, 13 (7), 979–991.