Selecting an optimal set of shared ridepooling stops

Francisco Vilches[1], Cristan Cortes[1], Andres Fielbaum[2]

[1]Department of Civil Engineering, Universidad de Chile
[2]School of Civil Engineering, University of Sydney

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

Keywords: Ridepooling, On-demand, Shared mobility, Virtual stops, PUDO points.

1 INTRODUCTION

Ridepooling systems have gathered attention from researchers and industry in recent years. They could combine the attractiveness of ridehailing, being flexible on routes and on-demand, and the efficiency of public transport, by having many passengers sharing the vehicle simultaneously.

Previous studies indicate that door-to-door services introduce inefficiencies in ridepooling (Gurumurthy & Kockelman, 2022, Fielbaum *et al.*, 2021, Fielbaum, 2022). The main reason is that driving door-to-door often implies entering slow streets or following long detours, which can be avoided if the users could walk even short distances. On the other hand, even the door-to-door version of ridepooling can be unreliable to users. As the shape of a route depends on circumstantial co-travellers, it is hard for users to accurately anticipate the conditions of their trip because they can get updated while a trip is being executed (Zhang *et al.*, 2024, Fielbaum & Alonso-Mora, 2020). When the system requires the users to walk to/from different locations each time, another layer of unreliability is introduced, potentially hindering ridepooling's broader adoption. For instance, an experiment reported by Martin *et al.* (2021) in collaboration with Lyft, showed that the adoption of ridepooling might be tripled if the pickup and dropoff points were informed before the user decides whether to accept a trip or not, as opposed to accepting a cloud of potential points.

Therefore, a natural question emerges: Is it possible to abandon the door-to-door scheme to increase the system's efficiency without making it more unpredictable? Specifically, in this paper, we study the problem of selecting a subset of locations where the pickup and dropoff (PUDO) can occur: the Shared Ridepooling Stops SRS. Crucially, this subset is decided beforehand, and then every user chooses their PUDO points from there.

In the traditional public transport nomenclature, this can be seen as a *tactical* decision, i.e., that could be changed on a monthly or yearly basis, but not every day. Moreover, we aim to develop methods based solely on the street network, independent of the characteristics of the ridepooling system, which may vary significantly over time (peak vs off-peak hours, week vs weekends, sunny vs rainy days, and so on); this is similar to the location of bus stops in a public transport network, whose position is rarely modified even if the routes or the demand changes.

2 METHODS

2.1 Problem statement

We consider an on-demand ridepooling system that is not door-to-door. Instead, the system offers a set of potential places where the vehicles can stop to pick up or drop off a passenger, called the

Shared Ridepooling Stops (SRS). From the users' perspective, when they select their trip, they must select one SRS for the pickup and another for the dropoff. Thus, the methods to assign and route door-to-door ridepooling (e.g. Alonso-Mora *et al.* (2017), Ramezani & Valadkhani (2023)) can be applied directly. The focus of this paper is how to select the set of SRS.

Preliminaries, decision variables, and constraints: We consider a strongly directed graph $G = (N, E)$. Each arc $e \in E$ is characterized by a distance d_e and a vehicle-time t_e. We assume that in every leg passengers cannot walk a distance longer than D_M. We must select a subset $P \subseteq N$, where nodes $u \in P$ are the SRS. The subset P must fulfil that $\forall u_1 \in N, \exists u_2 \in P$, such that $d(u_1, u_2) \leq D_M$, i.e., every node is at walking distance from P. We will consider the service rate of the system as our main KPI, i.e., the percentage of requests that get served (given hard waiting and in-vehicle time windows) and are not rejected. In the full version of the paper, other KPIs such as vehicles-kilometres-travelled and total travelling times are reported.

We face a challenging scheme. On the one hand, we aim for a method that takes the graph as its only input; on the other hand, the evaluation of P will depend on the fleet, the demand, the users' decisions about how to select their preferred SRS, and the routing-and-assignment mechanism. To deal with this puzzle, we will proceed as follows: In section 2.2, we propose methods that receive the graph G as input, and the set of SRS as output. Then, we test all these methods considering different fleets, demand patterns, and users' choices, utilizing the state-of-the-art routing and assignment method by Alonso-Mora *et al.* (2017). In this extended abstract, we only show results for one scenario.

2.2 Solution method

Our method to build the set of SRS is based on determining a *hierarchy for the nodes*. That is, we will assign each node u a distinct ranking $r_u \in \{1, ..., |N|\}$. We consider three ranking methods: (i) The **contraction hierarchy**, as defined by (Geisberger *et al.*, 2008), where intuitively the hierarchy of a node is greater if it belongs to more shortest paths; (ii) The **convex hierarchy**, where nodes are ranked according to the value of $VA(u) = \alpha \cdot D(u) + (1 - \alpha) \cdot V(u)$, with $D(u)$ its degree (number of neighbours), and $V(u)$ the average speed of the surrounding arcs; and (iii) a **random hierarchy** to be used as a benchmark.

Once the hierarchy has been decided, we build P as follows. We consider the nodes in order according to their ranking. For each of them, if it has not been *covered* yet, we add it to the list. A node is covered if another node, closer than D_M, has been previously added.

3 NUMERICAL SIMULATIONS

3.1 The scenario

We run simulations over a network representing the commune of Providencia in Santiago, Chile, consisting of 2,035 nodes. We generate a synthetic set of 1,000 requests, and use $D_M = 0.45$ kms, to be served by 100 vehicles of capacity 3. We assume that each time a vehicle needs to stop, it loses 60 seconds in total. Figure 1 shows the full network and the resulting 94 SRS when applying the contraction hierarchy.

3.2 The assignment method

The assignment between vehicles and users, together with the vehicles' routes, are decided following the well-known method by Alonso-Mora *et al.* (2017). This method requires receiving the pickup and drop-off point of each request[1], but they are not uniquely determined, as every

[1]It also requires a number of parameters, for which we use similar values as the original paper. The exact parameters are described in the full version of this paper.

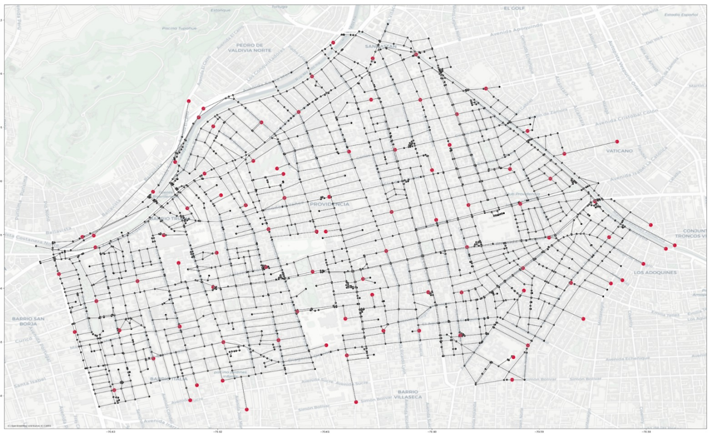

Figure 1 – *The full Providencia network (black), and the resulting SRS (red).*

user might have several SRS at walking distance. Which SRS to consider is in fact a behavioural question. We consider two scenarios for this: (i) The *min-walk* scenario, where every user selects the SRS that minimises the distance between origin and pickup (and the same for the destination), and (ii) the *min-time* scenario, where every user r select u_1, u_2 SRS to minimise $t_a(o_r, u_1) + t_v(u_1, u_2) + t_a(u_2, d_r)$, where t_a represents the walking times, t_v the in-vehicle time, and o_r and d_r are the exact origin and destination.

3.3 Results

Every simulation takes a few minutes. Results are summarised in Figure 2, where we show the service rate for the different hierarchies, varying α in the case of the convex hierarchy. The results of the convex and contraction hierarchies (CHR) are similar and much better than the door-to-door alternative (DtD). This is a notable result, as to avoid detours, we do not need to optimise the PUDO points dynamically - instead, an offline decision, where users can select which SRS to use, suffices. The result is robust with respect to how do users make their selection (min-walk or min-time), although results are obviously better when they minimise their in-vehicle time, as vehicles are occupied for shorter periods; the longer the D_M, the greater the improvement.

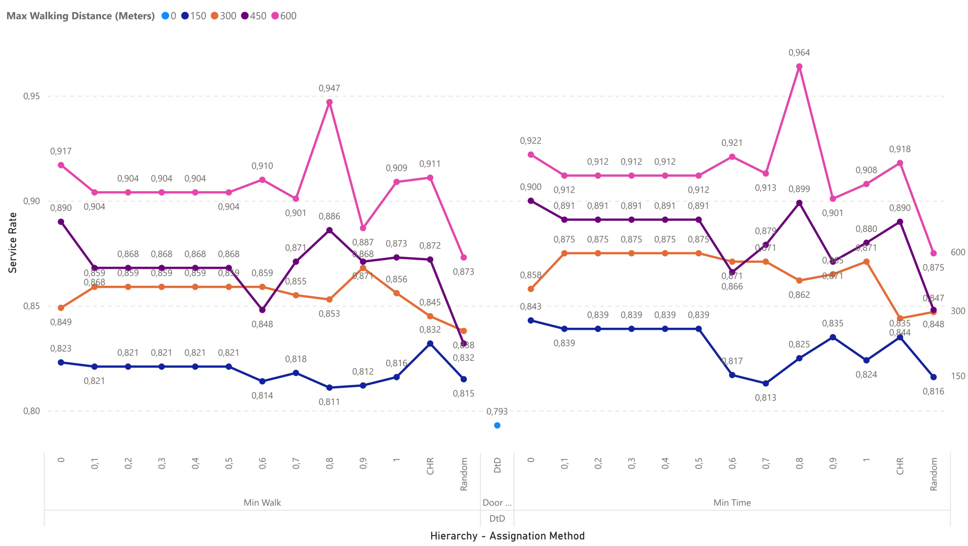

Figure 2 – *Results of the simulations.*

Using the SRS with any reasonable hierarchy is always better than the random hierarchy, but remarkably, the latter also outperforms the door-to-door service. This suggests that using SRS benefits the system in two different ways: first, by keeping the cars in the most useful arcs (achieved with the good hierarchies), and second, by the very fact of concentrating the users in a few places to stop. The reason for this is that vehicles stop less often. We measure this by computing the number of *coincidences*, i.e., how many times two users were served in the same stop. Results are summarised in Table 1, with $D_M = 0.45$ km. As evident, the random hierarchy achieves a much greater number of coincidences than door to door, which ultimately increases the system's efficiency and reduces the rejection rate.

Table 1 – *Number of coincidences depending on the hierarchy.*

Hierarchy	**Number of coincidences - Min walk**	**Number of coincidences - Min time**
CHR	166	167
Random	165	159
Door to door	18	18

4 CONCLUSION

We have proposed methods to select SRS that are decided offline, from which ridepooling users can choose their preferred PUDO points. Our results show a great improvement over a door-to-door alternative, showing that it is possible to unleash the potential of walking in ridepooling, without making the system less predictable for the users. A close look at the results reveals that walking benefits the system not only by avoiding detours but also by occasionally having users served at the same stop. The methods and the high-level conclusions bear the relevant potential to make on-demand ridepooling and on-demand public transport more widespread.

References

Alonso-Mora, Javier, Samaranayake, Samitha, Wallar, Alex, Frazzoli, Emilio, & Rus, Daniela. 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, **114**(3), 462–467.

Fielbaum, Andrés. 2022. Optimizing a vehicle's route in an on-demand ridesharing system in which users might walk. *Journal of Intelligent Transportation Systems*, **26**(4), 432–447.

Fielbaum, Andrés, & Alonso-Mora, Javier. 2020. Unreliability in ridesharing systems: Measuring changes in users' times due to new requests. *Transportation Research Part C: Emerging Technologies*, **121**, 102831.

Fielbaum, Andres, Bai, Xiaoshan, & Alonso-Mora, Javier. 2021. On-demand ridesharing with optimized pick-up and drop-off walking locations. *Transportation Research Part C: Emerging Technologies*, **126**, 103061.

Geisberger, Robert, Sanders, Peter, Schultes, Dominik, & Delling, Daniel. 2008. Contraction hierarchies: Faster and simpler hierarchical routing in road networks. *Pages 319–333 of: Experimental Algorithms: 7th International Workshop, WEA 2008 Provincetown, MA, USA, May 30-June 1, 2008 Proceedings 7*. Springer.

Gurumurthy, Krishna Murthy, & Kockelman, Kara M. 2022. Dynamic ride-sharing impacts of greater trip demand and aggregation at stops in shared autonomous vehicle systems. *Transportation Research Part A: Policy and Practice*, **160**, 114–125.

Martin, Sebastien, Taylor, Sean J, & Yan, Julia. 2021. Trading flexibility for adoption: Dynamic versus static walking in ridesharing. *Available at SSRN 3984476*.

Ramezani, Mohsen, & Valadkhani, Amir Hosein. 2023. Dynamic ride-sourcing systems for city-scale networks-Part I: Matching design and model formulation and validation. *Transportation Research Part C: Emerging Technologies*, **152**, 104158.

Zhang, Xin, Zhong, Shiquan, Jia, Ning, Ling, Shuai, Yao, Wang, & Ma, Shoufeng. 2024. A barrier to the promotion of app-based ridesplitting: Travelers' ambiguity aversion in mode choice. *Transportation Research Part A: Policy and Practice*, **181**, 103971.