Reformulation of Drivers' Fixed Path Constraints in Ridesharing Problems

Vincent Armant and Kenneth N. Brown

Insight Centre for Data Analytics
Department of Computer Science, University College Cork, Ireland
{firstname.lastname}@insight-centre.org

Abstract. Ridesharing schemes, which match passengers to private car journeys with spare seats, are becoming popular for reducing costs, emissions and congestion. Current systems offer matches to participants based on trip details and on preferences, but do not optimise for global system objectives like maximising the number of served users. They are similarly not applicable to real-time matching for high demand ridesharing. In this paper, to limit the inconvenience of drivers who share their journeys, we respect in the solution the specific route and travel duration intended by each driver. The goal is to assign riders to drivers and maximise the number of participants who receive a match. We first present a basic model. We then consider two reformulations. The first reformulation is based on computing sets of time-consistent riders for each driver. The second is based on observation about driver speed along the route. We make two contributions. We first propose a first reformulation of the ridesharing problem based on the drivers' time-consistent sets of feasible rides. We then introduce an other reformulation based on a restriction of the drivers' paths constraints that significantly speeds up the solving time. We demonstrate the scalability of the approach on four real-world ridesharing clusters. We show empirically that our first reformulation performs better on instances with few shifters, i.e., drivers that are willing to become passengers. Our second reformulation is one order of magnitude faster than the initial problem description, and can return within one minute an optimal ridesharing plan for thousands of users.

1 Introduction

Ridesharing schemes are becoming popular in large city regions, where prospective passengers are matched to spare seats in private car journeys. The overall problem is a combination of a routing and scheduling optimisation with a matching problem. Successful schemes reduce travel costs for participants and take cars off the roads, thus reducing emissions and congestion. In flexible ridesharing schemes, some drivers may be willing to change role and accept a ride from other drivers, Agatz et al [1]. Allowing this flexibility enhances the sustainability benefit of the schemes, since it directly reduces the number of cars being used. Armant et al. [4] showed that increasing the number of flexible drivers in some real-world schemes could correct a critical imbalance between drivers and riders while increasing the number of participants who receive a match. Most deployed applications focus on the online problem of proposing attractive matches to individual participants, concentrating on user preferences, rather than addressing the system-wide optimisation problem. Fast and scalable solutions for maximising the number of served participants remains an open challenge that have not been tackle. As more cities introduce penalties to discourage drivers from using their cars during high pollution periods, or offer faster routes for high occupancy vehicles, the need for managing high-demand schemes will grow.

In this paper, we address the problem of returning in real time optimal plans in high-demand ridesharing context. We solve the problem of maximising the number of served users to ensure the long-term viability of the scheme. To encourage drivers to participate to the ridesharing scheme, we consider fixed driver path constraints. Drivers pick-up and drop-off passengers on their usual path and the travel duration is fixed (in the plan). We propose two reformulations for flexible ridesharing schemes, reformulating the core resource allocation constraints. The first reformulation is based on an aggregation of each driver's feasible rides into time-consistent sets of rides. We then introduce an other reformulation that simplified the driver's path constraint to speed up the solving of the reformulated problems. We evaluate the approach empirically, based on datasets of trip adverts and requests from real ridesharing schemes. We show that the new formulations are up to one order of magnitude faster than the initial ridesharing modelisation, finding system-wide matches for thousands of participants in under one minute.

2 Related work

The dial-a-ride problem has long been studied in the OR community Cordeau and Laporte [7]. Dial-a-ride typically assumes a single vehicle, picking up and dropping off riders at specified locations within time windows, although multiple vehicle problems have also been studied Berbeglia et al [6] and Attanasio et al [5]. The dial-a-ride drivers have no strong journey requirements of their own. For ride-sharing schemes, both the drivers and the riders have their own objectives, Furuhata et al [8]. Specific schemes vary as to whether the drivers move to the riders' locations or the riders move to and from the drivers' routes, and whether or not drivers take single or multiple riders on a trip. One extension includes participants known as *shifters*, who may either drive or ride as a rider, Agatz et al. [1]. Armant and Brown [3] also include shifters, but assume that each pure rider who is not served in the matching has a probability of driving on his own, included as a penalty in the objective function. Armant et al. [4] assess the performance of a deployed ride-sharing scheme and evaluate the potential of persuading drivers to become passengers. They show that increasing driver flexibility could have a significant impact, reducing the number of cars on the road while increasing the number of matched participants. Computing an optimal matching is hard, Agatz et al. [2], and the complexity increases as the number of shifters increases. Kamar and Horvitz [9] model the problem as one of collaborative planning, where agents must balance competing goals. Yousaf et al. [13] model the problem as multi source-destination path planning, with a wide range of competing objectives including privacy and incentives. Schilde et al. [11] and Manna and Prestwich [10] consider stochastic problems, in which trip requests arrive during the execution of the solution, using scenario-based methods to minimize expected delays or unserved requests. Simonin and O'Sullivan [12] focus on the matching problem, assuming an input graph of all feasible pairings, and establish the complexity of a number of variations, showing that in some cases polynomial time solutions are possible.

3 Notation

Drivers and riders will post adverts for their trips, specifying locations, earliest departures and latest arrivals. From these adverts, we compute time-windows for each possible ride share, and define time-consistent trips. Our goal is to formalize and solve the problem of high demand ridesharing, arising in the context of exceptional events by finding a match for the greatest number of users. We introduce the following notation. The set of ridesharing users U are partitioned into three sets, D, the set of drivers offering seats in their cars, R, the set of riders asking for a ride, and S, the set of shifters, i.e., drivers willing to change role and becoming passengers. $L = \{l^1, \ldots, l^n\}$ denotes the set of road node locations identified by their GPS coordinates. The path $\pi^u_{l^i,l^j} = (l^i, \ldots, l^j)$ is the ordered list of locations visited by a user $u \in U$ from location l^i to l^j. When there is no ambiguity, we denote by π^u the full path of u from the starting location $l^\star_u$ to the destination $l^\oslash_u$. The location $l^\uparrow_{d,r} \in \pi^d$ (resp. $l^\downarrow_{d,r} \in \pi^d$) denotes the pick-up (resp. drop-off) point of r in the path of d. For simplicity, when a driver d and a rider r already appear in the notation we denote the pick-up (resp. drop-off) by $l^\uparrow$ (resp. $l^\downarrow$). The time $t(\pi^u_{l^i,l^j})$ denotes the time for a user u to traverse π from l^j to l^j. The time $t(\pi^u_{l^i,l^j})$ is negative if l^i precedes l^j in π^u. In this case, it represents the reverse travel time from l^i to l^j. The time $t^{early}_{l^i,u}$, (resp. $t^{late}_{l^i,u}$) represents the earliest time (resp. latest time) the user u can be at the location l^i. From the earliest start time $t^{early}_{l^\star_d}$ of d, and the earliest start time $t^{early}_{l^\star_r}$ of r, we define the earliest pick-up time $t^{early}_{l^\uparrow,d,r}$ at the pick-up location $l^\uparrow$ as the earliest arrival of d and r at the pick-up point:

$$t^{early}_{l^\uparrow,d,r} = max(t^{early}_{l^\star_d} + t(\pi^r_{l^\star_d,l^\uparrow}), t^{early}_{l^\star_r} + t(\pi^r_{l^\star_r,l^\uparrow})).$$

Similarly, from the latest arrival time $t^{late}_{l^\oslash_d}$ of d, and the latest arrival time $t^{late}_{l^\oslash_r}$ of r, we define the latest pick-up time of $t^{late}_{l^\uparrow,d,r}$ as the latest time departure of d and r from the pick-up point:

$$t^{late}_{l^\uparrow,d,r} = min(t^{late}_{l^\oslash_d} - t(\pi^d_{l^\uparrow,l^\oslash_d}), t^{late}_{l^\oslash_r} - t(\pi^r_{l^\downarrow,l^\oslash_r}) - t(\pi^d_{l^\uparrow,l^\downarrow_r})).$$

For the sake of clarity, we illustrate the notations describing a feasible ridesharing trip between a driver d and a rider r in the Figure 1. The black plain line depicts the path of d, π^d, from its departure $l^\star_d$ to its destination $l^\oslash_d$. The first dashed grey line depicts the pick-up path of r from its departure $l^\star_r$ to the pick-up point $l^\uparrow$. The second dashed grey line depicts the drop-off path of r from its drop-off point $l^\downarrow$ to its destination $l^\oslash_r$. We also depict the delivery time window $[t^{early}_{l^i}, t^{late}_{l^i}]$ at the location l^i in the path π^d.

Given these notations we can now define time-consistent ridesharing trips between drivers and riders.

Definition 1 (Time-consistent ridesharing trip) *A ridesharing trip between a rider $r \in R \cup S$ and a driver $d \in D \cup S$, is* time-consistent *if:*

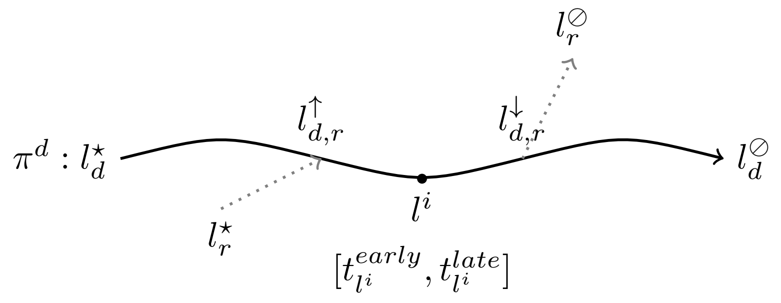


Fig. 1: Feasible ridesharing trip betwen a driver d and a rider r

1. $\pi^r_{l^\uparrow,l^\downarrow}$ *is a subpath of* π^d*,*

2. At any point l^i *in the path* π^d*, the consistent delivery time window* $[t^{early}_{l^i,d,r}, t^{late}_{l^i,d,r}]$ *is s.t. :*

(a) $t^{early}_{l^i,d,r} = t^{early}_{l^\uparrow,d,r} + t(\pi^d_{l^\uparrow,l^i})$

(b) $t^{late}_{l^i,d,r} = t^{late}_{l^\uparrow,d,r} + t(\pi^d_{l^\uparrow,l^i})$

(c) $t^{early}_{l^i,d,r} \leq t^{late}_{l^i,d,r}$

We denote by $G = (D \cup S, R \cup S, E)$ the graph of time-consistent ridesharing journeys between drivers and riders s.t. $E \subseteq (D \cup S) \times (R \cup S)$.

4 Maximizing the number of served users

The overall aim of ride-sharing is to improve mobility by providing rides to users without cars, and also to reduce the total driven kilometers on our roads. This depends on the long-term sustainability of the schemes. A ride-sharing scheme is only viable if there are sufficient participants, and from data on deployed schemes, users stop participating if they do not regularly receive suitable ride matches. Therefore, our objective in this research is to maximize the number of served participants. Although on any instance, this may not produce the minimal total driven distance over the scheme, it does increase the probability of repeat custom, and thus lower total driven distance over a longer timeframe. Thus, in the models that follow, we do not account for the driving distance of unmatched drivers nor for the alternative travel costs of unmatched passengers, and we do not gain any reward in the objective function for these unmatched participants. As an aside, we note that the presence of shifters does help to reduce the total driven distance, since these participants would drive themselves if unmatched, but by re-assigning some of them as passengers, we remove their individual driving distance and increase the number of matches.

Informally, the aim is to determine which passengers should ride with which drivers, and so we consider a solution do be a set of such assignments. To ensure the route and time window constraints are satisfied, and the number of assigned users maximised, we represent the problem as a MIP with auxiliary variables and constraints. In the initial model, the $\{0,1\}$ decision variables x_d, x_r, x_s represent the assigned driver, rider and shifter. The $\{0,1\}$ decision variable $y_{d,r}$ represents a ride-match assignment of the rider r to driver d's car s.t. $(d,r) \in E$. The integer variable $o_{d,l^i} \in [0, q_d]$ denotes the car occupancy of driver d at location l^i, bound by the car capacity q_d. The continuous auxiliary variable v_{d,l^i} represents the time the driver $d \in D \cup S$ leaves the location $l^i \in L$. For the driver path π^d, $pred_\pi(l^i)$ denotes the predecessor of l^i in the path π^d. The set of riders $pick_{\pi^d}(l^i)$ (resp. $drop_{\pi^d}(l^i)$) denotes the feasible ridesharing trip that can be picked up (resp. dropped off) at $l^i \in \pi^d$.

Our objective is to maximize the number served users (1). Each rider or shifter assigned to be a passenger is served if and only if (iff) they are assigned to exactly one driver (2). Each driver or shifter assigned to drive is served iff at least one rider has been assigned to them (3). A shifter can be served as driver or served as a passenger but not both (4). When a driver leaves a location, the car occupancy is equal to the difference between the picked up and dropped off passengers plus the car occupancy of the previously visited location (5). (6) states that when d is assigned the passenger r, the visiting time anywhere in the path satisfies the conditions of a time-consistent ridesharing trip (cf. Definition 1).
Our objective

$$maximize(\sum_{u \in U} x_u) \tag{1}$$

subject to:

$$x_r = \sum_{(d,r)\in E} y_{d,r}, \ \forall r \in R \cup S \tag{2}$$

$$(\sum_{(d,r)\in E} y_{d,r} \geq 1) \Leftrightarrow (x_d = 1), \ \forall d \in D \cup S \tag{3}$$

$$(\sum_{(s,r)\in E} y_{s,r}) \geq 1) \Rightarrow (\sum_{(d,s)\in E} y_{d,s} = 0), \ \forall s \in S \tag{4}$$

$$\begin{aligned} o_{d,l^j} = o_{d,l^i} + \sum_{r \in pick_{\pi^d}(l^i)} y_{d,r} - \sum_{r' \in drop_{\pi^d}(l^i)} y_{d,r'}, \\ \forall d \in D \cup S, \forall l^i = pred_{\pi^d}(l^j) \end{aligned} \tag{5}$$

$$y_{d,r} \Rightarrow (t^{early}_{l^i,d,r} \leq v_{d,l^i} \leq t^{late}_{l^i,d,r}), \forall (d,r) \in E, \forall l^i \in \pi^d \tag{6}$$

(2), (3) and (4) are specific to the flexible ridesharing problem and our chosen objective.

The remaining constraints (5) and (6) can be viewed as a multi resource allocation problem where riders have to be allocated to drivers' cars. In this context, the car capacity constraints (5), along the drivers' fixed paths, encode the limited resource capacities specific to the ridesharing problem. The time feasibility constraints of the rides, (6) define the drivers' fixed time path constraints. If a passenger and a driver share a ride, these constraints force the journey of the driver's journey to be consistent with the passenger's time window while allowing the driver to cover its journey within the same travel time than usual. In the first reformulation, our main contribution is to rewrite both the multi resource allocation constraints consisting of the time feasibility constraints and the car occupancy constraints of the ridesharing problem into a simpler form with fewer variables and easier to solve.

5 First reformulation: Elimination of the spatio-temporal resource-allocation constraints and variables

Our aim is to simplify the car capacity constraints and the time dependency constraints for the delivery of riders along the drivers path. The main idea is to generalize the definition of time-consistent ridesharing trips between one driver and one rider to one driver and a set of riders. If we are able to represent all the possible sets of riders a driver can deliver, at the solving step, when we allocate passenger to drivers'car, we will not need to determine a consistent time for the pick-up and drop-off of passengers. For this purpose, we compute for each driver a list of time-consistent sets of riders for which exists a consistent time window within which the driver can ensure the delivery of the riders along the path. The drivers's time-consistent set of riders is the main concept at the basis of our reformulation of the time feasibility constraints along the driver path. To tackle car occupancy constraints, we then extend the notion of time-consistent sets of overlapping riders. Implicitly, the ridesharing trips of two riders overlap if they share a common leg along the driver path. The overall rewriting represents a spatio-temporal reformulation of the ridesharing problem into time-consistent set of overlapping rides. First, we define for the drivers' time-consistent sets of riders before introducing the time-consistent sets of overlapping riders. At the end of the section, we formalize the resource allocation constraints and the car capacity constraints .

5.1 Drivers' time-consistent sets of riders

We denote by $R_d = \{r|(d, r) \in E\}$ the set of riders sharing a one-to-one time-consistent ridesharing trip with driver d. We define a driver's time-consistent set of riders as follows :

Definition 2 (driver's time-consistent set of riders) *R^k_d is a time-consistent set of riders for the driver d if :*

1. *$R^k_d \subseteq R_d$, each rider r in R^k_d has a time-consistent ridesharing trip with d,*
2. *at any location l^i in the path π^d, exists a consistent delivery time window $[t^{early}_{l^i,d,R^k_d}, t^{late}_{l^i,d,R^k_d}]$ for all riders in R^k_d s.t. :*
   (a) $t^{early}_{l^i,d,R^k_d} = max(\{t^{early}_{l^i,d,r} | r \in R^k_d\})$
   (b) $t^{late}_{l^i,d,R^k_d} = min(\{t^{late}_{l^i,d,r} | r \in R^k_d\})$
   (c) $t^{early}_{l^i,d,R^k_d} \leq t^{late}_{l^i,d,R^k_d}$

We establish the equivalence between the time dependency constraints of the original problem and the drivers' time-consistent set of riders (Definition 2) by the following proposition.

Proposition 1 *R^k_d is a time-consistent set of riders for driver d iff at any location $l^i \in \pi^d$ exists a visiting time v_{d,l^i} for d s.t. $\forall r \in R^k_d$, $t^{early}_{l^i,d,r} \leq v_{d,l^i} \leq t^{late}_{l^i,d,r}$*

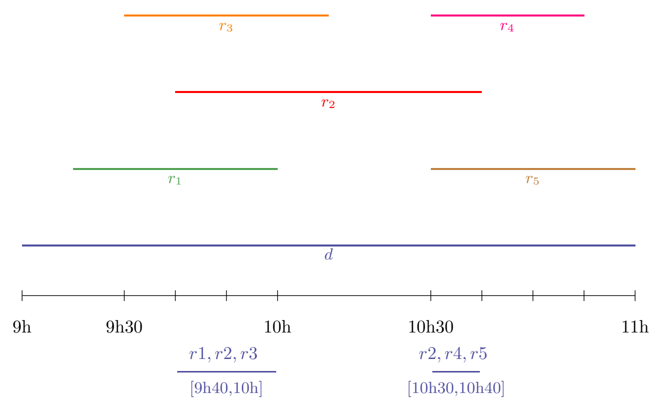


Fig. 2: Delivery time windows of a driver's time-consistent ridesharing trips

Proof.
⇒ In Definition 2, at any location in the driver's path, the delivery time window $[t^{early}_{l^i,d,R^k_d}, t^{late}_{l^i,d,R^k_d}]$ of a time-consistent set of riders R^k_d for a driver d is defined as the intersection of the delivery time window of each rider $r \in R^k_d$ (Definition 2a and 2b). Since the time window $[t^{early}_{l^i,d,R^k_d}, t^{late}_{l^i,d,R^k_d}]$ is consistent at any location in the path π^d (Definition 2c), there is a time within the time window $[t^{early}_{l^i,d,R^k_d}, t^{late}_{l^i,d,R^k_d}]$ for which the driver can visit the location while satisfying the delivery of all riders in R^k_d.
⇐ If at any location of the driver's path there is a consistent visiting time to deliver a set of passenger R^k_d, at any location in the driver's path there is a consistent time window for the delivery of the set of passengers. R^k_d is then a time-consistent set of riders. □

In Figure 2 we show two time-consistent sets of riders a driver d is able to deliver depending on the starting time at a specific location l^i in its path. If the driver starts from l^i between 9:30am and 10am he can pick up the riders $\{r_1, r_2, r_3\}$. The latter set is a time-consistent set of riders for the driver d. $\{r_3, r_2, r_5\}$ is another time-consitent for d. In this example there are no time-consistent sets containing both r_3 and r_4.

From a computational point of view, computing the time-consistent sets of riders for each driver and at each location in the path of the driver may be intractable. The next proposition shows that the size of the delivery time window of each time-consistent set of riders is constant along the driver path.

Proposition 2 *Given a time-consistent set of riders R^k_d of a driver d, at any location l_i in the path π^d, the size of the delivery time window for R^k_d is constant.*

Proof. By Definition 2, we know that for each time-consistent set of riders R^k_d of a driver d, exists at any location in the path π^d a consistent time window for the pick-up and the delivery of the riders in R^k_d . Let us consider two consecutive locations l^i, l^j in driver's path π^d and the travel time $t(\pi_{l^i,l^j})$ between the locations. We have the fact that the difference between the earliest start times of the delivery time window of the time-consistent set of riders for two consecutive locations l^i, l^j in driver's path π^d is equal to the travel time between the two locations : $t^{early}_{l^j,d,R^k_d} - t^{early}_{l^i,d,R^k_d} = t(\pi_{l^i,l^j})$. Similarly we have the fact that the difference between the latest start times of the delivery time window of the time-consistent set of riders for the two consecutive locations l^i, l^j is also equal to the travel time between the two locations : $t^{late}_{l^j,d,R^k_d} - t^{late}_{l^i,d,R^k_d} = t(\pi_{l^i,l^j})$. As a consequence, we have $t^{late}_{l^j,d,R^k_d} - t^{early}_{l^j,d,R^k_d} = t^{late}_{l^i,d,r} + t(\pi_{l^i,l^j}) - (t^{early}_{l^i,d,R^k_d} + t(\pi_{l^i,l^j})) = t^{late}_{l^i,d,r} - t^{early}_{l^i,d,R^k_d}$. Therefore the size of delivery time window remains the same from l^i to l^j. We deduce that the size of the delivery time window of a time-consistent set of riders $t^{late}_{l^i,d,R^k_d} - t^{early}_{l^i,d,R^k_d}$ at any point l^i in the path π^d remains the same. $\square$

As a consequence of this proposition, for each driver, we just have to represent the list of time-consistent sets of riders for one location in the path.

5.2 Driver's time-consistent set of overlapping riders

The time-consistent sets of riders represents the time-feasibility constraint along the driver path but not the car occupancy constraints of the driver. We tackle this issue in this section by the notion of time-consistent set of overlapping riders. We denote by $pos_{\pi^d}(l)$ the position of l in the path π_d.

Definition 3 (Driver's time-consistent set of overlapping riders) *O^k_d is a time-consistent set of overlapping riders for the driver d, if :*

1. *O^k_d is a time-consistent set of riders*
2. *$\{r, r'\} \subseteq O^k_d$ is s.t. :*
   (a) *the drop-off of r' is between the pick-up and the drop-off of r : $pos_{\pi^d}(l^{\uparrow}) \leq pos_{\pi^d}(l^{\downarrow}_{d,r'}) \leq pos_{\pi^d}(l^{\downarrow}_{d,r})$*
   (b) *or, the pick-up of r' is between the pick-up and the drop-off of r : $pos_{\pi^d}(l^{\uparrow}) \leq pos_{\pi^d}(l^{\uparrow}_{d,r'}) \leq pos_{\pi^d}(l^{\downarrow}_{d,r})$*

The example shown in Figure 3 extends the example of Figure 2 and shows the sets of overlapping riders along the driver's path within the time-consistent set of riders $\{r_1, r_2, r_3\}$. The maximal time-consistent overlapping sets of riders are $\{r_1, r_2\}$ and $\{r_2, r_3\}$. If the car capacity of d is greater than one, the overlapping riders will possibly share a common leg in the driver's path. Otherwise, if the driver offers only one spare seat, the only time-consistent sets of overlapping riders fitting to the car are the singletons $\{r_1\}$, $\{r_2\}$, $\{r_3\}$. The riders r_1 and r_3 are time consistent but do not overlap along the path π^d. Consequently driver d is able to deliver these two riders even if its car capacity is one.

Before presenting the reformulation, the following proposition establish the correspondence between the resource allocation constraints and the car capacity constraints of the initial ridesharing problem.

Proposition 3 *If a time-consistent set of overlapping riders O^k_d fits the car capacity of the driver d, the car occupancy constraints (5) and the time feasibility constraints (6) defined for d and O^k_d are satisfiable.*

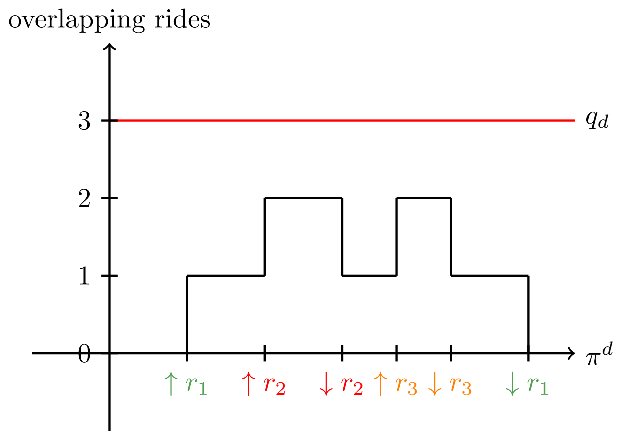


Fig. 3: Overlaping rides of the time-consistent set $\{r_1, r_2, r_3\}$

Intuitively, we have shown the correspondance between the notion of time-consistent set of riders and the resource allocation constraints (6) in Proposition 1. By Definition 3, a time-consistent set of overlapping riders represents a set of riders sharing a common leg in the driver's path. If the size of the set of the overlapping riders O_d^k does not exceed the driver's car capacity, it satisfies constraint (5).

5.3 Encoding the new ridesharing resources allocation constraints

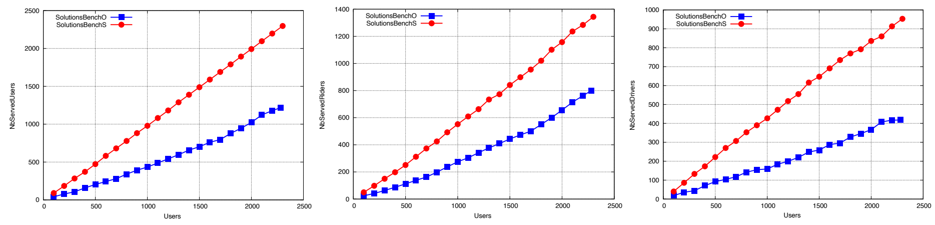
For each driver, computing all its time-consistent sets of riders even for one location, may be inefficient. To avoid this pitfall, we consider maximal time-consistent sets of riders.

Definition 4 (maximal time-consistent set of rides) *The set M_d^i is a maximal time-consistent set of riders for d if:*

1. $M_d^i \subseteq R_d$
2. $\forall R_d^k \subseteq R_d$ *if* $M_d^i \subseteq R_d^k$ *then* $M_d^i = R_d^k$

In Figure 2 the highlighted time-consistent sets $\{r_1, r_2, r_3\}$ and $\{r_4, r_2, r_5\}$ are maximal.

For each driver, the maximal time-consistent sets of riders represent all feasible subsets of assignments of riders to the driver's car. We introduce here the constraints corresponding to the time-feasible constraints (6). A maximal time-consistent set of riders represents a possible set of passengers the driver is able to deliver regardless the car capacity. We handle the car capacity constraints in the next paragraph. In our initial problem description, $y_{d,r}$ represents the assignation of a ridesharing trip between d and r. In the reformulation, for each driver, we indirectly encode the list of maximal time-consistent sets of riders by forbidding all non time-consistent riders. For this purpose, constraints (7) encode a conflict between each pair of non time-consistent riders belonging to two distinct maximal sets. Implicitly these conflicts force the ridesharing



a) Number of served users b) Number of users served as rider c) Number of users served as driver

Fig. 4: Optimal solutions found for benchO (having no shifters) and benchS (having shifters) in region 3

solution for each driver to a belong a subset of one of its maximal time-consistent set of riders.

$$\begin{aligned}&(y_{d,r} = 1) \Rightarrow (y_{d,r'} = 0),\\ &\forall d \in D \cup S, \forall r \in M_d^i \setminus M_d^j, \forall r' \in M_d^j \setminus M_d^i\end{aligned} \tag{7}$$

The constraints (7) add no new decision variables. The path consistency constraints of the ridesharing trip are indirectly described by the unauthorized pair of rides.

Similarly, for efficiency, we do not compute for each driver the list of all time-consistent sets of overlapping riders.We only compute the maximal time-consistent sets of overlapping riders: MO_d^k. To reformulate constraint (5) we restrict the number of assigned ridesharing trips represented by MO_d^k by the car capacity q_d.

$$\begin{aligned}&0 \leq \underset{r \in MO_d^k}{\Sigma y_{d,r}} \leq q_d\\ &\forall d \in D, \forall M_d^i \in M_d, \forall MO_d^k \in M_d^i\end{aligned} \tag{8}$$

In our reformulation of the original ridesharing model, we replace the time-feasibility constraints (6) and the car occupancy constraints (5) by constraints (7) and constraints (8).

6 Second Reformulation: Simplifying the path constraints

In the previous reformulation we eliminate the drivers' car occupancy variables and the drivers' path time variables. Our next contribution is a second reformulation of the drivers' fixed time path constraints into a simpler form. First, we introduce the following proposition:

Proposition 4 *If a driver shares a consistent time window with a set of passengers, it is able to deliver theses passenger without wasting time on the path.*

In the previous section, Proposition 2 shows that a driver can deliver a set of passengers when they share a consistent time windows at any location of the driver's path. We have also seen in Proposition 1 that the size of the time windows remain constant along the driver's. Consequently, if a driver starts its journey within this consistent time window and does not waste time between two locations, he will be able to deliver the set of passengers. Using the previous proposition, we can now reformulate the initial drivers path consistency constraints (6) by the two following constraints.

$$y_{d,r} \Rightarrow (t^{early}_{l^{\uparrow},d,r} \leq v_{d,l^{\uparrow}} \leq t^{late}_{l^{\uparrow},d,r}), \forall (d,r) \in E \tag{9}$$

$$v_{d,l^j} = v_{d,l^i} + t(\pi_{l^i,l^j}) \qquad \forall d \in D \cup S, \forall l^i = pred_{\pi^d}(l^j) \tag{10}$$

Constraints (9) force a consistent time window for drivers to visit the pick-up location of the passengers. Note that, here, the time window consistency is only enforced at the pick-up location. Constraints (10) force the time of the driver's journey to be equal to $t(\pi^d)$ and do not allow any waiting time between two locations. Together, the last constraints enforce the existence a no waiting path for a driver to deliver its passengers. There is no need to enforce a consistent time window for drop-off, it is implies by the conjunction of the two constraints.

7 Experimental protocol and benchmarks

To evaluate the time performance and the scalability of the different approaches, we measure the solving time for returning an optimal ridesharing plan while increasing the number of participants in the ridesharing problem. Each point in the plots represents the mean of 10 runs executed on a machine having 12 cores of 2.50GHz and 64GB of RAM, using the CPLEX solver. The data sets represent real ridesharing trips provided by our industrial partner and advertised for a specific day of the week. The advertised ridesharing trips have been collected for a period of two years for region 1 and region 2, nine months for region 3 and region 4. To check the scalability of our approach, we gradually increase the number of participants by selecting advertised ridesharing trips in chronological order from the data sets. Region 3 and region 4 represent big agglomerations where ridesharing is commonly used by commuters and contain road infrastructures such as High Occupancy Line to push drivers to share their journeys. Region 1 and region 2 are small areas where ridesharing is not consider yet as a daily alternative to public transport. For each dataset, we infer a flexible time window for each user based on the history of successful ridesharing trips from the same data sets. The observed features correspond to the maximal time changes drivers and riders are willing to accept to share the trip, or the maximal distance a rider will accept for a ride. From the inferred users' time windows, we build a graph of time-consistent ridesharing trips between drivers and riders. We use the graph of time-consistent collected ridesharing trips to build two benchmarks. The first benchmark benchO consists of the original ridesharing requests collected from the four different regions. In Table 1, we describe the parameters of an instance in benchO for each region. The notations Users*, Drivers*, ..., denote users having at least one match in the feasible match graph. The instances of benchO are characterized by a ratio $Riders(R{*}{+}S{*})/Drivers(D{*}/S{*}) < 1$ or a small number of shifters. As a consequence, the imbalance between riders and drivers may act as disincentive for participants that may not find a suitable match to share their rides.

To readjust the imbalance between riders and drivers, in the second benchmark, benchS, each trip initially advertised for driver role is considered as a potential shifter,

	Users	Users*	Drivers*	Riders*	Shifters*	(R*+S*)/	Edges	Matches
			D*	R*	S*	(D*+S*)		per user*
region 1	1000	955	564	326	65	0.62	7231	757
region 2	1000	950	524	378	48	0.70	10306	10.85
region 3	1000	973	698	275	0	0.40	11695	12.02
region 4	1000	808	503	306	1	0.60	2795	3.46

Table 1: Instance parameters in benchmark: benchO (original users' requests)

i.e., a driver that may accept to become a passenger when it can match as a passenger of an other driver. Considering shifters instead of drivers introduce two potential sustainable benefits. Users have more opportunities to find a match (cf. Matches per user in Table: 2). In addition, since original drivers will be proposed to share their journeys, cars will potentially be removed from the roads. As a consequence, the combinatorial complexity associated to shifter role make these ridesharing problem instances harder to solve.

	Users	Users*	Drivers*	Riders*	Shifters*	(R*+S*)/	Edges	Matches
			D*	R*	S*	(D*+S*)		per user*
region 1	1000	967	158	338	471	1.29	12464	12.89
region 2	1000	954	139	387	428	1.44	16957	17.77
region 3	1000	980	48	277	655	1.33	39172	39.97
region 4	1000	856	161	695	329	1.32	5968	6.97

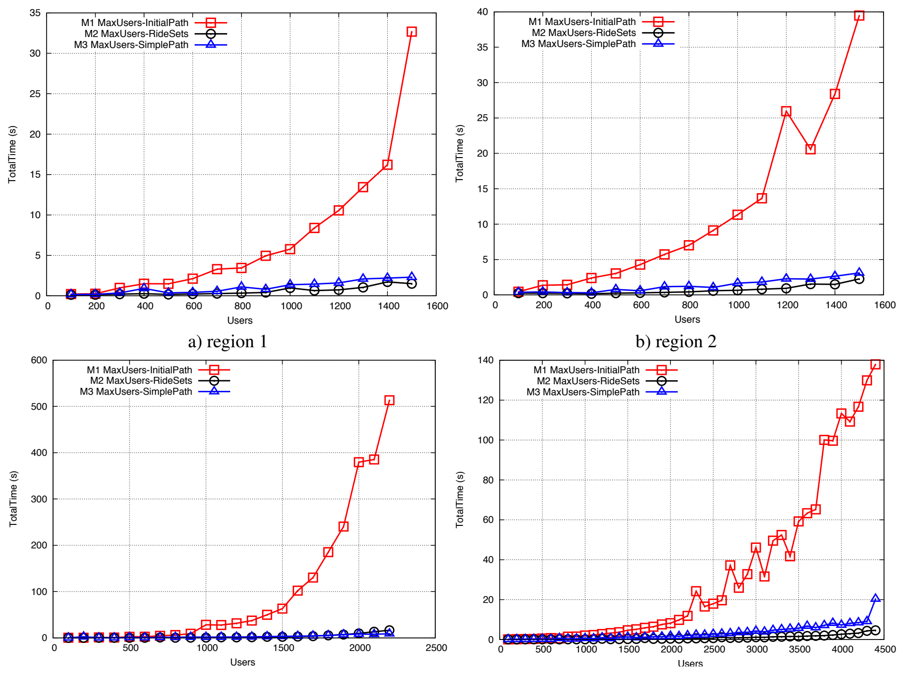
Table 2: Instance parameters in benchmark: benchS (drivers become shifters when it is possible)

8 Results

In Figure 4 we compare the optimal number of served users a), and the distribution of served users as riders b) and served users as drivers c) returned when solving ridesharing problems of same size for the two benchmarks. In Figure 4 a), the optimal solutions returned for the problem instances with shifters of benchS assign almost all the participants to a ridesharing trip. When there are no shifters, the optimal solution assigned around 50% of the users.

For the two benchmarks, (benchO, benchS) the proportion of assigned users as drivers represents respectively 30% and 60% of the users. Similarly, the proportion of assigned riders diverges for the two instances and represents up to 60% of the users for the instances with shifters and up to one third of the users for the instances of benchO. The same pattern can be observed for the percentage of assigned users as drivers. Intuitively, In the presence of shifters our model prefers to distribute the satisfied riders over different drivers' cars rather than merging them in one car. This characteristic is suitable for maintaining long-term sustainability of the ridesharing scheme.

In Figure 5 and Figure 6 we evaluate the solving performance of maximizing the number of served users. The original model *M1* is the ridesharing model introduced in section 4. The reformulated model *M2* represents the first reformulated ridesharing model introduced in section 5. It encodes the resource allocation constraints and the car



a) region 1 b) region 2

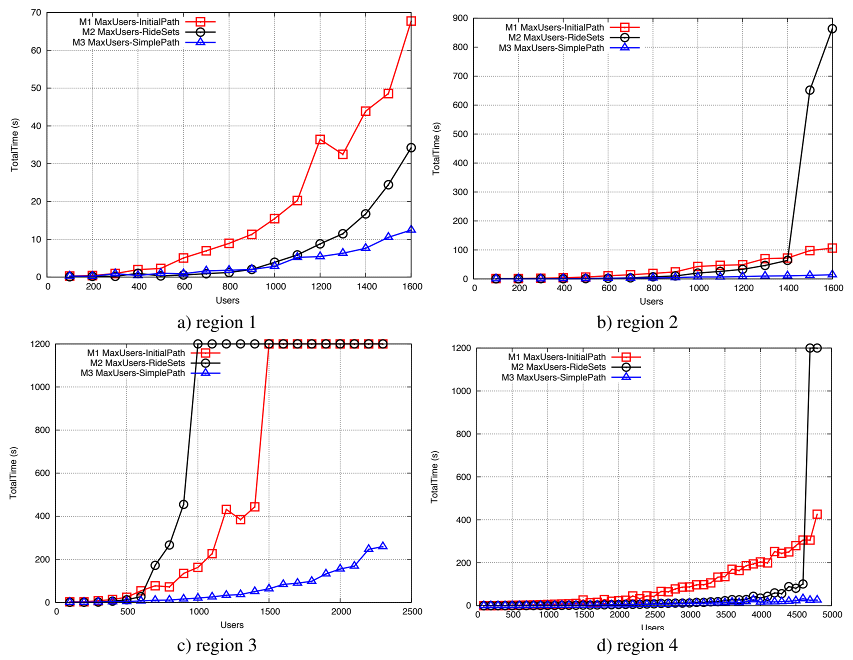
c) region 3 d) region 4

Fig. 5: CPU time for solving optimally the intances of benchO in the four regions

capacity constraints based on the drivers' maximal time-consistent sets of overlaping riders. The second reformulated model *M3* introduced in section 6 is based on a restriction of the ride-sharing path constraint on the auxiliary variables that does not remove any solutions.

8.1 Evaluation of solving time performance of instances where users have fixed role

In Figure 5, we evaluate the solving time performance of maximizing the served users on ridesharing problem instances of the benchmark benchO, where there are few shifters. For all the models and the regions, the time to find an optimal ridesharing plan for instances up to 1000 users is less than 20 seconds. At more than 1000 users in region 3 and 2000 users in region 4, the solving time of the initial model $M1$ grows rapidly and exceed one minute. In contrast to $M1$, the solving time of the reformulated models $M2$ and *M3* grows slowly and steadily for all instances and never exceed 10 seconds, resp. 20 seconds, in the worst case. Consequently, when we compare the solving time performance of the different models on the largest number of users, the new reformulated models, $M2$ and *M3* are up to one order of faster than the model original model $M1$. The efficiency of both methods can be partially explained by the absence of shifters making the instance easier to solve. For this benchmark the solving time of $M3$ is always slightly faster or similar to the solving time of *M3*. In the next section we consider



a) region 1 b) region 2

c) region 3 d) region 4

Fig. 6: CPU time for solving optimally the intances of benchS (with shifters) in the four regions

a more flexible ridesharing scheme where each driver is considered as shifter when the latter has a possible match as passenger.

8.2 Evaluation of solving time performance of instances where drivers can change role

In the second set of experiments we move to harder problems. In Figure 6, we evaluate the solving time performance of maximizing the served users on the ridesharing problem instances of the benchmarks benchS allowing drivers to change role when it is possible. Compare to the benchmark bench0, depending on the model and the region, the solving time can be multipled by a factor of ten. Here the experiments show the superiority of $M3$ over $M1$ and $M3$ for solving hardest instances. The model $M3$ is up to factor of 2 faster than $M2$ for solving the instances in region 1 and it is up to one order of magnitude faster than both $M2$ and $M1$ for solving the other instances.

In region 1 Figure 6 q), the model $M1$ using the initial formulation for encoding resource allocation problem solves efficiently small instances then it suddenly grows drastically while the solving time of the model $M2$ and *M3* remain very fast. For the largest instances of region 1, the solving time of $M3$ is a factor of 2 faster than $M2$ and a factor of 4 faster than $M1$.

In region 2 Figure 6 b), only the solving time of *M3* remains very fast and is up to 20 seconds for largest instance of the region. The solving time of $M2$ grows unexpectedly exponentially after 1400 users to reach 15 mins to solve the largest instance of the region. Inversely, the solving time of $M1$ grows steadily and finishes at 1min30s.

The ridesharing instances of region 3 in Figure 6 c) are characterized by the high number of possible ride matches per users, (cf. Table 2). These instances are harder to solve. The solving times of $M1$ and $M2$ do not scale and show rapidly an exponential pattern after 700 users. It is not the case for $M1$ that solves 1000 users in less than 30 secs and 2400 users in less than 5 minutes.

In region 4 Figure 6 c), the graphs of feasible matches show a lower connectivity (Table 2). In this case, the solving time of $M1$ grows steadily to reach 7 minutes for 4800 users. The solving time of $M2$ and $M3$ grow very slowly and steadily up to 4600 users. At this point $M2$ unexpectedly heats the time limit while the solving time of $M3$ remains very low.

In summary, the solving time of the ridesharing instances with shifters in shows the benefit of reformulating the drivers' path constraint using the model $M3$.The benefit of the model *M3* is more visible when solving the instances of region 4 and region 3 Figure 6 c) qnd d0. Only $M3$ is able to solve all the instances in less than one minute for 1000 users. The other models fail to solve the hardest instances (region 3) within this time limit and do not scale as nicely as $M3$ when increasing the size of the instances.

9 Conclusion and Future Work

We have designed new formulations for solving high-demand ridesharing to achieve a system-wide objective. We have demonstrated empirically that the new models achieve up to one order of magnitude improvement over a basic model for this problem, and we are able to find matches for thousands of users in under one minute. This fast and scalable performance is an important step towards future deployment of ridesharing schemes. The approach will be applied to more general cases where the aim is to achieve sustainability objectives such as minimising the number of cars entering an urban area, or minimising the total driven distance.

For successful deployment of ridesharing schemes, this optimisation approach should be integrated with user preference and matching optimisation, to ensure that participants are satisfied with the details of their trips. We have also begun to integrate the ridesharing optimisation model into a multi-modal transport planner. Finally, the improvements to the model are not specific to ridesharing, and we will investigate their application to a wider range of resource allocation problems.

References

1. Niels Agatz, Alan Erera, Martin Savelsbergh, and Xing Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295 – 303, 2012.
2. Niels A.H. Agatz, Alan L. Erera, Martin W.P. Savelsbergh, and Xing Wang. Dynamic ride-sharing: A simulation study in metro atlanta. *Transportation Research Part B: Methodological*, 45(9):1450 – 1464, 2011.
3. Vincent Armant and Kenneth N. Brown. Minimizing the driving distance in ride sharing systems. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus, November 10-12, 2014*, pages 568–575, 2014.

4. Vincent Armant, John Horan, Nahid Mabub, and Kenneth N. Brown. Data analytics and optimisation for assessing a ride sharing system. In *Advances in Intelligent Data Analysis XIV - 14th International Symposium, IDA 2015, Saint Etienne, France, October 22-24, 2015, Proceedings*, pages 1–12, 2015.
5. Andrea Attanasio, Jean-François Cordeau, Gianpaolo Ghiani, and Gilbert Laporte. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Comput.*, 30(3):377–387, March 2004.
6. Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15, 2010.
7. Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.
8. Masabumi Furuhata, Maged Dessouky, Fernando Ordóñez, Marc-Etienne Brunet, Xiaoqing Wang, and Sven Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57(C):28–46, 2013.
9. Ece Kamar and Eric Horvitz. Collaboration and shared plans in the open world: Studies of ridesharing. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 187–194, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
10. Carlo Manna and Steve Prestwich. Online stochastic planning for taxi and ridesharing. In *26th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2014, Limassol, Cyprus, November 10-12, 2014*, pages 906–913, 2014.
11. M. Schilde, K. F. Doerner, and R. F. Hartl. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Comput. Oper. Res.*, 38(12):1719–1730, December 2011.
12. Gilles Simonin and Barry O'Sullivan. Optimisation for the ride-sharing problem: a complexity-based approach. In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 831–836, 2014.
13. Jamal Yousaf, Juanzi Li, Lu Chen, Jie Tang, Xiaowen Dai, and John Du. Ride-sharing: A multi source-destination path planning approach. In Michael Thielscher and Dongmo Zhang, editors, *Australasian Conference on Artificial Intelligence*, volume 7691 of *Lecture Notes in Computer Science*, pages 815–826. Springer, 2012.