Fast Optimised Ridesharing: Objectives, Reformulations and Driver Flexibility

Vincent Armant, Kenneth N. Brown

Insight Centre for Data Analytics, Department of Computer Science, University College Cork, Ireland

preprint version accepted to Expert Systems With Applications in 2019

Abstract

Ridesharing schemes, which match passengers to private car journeys with spare seats, are becoming popular for improving access, and reducing costs, emissions and congestion. Current deployed ridesharing systems offer matches to participants based on trip details and on preferences. They do not optimise for system-wide objectives that generally incur a high computational cost. Academic studies tend to focus on specific single objectives, while stakeholders and policy-makers for practical deployments may have many different, possibly vague, objectives. In this paper, we try to reduce the gap between research and deployed applications, and tackle the problem of computing global plans for a flexible ridesharing scheme, for different objectives. In a flexible scheme, some drivers are willing to leave their cars and become passengers of other drivers. We consider objectives to improve access (maximising served passengers), reduce emissions and congestion (minimising total distance), and to sustain the long-term viability of the system (maximising served users). In the studied scheme, riders reach a meeting point and are picked-up along the intended routes of the drivers. We show that when drivers can change roles and become passengers, a state-of-the-art solver is not able to solve quickly ridesharing schemes optimizing for global sustainable objectives. To handle this issue, we introduce new spatio-temporal constraints, based on the notion of time- consistent sets of riders, from which we derive new logical rules and routing properties. We encode the new properties and the derived logical rules within new ridesharing problem formulations. We show that solving the introduced formulations encoding the routing properties and the logical rules are up to one order of magnitude faster than the basic model, and that even for the larger instances with 1000s of users, achieves results within 2% and 5% of the optimal value within a time limit of 300 seconds. Our study also shows the interaction between the different objectives and to which extent satisfying one objective also satisfies other objectives. This study demonstrates that the introduced method that integrates spatio-temporal constraints encoding the drivers' shared paths using logical rules within ridesharing problem formulations, help to solve faster, larger ridesharing schemes aiming at satisfying a variety of global sustainable objectives. The outcome of this study is of particular interest to both developers and researchers interesting in reformulation techniques using logical rules to speed-up the solving of real world applications. The study is also of interest to a variety of stakeholders (users, companies, public institutes) that aim at offering or maintaining ridesharing schemes as a sustainable solutions.

1 Introduction

Ridesharing schemes, where prospective passengers are matched to spare seats in private car journeys, are becoming popular in large city regions. Successful schemes increase transport options for citizens, reduce travel costs, take cars off the roads and reduce the total driven distance in the city region, and thus reduce emissions and congestion. 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. The underlying problem is a combination of route and schedule optimisation with a matching problem. Most deployed applications focus on the on-line problem of proposing attractive matches to individual participants, concentrating on user preferences, rather than addressing the system-wide optimisation problem, and so many proposed benefits of the schemes may not materialise. There are two open challenges for properly exploiting ridesharing schemes. The first is the societal and policy challenge of understanding the implications of focusing on different aims and incentives within the schemes,

establishing the tradeoffs between different possible objectives, and understanding how these schemes can be made sustainable in the long term. On the other hand, to show the feasibility of global sustainable ridesharing schemes, it is important to demonstrate that these schemes matches the scalability and the time requirements of deployed applications. The second challenge is technical, developing fast and scalable solutions for system-wide optimisation of real-world ridesharing problems. For this specific challenge, we explore the solution of integrating spatio-temporal constraints observed along drivers paths within different problem formulation, including conflicting assignment rules, in order to solve quickly larger ridesharing schemes.

In this paper, we make contributions towards both of these challenges. We develop mixed integer programming models with system-wide objectives for ridesharing schemes. We include the option of flexible drivers; that is, drivers who offer seats in their vehicles, but who are also willing to accept a ride as a passenger if a suitable match can be found. Allowing this flexibility enhances the environmental benefits of the schemes, since it directly reduces the number of cars being used, Agatz et al. (2012), and increasing the number of flexible drivers should increase the service options for all participants, Armant et al. (2015). Since the policy objectives of deployed schemes are often unclear, and may evolve over time, we investigate multiple different objectives applied to the same core combinatorial problem. We consider the two shorter-term objectives of supporting mobility and reducing emissions and congestion, and the longer term objective of maintaining the sustainability of the scheme by encouraging participation. Specifically, we develop objective functions which (1) maximise the (weighted) number of passengers placed in vehicles, (2) minimise the total driven distance across the scheme, and (3) maximise the number of participants, whether driver or passenger, receiving a match. Maximising the number of passengers views ridesharing as a public service to provide ubiquitous low-cost mobility, or to support exceptional events such as public transit disruptions or one-off sports events, festivals or concerts. The use of weighting allows us to adjust the priority between flexible drivers and pure passengers, and so can change the emphasis from increasing mobility services to removing cars from the road. Minimising the driven distance focuses directly on reducing total traffic and thus indirectly reducing emissions. A similar weighting can be used to account for participants who will drive outside the scheme if they do not receive a match. Finally, maximising the total number of matches is intended to encourage longer-term participation, since participants who rarely receive matches will stop interacting with the service. Although this objective might reduce benefits obtained from the scheme on a single day of operation, it should serve to increase those benefits in the longer term, by growing the scheme over time. We evaluate each objective applied to the core model in terms of the performance of the scheme measured against the other objectives, to assess the overall impact of choosing any particular option.

Secondly, we focus on the runtime performance of the models, assessing how well the models scale with increasing participation. We assume incoming requests are batched, with a required response time of a few minutes to compute an optimised plan for each additional batch. In each ridesharing plan, drivers pick up and drop off passengers at points on their intended path, within an intended travel time window, based on normal time-dependent traffic conditions. These points are selected from prospective passengers able to move between their source and destination and the driver's path within acceptable time windows. In practice, we assume drivers and passengers can negotiate detours and new pick-up points external to the plan, while respecting advertised time windows. Any change occurring in the route, travel time, or meeting time, are taken into account in the system-wide plan computed at the next update. Modelling and integrating spatio-temporal constraints within real-world applications has been a successful technique to assist decision making processes of various intelligent systems. We first develop a standard formulation of the ridesharing problems, encoding different sets of constraints encoding the drivers' path and time windows, the feasible matches and the driver's role flexibility. The model based on this formulation is able to solve, to optimality, ride sharing plans for over 4000 users in under 2 minutes on real-world datasets, if the problems exclude flexible drivers. When flexible drivers are included, the performance deteriorates, and the models, using a state-of-the-art solver, do not scale to real world problem sizes. To address this issue and improve the decision making process of deployed ridesharing schemes that aim at satisfying more sustainable objectives, we integrate new spatio-temporal constraints encoding the drivers paths within two reformulations of the core resource

allocation constraints. The first reformulation exhibits a method for derivating spatio-temporal constraints modelling each driver's feasible rides into a set conflicting assignment rules between riders. The second reformulation is based on a simplification of the driver's path constraints to speed up solving time.

We evaluate the different models and objectives empirically, on problem instances created from datasets of trip adverts and requests from real ridesharing clusters in Norway, Texas and California. We show that the new formulations are up to one order of magnitude faster than the initial MIP encoding, finding system-wide matches for thousands of participants, with flexible drivers, in under a few minutes. For time-limited scenarios, we also consider the optimality gap, and we show that the reformulated models can stay with 10% of the optimal solution for significantly larger problem instances. We assess the relative performance of the models on the different objectives. We show that the objective of maximising the number of matches is a significantly easier problem than the other two objectives, finding optimal solutions for much larger problems in shorter time. We also show that the two shorter-term objectives have similar performance to each other, each producing solutions that are within 2% of the optimal value when measured against the other objective, thus showing that increasing mobility service and minimising driven distance may be compatible goals, if drivers are flexible. Maximising the number of matches, however, does show a drop of between 10% and 20% on the other measurements, and so careful handling will be required to balance the short term objectives on a single day of a scheme with the longer term objective of maintaining the scheme's viability.

The integration of new spatio temporal constraints within different ridesharing problem formulations show that we can solve faster larger ridesharing scenarios satisfying sustainable objectives. The methodology for integrating the spatio-temporal constraints is as follow. First, we introduce the notion of time-consistent set of riders that represents a potential set of riders with which a driver can share its journey. Then, we exhibit new routing properties that show that maximal sets of time-consistent riders are a succinct representation of all the possible trips shared between the riders and a driver along the driver's intended path. From the representation of maximal sets of time-consistent and overlapping riders we derive a set of assignments rules. Each assignment rule takes the form of a conflict between two riders sharing a same driver. We encode the assignments rules and the new routing properties within two MIP formulations of ride-sharing problems.

In summary, we make a number of contributions to the development, optimisation and evaluation of ride sharing schemes:

- we develop ridesharing problem formulations to model real-world schemes, for thousands of users, including flexible drivers, with acceptable response times;
- we exhibit new methodology for deriving and integrating logical rules from spatio-temporal constraints within problem formalisation
- we propose and evaluate two reformulations, and demonstrate orders of magnitude improvement over the initial models;
- we study the impact and the interactions of three different objectives for ridesharing schemes;

Our results show that system-wide optimisation, for different environmental objectives, of ride matches in ridesharing is feasible for large real-world schemes, and is able to return optimal or close-to-optimal solutions with a response time of minutes for thousands of users at a time.

2 Related work

The dial-a-ride problem Cordeau and Laporte (2007), which has long been studied in the OR community, 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. (2010), Attanasio et al. (2004). The dial-a-ride drivers have no strong journey requirements of their own, whereas in ridesharing schemes, both the drivers and the riders have their own objectives Furuhata et al. (2013). Specific ridesharing

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. (2012). In addition to shifters, the idea of flexible roles is further extended Armant and Brown (2014) by assuming that each pure rider who is not served in the matching has a probability of driving on their own, included as a penalty in the objective function. Armant et al. (2015) assess the performance of a deployed ridesharing 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. (2011), and the complexity increases as the number of shifters increases. Kamar and Horvitz (2009) model the problem as one of collaborative planning, where agents must balance competing goals. Yousaf et al. (2012) model the problem as multi source-destination path planning, with a wide range of competing objectives including privacy and incentives. Schilde et al. (2011) and Manna and Prestwich (2014) 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 (2014) 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. Knapen et al. (2015) use a graph theory framework to show the scalability issues raised by carpooling matching services that aim to maximise the global expected value of successful negotiation among carpool members. In their formulation, each node of the graph represents a weekly carpool request or offer, and each edge models the probability of successful negotiation between the pair of users. This probability is estimated using different similarity measures including path and time similarity. The main focus is to analyse characteristics and identify scalability issues of a network of carpool members built from a synthetic dataset where the similarity measures have been learned or estimated.

Coltin and Veloso (2014) propose three heuristic algorithms to schedule ridesharing trips with transfers. Their problem formulation corresponds to a vehicle routing problem with pick-ups and deliveries with transfers. The goal is to assign a road to each vehicle. The simulated ridesharing problems based on grid networks and a map of San Francisco show the benefit of transferring passengers to increase the ridesharing service capability or to reduce the total driven distance. Passengers have no time window constraints.The experiments show the scalability of the approach and simulate ridesharing problems for up to 100 users. In our study, we assume that drivers pick up and drop off passengers along their usual paths. Although we do not consider transfers, we do consider flexible roles for drivers and realistic time trip duration and time windows. Stiglic et al. (2015) assess the potential benefit of introducing meeting points at a certain distance from riders' departure and destination. The benefit is measured in terms of saved mileage and an increase in the assigned matches. Stiglic et al. (2016) also investigate different types of matching flexibility such as the possibility for users to alter their departure time or to deviate from their usual route. In our study, we evaluate different problems and formulations based on ridesharing instances built from real trip adverts in different regions. The feasible match graph of each problem instance is based on the observed maximal distance within which a rider would accept to meet the driver's path and the observed maximal time a driver would alter a departure time in the different regions. Wang et al. (2016) propose an activity-based algorithm instead of a usual trip based formulation for solving optimally the problem of maximizing the number of one-to-one feasible matches between users. An activity is found in different spatial destinations e.g., a supermarket is an activity. The experimental results, based on a small set of trips<100 in Victoria State in Australia, show an increase in the matching rate of activity based ridesharing compared to usual trip based ridehsaring. In our approach, we do not consider multiple destinations for a trip, but we do consider multiple pick-ups and drop-offs and the possibility for drivers to change role.

Alonso-Mora et al. (2017) focus on a pick-up and delivery problem or a dial-a-ride problem operated by taxis. In this context, the proposed approach first solves the ridesharing problem by an heuristic solution before improving the quality of the solution by an exhaustive search. This approach allow any-time answers of system-wide ridesharing plans. In our study the main focus is a ride-sharing problem where drivers follow their own objectives and have their own path and time constraints while servicing passengers. In that sense

our work is closer a ride-sharing problem with Time windows (see below Herbawi and Weber (2012)). The approach of Alonso-Mora et al. (2017) solves both the rebalancing and the repositioning of fleet of vehicles the approach of Alonso-Mora et al. (2017) and can benefit from the reformulation principles of our approach.

Closer to our work, Herbawi and Weber (2012) described a Ridematching Problem With Time Windows (RMTW) where vehicles are not free to move everywhere and are limited to time-window constraints. In this piece of work the authors use a genetic algorithm to solve the introduced multi objective ILP ride-sharing problem. In the experiments the authors consider a synthetic benchmark consisting of 698 trips extracted from a travel survey conducted in Illinois. The quality of the approach is evaluated in term of assigned riders and drivers' and riders' travel time. The results show that the proposed genetic algorithm is able to solve reasonable size ride-matching problems. Compare to our approach, rather than formulating a complex multi-objective function, we model more simple objectives at the intent of specific stakeholder and show that our approach is able to answer in real time for all the introduced models. Furthermore, because Herbawi and Weber (2012) use a genetic algorithm approach, it is difficult to assess the quality of the solution with respect to the optimal. In our experiments we show that our approach is able to return optimal and close to optimal plans for reasonable size ride-sharing problems. Finally the time performance in Herbawi and Weber (2012) is given in term of number of generations. It is then also difficult to estimate the time period within which the approach returns optimized ride-sharing plans.

Wu et al. (2016) solve carpooling problems of at most three passengers up to 100k users. The experiments are based on synthetic instances built for the San Francisco Bay Area. The study covers different formulations including three set partitioning, dynamic programming, integer programming and different local search heuristics including hill-climbing, simulated annealing and large neighbouring search. Despite the variety of heuristics used and the problem simplification, the best solutions found still have an optimality gap of 100% over the lower bound. In this study we focus on real time approaches for solving to optimality (or close to optimal) different sustainable ridesharing problems including minimising the driving distance.

3 Preliminaries

In this section, we introduce the preliminary notation describing the data and derived expressions given as the input for the ridesharing problems. 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 trip and define time-consistent trips. We introduce the following notation.

The set of ridesharing users U is partitioned into three sets. D is the set of drivers offering seats in their cars. R is the set of riders asking for a ride. S is the set of shifters, i.e., drivers willing to change role and become passengers. By default r denotes a prospective rider in $R \cup S$, and d denotes a possible driver in $D \cup S$, unless stated otherwise. The car capacity of a driver d is denoted by q_d.

The set L contains the road node locations identified by their GPS coordinates. The path π^u is a sequence of locations $l \in L$ that the user $u \in U$ has planned to visit, from starting point to destination. For the driver path π^d, $pred_{\pi^d}(l)$ denotes the predecessor of l in the path π^d. If $u \in S \cup R$, then π^u is the path the user would take if they drove themselves. For a user $u \in U$, α_u and ω_u denote the starting point and the destination point of u in the path π^u. For a given driver d and a rider r, the location $\alpha_{d,r} \in L$ denotes the pick-up point in the path π^d, and $\omega_{d,r}$ denotes the drop-off point.

The time $\phi_{d,l,l'}$ denotes the travel time from l to $l^{'}$ in the path π^d. The travel time is positive when l precedes l in π^d, and negative otherwise. The travel time ϕ_d denotes the travel time of the path π^d from origin to destination. Similarly, the distance $\delta_{d,l,l'}$ denotes the travel distance from l to $l^{'}$ in the path π^d. The travel distance δ_d denotes the travel distance of the path π^d.

For a user u, $\varepsilon_{u,l}$ and $\lambda_{u,l}$ denote the earliest time and the latest time the user u can be at l. For a driver d and a rider r, $\varepsilon_{d,r,l}$ and $\lambda_{d,r,l}$ denote the earliest time and the latest time d and r can be together at l.

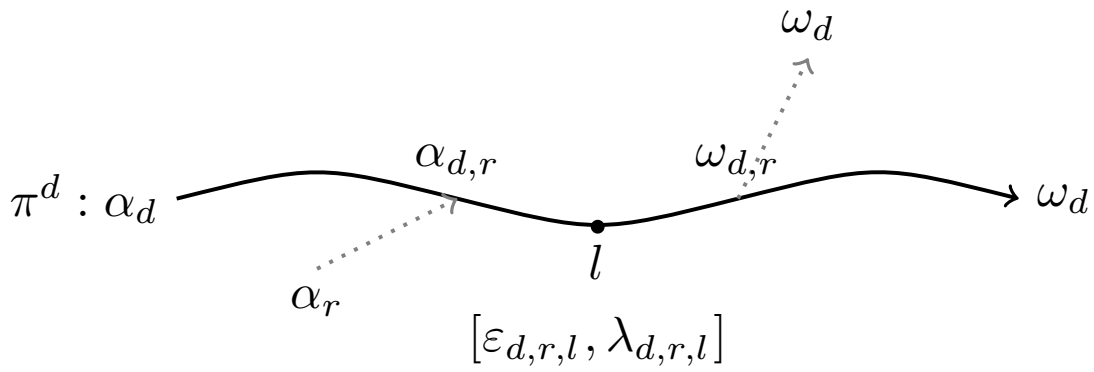

Figure 1: Time-consistent ridesharing trip betwen a driver d and a rider r

The earliest time a driver d and a rider r can be at l in π^d is the maximal time between the earliest time for d to arrive at l ($\varepsilon_{d,\alpha_d} + \phi_{d,\alpha_d,l}$), and the earliest time from the pick-up of r to l ($\varepsilon_{r,\alpha_{d,r}} + \phi_{d,\alpha_{d,r},l}$). Note that if $\alpha_{d,r}$ is before l in π^d then $\phi_{d,l,\alpha_{d,r}}$ is negative :

$$\varepsilon_{d,r,l} = max(\varepsilon_{d,\alpha_d} + \phi_{d,\alpha_d,l}, \varepsilon_{r,\alpha_{d,r}} + \phi_{d,\alpha_{d,r},l})$$

The latest time a driver d and a rider r can be at l in π^d is the minimal time between the latest time for d to arrive at l ($\lambda_{d,\omega_d} - \phi_{l,\omega}$), and the latest time from the drop-off of r to l ($\lambda_{r,\omega_{d,r}} - \phi_{l,\omega_{d,r}}$) :

$$\lambda_{d,r,l} = min(\lambda_{d,\omega_d} - \phi_{l,\omega}, \lambda_{r,\omega_{d,r}} - \phi_{l,\omega_{d,r}})$$

Given this notation, 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 $d \in D \cup S$, $d \neq r$, is time-consistent, if at any point $l \in \pi^d$, there exists a consistent time window $[\varepsilon_{d,r,l}, \lambda_{d,r,l}]$ s.t. $\varepsilon_{d,r,l} \leq \lambda_{d,r,l}$*

Implicitly a time-consistent ridesharing trip between a driver d and a rider r means d is able to pick-up and drop-off r while respecting earliest start and latest arrival constraints. A time-consistent ridesharing trip defines a feasible match between a driver and a rider. At each period of time, the feasible ridesharing trips between drivers and riders are represented by a time-consistent ridesharing graph.

Definition 2 (Time-consistent ridesharing graph) *The graph $G = (D \cup S, R \cup S, E)$ is a time-consistent ridesharing graph matching drivers to riders if $E \subseteq (D \cup S) \times (R \cup S)$ and $\forall (d, r) \in E$, $d \neq r, (d, r)$ is a time-consistent ridesharing trip.*

The time-consistent ridesharing graph is a common input of the different system-wide ridesharing problems we try to solve. The set of riders $pick_{\pi^d}(l)$ (resp. $drop_{\pi^d}(l)$) denotes the set of feasible riders that can be picked up (resp. dropped off) one at a time by the driver d.

For the sake of clarity, we illustrate the notation describing time-consistent ridesharing trips 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 α_d to its destination ω_d. The first dashed grey line depicts the pick-up path of r from its departure α_r to the pick-up point $\alpha_{d,r}$. The second dashed grey line depicts the drop-off path of r from its drop-off point $\omega_{d,r}$ to its destination ω_r. We also depict the consistent time window $[\varepsilon_{d,r,l}, \lambda_{d,r,l}]$ representing the earliest time $\varepsilon_{d,r,l}$, and the latest time $\lambda_{d,r,l}$, the driver d and the rider r can visit l if they share a ride together.

We summarise the notation in Table **??**.

	Users
U	the set of all users, partitioned into three sets D, R, S
D	the set of drivers
R	the set of riders
S	the set of shifters
	Locations
L	the set of locations
π^u	the path of u, i.e., a sequence of locations in L
$pred_{\pi^u}(l)$	the predecessor of l in π^u
α_u	the starting point of π^u
ω_u	the destination point of π^u
$\alpha_{d,r}$	the possible pick-up point of r in π^d
$\omega_{d,r}$	the possible drop-off point of r in π^d
$pos_{\pi^d}(l)$	the position of l in the path π^d
	Times and distances
$\varepsilon_{d,l}$	the earliest time d can be at l
$\varepsilon_{d,r,l}$	the earliest time d and r can be at l
$\lambda_{d,l}$	the latest time d can be at l
$\lambda_{d,r,l}$	the latest time d and r can be at l
ϕ_d	the driving time to cover π^d
$\phi_{d,l,l'}$	the driving time from l to l' in π^d
δ_u	the driving distance to cover π^u
$\delta_{d,l,l'}$	the driving distance from l to l' in π^d
	Other notations
q_d	the capacity of d's car
$pick_{\pi^d(l)}$	riders that can be picked up one at a time by d at l
$drop_{\pi^d}(l)$	riders that can be dropped off one at a time by d at l

Table 1: Input data notation

4 Problem Descriptions

In general, the aims of ridesharing are to improve mobility by providing rides to users without cars, to reduce travel costs, to address environmental concerns by reducing the total driven distance on roads, and to alleviate congestion by reducing vehicle numbers. Different stakeholders will have different performance measures to be applied to the schemes, and these will need to be addressed in any optimisation of the ridesharing matches. To begin this analysis, we introduce an informal ridesharing problem description that encapsulates three different objectives that we will consider in the study. A solution to a ridesharing problem is a set of assignments that determine which passengers should ride with which drivers, and for each pick-up or drop-off, a feasible time window. To ensure the route and time window constraints are satisfied, we will model and solve the problem as a mixed integer program (MIP).

optimising a global sustainable ridesharing objective selected from:

1. maximising the number of assigned passengers (MaxRiders),
2. maximising the number of assigned users (MaxUsers),or
3. minimising the total driven distance (MinDistance);

subject to:

1. Flexible Assignment constraints ensuring that:
   (a) each passenger can be assigned to at most one driver's car (AssignedRider),
   (b) each driver can have 0 or more passengers assigned (AssignedDriver),
   (c) each shifter can be assigned to at most one driver's car (AssignedShifterAsPassenger),
   (d) each shifter can have 0 or more passengers assigned (AssignedShifterAsDriver),
   (e) each shifter is assigned to exactly one role, rider or driver (ShifterMutEx),
2. Drivers Intended Route Scheduling constraints ensuring that:
   (a) carriage of passengers by a driver does not exceed the car's capacity (CarCapacity),
   (b) users complete their journeys within a time window (TimeWindow).

In the following sections we describe in detail each of the three MIP models we use to formalise the different ridesharing problems. First we introduce the decision variables and constraints present in the common description of each of the ridesharing problems.

4.1 Variables

$x_r \in \{0,1\}$ states whether rider r is assigned a ride, for $r \in R$.

$x_d \in \{0,1\}$ states whether driver d is assigned a rider, for $d \in D$.

$w_s \in \{0,1\}$ states whether a shifter s is assigned as a rider, for $s \in S$.

$z_s \in \{0,1\}$ states whether a shifter s is assigned as a driver, for $s \in S$.

$y_{d,r} \in \{0,1\}$ states whether r is assigned to d's car, $d \in D \cup S, r \in R \cup S$

$o_{d,l} \in [0, q_d]$ represents the car occupancy for d as it leaves l. $o_{d,l}$ is bound by the car capacity q_d.

$v_{d,l} \in \mathcal{R}^+$ represents the time the driver $d \in D \cup S$ leaves the location l.

4.2 Constraints

The flexible assignment constraints state that each rider assigned to be a passenger is served if and only if (iff) they are assigned to exactly one driver, for both pure riders (1) and for shifters (2). A driver is assigned iff it has been assigned at least one passenger (3) and (4). A shifter is assigned and served as a driver if it is assigned at least one passenger (5) and (6). A shifter can be an assigned driver or an assigned passenger, or neither, but not both (7). The intended route scheduling constraints state that 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 when leaving the previously visited location (8). (9) and (10) states that when a driver d is assigned a passenger r, the visiting time anywhere in the path satisfies the conditions of a time-consistent ridesharing trip (cf. Definition 1). In (10) M is a big integer greater than the integer encoding the latest time of a day.

Flexible Assignment Constraints:

$$x_r = \sum_{(d,r)\in E} y_{d,r}, \quad \forall r \in R \qquad (AssignedRider) \tag{1}$$

$$w_s = \sum_{(d,s)\in E} y_{d,s}, \quad \forall s \in S \qquad (AssignedShifterAsRider) \tag{2}$$

$$y_{d,r} \leq x_d, \quad \forall (d,r) \in E \qquad (AssignedDriver\ a) \tag{3}$$

$$x_d \leq \sum_{(d,r)\in E} y_{d,r} \geq 1, \quad \forall d \in D \qquad (AssignedDriver\ b) \tag{4}$$

$$y_{s,r} \leq z_s, \quad \forall (s,r) \in E \qquad (AssignedShifterAsDriver\ a) \tag{5}$$

$$z_s \leq \sum_{(s,r)\in E} y_{s,r}, \quad \forall s \in S \qquad (AssignedShifterAsDriver\ b) \tag{6}$$

$$w_s + z_s \leq 1, \quad \forall s \in S \qquad (ShifterMutEx) \tag{7}$$

Drivers' Intended Path Scheduling Constraints:

$$o_{d,l'} = o_{d,l} + \sum_{r\in pick_{\pi^d}(l)} y_{d,r} - \sum_{r'\in drop_{\pi^d}(l)} y_{d,r'}, \quad \forall d \in D \cup S, \forall (l = pred_{\pi^d}(l')) \qquad (CarCapacity) \tag{8}$$

$$\varepsilon_{d,r,l} \times y_{d,r} \leq v_{d,l}, \quad \forall (d,r) \in E, \forall l \in \pi^d \qquad (TimeWindow\ a) \tag{9}$$

$$v_{d,l} \leq \lambda_{d,r,l} \times y_{d,r} + M \times (1 - y_{d,r}), \quad \forall (d,r) \in E, \forall l \in \pi^d \qquad (TimeWindow\ b) \tag{10}$$

The Drivers' Intended Path Scheduling constraints (8), (9)and (10) 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 (8), along the drivers' intended paths, encode the limited resource capacities specific to the ridesharing problem. The time feasibility constraints of the rides (9) and (10) define both the drivers' intended time path constraints and the time consistency of the ridesharing trip along the driver's route. In the next section, we will reformulate these constraints into simpler formulations using fewer variables, intended to make them easier to solve. In the immediate subsections below, we consider the different objective functions. Some of these objectives may allow us to omit some of the constraints.

4.3 Maximising the number of assigned passengers

Maximising the number of assigned passengers is intended to increase access to mobility services, by providing rides for as many people as possible, reducing the need for people to own cars, and offering cheaper services than last-minute taxi or public transit tickets. However, when we have flexible drivers, optimising solely for the number of assigned passengers may serve to increase the number of cars on the road, since we have an incentive to keep the shifters as drivers, to provide more passenger seats. Therefore, we introduce a weighting factor, β, to balance the reward obtained for assigning pure riders as passengers and assigning shifters as passengers, and the objective then becomes maximising the weighted sum of assigned passengers (11). If we choose $\beta > 1$, we prioritise assigning pure riders who might not have a car; if we choose $\beta = 1$ pure riders and shifters are treated equally; if we choose $\beta < 1$, then we prioritise assigned shifters as passengers and implicitly prefer taking cars off the road. Note that in that final case, β could be considered as an informal probability that the rejected pure rider decides to drive themselves outside the scheme, and so the objective becomes maximising the expected number of cars taken off the road.

More formally, the objective is to:

$$maximize(\sum_{s \in S} w_s + \beta \sum_{r \in R} x_r) \qquad (MaxPassengers) \tag{11}$$

For this objective, the number of assigned drivers is irrelevant, and so constraints (3) and (4) can be omitted from the model.

4.4 Maximizing the number of assigned users

Ridesharing schemes are only viable if there are enough participants: the more drivers we have, the higher the chance of a prospective passenger getting a ride; the more riders we have, the higher the chance of a driver getting a passenger, and thus offsetting some of their costs. Typically, users stop participating in a scheme if they receive no benefit. If we focus exclusively on riders, as in the previous objective, we run the risk of losing drivers from the scheme. Instead, we should consider also trying to maximise the number of drivers that get an offer, in order to ensure the long-term sustainability of the ridesharing scheme (which should also indirectly maximise the total number of assigned passengers over a longer time frame). More formally, the objective is to maximise the number of served users (12):

$$maximize(\sum_{d \in D} x_d + \sum_{s \in S} (w_s + z_s) + \sum_{r \in R} x_r) \qquad (MaxUsers) \tag{12}$$

4.5 Minimising the total driving distance

Minimising the total driven distance of cars in an urban area is an important objective that directly addresses the environmental sustainability goal of reducing carbon emissions and pollution. The previous objectives should direct solutions towards a smaller driven distance, by reducing the incentive for riders to use their own cars, and by turning shifters from drivers into passengers, and removing cars from road. However, the

actual outcomes will depend on the distribution of ride offers and requests, and may prefer solutions which involve many short rides with many different passengers, rather than a smaller number of longer shared journeys, or may force shifters to remain as drivers. In this case, we must take account of all behaviours by participants who do not get a match. Specifically, pure drivers ($\in D$) will definitely drive, regardless of whether or not they are given a passenger. Since we assume drivers follow their intended path, we cannot reduce their driving distance, and so are not directly involved in the objective. Shifters ($\in S$) will drive if not matched, and will leave their cars at home if they are matched, and so we receive a penalty of the length of the shifter's path for every shifter we do not assign as a rider. Finally, some pure riders ($\in R$) may have their own cars, and may choose to drive outside the scheme if they do not receive a match. We use the weighting factor γ to represent the probability that a denied rider decides to drive outside the scheme. Thus the formal objective becomes (13):

$$minimize(\sum_{s \in S} \delta_s \times z_s + \gamma \sum_{r \in R} \delta_r \times (1 - x_r)) \qquad (MinDist) \tag{13}$$

Since the number of matched drivers does not affect the objective value, we can again omit constraints (3) and (4) from the model.

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

As will be shown in the experiments in the following section, the above encoding of the different ridesharing problems is not sufficient, even with a state-of-the-art solver, to meet the time performance requirement for solving large real-world ridesharing instances with flexible drivers to optimality. We therefore attempt to reformulate the models to allow faster solving. Our first approach focuses on simplifying 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 a driver's 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 there exists a consistent time window within which the driver can ensure the delivery of the riders along the path. To tackle car occupancy constraints, we then extend the notion of time-consistent sets to overlapping riders. Implicitly, the ridesharing trips of two riders overlap if they share a common leg along the driver path. First, we define 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 reformulation of the resource allocation constraints and the car capacity constraints.

5.1 Driver's time-consistent sets of riders

We recall that $G = (D \cup S, R \cup S, E)$ denotes the graph of time-consistent ridesharing journeys between drivers and riders. We define a driver's time-consistent set of riders as follows :

Definition 3 (driver's time-consistent set of riders)
R_d is a time-consistent set of riders for the driver d if :

1. *each rider r in R_d shares a one-to-one time-consistent ridesharing trip with d,*
2. *at any location l in the path π^d, there exists a consistent visit time window $[\varepsilon_{d,R_d,l}, \lambda_{d,R_d,l}]$ for all riders in R_d s.t. :*
   (a) $\varepsilon_{d,R_d,l} = max(\{\varepsilon_{d,r,l} | r \in R_d\})$
   (b) $\lambda_{d,R_d,l} = min(\{\lambda_{d,r,l} | r \in R_d\})$

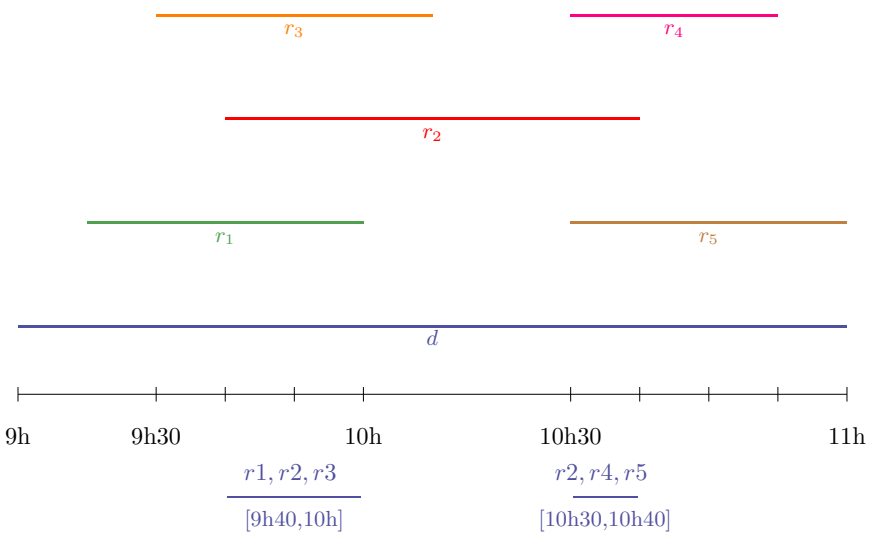

Figure 2: Consistent delivery time window of a driver d at a location l in π^d

(c) $\varepsilon_{d,R_d,l} \leq \lambda_{d,R_d,l}$

We establish the equivalence between the time dependency constraints of the original problem and the drivers' time-consistent set of riders (Definition 3) by the following proposition.

Proposition 1 *R_d is a time-consistent set of riders for driver d iff at any location $l \in \pi^d$ there exists a visiting time $v_{d,l}$ for d s.t. $\forall r \in R_d, \varepsilon_{d,r,l} \leq v_{d,l} \leq \lambda_{d,r,l}$*

Proof.
$\Rightarrow$ In Definition 3, at any location in the driver's path, the delivery time window $[\varepsilon_{d,R_d,l}, \lambda_{d,R_d,l}]$ of a time-consistent set of riders R_d for a driver d is defined as the intersection of the delivery time window of each rider $r \in R_d$ (Definition 3.2a and 3.2b). Since the time window $[\varepsilon_{d,R_d,l}, \lambda_{d,R_d,l}]$ is consistent at any location in the path π^d (Definition 3.2c), there is a time within the time window $[\varepsilon_{d,R_d,l}, \lambda_{d,R_d,l}]$ for which the driver can visit the location while satisfying the delivery of all riders in R_d.
$\Leftarrow$ If at any location of the driver's path there is a consistent visiting time to deliver a set of passenger R_d, at any location in the driver's path there is a consistent time window for the delivery of the set of passengers. R_d is then a time-consistent set of riders. $\square$

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 in its path. If the driver starts from l between 9:40am and 10am the riders $\{r_1, r_2, r_3\}$ can be picked up. The latter set is a time-consistent set of riders for the driver d. $\{r_3, r_2, r_5\}$ is another time-consistent 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_d of a driver d, at any location l in the path π^d, the size of the delivery time window for R_d is constant.*

Proof.
By Definition 3, we know that for each time-consistent set of riders R_d of a driver d, there 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_d . Let us consider two consecutive locations $l, l^{'}$ in the driver's path π^d and the travel time $\phi_{l,l'}$ between the two

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 $l, l^{'}$ in driver's path π^d is equal to the travel time between the two locations : $\varepsilon_{d,R_d,l^{'}} - \varepsilon_{d,R_d,l} = \phi_{l,l^{'}}$. 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 $l, l^{'}$ is also equal to the travel time between the two locations : $\lambda_{d,R_d,l^{'}} - \lambda_{d,R_d,l} = \phi_{l,l^{'}}$. As a consequence, we have $\lambda_{d,R_d,l^{'}} - \varepsilon_{d,R_d,l^{'}} = \lambda_{d,R_d,l} + \phi_{l,l^{'}} - (\varepsilon_{d,R_d,l} + \phi_{l,l^{'}}) = \lambda_{d,R_d,l} - \varepsilon_{d,R_d,l}$. Therefore the size of the delivery time window remains the same from l to $l^{'}$. We deduce that the size of the delivery time window of a time-consistent set of riders $\lambda_{d,R_d,l} - \varepsilon_{d,R_d,l}$ at any point l in the path π^d remains the same. □

As a consequence of this proposition, we do not need to express the TimeWindow constraints, for each rider, and for each location in the driver's path as it is suggested by (9) and (10); we just have to represent in the problem formulation the list of time-consistent sets of riders for just one location in the driver's path.

5.2 Driver's time-consistent sets of overlapping riders

A time-consistent set of riders represents a set of riders that can be picked up and dropped off within a consistent time-window by a driver along its path. A time-consistent set of riders does not encode the car occupancy constraints of drivers' vehicle. We tackle this issue by the notion of time-consistent set of overlapping riders. We recall that we denote by $pos_{\pi^d}(l)$ the position of l in the path π^d.

Definition 4 (Driver's time-consistent set of overlapping riders)
O_d is a time-consistent set of overlapping riders for the driver d, if :

1. *O_d is a time-consistent set of riders for d,*
2. *$\forall \{r, r'\} \subseteq O_d$:*
   (a) *either, the drop-off of r' is between the pick-up and the drop-off of r:*
   $pos_{\pi^d}(\alpha_{d,r}) \leq pos_{\pi^d}(\omega_{d,r'}) \leq pos_{\pi^d}(\omega_{d,r})$
   (b) *or, the pick-up of r' is between the pick-up and the drop-off of r:*
   $pos_{\pi^d}(\alpha_{d,r}) \leq pos_{\pi^d}(\alpha_{d,r'}) \leq pos_{\pi^d}(\omega_{d,r})$

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 one after the other even if the car capacity is one.

Before presenting the reformulation, the following proposition establishes the correspondence between the $TimeWindow$ and the $CarCapacity$ constraints of the initial ridesharing problem.

Proposition 3 *If a time-consistent set of overlapping riders O_d fits the car capacity of the driver d, the car occupancy constraints (8) and the time feasibility constraints (9) and (10) defined for d and O_d are satisfiable.*

Intuitively, we have shown the correspondance between the notion of time-consistent set of riders and the $TimeWindow$ constraints (9) and (10) in Proposition 1. By Definition 4, 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 does not exceed the driver's car capacity, it satisfies constraint (8).

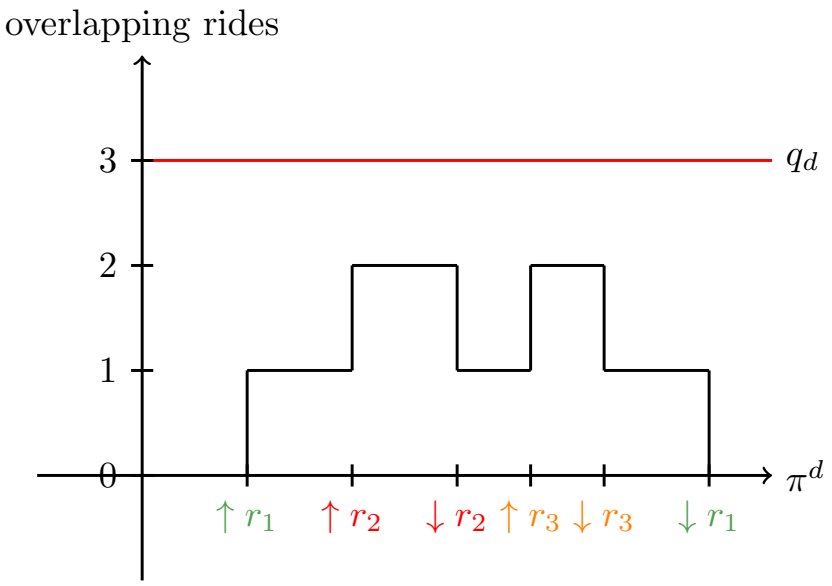

Figure 3: Overlaping rides of the time-consistent set $\{r_1, r_2, r_3\}$

5.3 Encoding the new ridesharing intended route scheduling 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 5 (maximal time-consistent set of rides)
M_d is a maximal time-consistent set of riders for d if:

1. *M_d is a time-consistent set of riders for d,*
2. *for all time-consistent set of riders R_d, if $M_d \subseteq R_d$ then $M_d = R_d$*

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 (9) and (10). A maximal time-consistent set of riders represents a possible set of passengers the driver is able to deliver regardless of 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 (14) encode a conflict between each pair of non time-consistent riders belonging to two distinct maximal sets. Implicitly these conflicts force the ridesharing solution for each driver to belong to a subset of one of its maximal time-consistent set of riders.

$$\begin{aligned} 0 \leq y_{d,r} + y_{d,r'} \leq 1, \\ \forall d \in D \cup S, \forall r \in M_d \setminus M'_d, \forall r' \in M'_d \setminus M_d, \nexists M"_d \text{ s.t. } \{r, r'\} \nsubseteq M"_d \end{aligned} \tag{14}$$

Constraints (14) 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. To reformulate constraint (8) we restrict the number of assigned ridesharing trips represented by MO_d by the car capacity

q_d.

$$\begin{aligned} 0 \leq \Sigma_{\substack{r \in MO_d}} y_{d,r} \leq q_d \\ \forall d \in D, \forall M_d, \forall MO_d \subseteq M_d \end{aligned} \tag{15}$$

In first reformulation of the original ridesharing model, we replace the $TimeWindow$ constraints (9), (10) and the $CarOccupancy$ constraints (8) by constraints (14) and constraints (15).

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' intended 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 these passengers 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 window 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 path. Consequently, if a driver starts their journey within this consistent time window and does not waste time between two locations, the time windows for the set of passengers will be respected. Using the previous proposition, we can now reformulate the initial drivers path consistency constraints (9) and (10) by the two following constraints.

$$y_{d,r} \Rightarrow (\varepsilon_{d,r,\alpha_{d,r}} \leq v_{d,\alpha_{d,r}} \leq \lambda_{d,r,\alpha_{d,r}}), \quad \forall (d,r) \in E \tag{16}$$

$$v_{d,l'} = v_{d,l} + t(\pi^d_{l,l'}), \quad \forall d \in D \cup S, \forall l = pred_{\pi^d}(l') \tag{17}$$

Constraints (16) force the existence of 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 (17) 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 of a no waiting path for a driver to deliver its passengers. There is no need to enforce a consistent time window for drop-off, since it is implied by the conjunction of the two constraints.

7 Datasets of Ridesharing Clusters in Norway, Texas, California

We evaluate the system-wide ridesharing models on problem instances created from datasets of real ridesharing trips collected for a period of two years in ridesharing clusters in Norway, and nine months in California and Texas. To check the scalability of our approach, we gradually increase the number of users by selecting randomly advertised ridesharing trips for each cluster. At each step we generate 20 ridesharing problem instances [1] . Clusters in Texas and California represent large agglomerations where ridesharing is commonly used by commuters and contain road infrastructures such as the High Occupancy Line to push drivers to share their journeys. Ridesharing in Norway's cluster is not yet considered a daily alternative to public transport. For each dataset, we infer a flexible trip time window for each user based on the history of

[1] https://github.com/ElVinto/ridesharing_pb_instances

	Users	Drivers	Riders	Shifters	R/ D	Edges	Matches per user*
Norway	1000	595	405	0	0.68	7993	8.0
Texas	1000	641	359	0	0.56	14647	14.7
California	1000	553	447	0	0.8	3367	3.4

Table 2: Instance parameters in benchmark: benchO (original users' requests)

	Users	Drivers	Riders	Shifters	(R+S)/ (D+S)	Edges	Matches per user*
Norway	1000	150	405	445	1.43	15462	15.5
Texas	1000	54	359	587	1.48	36167	36.1
California	1000	241	447	312	1.10	4980	4.9

Table 3: Instance parameters in benchmark: benchS (drivers become shifters when it is possible)

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 a trip, or the maximal distance a rider will accept for a ride.

We consider a common ride-sharing scheme where riders reach a meeting point on a driver intended path. In this context, a rider begins its journey from a start location and reaches a pick-up location to get in a driver's car. Similarly, to reach its final destination, the rider gets off the car at a drop-off location and reaches its intended destination. This corresponds to a real-world scenario of a ridesharing system where ridesharing companies show to the prospective riders both the expected path of drivers and the list of possible pick-ups and drop-offs points described by drivers. From a scenario assessment viewpoint, in the experiments, the pick-up location corresponds to the end point of the shortest path from a rider's start to the expected path of a driver. Similarly a drop-off point corresponds to the shortest path from the driver expected path to rider final destination. The computational effort required is the computation of Dijkstra Algorithm from a point, the rider start/end, to a set of points, the driver expected path. We do not need to run the Dijkstra algorithm for all the pairs of end points. In practice, for the start and end points of rider's journey, we run Dijkstra and store the nodes and the times within the reachable area of the rider's end points. The reachable area from a rider's start corresponds to the maximal distance within which a rider will be willing to join a ride. Reachable areas of riders is inferred from successful ride-matches observed each region, Armant et al. (2015). Note that with the help of reachable areas, the computation the shortest path from a rider end point to a driver path can be computed in $O(n^2)$, with n representing the number of point within the reachable area of a rider end.

From the inferred users' time windows, and the reachable areas of riders, we build a graph of time-consistent ridesharing trips between drivers and riders. We use the graph of time-consistent ridesharing trips to build two benchmarks. The first benchmark benchO consists of the original ridesharing requests collected from the different regions. In Table 2, we describe the average the indicators over 20 instances of 1000 users in benchO for each region. When building the instances we ensure that each user has at least one match. The instances of benchO have no shifters and are characterized by a low ratio Riders(R)/Potential Drivers(D)<1. As a consequence, it shows an imbalance between riders and drivers that may act as disincentive for participants if they frequently do not receive a suitable match to share their rides.

Each generated instance in the second benchmark, benchS, is built from an instance from the first benchmark benchO where driver is now considered as a shifter. We recall that a shifter a is driver that may become a passenger when they can be picked up by an other driver. As a result, the imbalance between riders and drivers appearing in the original instances now switches to be in favour of potential riders.

More precisely in Table 3 the ratio Potential Riders (R+S)/ Potential Drivers (D+S) is now higher > 1 and may lead to satisfying more users. We will see in the results section that allowing for shifters significantly increases the solving time. Finally, in all the experiments below, we set $\beta = \gamma = 0.3$.

8 Solution Quality of the Sustainable Ridesharing Schemes

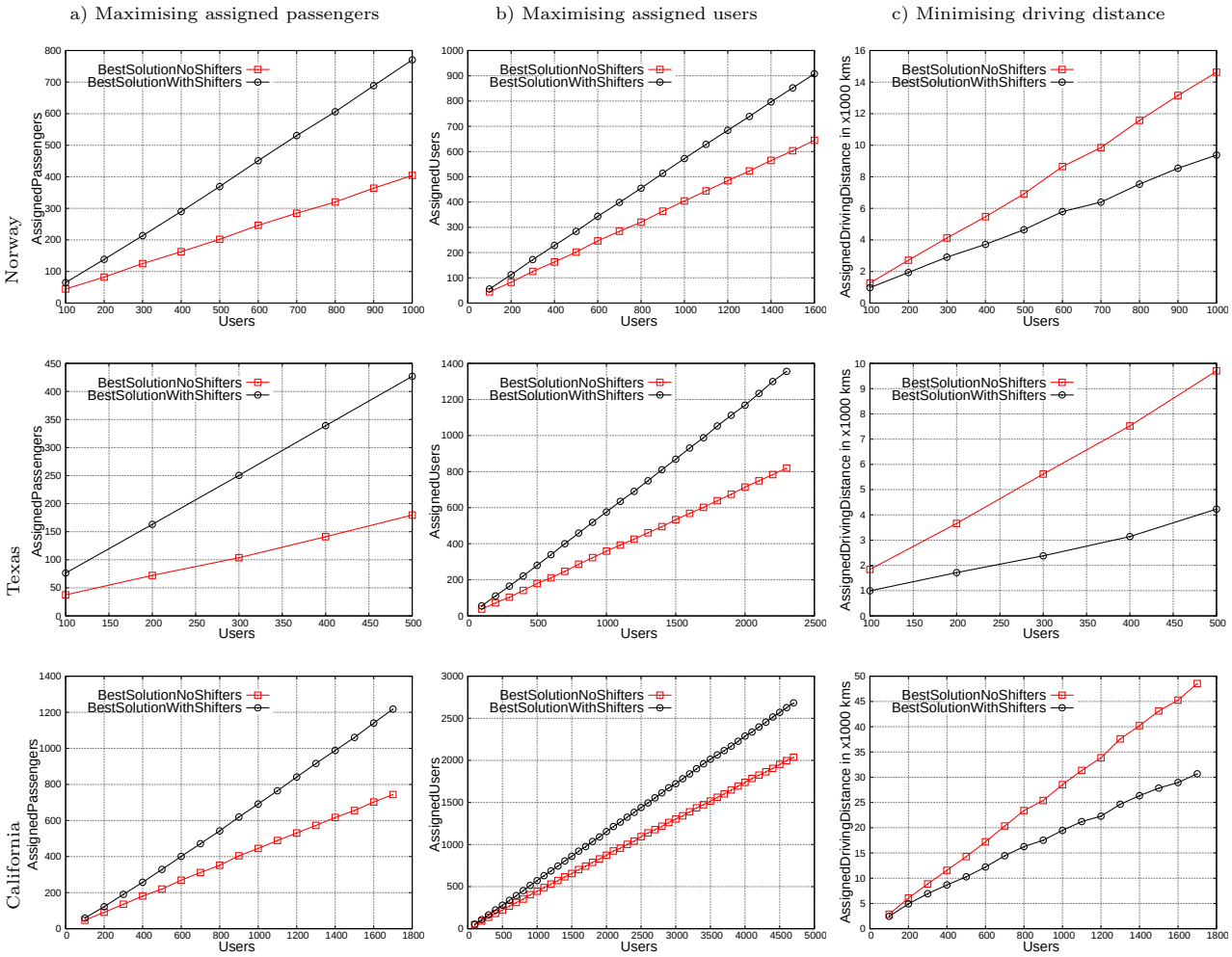

Figure 4: Comparison of the solution quality of instances with and without shifters

We measure the solution quality of the different ride-sharing schemes in term of assigned passengers, assigned users and driving distance. We first compare the solutions of ridesharing schemes without shifters in Figure 4 and highlight for benefit of shifters the three objectives respectively. We then compare to what extent satisfying one objective optimally also satisfies the other objectives for the instances without shifters Figure 5 and with shifters Figure 6. Each point on the following plots represents the average value of 20 instances.

In Figure 4 all instances were solved to optimality except the instances with shifters in Texas and California for which the time limit was exceeded. For these regions, the solutions displayed are within a relative optimal gap of 5% and 1% respectively. For all studied regions and all objectives, Figure 4 shows the benefits of high numbers of shifters in the scheme compared to instances with no shifters. The comparison is measured in term of assigned passengers, assigned users and driving distance for the three objectives respectively. For these regions, the solutions displayed are within a relative optimal gap of 5% and 1% respectively. Each point on the plots represents the average value of 20 instances.

For all the objectives and all the regions, the difference between the solution quality of instances with shifters and instances without shifters linearly increases to the advantage of shifters. When maximising the number of assigned passengers the solutions with shifters assign up to one third more passengers in California's

cluster, and approximately twice the number of passengers in Norway and Texas. When maximising the number of assigned users the gain in term of assigned users also increases by up to 45%, 40%, and 35% in Norway, Texas and California. Similarly, when minimising the total driving distance, the presence of shifters saves up to 6000km in Norway and Texas for 1000 and 500 users respectively while in Calfornia up to 18000km can be saved for 1700 users. The kilometers saved represent a reduction in driving distance of 30% in Norway, 50% in Texas and 38% in California. We have seen that shifters fix the critical imbalances observed in the real datasets of ridesharing trip adverts. They also represent the expected behaviour of drivers in case of congestion peaks, or in response to limitations on car usage enforced by public authorities. Although the objective values show a strong improvement (Fig. 4), it does come with price to pay in term of computing time, as will be shown in later results. Before that, we compare the quality of the solutions returned by the different models in terms of the other objectives.

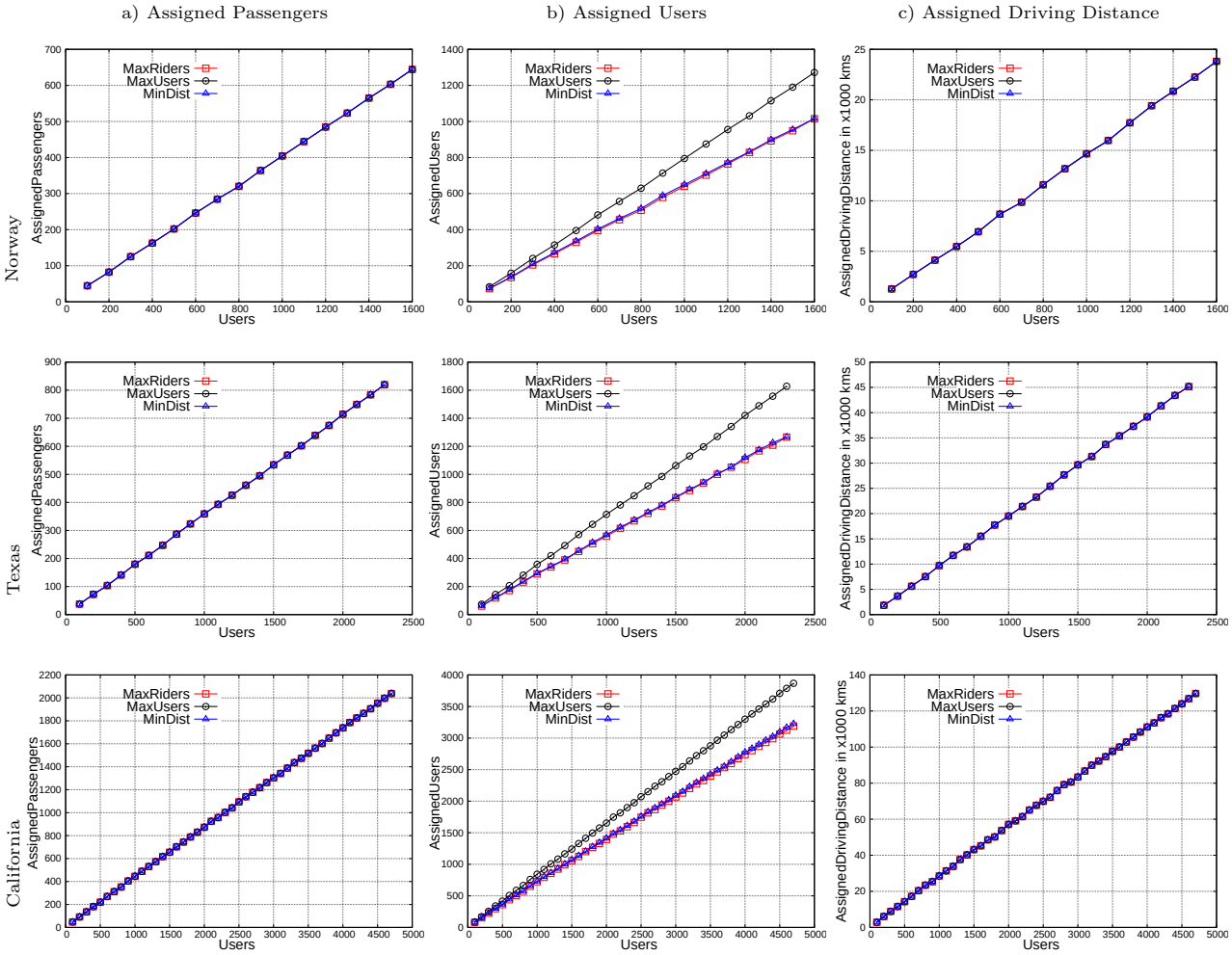

Figure 5: Comparison of the solutions (without shifters) returned by the three objectives MaxRiders, MaxUsers, MinDist in term of assigned passengers, assigned users, driving distance

We compare in Figure 5 the three objectives: a) maximising the number of assigned passengers (MaxRiders), b) maximising the number of assigned users (MaxUsers) and c) minimising the total driving distance (MinDist) in term of assigned passengers, assigned users and driving driving distance for all the regions, when solving instances without shifters (benchO). For all instances, we are able to compute optimal solutions within the time limit of 20 minutes.

We first observe that all objectives assign the same number of passengers (first column). Here, it is important

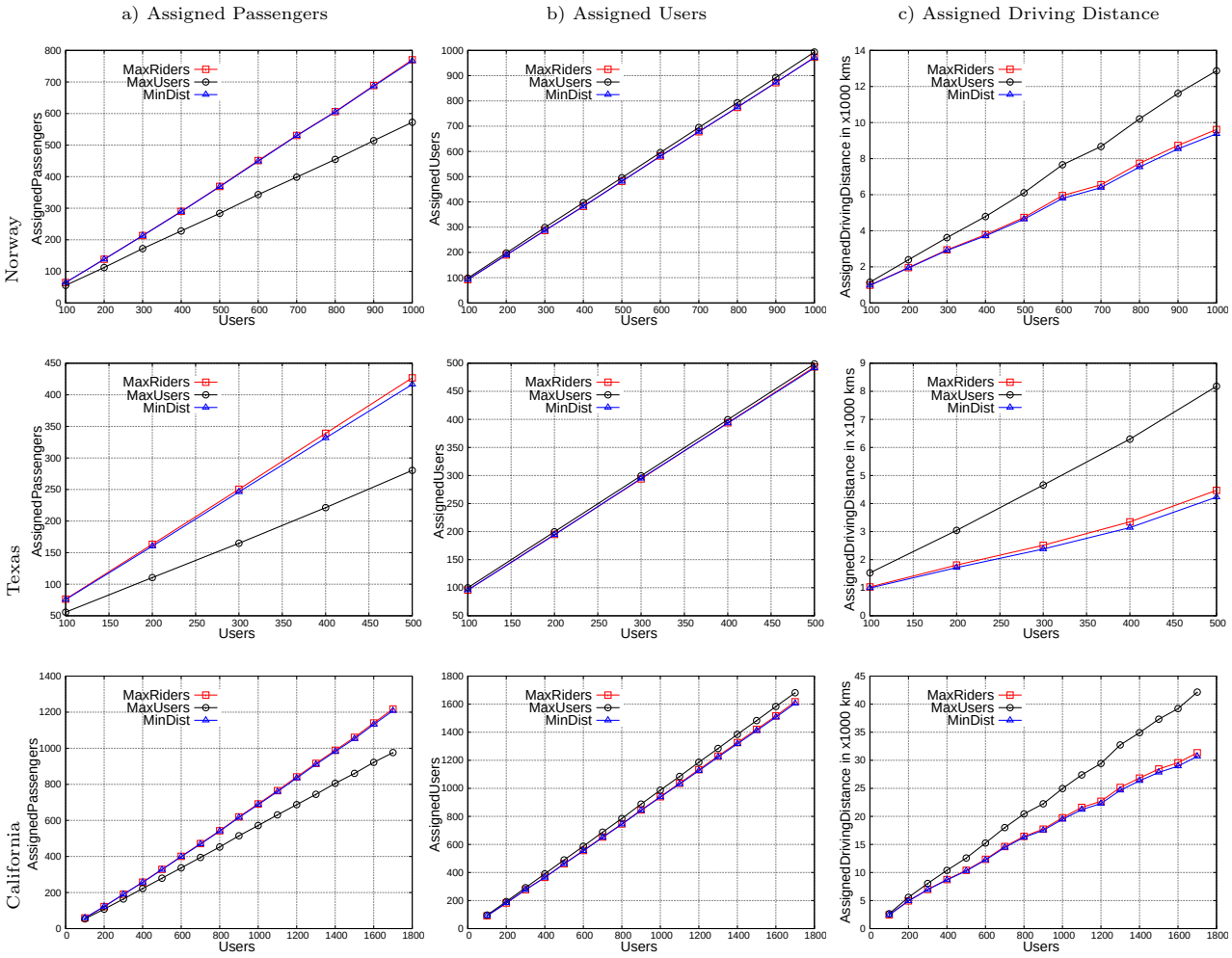

Figure 6: Comparison of the solutions (with shifters) returned by the three objectives MaxRiders, MaxUsers, Mindist in term of assigned passengers, assigned users, driving distance

to recall the noticeable imbalance playing in favour of drivers in the collected datasets of advertised trips. As a result, in the optimal solutions, almost all riders can be assigned to a car. This imbalance also influences the solving time of the different models making the ridesharing instances easy enough to be solved in real-time. We then observe that all models generate similar driving distances (third column). This is due to both the imbalance between riders and drivers and the absence of shifters. As before, all the pure riders can be assigned as passengers. Moreover, no driving distance can be saved from pure drivers, as they will drive anyway. The last observation is that maximising the number of assigned users satisfies noticeably more participants than the other objectives. Having noticed before that the objectives assign the same number of passengers, the difference in the assigned users can only come from assigned drivers. Implicitly, maximising the number of assigned users distributes assigned passengers over the drivers' cars. The other objectives do the opposite, and do not try to distribute passengers over cars even if some drivers may have no passengers.

In summary, because of the noticeable imbalance in favour of drivers, maximising the number of passengers or users, or minimising the total driving distance generates similar numbers of passengers and similar driving distances. The solution of the objectives differs in term of assigned users. The objective maximising the number of users will implicitly try to distribute passengers over cars.

In the Figure 6 we compare to what extent satisfying one objective optimally also satisfies the other objectives

for the instances with shifters (benchS). Here the Norway instances were solved to optimality, while the Texas and California instances were solved until an optimality gap of up to 5% and 1% was reached. In contrast to the no-shifters case, the solutions to the three models all assigned similar numbers of users (column b)), but the MaxUsers objective assigned fewer passengers than the other two objectives (column a)). For this case, as shown in Table 3, compared to Table 2, the proportion of pure drivers drastically decreases in favour of shifters. As a result, most of the remaining users that have to be assigned are shifters. In the expression of MaxUsers (12), a shifter assigned as a driver receives the same reward as a shifter assigned as passenger. For MaxPassengers and MinDistance, however, the objectives will prefer to assign shifters as passengers, unless the vehicles can be fully packed with passengers. This common behaviour shared by those objectives also explains why maximising the number of participants saves almost as many driving kilometres than the objective that explicitly minimise the distance and vice versa. Even if these two objectives are close in term of solution quality, in the next section we show that they differ in term of solving time. Note that MaxUsers saves less driving kilometres than the other objectives. As discussed before, maximising the number of users will implicitly try to distribute passengers over as many cars as possible, and thus more cars and more kilometres, are driven. In presence of shifters, the other objectives MinDistance and MaxPassengers do the opposite. They prefer to gather shifters into cars as passengers and save driving distance by removing cars from the roads.

9 Solving Time Performance

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 20 runs executed on a machine having 12 cores of 2.50GHz and 64GB of RAM, using the CPLEX solver. We compare for each objective the total CPU time for solving the ridesharing instances by the three formulations. M-TimeWindow represents the initial formulation encoding consistent time windows between drivers and riders along the drivers' paths. M-RiderSets represents the first reformulation based on time consistent sets of riders for each driver. M-IntendedPathTime represents the formulation that explicitly encodes a possible non-waiting path along the driver route.

9.1 Optimal ridesharing plans within 20 mins

Figure 7 compares the time performance on the problem instances without shifters. The main observation is that two reformulations, M-RiderSets and M-IntendedPathTime, are noticeably faster than the initial encoding M-RiderSets. They solve all three ridesharing problem types for all regions in a small number of seconds. For both models, the time performance grows slowly and steadily and never takes more than a few seconds even for the largest instances in California and the more dense instances in Texas. For the largest instances of each region, the observed mean time to solve the instances to optimality is up to 10 seconds and never exceeds 20 seconds. M-RiderSets slightly outperforms M-IntendedPathTime for Norway and California, while the two formulations show similar time performance for Texas. M-TimeWindow, the initial model, takes up to 10 minutes to solve the larger Texas instances. The reformulations M-RiderSets and M-IntendedPathTime are an order of magnitude faster.

Figure 8 compares the time performance on the instances with shifters. As can be seen by comparison with Figure 7, the presence of shifters makes the combinatorial problem much harder to solve, and reduces the size of the instances able to be handled in reasonable time down from thousands of users to a few hundred users. Maximising the number of passengers shows the most significant impact. For Texas, the region with the more dense feasible-match graph, all three formulations are only able to solve to optimality up to 100 users before hitting the timeout. In Norway and California the optimal is found up to 200 and 500 users. The M-IntendedPathTime formulation is fastest, but exceeds the time threshold for the same number of users as the original model M-TimeWindow. We note that, in California cluster, the second formulation,

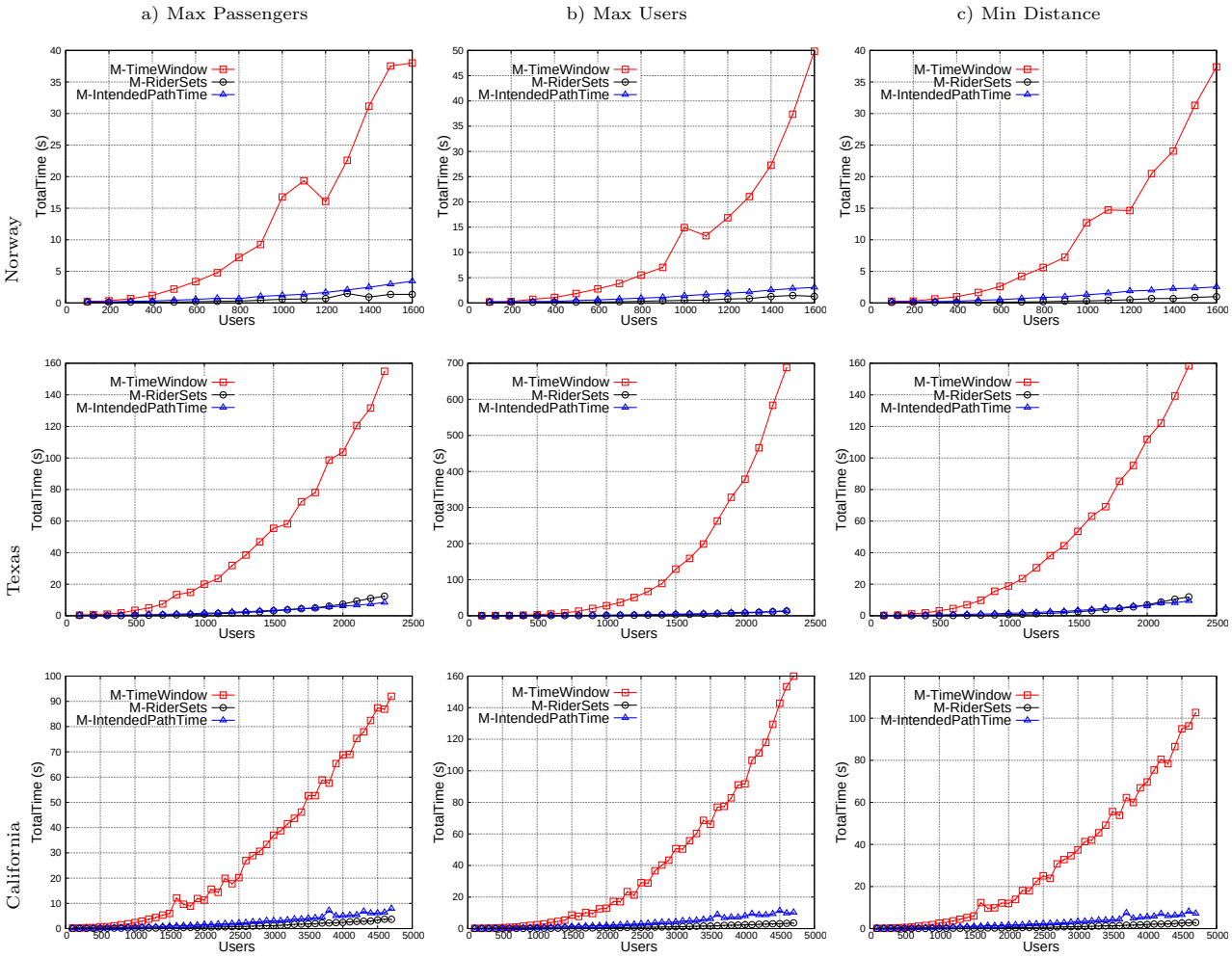

Figure 7: Total CPU time for solving optimally the intances (without shifters)

M-RiderSets, appears to be poorer than the original model, reaching the time threshold before solving for 500 users. In M-RiderSets the maximal sets of time-consistent of riders are encoded by a set of conflicts identifying the riders that cannot share the same trip. Adding these conflicts in the problem formulation allow faster convergence to optimal as shown in next section. However, after a longer solving time the quality of the solutions does not improve. M-IntendedPathTime slightly improves the quality of its solutions until a certain limit.

When minimising the driving distance (column c)), the time performance improves, with the best formulation (M-IntendedPathTime) solving to optimality instances with up to 200, 500, and 1300 users, within 200 seconds, for the ridesharing clusters from Texas, Norway and California respectively. In each cluster, M-IntendedPathTime is able to handle between 50% and 100% more users than the other two models.

For maximising the number of users (column b), the results are different, and all three models scale much better, handling over a thousand users in each region before timing out. This change in behaviour is consistent with the results from the previous section, where maximising passengers and minimising distance seemed to produce similar quality solutions by either metric, but maximising users was different. The original model (M-TimeWindow) is now noticeably poor, with the third model (M-IntendedPathTime) offering up to an order of magnitude improvement, and M-RiderSets being inbetween. M-IntendedPathTime is able to handle over 4500 users for California without timing out, returning optimal solutions in under 30 seconds.

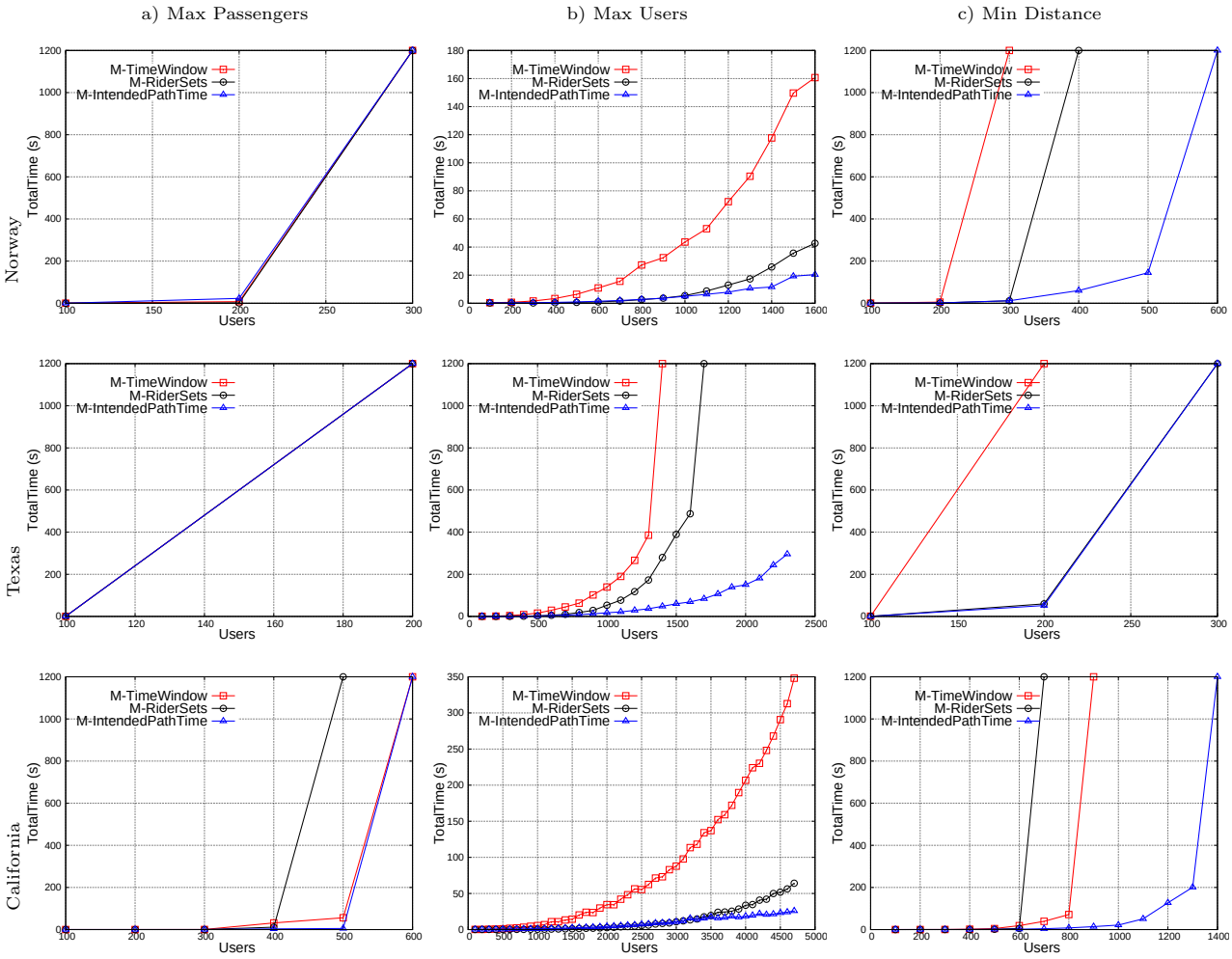

Figure 8: Wall CPU time for solving optimally the intances with shifters (benchS)

9.2 Optimized ridesharing plans within 5 mins

In the previous section, we saw that for maximising the number of passengers and for minimising the total driven distance, when there are flexible drivers (i.e. shifters), even the best model was unable to solve larger problems to optimality within 20 minutes. In this section, we examine the optimality gap (the gap between the best solution so far and the upper or lower bound), as an indication of how close the returned solutions are to the optimal result, after a fixed time period of 5 minutes. The results on the three regions, for the three objectives, for each model formulation, are shown in Figure 9. To compare performance on minimisation and maximisation problems, we measure the results on a consistent scale between 0 and 1, by computing the relative optimality gap as:

$$RelativeOptimalGap = \frac{|BestBound - BestSolution|}{max(BestBound, BestSolution)}$$

The value $BestBound$ (resp. $BestSolution$) corresponds to the best upper or lower bound (resp. best solution) found so far during the search. When $RelativeOptimalGap = 0$ the optimal solution has been found. However when $RelativeOptimalGap$ approaches 1, either the best solution or the best bound or both remain far from the optimal. When a models fails to find any solution with the time-limit, we report the optimality gap as 1.

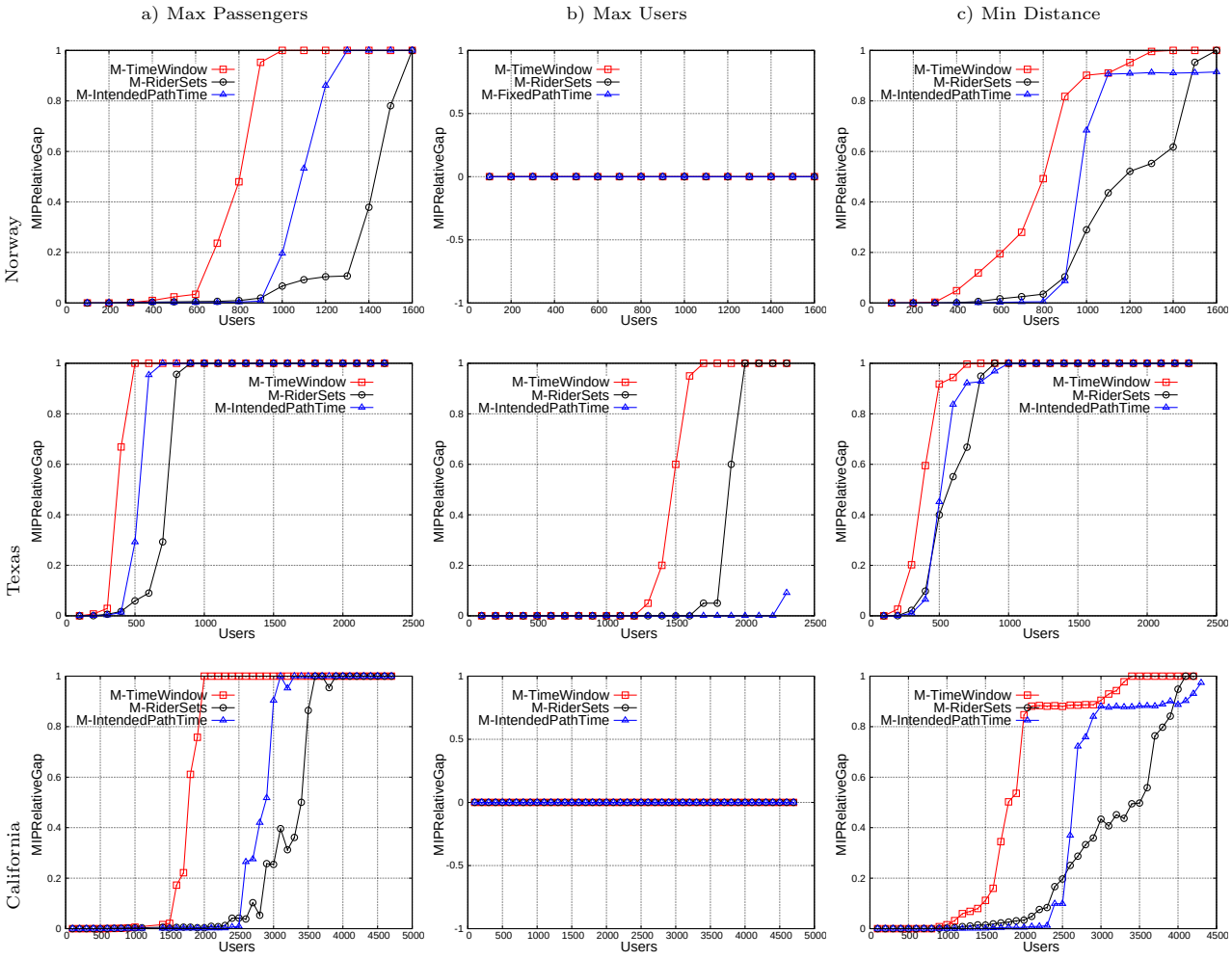

Figure 9: Relative optimality gap observed after 5 mins when solving the intances of benchS (with shifters)

As noted before, when maximising the assigned users, all formulations find the optimal solution quickly, and so the optimality gap is 0, except for the Texas region, where M-TimeWindow and M-RiderSet timeout after 1600 and 2000 users. When maximising the number of passengers, all models are able to report results, within 300 seconds, for significantly larger problems than those able to be solved to optimality within 1200 seconds (as was shown previously in Figure 8 column a)). The best formulation is now M-RiderSets, which is able to find solutions within 10% of the optimal for California for 2700 users, and returns solutions with 2% of the optimal for up to 2000 users. The results for minimising the driving distance(column c)) are similar, although the best model now varies between M-RiderSets and M-IntendedPathTime, depending on the desired optimality gap. In particular, for California, M-IntendedPathTime is returning solutions that are under 1% above the optimal up to 2300 users.

In summary, when the ridesharing problems include shifters, the three different problem types are all much harder to solve. When maximising the number of passengers or minimising the driving distance the best formulation is only able to solve to optimality instances with a few hundred of users in reasonable time. The response time and the size of the instances able to be solved significantly improve when maximising the number of assigned users. For each of the objectives, the formulation based on fixed path time encodings, M-IntendedPathTime, outperforms the other formulations, sometimes being at least an order of magnitude faster than the other formulations. In cases where we are required to return solutions within

a more limited time frame, we see that the models can return close to optimal solutions for much larger problem instances. In particular, for maximising the number of passengers or minimising the total distance, the M-IntendedPathTime model is able to return solutions with objective values within 1% of the optimal for each of the three objectives, for problems of up to 400 users in Texas, up to 900 users in Norway, and up to 2300 users in California, within a fixed time limit of 300 seconds. For larger optimality gaps, the M-RiderSets model is able to handle larger problem instances before timing out. For maximising the number of users, all models are able to solve to optimality within 5 mins for Norway and California for 1600 and 4400 users respectively, and M-IntendedPathTime is able to solve to optimality in Texas for up to 2200 users.

10 Limit and discussion

The methodology presented in this paper shows how to integrate spatio-temporal constraints, observed along drivers' paths, within two new ridesharing problem formulations. Even if the experiments show that both formulations are able to solve faster different realistic schemes satisfying global sustainable objectives, the problem reformulations are not equivalent with respect to their scalability capability and their capacity of generalisation.

From a scalability viewpoint, the first rewriting method results in a set of assignment rules derived from spatio-temporal constraints encoding the drivers' paths. In the case of an exceptional increase in the trip demand, during an extended time period, for a specific geographical area traversed by drivers, the number of conflicting assignments between pairs of riders may grow exponentially. This is not the case of second formulation that is based on the encoding of the locations traversed by the drivers rather than conflicting trips of riders. This remark may impact the solving performance for very specific scenarios. However the relative gain in performance has to be balanced with the fact we are in a context of ridesharing that remain hard to solve.

From a generalisation viewpoint, since the second formulation is a direct encoding of the spatio temporal properties observed along the drivers' intended path, it cannot be applied to solve more general pick-up and delivery scenarios of vehicle routing problems where drivers follows a flexible path. It is not the case of the method presented for the first reformulation that can be extended to handle flexible drivers' routes. In this case, the conflicting assignments rules among riders, or more generally, the conflicting assignment among deliverable objects, should not be inferred only from one path but a set of paths. In this context, the derivation of assignment rules may required a greater computational effort.

11 Conclusion

Ridesharing schemes, where people requesting transport are matched with drivers offering seats in their cars in existing planned journeys, are now well established in many city regions worldwide. The schemes offer many benefits to society, including increased access to travel services, reduced costs, fewer cars, and lower pollution and emissions. However, the schemes typically work in a real-time and interactive fashion, either leaving service discovery up to users, or offering real-time matches based on stated or inferred user preferences. Few schemes offer system-wide optimisation, and, indeed, clear objectives are often not clearly expressed. The main contribution of this work shows computationally efficient formulations of ride-sharing constraints. We show that the proposed encoding is able to tackle different ride-sharing schemes at the intent of different stakeholders, users, deployed applications, developers. By solving efficiently system wide ride-sharing schemes, we narrow the gap between research and deployed applications. We have focused on system-wide optimisation with sufficiently fast response times to be used in dynamic ridesharing systems.

We have proposed a core combinatorial problem, and presented three different ride-sharing schemes, capable of representing different social, environmental and sustainable goals. Maximising the number of assigned passengers addresses the problem of finding rides for users to increase access to transport, and in cases of of

exceptional events such as public transit disruption, sport events, or climatic events. Maximising the number of assigned users is a long term objective for maintaining the viability of deployed ridesharing applications. Minimising the driving distance addresses the ecological impact of driving on our city regions. We have developed mixed integer programming models for these problems, and we have developed two different reformulations, aimed at speeding up computation for difficult problem instances. We have developed two linked suites of benchmark problem instances, extracted from real-world ridesharing datasets from three city regions, and we evaluate all of our models and objectives on instances from these benchmarks of increasing scale. We have shown that the presence of flexible drivers, who are willing to act as passengers, improves the performance of all objectives in all regions. One of the novelties of our study shows to which extent satisfying one objective also satisfies other objectives. We show that, for suitable parameter choices, and for schemes with flexible drivers, maximising passenger numbers and minimising driving distance are compatible objectives, and that each one approximately optimises the other. Minimising the driving distance allows shorter times to prove optimality, while maximising passengers tends to approach the optimal values more quickly, but takes longer to complete the search. Maximising the matched users shows different results; it is faster to find and prove optimality, and so can handle larger problems, but does decrease the metrics for the other objectives.

The original model handles thousands of users in minutes for problems with no driver flexibility, but does not scale when we introduce flexible drivers. The two reformulations are considerably faster on these more difficult problems, showing an order of magnitude improvement over the original model. When we impose a time constraint of 5 minutes, and inspect the solutions, we show that the reformulated models achieve an optimality gap of below 2% and 5% for the two most difficult regions. These results demonstrate that fast near-optimal solutions are feasible for global system-wide optimisation of realistic size ridesharing schemes.

Further work is still required to exploit the full potential of such schemes. Our scheme assumes that drivers follow their intended path and imposes a schedule, but allows participants to modify the details of the route and schedule. Further development is required to include alternative routes directly into the optimisation model, as route flexibility should serve to improve the ability to serve more participants. Our scheme allows re-optimisation every few minutes, and so is to some extent robust to uncertainty and to breaks in the contract between drivers and passengers. However, more work is required to incorporate explicit models of uncertainty and risk, and to allow optimisation for robustness as well as the existing objectives. In addition, the scheme should be extended to allow transfers, both for improving the initial solutions, and for adapting to changes as the travel proceeds. Our models incorporate weights which represent the importance to the stakeholders of serving different types of participant, and also represent the probability that participants who are denied service will resort to driving outside the scheme. We have only explored uniform weights for pure passengers; the models and evaluation should be extended to allow different weights per passenger, to establish the impact of different weighting factors. Finally, we have compared the impact of different objectives, and we have shown that some are approximately interchangeable, but that the policy of maximising service to all types of participant has a negative impact on the more direct objectives. Maximising all served participants is intended to achieve longer term viability of the schemes, so this apparent reduction in performance may be offset by longer term growth in participant numbers and thus overall improvements in the overall objectives. This would require more extensive and perhaps longitudinal study on real-world deployments and on social attitudes to ridesharing, and incentives and barriers to participation.

12 Acknowledgment

This work has been supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289, Insight-Centre for Data Analytics, and Enable. The datasets were provided by Carma Technology Limited (http://www.gocarma.com/)

References

Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. European Journal of Operational Research, 223(2):295 – 303.

Agatz, N. A., Erera, A. L., Savelsbergh, M. W., and Wang, X. (2011). Dynamic ride-sharing: A simulation study in metro atlanta. Transportation Research Part B: Methodological, 45(9):1450 – 1464.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, 114(3):462–467.

Armant, V. and Brown, K. N. (2014). 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.

Armant, V., Horan, J., Mabub, N., and Brown, K. N. (2015). 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.

Attanasio, A., Cordeau, J.-F., Ghiani, G., and Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Comput., 30(3):377–387.

Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. European journal of operational research, 202(1):8–15.

Coltin, B. and Veloso, M. (2014). Ridesharing with passenger transfers. In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3278–3283. IEEE.

Cordeau, J.-F. and Laporte, G. (2007). The dial-a-ride problem: models and algorithms. Annals of Operations Research, 153(1):29–46.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. Transportation Research Part B: Methodological, 57(C):28–46.

Herbawi, W. and Weber, M. (2012). The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm. In Evolutionary Computation (CEC), 2012 IEEE Congress on, pages 1–8. IEEE.

Kamar, E. and Horvitz, E. (2009). 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. Morgan Kaufmann Publishers Inc.

Knapen, L., Hartman, I. B.-A., Keren, D., Cho, S., Bellemans, T., Janssens, D., Wets, G., et al. (2015). Scalability issues in optimal assignment for carpooling. Journal of Computer and System Sciences, 81(3):568–584.

Manna, C. and Prestwich, S. (2014). 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.

Schilde, M., Doerner, K. F., and Hartl, R. F. (2011). Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. Comput. Oper. Res., 38(12):1719–1730.

Simonin, G. and O'Sullivan, B. (2014). 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.

Stiglic, M., Agatz, N., Savelsbergh, M., and Gradisar, M. (2015). The benefits of meeting points in ride-sharing systems. Transportation Research Part B: Methodological, 82:36–53.

Stiglic, M., Agatz, N., Savelsbergh, M., and Gradisar, M. (2016). Making dynamic ride-sharing work: The impact of driver and rider flexibility. Transportation Research Part E: Logistics and Transportation Review, 91:190–207.

Wang, Y., Kutadinata, R., and Winter, S. (2016). Activity-based ridesharing: increasing flexibility by time geography. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, page 1. ACM.

Wu, C., Shankari, K., Kamar, E., Katz, R., Culler, D., Papadimitriou, C., Horvitz, E., and Bayen, A. (2016). Optimizing the diamond lane: A more tractable carpool problem and algorithms. In Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on, pages 1389–1396. IEEE.

Yousaf, J., Li, J., Chen, L., Tang, J., Dai, X., and Du, J. (2012). Ride-sharing: A multi source-destination path planning approach. In Thielscher, M. and Zhang, D., editors, Australasian Conference on Artificial Intelligence, volume 7691 of Lecture Notes in Computer Science, pages 815–826. Springer.