Maximising the number of participants in a ride-sharing scheme: MIP versus CP formulations

Vincent Armant and Nahid Mahub and Kenneth N. Brown
Insight Centre for Data Analytics
School of Computer Science and IT
University College Cork
Cork Ireland

***Abstract*—**

Ride sharing schemes aim to reduce the number of cars in congested cities, while providing the participants with a cheaper alternative to solo driving. To ensure a ride-sharing scheme thrives, it is important to maintain a high participation rate. This requires an adequate balance between drivers and riders. and thus ride matches should be proposed which maximize the number of participants. Different variants of the ride sharing problem have been solved using mixed integer programming. In this paper, we introduce a constraint programming formulation for the problem that uses cumulative constraints with dependencies between trip times. In experiments based on collected trip schedules from four different regions, the constraint model outperforms the MIP model. However, when we change the problem by assuming all drivers have flexible roles, the MIP model allows faster solution times than the CP model.

INTRODUCTION

Road traffic is one of the main generators of carbon emissions, and traffic congestion is a significant contributor to pollution around major cities and urban areas. Partly motivated by these issues, there has been a recently a strong growth in ride-sharing schemes (e.g. Blabla car[1], Carma[2], Lyft[3], Sidecar[4], Uber[5]), where participants post details of intended trips, and the system then proposes possible matches between drivers and prospective passengers. As more matches are agreed, the number of car journeys decreases, and the total driven distance also decreases, helping to reduce congestion, emissions and energy consumption. Consequently, ride-sharing schemes are considered a benefit to society, but are also revenue generators for system operators. To tackle this ride sharing-problem, recent approaches in the literature have proposed different formulations aimed at enhancing the user experience in ride-sharing schemes. As shown in [5], maintaining both a high participation of commuters and an adequate balance between the number of potential passengers and prospective drivers is crucial for a successful deployed ride-sharing system. Therefore, there is a need to efficiently identify the ride matches that maximize the number of participants and advertise those matches through incentives, reducing the cost of a ride shared trip.

For this purpose, we evaluate two ride-sharing problem formulations on data sets of ride share trips from four different regions. To do so, we introduce similar ride-sharing problem formulations using a mathematical integer programming (MIP) encoding and constraint programming encoding, where we are interested in maximizing the number of satisfied riders while satisfying time-window constraints on the ride segments. We first observe the characteristics of the feasible ride-match graph inferred from real user trip schedules collected in four different regions over six months. Then, we compare the modelling aspects of the two formulations, followed by the solving time for both models. The CP model is smaller and significantly faster than the MIP model. Then we consider a situation where drivers are encouraged to be flexible and act as riders if required, to improve participation. For this problem variant, the model sizes and solving times all increase. Neither model finishes within the time limit for the larger problems, but the MIP model now outperforms the CP model on the smaller regions.

RELATED WORK

The dial-a-ride problem has long been studied in the OR community [8]. 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 [7], [6]. The dial-a-ride drivers have no journey requirements of their own. For ride-sharing schemes [9], both the drivers and the riders have their own objectives. Specific schemes vary as to whether the drivers move to the riders locations or the riders move to and from the driver routes, and whether or not drivers take single or multiple riders on a trip. One extension includes

[1] www.blablacar.com
[2] https://carmacarpool.com
[3] www.lyft.com
[4] http://www.side.cr
[5] www.uber.com

participants known as *shifters*, who may either drive or ride as a rider [2]. Armant *et al.* [4] also include shifters, but also assume 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.* [5] assess the performance of a deployed ride-sharing scheme and evaluate the potential of persuading drivers to become passengers. The inferred constraint programming model used in that study models only part of the ride-sharing problem and it does not need to take into account time dependencies between the riders pick-ups and drop-off for the assessment. Computing an optimal matching is hard [3], and the complexity increases as the number of shifters increases. Kamar and Horvitz [10] model the problem as one of collaborative planning, where agents must balance competing goals. Yousaf *et al.* [14] model the problem as multi source-destination path planning, with a wide range of competing objectives including privacy and incentives. Schilde *et al.* [12] and Manna and Prestwich [11] 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 [13] 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.

RIDE SHARING PROBLEM IN CORK HARBOR

Due to the privacy of the data collected by our industrial partner we motivate our study and introduce the notations by presenting a toy ride-sharing problem in Cork Harbor.

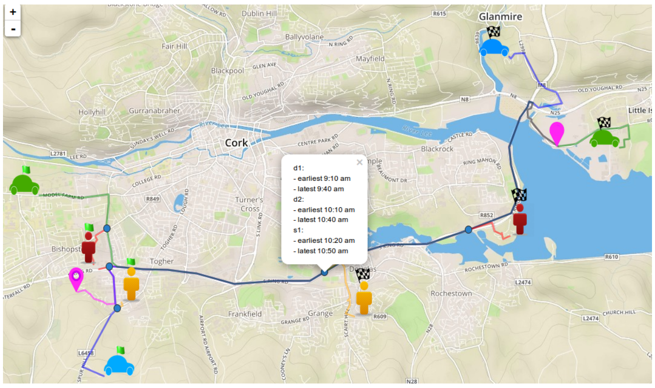

Figure 1: Example of ride-sharing problem Cork

In the Figure 1, we show two drivers' trip schedules drawn in green (d_1) and blue (d_2) lines with green-flag-car icon as starting location and check-flag-car icon as destination. We show two riders in red (r_1) and orange (r_2) having similar kind of flags to denote their start and end points. We are also showing a shifter (s_1) in pink-marker-icon who is willing to drive or becoming a passenger. Geographical constraints allow driver d_1 and driver d_2 to pick-up and drop-off r_1, r_2 and s_1 if the latter is willing to be a passenger. Similarly, if we only consider geographical constraints, if s_1 chooses to drive he will be able to share his ride with r_1 and r_2. In Figure 2 we show the overlapping of the time windows (i.e., earliest and latest start times) for each user trip schedule.

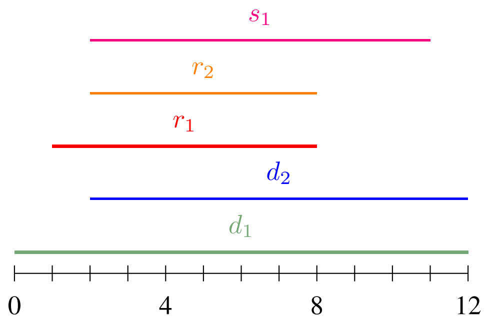

Figure 2: Time window constraints

When considering both geographical and the time window constraints, driver d_2 can no longer pick-up rider r_1. The intersection of the geographical and the time window constraints is shown by feasible match graph in Figure 3. In Figure 1 we show in the bubble the possible time windows for the drop-off of r_2 for each of its feasible rides.

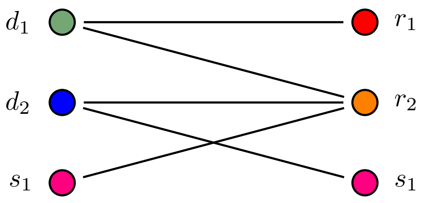

Figure 3: Feasible match graph

In this study, our goal is to find an optimal assignment of riders to drivers' car that maximizes the number of passengers.

NOTATIONS AND INFERRED FEASIBLE RIDE-MATCH GRAPH

To describe the trip schedules and the parameters inferred from the history of advertised trip schedules of four regions during a period of 6 months, we introduce the following notation. D denotes the set of possible drivers, R the set of possible riders, and $U = D \cup R$ the set of all users. S represents the set of shifters i.e., the drivers that are willing to change role. A trip schedule is a tuple $ts_u = (et_u^{start}, lt_u^{dest}, l_d^{start}, l_d^{dest})$ describing for the user u his inferred earliest start time et_u^{start}, his latest arrival time lt_u^{dest}, his start location l_u^{start}, and his destination l_u^{dest}. $TS = \{ts_{u_1}, \ldots, ts_{u_n}\}$ denotes the set of users' trip schedules sent to the system. q_d represents the car capacity of $d \in D$.

To infer the time and geographical constraints, we use Open Street Map data to deduce minimal path distances and times between two locations. $L = \{l_1, \ldots, l_n\}$ denotes the set of road node locations identified by their GPS coordinates. A path $\pi = (l_i, \ldots, l_j)$ is an ordered list of locations, and $time(\pi)$ (resp. $dist(\pi)$) returns the driving path time (resp distance) for π. The path $\pi^{*}_{l_i,l_j}$ (resp. $\pi^{\diamond}_{l_i,l_j}$) denotes a minimal time (resp. distance) path from l_i to l_j. The path $\pi^{\diamond}_{l_i,\pi'}$ represents the shortest path distance from the location l_i to the path π'. It is used to model the walking path from the riders' intended start (resp. intended destination) to a pick-up (resp. drop-off) location on a driver path. For the driver path π_d, $pred_{\pi}(l)$ denotes the predecessor of l in the path π_d, $pick_{\pi_d}(l)$ the set of rider trip schedules starting at the pick-up location $l \in \pi_d$, $drop_{\pi_d}(l)$ denotes the set of rider trip schedules ending at the drop-off location $l \in \pi_d$. For a driver trip schedule ts_d, π_d denotes the inferred driver path from start l_d^{start} to destination l_d^{dest}. For a rider trip schedule ts_r, m_r^{pick} denotes the inferred maximal path distance r is willing to walk from his intended start l_r^{start} to a pick-up location l_r^{pick} on the driver path π_d. Similarly m_r^{drop} denotes the inferred maximal path distance the rider is willing to walk from a drop-off location l_r^{drop} to his destination l_r^{dest}. For the sake of simplicity, for each rider r, we consider his earliest pick-up time equal to his earliest start time et_r^{start}, and his latest drop-off time equal to his latest arrival time lt_r^{dest}.

Given the above notation, we define the feasible matches relaying both on the users' inferred path constraints and the users' inferred time constraints.

Definition 1 (feasible ride-match): A driver's trip schedule and a rider's trip schedule, ts_d and ts_r, $d \neq r$, represent a feasible ride match if:

1) their inferred time windows tw_d, tw_r are consistent with the rider's pick-up and drop-off time:
   a) $lt_d^{dest} - et_r^{start} > time(\pi^{*}_{l^{pick},l^{drop}})$, the time interval between the driver's latest arrival and the rider's earliest start is greater than the fastest path from the rider's inferred pick-up to his inferred drop-off, or,
   b) $lt_r^{dest} - et_d^{start} > time(\pi^{*}_{l^{pick},l^{drop}})$, the time interval between the earliest driver's start and the latest rider's arrival is greater than the fastest path from the rider's inferred pick-up to the inferred drop-off.
2) The expected driving path intersects the rider's possible pick-up and drop-off points.
   a) $dist(\pi^{\diamond}_{l_r^{start},\pi_d}) < m_r^{pick}$, the shortest path distance between the rider's intended start and the expected driver's path is lower than the maximal distance that the rider is willing walk to reach the pick-up location.
   b) $dist(\pi^{\diamond}_{l_r^{start},\pi_d}) < m_r^{drop}$, the shortest path distance between the rider's intended destination and the expected driver's path is lower than the maximal distance that the rider is willing to walk to reach the drop-off.

Parameters such as the maximal distance a rider is willing to walk to join the driver's path or the users' time window appearing in the above definition and describing the users' behavior have been processed through the analysis of a set of successful ride matches, observed between riders' and drivers' trip schedules that have led to a effective ride-sharing [5]. Given a set of unmatched trip schedules observed in each region, we discover the set of feasible ride-matches and build the feasible match-graph defined as follows :

Definition 2 (inferred feasible ride-match graph G): Given a set TS of trip schedules, $G = (TSD, TSR, E)$ is an inferred feasible ride-match graph induced by TS if:

1) $TSD \subseteq TS$ is a set of drivers' trip schedules,
2) $TSR \subseteq TS$ is a set of riders' trip schedules,
3) $TSD \cup TSR = TS$ s.t. $TSD \cap TSR = TSS$ is the set of shifter trip schedules,
4) $\forall (ts_d, ts_r) \in E$, (ts_d, ts_r) is feasible ride match.

G is the input parameter of the constraint programming and the mathematical integer formalizations.

Mathematical Integer Programming formulation

In the MIP formulation of the ride-sharing problem described in [4] the objective is to minimize the total driven distance by the drivers. In this work, we are interested in maximizing the number of participants. Apart from the different objectives, and in addition to taking as input a feasible ride match graph inferred from real data, in the new MIP formulation we are interested in returning a solution in which each matched rider is given a time window within which he or she can be picked up and dropped off by the corresponding matched driver, rather than a single time point.

In the MIP formulation, the decision variable y_{ts_d,ts_r} represents a ride match variable between a driver trip schedule $ts_d \in TSD$ and a rider $ts_r \in TSR$. When $y_{ts_d,ts_r} = 1$ in a solution, d and r are proposed to share a ride , $y_{ts_d,ts_r} = 0$ otherwise. The decision variable $t^{early}_{ts_d,l}$ represents the earliest departure time of the driver d from the location l while the decision variable $t^{late}_{ts_d,l}$ denotes the latest departure from the location l. From a proposed earliest and latest departure time from each location visited by a driver, one can easily deduce the earliest and latest pick-up (resp. drop-off) time for feasible rider matches. Indeed, the original rider time window at a pick-up location can be updated with the proposed driver time window at the location. Moreover, modelling directly the earliest and the latest pick-up (resp. drop-off) time for each rider is potentially prohibitive since we would like to introduce at least as many variables as edges in G as it is already the case for y_{ts_d,ts_r}.

In addition, the auxiliary variable x_{ts_s} represents the role of a shifter $ts_s \in TSS$ s.t. $x_{ts_s} = 1$ iff ts_s is proposed to be a driver trip schedule in a solution , $x_{ts_s} = 0$ otherwise. The

value of x_{ts_s}, entirely depends on the rideshare variables y_{ts_s,ts_r} and y_{ts_d,ts_s}. The auxiliary variable z_{ts_r} denotes a served rider $ts_r \in TSR \setminus TSS$ s.t. $z_{ts_r} = 1$ iff ts_r is proposed a match in a solution, $z_{ts_r} = 0$ otherwise. The value of x_{ts_r} entirely depends on the rideshare variables $y_{ts_d,ts_r}, \forall ts_d \in TSD$. The auxiliary variable $o_{d,v}$ denotes the car occupancy of driver $ts_d \in TSD$ when leaving the location $l \in \pi_d$. It also depends on the rideshare variables y_{ts_d,ts_r}, $\forall ts_d \in TSD$.
Our objective is to maximize:

$$\sum_{r \in TSR} z_{ts_r} \tag{1}$$

subject to:

$$(\sum_{(ts_d,ts_r)\in E} y_{ts_d,ts_r}) = z_{ts_r}, \quad \forall ts_r \in TSR \tag{2}$$

$$z_{ts_r} \leq 1, \quad \forall ts_r \in TSR \tag{3}$$

$$z_{ts_s} = 1 - x_{ts_s}, \quad \forall ts_s \in TSS \tag{4}$$

$$(\sum_{(ts_s,ts_r)\in E} y_{ts_s,ts_r} \geq 1) \Rightarrow (x_{ts_s} = 1), \\ \forall ts_s \in TSS \tag{5}$$

$$o_{ts_d,l} = o_{ts_d,l'} + \sum_{ts_r \in pick_{\pi_d}(l)} y_{ts_d,ts_r} - \sum_{ts_{r'} \in drop_{\pi_d}(l)} y_{ts_d,ts_{r'}}, \\ \forall ts_d \in TSD, \forall l = pred_{\pi_d}(l') \tag{6}$$

$$t^{early}_{ts_d,l} \leq t^{late}_{ts_d,l}, \quad \forall ts_d \in D, l \in \pi_d \tag{7}$$

$$y_{ts_d,ts_r} \Rightarrow et^{start}_r \leq t^{early}_{ts_d,l^{pick}_r}, \\ \forall (ts_d, ts_r) \in E \tag{8}$$

$$y_{ts_d,ts_r} \Rightarrow t^{late}_{ts_d,l^{pick}_r} \leq lt^{dest}_r - \pi*_{l^{pick}_r,l^{drop}_r}, \\ \forall (ts_d, ts_r) \in E \tag{9}$$

$$y_{ts_d,ts_r} \Rightarrow et^{start}_r + \pi*_{l^{pick}_r,l^{drop}_r} \leq t^{early}_{ts_d,l^{drop}_r}, \\ \forall (ts_d, ts_r) \in E \tag{10}$$

$$y_{ts_d,ts_r} \Rightarrow t^{late}_{ts_d,l^{drop}_r} \leq lt^{dest}_r, \quad \forall (ts_d, ts_r) \in E \tag{11}$$

$$t^{late}_{ts_d,l} + \pi^*_{l,l'} \leq t^{late}_{ts_d,l'}, \forall d \in D, \forall l = pred_{\pi_d}(l') \tag{12}$$

$$t^{early}_{ts_d,l} + \pi^*_{l,l'} \leq t^{early}_{ts_d,l'}, \forall d \in D, \forall l = pred_{\pi_d}(l') \tag{13}$$

$$y_{ts_d,ts_r} \in \{0,1\}, o_{ts_d,l} \in [0, q_d] \tag{14}$$

The aim is to maximize the total number of passengers in the solution (1). The objective only represents the matched riders. The constraints (2) force each rider to be a passenger of one of the feasible drivers. The constraints (3) force each rider to be a passenger of at most one driver. A shifter assigned to one driver as a passenger is a rider, otherwise a driver (4). A shifter picking up at least one passenger is a driver (5). When a driver leaves a location its car occupancy is equal to the difference between picked up and dropped off passengers plus the car occupancy of the previously visited location (6). A driver leaves a passenger's pick-up location, at least after the passenger's earliest departure time (8) and, at most before the passenger's latest departure time (9). Similarly, he leaves a passenger's drop-off location at least after the passenger's earliest arrival time (10) and at most before the passenger's latest arrival time (11). The time spent between two consecutive locations on a path is not less than the minimum time to travel between the two locations (13,12).

Constraint Programming formulation

In this section, we introduce a similar formulation of the ride-sharing problem based on a constraint programming formulation. This formulation extends the CP formulation used in [5] and proposes a more realistic model that aims at maximizing the total number of rider participants while modelling time dependencies between the pick-ups and drop-offs. As before, the time window for the pick-up and the drop-off of each passenger can be quickly computed from a solution. More specifically we use the well known cumulative constraint to describe the ride-sharing assignment problem as a resource allocation problem where the drivers' cars play the role of resources and feasible rides can be seen as a task that requires a time slot in the drivers' car. In addition to the classical scheduling problem, to tackle the possibility for drivers to becoming passengers, our formulation of the ride-sharing assignment problem introduces new constraints that allow drivers to shift roles and become riders. As in the MIP formulation $G = (TSD, TSR, E)$ is the input parameter of the constraint programming model. To differentiate the decision variable notation from the MIP model we use capital letter in the CP model. Each feasible match (ts_d, ts_r) in E is associated to a rideshare trip Y_{ts_d,ts_r} encoded as a collection of decision variables s.t.:

$Y_{ts_d,ts_r}.start$ represents the pick-up time of r,
$Y_{ts_d,ts_r}.end$ denotes the drop-off time of r,
$Y_{ts_d,ts_r}.duration$ denotes the rideshare trip duration,
$Y_{ts_d,ts_r}.presence$ denotes presence of the rideshare trip in the solution.

We model a served rider using Z_{ts_r} s.t. Z_{ts_r} equal 1 when the rider is allocated to exactly one of the feasible share rides Y_{ts_d,ts_r}. To assess the potential of a ride-sharing scheme, our objective is to maximize:

$$\sum_{(ts_r \in TSR} Z_{ts_r} \tag{15}$$

subject to:

$$Y_{ts_d,ts_r}.start \geq max(t^{early}_d + time(\pi*_{l^{start}_d,l^{pick}_r}), t^{early}_r), \\ \forall (ts_d, ts_r) \in E \tag{16}$$

$$Y_{ts_d,ts_r}.end \leq min(t_d^{latest} - time(\pi*_{l_r^{drop},l_d^{dest}}), t_r^{latest}), \quad \forall(ts_d, ts_r) \in E \tag{17}$$

$$Y_{ts_d,ts_r}.duration = y_{ts_d,ts_r}.end - y_{ts_d,ts_r}.start, \quad \forall(ts_d, ts_r) \in E \tag{18}$$

$$Y_{ts_d,ts_r}.duration \geq \pi *_{l_r^{pick},l_r^{drop}}, \quad \forall(ts_d, ts_r) \in E \tag{19}$$

$$CUMULATIVE(\{Y_{ts_d,ts_r}\}, q_d, \leq), \quad \forall ts_d \in TSD \tag{20}$$

$$ALTERNATIVE(Z_{ts_r}, \{Y_{ts_d,ts_r} | (ts_d, ts_r) \in E\}), \quad \forall ts_{ts_r} \in TSR \tag{21}$$

$$(Z_{ts_s}.presence \Rightarrow \neg Y_{tr_s,ts_r}.presence), \quad \forall(ts_s, ts_r) \in E \tag{22}$$

$$\begin{aligned} &StartBeforeStart(Y_{ts_d,ts_r}, Y_{ts_d,ts_{r'}}, \pi^*_{l_r^{pick},l'^{pick}_{r'}}),\\ &\forall tsd \in TSD, \forall l = pred_{\pi_d}(l'),\\ &\forall ts_r = pick_{\pi_d}(l), \forall ts_{r'} = pick_{\pi_d}(l') \end{aligned} \tag{23}$$

$$\begin{aligned} &StartBeforeEnd(Y_{ts_d,ts_r}, Y_{ts_d,ts_{r'}}, \pi^*_{l_r^{pick},l'^{drop}_{r'}}),\\ &\forall tsd \in TSD, \forall l = pred_{\pi_d}(l'),\\ &\forall ts_r = pick_{\pi_d}(l), \forall ts_{r'} = drop\pi_d(l') \end{aligned} \tag{24}$$

$$\begin{aligned} &EndBeforeStart(Y_{ts_d,ts_r}, Y_{ts_d,ts_{r'}}, \pi^*_{l_r^{drop},l'^{pick}_{r'}}),\\ &\forall tsd \in TSD, \forall l = pred_{\pi_d}(l'),\\ &\forall ts_r = drop_{\pi_d}(l), \forall ts_{r'} = pick_{\pi_d}(l') \end{aligned} \tag{25}$$

$$\begin{aligned} &EndBeforeEnd(Y_{ts_d,ts_r}, Y_{ts_d,ts_{r'}}, \pi^*_{l_r^{drop},l'^{drop}_{r'}}),\\ &\forall tsd \in TSD, \forall l = pred_{\pi_d}(l'),\\ &\forall ts_r = drop_{\pi_d}(l), \forall ts_{r'} = drop_{\pi_d}(l') \end{aligned} \tag{26}$$

The aim is to maximize the total number of served riders (15). The constraints (16) force each rideshare trip to start after the earliest rider start and the earliest driver arrival time at the rider's pick-up. Similarly, the constraints (17) force each rideshare trip to end before the latest rider arrival and the latest driver arrival time at the rider's drop-off. The duration of the rideshare trip is the difference between the end and the start (18) and it is greater than the rider shortest path (19). The cumulative constraints (20) restrict each driver's car occupancy to not exceed the number of available seats at any moment of the trip. The alternative constraints (21) enforce that exactly one Y_{ts_d,ts_r} rideshare trip has to be chosen for r to be a served rider, i.e., a passenger. In the successful case of the rider rideshare trip Z_{ts_r} is equal to the chosen rideshare Y_{ts_d,ts_r} otherwise the rider is not chosen. The constraints (22) state that a shifter assigned to be a rider does not drive. The last four constraints set up path time dependencies between starts and ends of rideshare trips. For each consecutive location l, l' visited in the driver's path the constraints (23) (resp. 24) force the time between rideshare startings at l and the rideshare startings (resp. endings) at l' to be greater than the shortest time between l and l'. Similarly, for each consecutive location pair l, l' visited in the driver path the constraints (25)(resp. 26) force the time between rideshare end at l and the rideshare start (resp. end) at l' to be greater than the shortest time between l and l'.

MODELLING AND SOLVING THE OPTIMIZATION PROBLEMS INDUCED BY THE FEASIBLE RIDE-MATCH GRAPH: G

In the first set of experiments we observe the characteristics of the feasible ride-match graph inferred from user trip schedules collected from four different regions. Then, we compare the modelling aspects of the constraint programming formulation (CP) and the mathematical integer programming formulation (MIP) previously introduced. Finally, we compare the solving time and the difference between the optimal solutions returned by both models. In [5] it has been shown that allowing drivers to change role drastically increases the number of possible ride matches. In the second set of experiments, we follow this heuristic, and as previously we analyze the characteristics and the time performance for building the feasible ride match graph, as well as the modelling and solving time performance and of the ride-sharing problem for the two formulations. In the following, the regions 1 and 2 represent collected trip schedules of western European countries during a period of 6 months while regions 3 and 4 represent collected trip schedules from western American states during the same period. We solved the constraint problem using the CP OPTIMIZER solver of IBM [1] while we solved the integer programming problem using CPLEX solver of the same distribution. For boeth solvers, we use the default parameters and search heuristics. The experiments were run on a machine with 2 processors of 2.5GHz, 12 cores, and 64 GB of memory, with a time limit of 1 hour for the solving. When the optimal solution has not been found within the time limit we recorded the best objective value reached so far.

Characteristics of the feasible ride-match graph: G per region

In Table I we show for each region, the characteristics of the feasible ride-match graph inferred from weekly user trip schedules collected for a specific day during a period of 6 months by our industrial partner. Regions are numbered by the increasing number of observed users. Region 1 and 2 are small instances ($\leq$ 1000 users) where ride-sharing is not yet well established while in region 3 and 4 the number of users (resp. $\geq$ 1500 and more $\geq$ 4000) indicate that the ride-sharing is a viable solution among the various transit modes. None of the users advertised weekly trip schedules in which they offered to be shifters. For all regions, the main characteristic is the imbalanced between the drivers and riders shown by the ratio of the number of riders to the number of drivers. In the worst case, region 2, there are at least two drivers for one rider. In this state, participants may lose interest in the ride-sharing scheme since most of them cannot be satisfied. This is why, in the second set of experiments we analyse the heuristic where drivers have an incentive to switch to being riders. In the inferred feasible match graph, the proportion of poorly connected users (i.e., users belonging to a connected component of G smaller or equal to two) is significant in region 1 (47%), region 2 (36%), and region 4 (26%) (cf line $|CC \leq 2|$). These connected components represent trivial instances and there is no need to formulate them in the ride-sharing problem. On the other hand, users have a high probability of belonging to the maximal connected component (cf line max $|CC|$) of Table I. The maximal connected component represents 24% of the users in region 1, 56% of the users in region 2, 84% of the users in region 3 and 70% of the users in region 4. In the last line of Table I we observe the cpu time required for building G. The building time starts from four minutes for the smaller instances and drastically increases to 1h38mins for the region 3 and reaches more than 4h28mins for the biggest instance. This total time prevents us from building the whole feasible ride match graph online. In the case of an incremental alternative, the average building time per user is 0.5 sec in the best case for region 1, and in the worst case we reach 3.7 sec for region 3 and region 4. This last metric supports the possibility of an incremental alternative for maintaining G.

Graph: G	region 1	region 2	region 3	region 4
# users	512	838	1578	4250
# drivers	308	545	917	2319
# riders	204	293	661	1931
# edges	849	2109	28353	26536
ratio $\frac{\#riders}{\#drivers}$	0.66	0.53	0.72	0.83
# $|CC \leq 2|$	240	308	164	1124
# max $|CC|$	123	470	1340	3016
building time	4m 10s	4m 33s	1h 38m 18s	4h 28m 11s

Table I: Characteristics of the feasible ride-match graph G per regions

modelling times of the ride-sharing problem induced by G

The the first three lines of Table II shows that the MIP formulation contains from 1.17 (region 3) to 3.2 (region 2) times more decision variables than the CP formulation. Similarly the MIP formulation contains from 1.5 (region 3) to 6.7 (region 2) times more constraints than the CP formulation. Then we compare the required encoding time from the input feasible ride match graph G while considering a discretization of time in minutes and in seconds in each of the formulations. The encoding time never exceed 7 seconds. Surprisingly using a discretization in seconds which implicates a larger domain for the time variable than a discretization in minutes is faster to encode for both formulations. Nevertheless, as expected, the CP formulation is always faster to encode than the MIP formulation since it requires fewer variables and fewer constraints. The CP modelling time ranges from two times faster than the MIP modelling time to over 5 times faster.

Models	region 1	region 2	region 3	region 4
Number of variables				
MIP	8427	21552	100656	196516
CP	2662	6543	85630	81041
Number of constraints				
MIP	12478	31780	179656	303501
CP	2136	4710	115066	88771
Encoding time when the models distcretize time in minutes				
MIP	711ms	1s 153ms	4s366ms	6s37ms
CP	197ms	283ms	2s672ms	2s882ms
Encoding time when the models distcretize time in secondes				
MIP	281ms	523ms	3s374ms	4s760ms
CP	54ms	104ms	574ms	839ms

Table II: Characteristics of the problem formulations induced by the feasible ride-match graph G

Solving times of the ride-sharing problem induced by G

In Table III we compare the solving times and the characteristics of optimal solutions observed for each formulation. To have a fair comparison of the different encodings we choose the default search strategy proposed by the CP and MIP solvers. The first lines show that the optimal number of served riders returned by each formulation only differs by at most three riders in the worst case. The almost equal numbers of served riders in the optimal solution show the similarity of the two models. Since different optimal solutions may exist, and both the resolution algorithms and the search heuristics may differ, the number of matched drivers found in the different formulation may differ. As expected, since there are no shifters in the input feasible ride match graph there are no shifters in the optimal solution. Concerning the solving time, we first observe the proportion of the time needed to solve the sub problem corresponding to the maximal connected component (cf line max CC) in the total time required for solving the MIP or the CP formulation with different heuristics of time discretization. For any formulations and for the larger instances (i.e., regions 3 and 4), we observe that almost all the time required for solving the global instance is spent solving the maximal connected component. For the smaller instances, solving the maximal connected component occupies at least 28% of the time. Moreover, for the largest instances, the MIP formulation discretizing the time in minutes is always solved faster than the MIP formulation discretizing the time in seconds. However, for the CP formulation and for the largest instances we observe that solving time remains more or less constant. These last observations highlight a relative independence of the CP encoding with respect to the size of the domain of the time variables which is not the case for the MIP encoding. This relative independence of the CP formulation with respect to the size of the domain of the time variable is confirmed by the superiority of the solving time of the CP formulations encoding time in seconds over the CP formulations encoding time in minutes. Finally, when we compare the time performance of the different encodings, Table III shows the CP formulations consistently outperforms the MIP formulations for all regions. In the worst case, solving the MIP encoding is at least 1.6 times slower than solving the similar CP encoding while in the best case solving the CP formulation is 8 times faster for region 3 and at least one order of magnitude faster for region 1.

CPU time	region 1	region 2	region 3	region 4
Number of matched riders				
MIP	106	206	567	1388
CP	106	207	566	1390
Number of matched drivers				
MIP	61	119	330	782
CP	64	109	280	646
Solving time when the models distcretize time in minutes				
MIP	568ms	1s 333ms	31s 620ms	30s 884ms
max CC	244ms	1s 253ms	31s 390ms	30s 634ms
CP	247ms	745ms	7s 486ms	13s 174ms
max CC	70ms	620ms	7s 379ms	12s 939ms
Solving time when the models distcretize time in secondes				
MIP	629ms	1s 313ms	58s 416ms	54s 1ms
max CC	306ms	1s 242ms	58s 175ms	53s 709ms
CP	54ms	775ms	7s 524ms	12s 765ms
max CC	21ms	620ms	7s 169ms	12s 480ms

Table III: formulation and solving times of the ride-match graph G

MODELLING AND SOLVING THE FEASIBLE RIDE-MATCH GRAPH G' ALLOWING DRIVERS TO CHANGE ROLE

In [5], the data analysis of real world ride-sharing instances has revealed an insufficient number of riders with respect to the number of drivers. This imbalance may lead the participants to lose interest in the scheme. To tackle this issue, in the second set of experiments, we analyze the heuristic where drivers have an incentive to becoming riders. As in the previous section, we first consider the characteristics of the feasible ride-match graph G' derived from G before comparing the modelling and the solving characteristics of the CP and MIP formulations.

Characteristics of the feasible ride-match graph G' per region

The only difference between Table I and Table IV is the number of shifters is increased to be the number of drivers. This heuristic not only increases the number of potential riders but also increases the number of edges of the new feasible ride match graph G' and also increases the building time by a factor of 2.

Graph: G'	region 1	region 2	region 3	region 4
# users	512	838	1578	4249
# drivers	308	545	917	2319
# riders	512	838	1578	4249
# shifters	308	545	917	2319
# edges	1556	3858	61520	43442
ratio $\frac{\#riders}{\#drivers}$	1.66	1.53	1.72	1.83
# $\|CC \leq 2\|$	197	217	121	766
# max $\|CC\|$	136	523	1386	3633
building time	9min22s	10min39s	3h45m38s	8h46m36s

Table IV: Characteristics of the inferred feasible ride-match graph G' per region

modelling times of the ride-sharing problem induced by G'

Similarly, in the Table V the heuristic allowing drivers to becoming riders also involves a cost on the modelling. With respect to the ride-sharing problems induced by the graph G, the problems induced by the graph G' increases by a factor of 1.2 to 2 in the number of variables, the number of constraints and the different encoding times. The CP formulation still requires a smaller number of variables, a smaller number of constraints and a faster encoding compared to the MIP formulations.

Models	region 1	region 2	region 3	region 4
Number of variables				
MIP	11738	30067	146288	270198
CP	4901	12062	185941	133305
Number of constraints				
MIP	18121	46218	296986	431433
CP	4834	11768	476396	197586
Encoding time when the models distcretize time in minutes				
MIP	982ms	1s 498ms	9s 59ms	9s 457ms
CP	224ms	543ms	3s 411ms	4s 411ms
Encoding time when the models distcretize time in secondes				
MIP	252ms	694ms	12s 424ms	8s 205ms
CP	155ms	185ms	1s 929ms	1s 787ms

Table V: Characteristics of the problem formulations of the feasible ride-match graph G'

Solving times of the ride-sharing problem induced by G'

However, the increase in the number of shifters does change the relative performance of teh two formulations in terms of solving time. Creating more shifters implicitly introduces new combinatorial variables that make the ride-sharing problem harder to solve. Table VI show that for the larger instances, i.e., regions 3 and 4, for both formulations the solvers hit the time limit and return their best solutions found so far with no optimality guarantee. For the smaller instances, i.e., regions 1 and 2, the MIP problem formulations are solved within 20 seconds while the solving of the CP problem formulations never terminate before the time limit.

CONCLUSION AND PERSPECTIVE

In this study, we introduce a MIP formulation and a CP formulation to maximize the number of riders served in a deployed ride-sharing scheme. Based on the user trip schedules collected from four different regions during a period of 6 months, we first observe the characteristics and the building time of the feasible match graph inferred from the collected trip schedules. We compare the modelling

CPU time	region 1	region 2	region 3	region 4
Number of matched riders				
MIP	198	420	57	2615
CP	201	428	1167	2586
Number of matched drivers				
MIP	89	152	22	756
CP	76	144	268	721
Number of matched shifters				
MIP	88	204	20	1257
CP	90	203	687	1232
Solving time when the models distcretize time in minutes				
MIP	1s535ms	10s 783ms	> 1h	> 1h
CP	> 1h	> 1h	> 1h	> 1h
Solving time when the models distcretize time in secondes				
MIP	2s57ms	17s 827ms	> 1h	> 1h
CP	> 1h	> 1h	> 1h	> 1h

Table VI: formulation and solving times of the ride-match graph G'

and the solving performance of two ride-sharing problem encodings under two hypothesis. The first hypothesis considers the different encodings of the ride-sharing problem induced by the feasible ride match graph inferred from the collected user trip schedules without transformation. In this case, our experiments show the superiority of the CP problem formulations over the MIP problem formulations with respect to the succinctness of encoded problems, the modelling time and the solving time for all the instances. For the second hypothesis we allow drivers to change role and to becoming riders. In this case, we observe an increase on the numbers of variables, the number of constraints, the modelling time and the solving time. For the smaller instances, the experiments show the superiority of the MIP formulation over the CP formulation with respect to the solving time, despite a larger encoding. Unfortunately, none of the problem formulations were able to solve the larger instances within the time limit. Future work will focus on efficient local search heuristics and the definition of a global constraint for handling the special case where resources can also become tasks which corresponds to the hypothesis allowing drivers to become passengers.

Acknowledgements

This work is funded by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

[1] Ibm ilog cplex optimization studio v12.6, 2014.

[2] 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.

[3] 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.

[4] 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.

[5] Vincent Armant, John Horan, Nahid Mahub, and Kenneth N. Brown. Data analytics and optimisation for assessing a ride sharing system (to appear). In *IDA 2015, 14th International Symposium on Intelligent Data Analytics*, 2015.

[6] 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.

[7] Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15, 2010.

[8] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.