# Online Stochastic Planning for Taxi and Ridesharing

Carlo Manna
*Insight Centre for Data Analytics*
*University College Cork*
*Cork, Ireland*
*carlo.manna@insight-centre.org*

Steve Prestwich
*Insight Centre for Data Analytics*
*University College Cork*
*Cork, Ireland*
*s.prestwich@cs.ucc.ie*

***Abstract*—In this paper we consider the problem of on-line stochastic ride-sharing and taxi-sharing with time windows. We study a scenario in which people needing a taxi, or a ride, assign their source and destination points plus other restrictions (such as earlier time to departure and maximum time to reach a destination); at the same time, there are taxis or drivers interested in providing a ride (also with departure and destination points, vehicle capacity and time restrictions). We model the time window restrictions as a soft constraint (a reasonable delay might be acceptable in a realistic scenario), and consider the problem as an on-line continual planning problem, in which additional ride requests may arrive while plans for previous ride-matching are being executed. Finally, such new requests may arrive at each time step with some probability. The aim is to maximize the shared trips while minimising the expected travel delay for each trip. In this paper we propose an on-line stochastic optimization planning approach in which instead of myopically optimising for the offered trips and requested trips that are known, incorporate information that partially describes the stochastic future into the model in order to improve the quality of the solution. We prove the effectiveness of the method in a real world scenario using a number of instances extracted from a travel survey in north-eastern Illinois (USA) conducted by the Chicago Metropolitan Agency for Planning.**

***Keywords*-stochastic optimization, on-line scheduling, ridesharing;**

## I. INTRODUCTION

Private cars represent the most widespread transportation vehicle and their occupancy rate is relatively low [17]. Many cars only carry a single occupant and have three or more empty seats. During the peak-hours such a large demand for automobile transportation along with high vacancy rate leads to traffic congestion in many urban and sub-urban areas for an annual cost in terms of wasted fuel that in US was estimated to be around $78 billion in 2007 [18]. At this, we must then add the high cost in terms of environmental impact due to the resulting gas emissions.

Ridesharing and taxi sharing services provide an effective usage of cars capacity by using a common vehicle for shared trips. Conceptually, ridesharing refers to a mode of transportation in which individual travelers share a vehicle for a trip and split travel costs such as gas, toll, and parking fees with others that have similar itineraries and time schedules. These services could drammatically increase the vehicle occupancy rates, as well as the efficiency of urban transportation systems, reducing traffic congestion, fuel consumption, and pollution. They also represent an economical advantage for both passengers and drivers (or taxi drivers) as the users can share the travel related expenses. Currently, existing ridesharing systems can be classified depending on their matching search criteria and demand target segments [7]. However, in this paper we are only interested in ones, the so-called dynamic or real-time ridesharing system which supports an automatic ride-matching process between particpants on very short notice or even *en-route* by effectively using new communication capabilities, including mobile technology and global positioning system. In addition, we consider the scenario in which stochastic information about future requests are available in real-time with a certain probability.

The purpose of this paper is to investigate, whether using this information in designing the routes has a significant positive effect on the solution quality and, at same time, we propose an on-line procedure to accomplish this task.

To the best of our knowledge, there is no previous work on the dynamic taxi/ride sharing with time windows considering stochastic requests.

On the other hand, literature on the deterministic version of such a problem has attracted many researchers in the field of AI (see [1], [3], [4], [5], [11] and [6]), to propose algorithms able to maximize the number of shared trips while minimising other parameters (delay time per trips, cost of a shared trips, travel time etc.). It can be formulated as an on-line optimization problem and has be proven to be NP-hard [1].

More precisely, the deterministic problem consists of a set of taxi/drivers offers (set of vehicles) and riders' requests. Each participant (driver or rider) specifies a source and a destination location for her/his trip, along with an earliest departure time and a latest arrival time which define a time window at each location (notice that in the case of taxi drivers, these conditions are not applied. This removes only few constraints to the problem, but the overall formulation and complexity is still valid [1]). Finally, each vehicle has its own capacity (maximun number of carrying passengers). The aim is to assign riders to drivers and determine the

timing and order of the pickup and delivery of riders in real-time. Because new requests may arrive at each time while other are already assigned, the problem is dynamic and has to be solved on-line.

This is a challenging problem even in deterministic domain. However, the advancements in the field of human mobility [14], in the *trip based models* [13] for forecasting urban passenger travel demand and finally the large use of phone app for the use of any transportation service (which potentially represents a massive travelers data source), suggests that the use in real-time of travel demand prediction in transportation system design should be even considered in order to improve the quality and the performance of the service in the future.

In this paper, we consider an on-line stochastic optimization approach that explicitly exploits information about future request arrivals. It assumes that the probability distribution over incoming requests is either known or learn-able. We propose an *on-line anticipatory approach* previously tested to the on-line stochastic packet scheduling problem [12], for dynamic taxi and ridesharing problem with time window. By means of this technique, we first samples from the distribution of possible future *request* arrivals, and then optimize each scenario deterministically. Finally we select the best *decision* by combining the solutions found in all previously sampled scenarios. Using this anticipatory technique, our planner is able to take future requests into account. We test our approach to solve instances using real world dataset extracted from a travel survey for northeastern Illinois (USA) provided by Chicago Metropolitan Agency for Planning [15].

The rest of the paper is organised as follows. First, in the next section, we describe the related work, in which we consider some on-line (deterministic) taxi and ridesharing solvers recently developed and works related to stochastic planning for similar problems. Then, in the sections 3 and 4 we give respectively, a formal description of the problem and the proposed method in detail. Finally, we present our results in the aforementioned real world scenario in section 6, while in section 7 we detail future work and conclusions.

## II. RELATED WORK

Taxi and ridesharing is a relatively new emerging transportation system. [7] reports a comprensive state-of-art, providing a taxonomy and a classification of the existing systems and analysing the current limitations. Among different ridesharing systems, in this paper we consider the so-called dynamic taxi and ridesharing [11] which aims to bring together travelers with similar itineraries and time schedules on short-notice or en-route. On the authors knowledge there is no any previous work for on-line stochastic planning in dynamic ridesharing. However, many AI researchers provided different approaches to solve the deterministic version of such a problem. [3] and [4] proposed interesting framework models to compute route matching between users by means of simple greedy heuristics. However, both of them do not take into account any complicated constraints. A more sofisticated approach to the deterministic version for dynamic ridesharing was proposed by [6]. They take into account the time constraint and model the problem as a maximum-weight bipartite matching problem, testing it in a real world scenario (using dataset from Atlanta metropolitan area). However, in their formulation they consider each driver may pass through only one pick up and delivery location and optimize just the vehicle miles to evaluate the routes. In [5], the authors propose a two stage heuristic algorithm for solving the problem. In the first off-line stage, the algorithm works as a genetic algorithm while in the second stage it works as an insertion heuristic that modifies the solution of the genetic algorithm to do ridematching in real-time. They consider multiple constraints and time windows. A similar approach is proposed in [1] in which they add the minimisation of shared cost between the partecipants in the objective function using a local search based method for the on-line optimization and prove the NP-hardness of the problem.

The main difference with the last two mentioned works is that we consider the time window as soft constraints (because in real world we can deal with a reasonable delay). Furthermore, we take into account each possible future requests, while they evaluate routes considering only requests that are known.

The only work similar to the one proposed in this paper is provided in [16]. They analize the problem in a dial-a-ride setting, considering the daily operation of the Austrian Red Cross, in which each request requires transportation from a patient's home location to a hospital or back home from the hospital and in which some requests are stochastic. In order to solve the on-line stochastic problem they use the Multiple Scenario Approach (MPA) proposed in [9] for the vehicle routing problem, considering each vehicle pass through only one pick up and delivery location and minimising the total vehicle-miles driven. Conversely, we consider a scenario in which each vehicle has higher capacity (maximum number of passengers) and the time window as soft constraints. Then we maximize the number of participants (ride matching) while minimize the delay time per passengers. Finally [8] proposed a dynamic programming based approach to solve a stochastic dynamic assignment problem for truck load motor carrier. In order to deal with stochastic scenario, they model the problem as a dynamic network with random arc capabilities to deal with uncertainty and solving it with approximate dynamic programming based techniques.

## III. PROBLEM FORMULATION

The on-line stochastic planning for taxi and ridesharing problem can be defined as a 7-tuple $\langle \mathbf{R}, \mathbf{D}, \mathbf{S}, \Omega, H, W, C \rangle$, where $\mathbf{R}$ and $\mathbf{D}$ are the set of riders and drivers respectively.

$\mathbf{S}$ is the *world state*, the configuration of the world at generic time $t$, $\Omega$ is a known probability distribution for the future requests, while $H = [1,h]$ is the time horizon and specifies a discrete model of time $t$. Finally, $W$ and $C$ are a profit function and the problem specific constraints, respectively.

The problem can be described as follow. At each time $t \in H$, there is a set of $R_t \in \mathbf{R}$ riders wanting a ride and a set of $D_t \in \mathbf{D}$ available drivers offer a ride. The set of requests $R_t$ are not known a priori but are revealed online at time $t$ and are drawn from a distribution $\Omega$.

Let define a *matching* $a(r,d)$ between a rider $r \in R_t$ and a driver $d \in D_t$ as an assignment of the rider $r$ to $d$, and the *plan* $\gamma_t$ be the set of all the couple $a(r,d)$ at time $t$, such that $a \in \gamma_t$ for each $a$. Then, the configuration of the world $\mathbf{S}$ at time $t$ is specified by all the elements $s(a,p) \in \mathbf{S}$ defined as the combination of the matching $a(r,d) \in \gamma_t$ and the position $p$ of the driver $d$ in $a$ on the world map.

In the problem we consider here, new requests may arrive at each time step with the known distribution $\Omega$. Hence, at time $t$ the on-line algorithm has at its disposal the set of requests $R_t$ (which include both requests revealed in $t$ and those previously *not assigned* to any vehicle) and available drivers $D_t$ (i.e new drivers and those previously revealed but still available to give a ride). The goal is to find a set of new matching $\gamma_t$ (or plan) in $t$ between the elements $r \in R_t$ and $d \in D_t$, starting from the previous world state $\mathbf{S}_{t-1}$.

The criteria by which the online algorithm assigns such a requests depends upon a profit function $W(\cdot)$ subjects to some problem specific constraints $C(\cdot)$ (both are detailed in the following subparagraph). The optimization problem is stochastic because the maximization of the profit $W(\cdot)$ take into account not only the immediate profit at the current time $t$ (as in a myopic strategy), but the overall expected profit over the total time horizon $H = [1,h]$ considering all possible future requests from the distribution $\Omega$. The solution of the problem can be represented by the sequence of plans $\Gamma = \langle \gamma_1, \dots \gamma_h \rangle$ over the total time horizon $H$. Therefore, at each time $t \in H$, the online optimization strategy selects a $\gamma_t$ to maximize the expected profit:

$$\max \quad \mathbb{E}(W(\langle \gamma_1, \dots \gamma_h \rangle)) \tag{1}$$

such that the sequence $\langle \gamma_1, \dots \gamma_h \rangle$ satisfies the problem specific constraints:

$$C(\langle \gamma_1, \dots \gamma_h \rangle); \tag{2}$$

### A. Profit function

The sequence of plans $\Gamma = \langle \gamma_1, \dots \gamma_h \rangle$ over the time horizon $[1,H]$ in the eq. (1), is computed in order to maximize the number of matching $a \in \Gamma$, that is the number of shared trips over the entire horizon $H$, and at same time, it minimize the travel delay for each user (both riders and drivers). In particular, we consider the situation in which a delay might be possible because consider such situation more realistic. Anyway such a delay can be seen as a penalty for the quality of service [10].

In particular, each user $u \in (\mathbf{R} \cup \mathbf{D})$ (i.e. both riders and drivers) is characterized by the couple $(e_u, l_u)$, where $e_u$ is the earliest time to departure (i.e. the user cannot leave its starting location before that time), while $l_u$ is the latest time to arrival (i.e. if the time to arrival is greater than $l_u$, such delay decreases the quality of service).

We penalize each travel delay as follow: for each user $u$, let $T_u^e = (l_u - e_u)(1+\alpha)$ be the *estimated* travel time with $\alpha \in [0,1]$, and $T_u^a$ the *actual* travel time. The penalty $c(u)$ for the user $u$ is:

$$c(u) = \begin{cases} e^{\frac{T_u^a - T_u^e}{T_u^e}}, & \text{if } T_u^a > (l_u - e_u), \\ 0, & \text{otherwise.} \end{cases}$$

Hence, the sequence of plans $\Gamma$ maximize the following profit function:

$$W(\Gamma) = \mid \Gamma \mid - \sum_{u \in \Gamma} c(u) \tag{3}$$

for all the users $u \in \Gamma$ (riders and drivers) for which a suitable match has been found over the entire time horizon $[1,H]$.

### B. Constraints

The online optimization problem described above is subject to the following constraints:

1) Each driver (rider) must start (be picked up) at the specified source point, after its earliest departure time $e_u$;
2) Each vehicle cannot load more passengers than its capacity.

## IV. PROPOSED METHOD

The on-line stochastic planning algorithm proposed for the taxi and ridesharing problem consists in three main steps:

1) draw samples from the probability distribution over the entire time horizon to obtain future scenarios;
2) find an optimal plan for each scenario ;
3) select the best decision combining optimal plans over all scenarios.

In particular, we assume a discrete model for the time $t \in H$. At each time step $t$, there are $R_t$ revealed riders to assign to $D_t$ available drivers. The algorithm samples from the distribution $\Omega$ a number $i$ possible future scenarios (step 1). Each sampled scenario includes possible future requests $r \in \mathbf{R}$ and available drivers $d \in \mathbf{D}$ from $t$ to $H$ (over the entire time horizon). Notice that, among them, only riders and drivers revealed at time $t$ ($R_t \in \mathbf{R}$ and $D_t \in \mathbf{D}$ respectively) can finally be assigned from the on-line algorithm. The number of sampled scenarios is an algorithm parameter.

Then, each sampled scenario $i$ can be optimized deterministically using an appropriate optimization procedure (step 2). Hence, for each scenario $i$ there is a *deterministic plan* $\pi_i$ that match riders to drivers revealed in the scenario $i$ from $t$ to $H$. However, we have two issues. First we have as many *deterministic plans* $\pi_i$ as the number of sampled scenarios. How can we *combine* those to select the *best* plan ? Finally, each *deterministic plan* includes matching between riders and drivers available over the entire time horizon from $t$ up to $H$. How can we extract *only* matching between riders and drivers available at time $t$ (we cannot match users before they are not revealed)?

**Algorithm 1** ONLINE STOCHASTIC PLANNING

1: **for** $t$ from 1 to $H$ **do**
2: $\quad 0 \leftarrow Q(r,d)$
3: $\quad$ **for** $i$ from 1 to $Width$ **do**
4: $\quad\quad f_i \leftarrow draw\_future(\mathbf{R},\mathbf{D},\Omega,t,H)$
5: $\quad\quad \pi_i \leftarrow optimize(f_i, s_{t-1}, C)$
6: $\quad\quad$ **for all** $(r,d) \in \pi_i$ **do**
7: $\quad\quad\quad Q(r,d) \leftarrow Q(r,d)+1$
8: $\quad \gamma_t \leftarrow compute$ eq.(5)
9: $\quad s_t \leftarrow \langle s_{t-1}, \gamma_t \rangle$

**Definition**: given a time slot $t$, for each matching $a = (r,d)$, with $r \in R_t$ and $d \in D_t$, we define a *plan aggregation rank*, a function such that:

$$Q(r,d) : (R_t \times D_t) \rightarrow \mathbb{N} \tag{4}$$

The plan aggregation rank $Q$ is a function of each matching between the rider requests and drivers *revealed* at time $t$.

$Q(r,d)$ can be easely represented as a *lookup table* with $r$ rows and $d$ columns for each $r$ and $d$ defined in $R_t$ and $D_t$ respectively. The value of each cell $q(r,d)$ is a non negative integer.

At start of the sample procedure (in step 1), we inizialize each element of $q(r,d) = 0$. Then, during the deterministic optimization step, for each couple of elements $(r,d)$ such that $r \in (R_t \bigcup \pi_i)$ and $d \in (D_t \bigcup \pi_i)$ we update the element $q(r,d) = q(r,d) + 1$. Put in other words, we select the drivers and riders available at time $t$ and included in the deterministic plan $\pi_i$ (i.e. drivers and riders for which a match has been found) and increment those elements $(r,d)$ of $Q$ by 1. We continue to update $Q$ after optimizing each sampled scenario $i$.

At the end of the sampling/optimization step, the final value of the all elements $q(r,d)$ denote the number of times those matching have been selected for a plan $\pi_i$. Now, we can evaluate each deterministic plan $\pi_i$ individually by means of $Q(r,d)$.

We choose the best plan $\gamma_t$ among all the deterministic plans $\pi_i$ as follow:

$$\gamma_t = \arg\max_{\pi_i} \sum_{r,d \in (R_t \times D_t) \bigcup \pi_i} Q(r,d) \tag{5}$$

Basically, by means of eq.(5), we select the deterministic plan $\pi_i$ which includes the matching between riders and drivers available in $t$ that maximize $Q$. We choose such a plan as the best plan. Finally, we select to match, among riders and drivers available in $t$, only those included in the best plan $\gamma_t$. Notice that in doing this, we select only users available (or revealed) at time $t$.

The on-line stochastic planning procedure is reported in the box Algorithm 1. At each time step $t \in H$ (line 1), the algorithm finds the next matching between users available in $t$. In line 2, the plan aggregation rank $Q(r,d)$ is inizialized for each $r \in R_t$ and $d \in D_t$. We generate a set of *Width* samples $f_i$, using the *draw_future* function (line 3 and 4), where each $f_i$ is a set of possible drivers and riders available over the entire time scenario from $t$ to $H$. Then, in line 5, each scenario is deterministically optimized finding a plan $\pi_i$ by means of some optimization algorithm (which we detail in the next section). In line 6 and 7, we update the plan aggregation rank $Q(r,d)$ according to the users matched in each deterministic plan $\pi_i$. Finally, il line 8, we compute the eq.(5) to select the best plan $\gamma_t$ and then, il line 9, we update the world state $s_t$ *including* the new assignment $\gamma_t$ of riders to drivers to the previous state $s_{t-1}$.

In the next section we detail the optimization algorithm used to find each deterministic plan $\pi_i$.

### A. Optimization algorithm

For each sampled scenario, we solve a deterministic optimization problem of maximizing the objective function in eq. (3) under the problem specific constraints reported in section III.B. We propose a two-step optimization procedure. We first compute a *greedy* initial solution and, in the second step, we make an attempt to improve this starting solution using a *local search* based procedure.

**Initial solution**: An initial solution is created as follows. First, we randomly select an available driver $d$. Then, we insert in its route a rider $r$ and evaluate the profit function $W$ and store this value. We repeat this operation on the same $d$ for each available rider $r$. Then we insert in the $d$ route a number of riders (as many as to fill its capacity) according to a probability function proportional to the profit function $W$ obtained for each $r$. We repeat the procedure above for each driver $d$.

**Local Search**: The local search starts from the current solution as an initial feasible solution, and explore new solutions that are in its neighborhood. Solutions in the neighborhood are obtained by performing some modifications in the current solution. In our approach we use three types of operations to obtain new solutions:

- Permutation of two requests from different routes;

- Permutation of two consecutive points of the same route;
- Removal of some request from some route.

In the first operation, we randomly select two served requests ($r1$ and $r2$) served by two different drivers ($d1$ and $d2$ respectively). Then, we try to insert $r1$ in $d2$ and $r2$ in $d1$ in any permutation order. In the second operation, we select randomly some driver $d$. Then, we randomly select a point of this route (which could corrisponds either to a pick up or a delivery point of a passenger). Notice that, in doing so, we never select the first point or the last point of its route (which corrisponds to the source point or the arrival point of the driver itself respectively). Finally, we change the order of the choosen point with its successor. In order to do this, we insert again all the points of the route in the same order, except the two points, which will be permuted. In the third operation, we randomly select a route and remove one passenger (one pick up and the relative delivery point). We repeat this operation for each passenger in this route.

In the aforementioned local search procedure, at each iteration, we execute each of the three operations defined above, obtaining three new solutions. If the best of them is better than the current one, then the method sets this solution as the new current one. This procedure is repeated until a predefined maximum number of iterations is achieved.

## V. EXPERIMENTS

In this section we describe the experimental testing to validate our approach. First we describe the real-world data sets used and then we present results on this set.

### A. Data preparation

To validate our approach we use data from a travel survey in north-eastern Illinois (USA) conducted by Chicago Metropolitan Agency for Planning. This data collection took place between January 2007 and February 2008 and covered a total of 10,552 households participant [15]. This survey was carried out on a large area covering 11 counties (fig. 1).

A trip in the study is defined as a pair of source and destination positions (using latitude and longitude) with departure and arrival timestamps. From the whole dataset, we prepared 12 instances as follow. First, we have extracted four travel patterns, in which each trip is at least 20 km long. These initial four travel patterns consist in, respectively, 219, 396, 541 and 843 individual trips. Because all source and destination points are in latitude and longitude coordinates, we used the Haversin formula to evaluate the distance between two pair of points (evaluated in a straight line). Then, for each travel pattern we select the 85 most long trips and assumed these as drivers while the rest as rider requests. Finally, we randomly divided each group of drivers in three sets of 10, 25 and 50 respectively, obtaining the 12 instances reported in table 1.

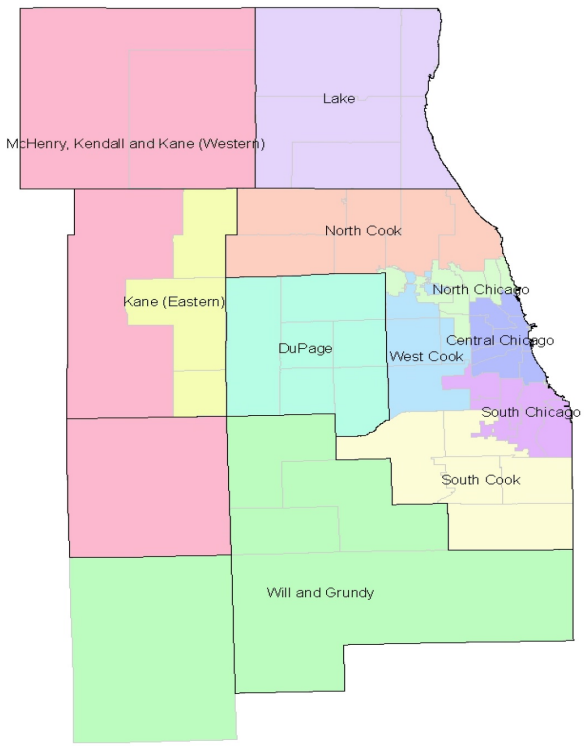


Figure 1. CMAP travel survey was carried out on 11 counties, between January 2007 and February 2008, and involved 10,552 households participant. For more detail see [15].

Table I
LIST OF INSTANCES USED DURING THE EXPERIMENTS

| | **riders** | **drivers** |
|---|---|---|
| instance 1 | 134 | 10 |
| instance 2 | 134 | 25 |
| instance 3 | 134 | 50 |
| instance 4 | 311 | 10 |
| instance 5 | 311 | 25 |
| instance 6 | 311 | 50 |
| instance 7 | 456 | 10 |
| instance 8 | 456 | 25 |
| instance 9 | 456 | 50 |
| instance 10 | 758 | 10 |
| instance 11 | 758 | 25 |
| instance 12 | 758 | 50 |

Each extracted trip take place within a time window between 8 : 00 and 20 : 00. For each partecipant (both drivers and riders), the earliest departure time $e_u$ corresponds to that reported in the survey as departure time. The travel time between two pair of points was calculated considering a trip in a straight line at a vehicle speed of 60km/h. Then, the latest arrival time $l_u$ of each partecipant was set as the earliest departure time, plus the travel time between the source and the destination. Finally, to simulate a distribution of possible future request arrivals, in each instance, we randomly divided each group of rider requests in 3 sub-

groups with probability of occurrence of 0.90, 0.75 and 0.50 respectively.

### B. Parameter settings

Table II
NUMBER OF MATCHED PARTECIPANTS FOR THE **OSP** AND **OPnS** APPROACH. THE RESULTS ARE AN AVERAGE OVER 30 RUNS, THEN WE ROUND EACH RESULT TO THE NEAREST INTEGER. THE IMPROVEMENT RATIO (I.R.) IS THE RATIO BETWEEN THE NUMBER OF MATCHED PARTECIPANTS OBTAINED WITH **OSP** AND THOSE OBTAINED FROM **OPnS**

| | **OSP** | **OPnS** | **I.R.** |
|---|---|---|---|
| instance 1 | **21** | 12 | 1.75 |
| instance 2 | **65** | 40 | 1.62 |
| instance 3 | **72** | 44 | 1.63 |
| instance 4 | **36** | 19 | 1.89 |
| instance 5 | **58** | 37 | 1.57 |
| instance 6 | **123** | 73 | 1.68 |
| instance 7 | **22** | 10 | 2.22 |
| instance 8 | **72** | 39 | 1.85 |
| instance 9 | **107** | 62 | 1.73 |
| instance 10 | **32** | 18 | 1.78 |
| instance 11 | **84** | 52 | 1.61 |
| instance 12 | **131** | 77 | 1.70 |

As noted above, the experiments were carried out on the 12 instances reported in table 1. We assume a discrete model of the time $t$, and consider 12 time slots, 1 for each hour, starting from $8:00$ up to $20:00$. At each time slot $t$, new rider requests are revealed drawn from a probability distribution described in the last section. Basically, for each instance, we compare two approaches. In the first, we consider the proposed on-line stochastic planning (**OSP**), in which we use stochastic information into the solver. In the second, we use an on-line planning without stochastic information (**OPnS**) in which we only consider the requests that are known at time $t$ (myopic strategy). We evaluate this non stochastic approach using the same optimization procedure used in the **OSP** and reported in section IV.A. The parameter $\alpha$ for the estimated travel time $T_u^e$ was set to 0.1 for rider requests and 0.3 for drivers. This mean that for a passenger we assume that the *estimated* travel time $T_u^e$ is given by the difference $(l_u - e_u)$, plus a maximum delay of 10% of this quantity. Such a maximum delay raise up to 30% for a driver.

For the **OSP** algorithm setting, after a number of small test, we finally decided to set the parameter *Width* (the number of sampled scenarios) to 70 and the maximum number of iterations for the local search to 100. Conversely, the maximum number of iterations for the local search in the **OPnS** algorithm was 500.

### C. Results

We executed each instance 30 times, and we took the average of the results.

Table III
AVERAGE DELAY TIME FOR EACH PARTECIPANT FOR THE **OSP** AND **OPnS** APPROACH. THE RESULTS ARE AN AVERAGE OVER 30 RUNS AND ARE REPORTED IN PERCENTAGE

| | **OSP [%]** | **OPnS [%]** |
|---|---|---|
| instance 1 | **0.0532** | 0.0887 |
| instance 2 | 0.1363 | **0.0341** |
| instance 3 | **0.0866** | 0.0982 |
| instance 4 | 0.1256 | **0.0171** |
| instance 5 | 0.1143 | **0.0284** |
| instance 6 | **0.0507** | 0.0570 |
| instance 7 | 0.1043 | **0.0980** |
| instance 8 | 0.0996 | **0.0518** |
| instance 9 | 0.1432 | **0.0763** |
| instance 10 | 0.1269 | **0.0188** |
| instance 11 | 0.1573 | **0.0399** |
| instance 12 | 0.0824 | **0.0752** |

Table IV
ITERATION TIME PER TIME SLOT FOR THE **OSP**. THE RESULTS ARE AN AVERAGE OVER 30 RUNS AND ARE REPORTED IN SECONDS

| | **Iter. time [s]** |
|---|---|
| instance 1 | **32.26** |
| instance 2 | **33.92** |
| instance 3 | **29.22** |
| instance 4 | **46.09** |
| instance 5 | **52.21** |
| instance 6 | **73.66** |
| instance 7 | **113.17** |
| instance 8 | **142.60** |
| instance 9 | **179.77** |
| instance 10 | **277.77** |
| instance 11 | **332.51** |
| instance 12 | **491.13** |

The experiments were carried out on an Intel i7-4800MQ 2.7 GHz x 8, with 8Gb of memory, using Window 7 operating system. The algorithms were written in Matlab.

Table 2 shows the results in terms of number of partecipants matched. In all the instances **OSP** achieves a better solution compared to the case in which we use a myopic strategy.

In the worst case (instance 11, which consists of 758 rider requests and 25 drivers) we achieve an improvement ratio of 1.61, while in the better case, the instance 7 (which corresponds to 456 rider requests and 10 drivers) the improvement ratio raise up to 2.22.

Table 3 shows the average delay time per partecipant in percentage. Although, in the majority of the instances, the myopic strategy achieves better results (9 out of 12), the average delay time per partecipant achieved by the **OSP** strategy is about 0.15% in the worst case (instance 11), which is negligible in a real world scenario.

In table 4 we report the average iteration time per time slot. It shows that the proposed method are generally fast (less than 3 minutes) for the majority of the instances, while for the last 3 instances, which consist of 758 rider requests each, the average iteration time per time slot raise up to 491

seconds (around 8 minutes).

Finally, we want to emphatize here that the overall results show different insights. First, based on our real-world based test, we could significantly improve the number of customers in taxi and ridesharing system (or similar transportation system), taking into account information about the *future travel demand*. Although we are conscious that this is just a first experimentation (maybe further tests might be carried out in the future), this first result is encouraging. Secondly, the proposed approch to accomplish this task has a low runtime complexity, and hence is feasible to be used in real-time.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we studied the problem of on-line stochastic ridesharing and taxi sharing with time windows in which we explicitly exploit information about future request arrivals. We also proposed a general method to accomplish this task. Using real world dataset, we proved that using stochastic information, we can maximize the number of partecipants, from 1.61 up to 2.22 times, compared to those achieved with a myopic strategy, while, assuming flexible time window constraints, the resulting travel delay per partecipant is, on average, negligible. Furthermore, our method is characterized by a fast iteration time, which, also with large instances (consisting in 758 requests in the largest case) it achieves a solution in about 8 minutes on average.

As future works, we believe that the proposed on-line stochastic planning procedure could be improved in many aspects.

First of all, in order to achieve better results in terms of iteration time and quality of solution, we can employ many fast heuristics for the deterministic optimization step. For example we can use fast solvers as those proposed in [1] and [5] for the dynamic ridesharing problem.

Finally, as further future work, experimentations using real map of the a city can be done. However, in order to do this, we need to compute shortest paths in a fast way. For this we could use fast algorithms as proposed [2] to compute distance tables for road networks.

## VII. ACKNOWLEDGEMENT

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

## REFERENCES

[1] D. O. Santos and E. C. Xavier, *Dynamic Taxi and Ridesharing: A Framework and Heuristics for the Optimization Problem*, IJCAI'13 Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, 2103.

[2] S. Knopp, D. Wagner,and et al.,*Computing many-to-many shortest paths using highway hierarchies*, ACM Journal of Experimental Algorithmics 15, 2010.

[3] P. Lalos, A. Korres, C.K. Datsikas, G.S. Tombras, and K. Peppas. *A framework for dynamic car and taxi pools with the use of positioning systems*. In Fu- ture Computing, Service Computation, Cognitive, Adap- tive, Content, Patterns, 2009. COMPUTATIONWORLD 09. Computation World:, pages 385 391, nov. 2009

[4] C.Tao. *Dynamic taxi-sharing service using intelligent transportation system technologies*. In Wireless Communications, Networking and Mobile Com- puting, 2007. WiCom 2007. International Conference on, pages 3209 3212, sept. 2007.

[5] W.M. Herbawi and M. Weber. *A genetic and insertion heuristic algorithm for solving the dynamic ridematching problem with time windows*. In Proceedings of the fourteenth international conference on Genetic and evolutionary computa- tion conference, GECCO 12, pages 385392, New York, NY, USA, 2012. ACM.

[6] N. A.H Agatz, A. L. Erera, M. W.P Savelsbergh, and X. Wang. *Dynamic ride-sharing: A simulation study in metro atlanta*. Transportation Research Part B: Methodological, 45(9):1450(15), 2011

[7] M. Furuhata, M. Dessouky, F. Ordonez, M. Brunet, X. Wang and S. Koenig. *Ridesharing: The State-of-the-Art and Future Directions*. Transportation Research Part B: Methodological, 57, 28-46, 2013.

[8] W. B. Powell, *A Stochastic Formulation of the Dynamic Assignment Problem, with an Application to Truckload Motor Carriers*, Transportation Science, Volume 30 Issue 3, pp. 195-219, 1996.

[9] R. W. Bent, P. Van Hentenryck, *Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers*, Operations Research, V. 52 Issue 6, pp. 977-987, 2004.

[10] Y. Dumas, F. Soumis, J. Desrosiers, *Optimizing the Schedule for a Fixed Vehicle Path with Convex Inconvenience Costs*, Technical Note, Transportation Science, 24(2):145-152, 1990.

[11] N. Agatz, A. Erera, M. Savelsbergh, X. Wang, *Optimization for dynamic ride-sharing: A review*, European Journal of Operational Research 223, 295303, 2012.

[12] L. Mercier, P. Van Hentenryck, *Performance analysis of online anticipatory algorithms for large multistage stochastic integer programs* Proceeding IJCAI'07 Proceedings of the 20th international joint conference on Artifical intelligence Pages 1979-1984, 2007.

[13] J.L. Bowman, M.E. Ben-Akiva, *Activity-based disaggregate travel demand model system with activity schedules*, Transportation Research Part A 35, 1-28, 2000

[14] F. Asgari, V. Gauthier, M. Becker, *A survey on Human Mobility and its Applications*, arXiv:1307.0814, 2013

[15] Chicago Metropolitan Agency for Planning (CMAP), Travel Tracker Survey. http://www.cmap.illinois.gov/data/transportation/travel-tracker-survey

[16] M. Schilde, K. F. Doerner, R.F. Hartl, *Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports*. Computers and Operations Research 38 (12), 17191730, 2011

[17] A. Santos, N. McGuckin,H.Y. Nakamoto, D. Gray, S. Liss, *Summary of travel trends: 2009 national household travel survey*. Technical Report, US Department of Transportation Federal Highway Administration, 2011.

[18] D. Schrank, T. Lomax, *2007 urban mobility report*. Technical report, Texas Transportation, 2007