UNIVERSITY OF BIRMINGHAM
University of Birmingham Research at Birmingham

Vehicle dispatch in on-demand ride-sharing with stochastic travel times

Li, Cheng; Parker, David; Hao, Qi

DOI:
10.1109/IROS51168.2021.9636499

License:
Other (please specify with Rights Statement)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Li, C, Parker, D & Hao, Q 2021, Vehicle dispatch in on-demand ride-sharing with stochastic travel times. in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 5966-5972, 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems, Prague, Czech Republic, 27/09/21.
https://doi.org/10.1109/IROS51168.2021.9636499

Link to publication on Research at Birmingham portal

Publisher Rights Statement:
© 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

General rights
Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

•Users may freely distribute the URL that is used to identify this publication.
•Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
•User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
•Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy
While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Download date: 26. Apr. 2024

Vehicle Dispatch in On-Demand Ride-Sharing with Stochastic Travel Times

Cheng Li[1,2], David Parker[1] and Qi Hao[2,3]

***Abstract*— On-demand ride-sharing is a promising way to improve mobility efficiency and reliability. The quality of passenger experience and the profit achieved by these platforms are strongly affected by the vehicle dispatch policy. However, existing ride-sharing research seldom considers travel time uncertainty, which leads to inaccurate dispatch allocations. This paper proposes a framework for dynamic vehicle dispatch that leverages stochastic travel time models to improve the performance of a fleet of shared vehicles. The novelty of this work includes: (1) a stochastic on-demand ride-sharing scheme to maximize the service rate (percentage of requests served) and reliability (probability of on-time arrival); (2) a technique based on approximate stochastic shortest path algorithms to compute the reliability for a ride-sharing trip; (3) a method to maximize the profit when a penalty for late arrivals is introduced. Based on New York City taxi data, it is shown that by considering travel time uncertainty, ride-sharing service achieves higher service rate, reliability and profit.**

I. Introduction

With the rapid development of e-hailing platforms, on-demand ride-sharing services such as UberPool, Lyftline and GrabShare are playing an important role in urban mobility. By simultaneously allocating multiple requests that are travelling in similar directions to the same vehicle, ride-sharing brings multiple benefits: passengers pay lower costs by sharing the vehicle with others; and the service providers can increase their profit by serving more requests with the same number of vehicles.

When evaluating the feasibility and benefits of schedules for ride-sharing trips, the most important consideration is the timing of passengers, because travel delay is a more constraining factor than others, e.g. spare capacity [1]. To have accurate estimations of the benefit from on-demand ride-sharing and get the best vehicle dispatch policy, the following technical challenges need to be properly tackled:

1) *Travel time uncertainty.* Due to various stochastic factors, travel times on roads can exhibit considerable variability in urban environments. Solutions based on deterministic routing would deteriorate when deployed in real-life scenarios.

*This work is partially supported by the Shenzhen Fundamental Research Program (No:JCYJ20200109141622964) and the Intel ICRI-IACV Research Fund (CG#52514373).

[1]School of Computer Science, University of Birmingham, Birmingham, UK {cxl776, d.a.parker}@cs.bham.ac.uk

[2]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China haoq@sustech.edu.cn (Corresponding author: Qi Hao)

[3]Research Institute of Trustworthy Autonomous System, Southern University of Science and Technology, Shenzhen, China

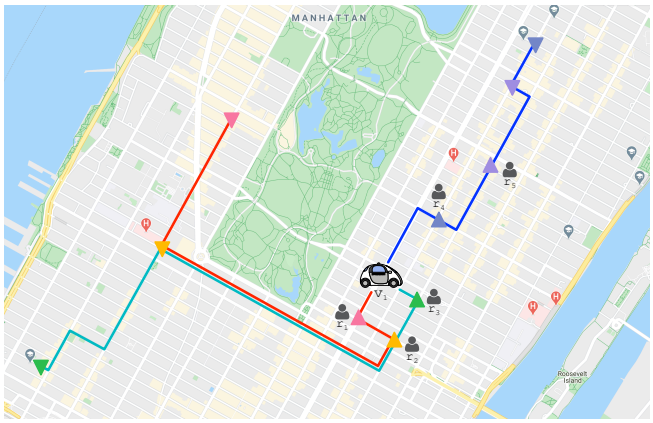

Fig. 1: An example with five requests r_1, r_2, r_3, r_4, r_5 and one vehicle v_1. Triangles ($\triangle$) and inverted triangles ($\triangledown$) represent the origins and destinations, respectively. There are three possible dispatch policies for v_1 and they exhibit no difference in service rate when travel time uncertainty is not considered. When taking stochastic travel time into account, it would be found that serving r_2 and r_3 is the best allocation as it has the highest reliability in terms of arrival.

2) *Reliability estimation.* In reality, vehicles may not arrive at their planned destinations on time. Passengers are more concerned about the reliability of their trip than its duration. Unreliable service would result in the loss of passengers.
3) *Profit estimation.* The prices of trips are determined immediately after their request and remain fixed, regardless of the results of allocation and travel time. But travel costs are related to actual travel routes and penalties may be applied for late arrivals, so inaccurate estimation of profit during dispatch may differ from the profit-optimal solution.

Ride-sharing has been receiving increased attention from the academic community. Existing work includes scheduling and allocating vehicles to serve requests [2], [3], repositioning idle vehicles to high demand areas [4], [5] and predicting future demands for long-term optimization [6], [7]. However, most of the previous studies fail to consider the uncertainty of travel time. They only compute assignments and routes with the shortest expected time for the sake of simplicity, and ignore the reliability of on-time arrival, which of high value for passengers. Furthermore, request price and platform profit, which are important references for the platform, have also received little attention.

In this paper, we study the role of travel time uncertainty in on-demand ride-sharing services and propose a

constrained optimization scheme, which considers stochastic travel time information to increase the passengers' on-time arrival reliabilities and further incorporates the concept of late arrival compensation to increase the platform's profit. The contributions of our work are summarized as follows:

1) Developing a method for reliability-aware vehicle dispatch, which takes travel time uncertainty into account in both the allocation of requests to vehicles and the vehicle routing.
2) Developing an algorithm to estimate the probability of on-time arrival for a ride-sharing trip, which is then incorporated into the main dispatch method to optimize the reliability of service.
3) Developing a method to estimate the profit of a trip and maximize the profit of the platform, including the penalty costs due to late arrivals.
4) Conducting a case study to investigate the benefit of our travel time uncertainty aware dispatchers and compare them to the state-of-the-art deterministic approach.

II. Related Work

A recent study for Manhattan shows that up to 80% of taxi trips can be pairwise shared with very little increase in travel time, which translates into a 40% reduction of the taxi fleet [8]; this was later validated in multiple cities [9]. Inspired by the work of [8], a real-time algorithm for on-demand vehicle dispatch and large ride-sharing fleet management is developed in [3], which allows more than two passengers to share a vehicle. The efficiency of dispatch is further improved by exploring more possible ride-sharing combinations in limited computational time [10], taking into account predictions of future demands [6], routing idle vehicles to high demand areas [5], and considering the future effects of request allocations [7]. However, none of these approaches addresses the uncertainty of travel time.

There has been some work focusing on the vehicle routing problem with stochastic travel times. But it is limited to dispatching tens of vehicles, and cannot be applied to real-time systems because the computation time is up to 5 hours [11], [12]. More relevant to this paper is a stochastic ride-sharing model presented in [13], where a generalized trip cost is introduced and analysed for both driving-alone and ride-sharing trips. However, more than two passengers sharing a vehicle is not allowed and on-demand requests are not supported as passengers need to announce their trip schedule one day before. Our work differs from [13] by providing high-capacity on-demand ride-sharing.

A similar approach to our work, which also considers on-time arrival reliability in on-demand ride-sharing, is presented in [14]. They introduce a reliable path concept for selecting a vehicle with the maximum reliability for each request. However, the reliable path is computed from the precomputed k-shortest paths, which may not be the optimal one. Only pairwise sharing is allowed and the fleet size considered is less than 200. In contrast, we allow the dispatch of 3000 vehicles with a capacity of six. Moreover, we optimize the provider's profits considering the penalty costs due to late arrivals.

Unlike most previous work that optimizes the number of served requests, a dispatcher considering request price is developed in [15] to maximize the platform's profit. But it does not take into account the penalty costs arising from failures to arrive on time.

III. Reliability-Aware Ride-Sharing

We start by formulating the problem we are tackling and giving an overview of our proposed solution.

A. Definitions

Since journey requests appear throughout the day, the vehicle dispatch method presented in this paper adopts the industrial practice [16] where submitted requests are periodically packed and allocated to suitable vehicles. The dispatcher assigns requests at a time epoch ΔT, at which it batches a set of n new requests $R = \{r_1, \ldots, r_n\}$ and computes allocations. Each request is submitted by a passenger and defined as a tuple $\langle o_r, d_r, t_r \rangle$, where o_r is the origin (pick-up location), d_r is the destination (drop-off location) and t_r is the time of submission. Requests that can be served by a single vehicle through ride-sharing are grouped as a trip $\Gamma = \{r_1, \ldots, r_{n_\Gamma}\}$.

The dispatcher considers a fleet of m vehicles $V = \{v_1, \ldots, v_m\}$, of which the capacity is κ. The state of each vehicle is defined as a tuple $\langle q_v, s_v \rangle$, where q_v is the current position and s_v is the planned schedule, comprising a sequence of pick-up and drop-off tasks.

The dispatcher computes all feasible schedules $\mathit{\Gamma}_v = \{s_{v,\Gamma_1}, s_{v,\Gamma_2}, \ldots\}$ for each vehicle. A schedule $s_{v,\Gamma} = \{q_v, \ldots, o_1, \ldots, d_1, \ldots, o_2, \ldots, d_{n_\Gamma}\}$ is a sequence of visiting positions for a vehicle v to pick up and drop off passengers in a trip Γ. A detailed route travelling from q_v to d_{n_Γ} is denoted by π. A portion of the route that travels from q_v to d_r ($r \in \Gamma$) is denoted by $\pi(r)$. There is more than one possible route for a specific schedule, of which the optimal one is denoted by π^*, i.e., the one serving the requests with the highest reliability.

Following the assigned schedules, vehicles travel on a predefined road network $G = (I, E)$, where the travel times for each edge follow a Gaussian distribution $N(\mu_e, \sigma_e^2)$ and are independent of each other. It is shown in [17] that the independent Gaussian assumption is very similar to real world empirical data and holds well for stochastic planning. We define two functions, $\tau(i_1, i_2)$ and $\varrho(i_1, i_2)$, to compute the mean travel time and the travel distance of the minimum expected time path from i_1 to i_2, respectively.

B. Problem Formulation

Taking travel time uncertainty into consideration, we opt to optimize the service rate and the overall reliability at each round. Using R_{miss} to denote the set of requests that cannot be served at the current dispatch epoch, we define the objective of a dispatch policy as:

$$O_{Reliab} = \sum_{v \in V} best_prob(s_v) - \sum_{r \in R_{miss}} p_{miss} \qquad (1)$$

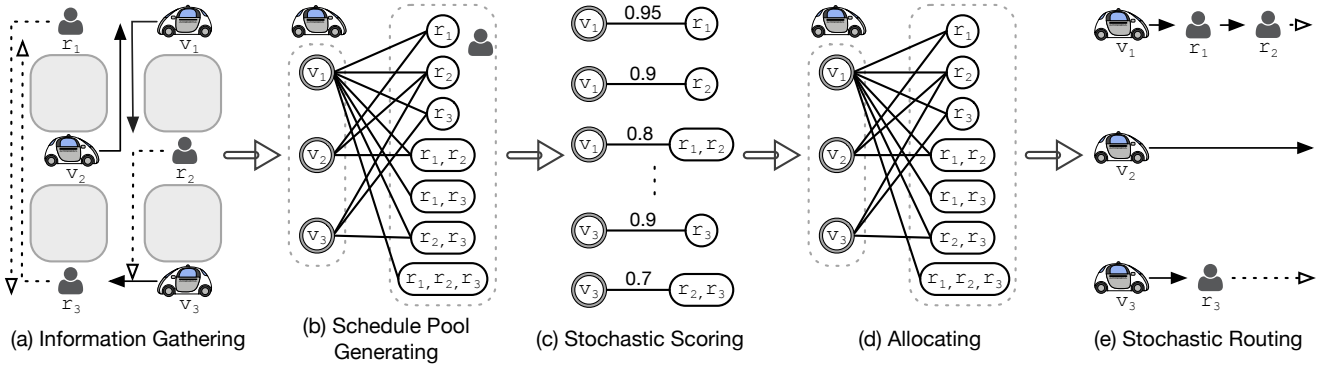

Fig. 2: Schematic overview of our proposed approach. (a) An example with three vehicles and three requests. The solid lines presents the current planned routes for vehicles and the dashed lines presents the shortest paths for requests. (b) A schedule pool that connects vehicles to servable ride-sharing trips. (c) Scored schedules with reliability information. (d) Allocation of requests to vehicles that maximizes the sores, where requests 1 and 2 are served by vehicle 1 and request 3 is served by vehicle 3. (e) Vehicles travelling on the stochastic optimal routes following the assigned schedules.

where $best_prob(s_v)$ is the maximum expected mean probability of dropping off requests on-time for a schedule and is affected by the routing policy (discussed in Section IV), p_{miss} is a large cost for not serving a request.

Problem 1 (Reliability-aware dispatch): Given a set of new requests R and a set of vehicles V for a dispatch epoch ΔT, the problem of reliability-aware dispatch is to assign vehicles particular schedules to serve requests, so that the objective (Eq. (1)) is maximized, subject to the following constraints:

- *Capacity constraint.* For each vehicle, the number of onboard passengers cannot be larger than its capacity.
- *Delay constraint.* For each request, its waiting time ω_r (the difference between when it is actually picked up and when it is submitted) and total travel delay δ_r (the difference between when it is actually dropped off and when it is expected to arrive if travelling alone) must be lower than two thresholds, Ω and Λ, respectively.
- *Precedence constraint.* For each request in a schedule $s_{v,\Gamma}$, its origin must be visited before its destination.

C. Method Overview

Fig. 2 illustrates the method to solve Problem 1. The processes for schedule generation (b) and request allocation (d) are inspired by [3]. The stochastic travel time information is incorporated in schedule routing (c) and vehicle routing (e). The reliability of a schedule is affected by the routing plan and the term $\sum_{v\in V} best_prob(s_v)$ in Eq. (1) is affected by the allocation results, which cannot be omitted. Fig. 1 shows an example. The main steps of the method are:

- Computing feasible ride-sharing combinations for each vehicle, along with schedules that satisfy the constraints defined in Problem 1, using the method proposed in [3].
- Scoring the feasible schedules using travel time distribution information.
- Allocating requests by solving an Integer Linear Program (ILP) to select each vehicle a schedule.
- Routing vehicles to follow the assigned schedules with optimal routes.

As stated in [3], given enough time, all possible ride-sharing trips can be found and the ILP-based allocations are optimal. However travel time distributions are not considered when generating the schedule pool, which could cause some potentially feasible trips to be mistakenly ignored [13]. In practice, longer delay constraints can be set so that searching for feasible ride-sharing combinations has a probabilistic guarantee. For example, setting the total travel delay constraint as $\Lambda^{'} = \Lambda + 3\sigma_\pi$, where σ_π is the largest standard variance of any possible ride-sharing route, will mean that less than 0.27% trips can be ignored.

IV. On-Time Arrival Probability Optimization

To maximize the reliability for a ride-sharing schedule, we adopt the approximate stochastic shortest path query algorithm of [18] and the multi-hop routing algorithm of [19] to on-demand ride-sharing systems. The generalized method consists of two procedures: preprocessing and online scoring. It can compute the maximum probability of dropping off requests on-time and return the corresponding optimal route for a schedule.

A. Preprocessing

Estimating the reliability of a schedule relies on computing the routes that maximize the probability of on-time arrival for each request in the schedule. For a request r travelling alone with a travel delay constraint Λ, we need to solve:

$$\pi^* = \underset{\pi\in\Pi}{\operatorname{argmax}} \quad \frac{\tau(o_r, d_r) + \Lambda - \mu_\pi}{\sigma_\pi} \tag{2}$$

where Π is the set of all possible routes from o_r to o_d.

Eq (2) can be solved exactly by iteratively finding the α-shortest path at most $n^{O(\log n)}$ times, of which the edge weight is defined as $\alpha\cdot\mu_e+\sigma_e^2$. There is a $\sqrt{1-\epsilon^2/(2+\epsilon^2)}$-approximation algorithm that runs in time polynomial in $1/\epsilon$, for any user-specified $\epsilon > 0$. Generalized from [18], the approximate solution can be found by enumerating α-shortest paths for $\alpha \in \{L, (1+\xi)L, (1+\xi)^2L, \ldots, U\}$, where $\xi = \epsilon/2$, $L = 2\min_e \sigma_e^2/d_L \geqslant 2\min_e \sigma_e^2/(\max_r \tau(o_r, d_r)+$

Algorithm 1 Find α Optimal Route

Input : a schedule $s_{v,\Gamma}$ and its pre-route π_{pre}
Output: the mean probability of on-time arrival $mean_prob$ and the α optimal route π^*_α

$mean_prob \leftarrow 0$
$\pi^*_\alpha \leftarrow \emptyset$
for each $\alpha \in A$ **do**
  $\pi \leftarrow Find\alpha ShortestPath(s_{v,\Gamma}, \pi_{pre})$
  $prob \leftarrow ComputeMeanProbability(\Gamma, \pi)$
  if $prob > mean_prob$ **then**
    $mean_prob \leftarrow prob$
    $\pi^*_\alpha \leftarrow \pi$

Algorithm 2 Schedule Reliability Estimation (SRE)

Input : a schedule $s_{v,\Gamma}$ and its pre-route π_{pre}
Output: the maximum mean probability of on-time arrival $best_prob$ and the optimal route π^*

$R_s \leftarrow GetRequestsInSchedule(s_{v,\Gamma})$
$best_prob \leftarrow 0$
$\pi^* \leftarrow \emptyset$
while $(r \leftarrow R_s.pop()) \neq \emptyset$ **do**
  $n_1, s^1_{v,\Gamma}, n_2, s^2_{v,\Gamma} \leftarrow CutSchedule(d_r, s_{v,\Gamma})$
  $prob_1, \pi_1 \leftarrow Find\alpha OptimalRoute(s^1_{v,\Gamma}, \pi_{pre})$
  $prob_2, \pi_2 \leftarrow SRE(s^2_{v,\Gamma}, \pi_1)$
  $mean_prob \leftarrow (prob_1 \cdot n_1 + prob_2 \cdot n_2)/(n_1 + n_2)$
  if $mean_prob > best_prob$ **then**
    $best_prob \leftarrow mean_prob$
    $\pi^* \leftarrow \pi_1 + \pi_2$

Λ) and $U = 2\sum_e \sigma_e^2/(\epsilon d_U) \leqslant 2\max_\pi \sigma_\pi^2/(\epsilon \min_e \mu_e)$. The term $\max_\pi \sigma_\pi^2$ is the largest variance of any possible ride-sharing route.

Denoting the set of the approximation parameters α as A, we precompute the shortest paths for all these values $\alpha \in A$ and store them in look-up tables. Therefore, the reliable route for a request can be found by checking the tables $|A|$ times.

B. Online Scoring

The maximum probability of visiting a schedule's last position $s_{v,\Gamma}[-1]$ and the correspond route can be found by enumerating α-shortest paths, as presented in [19]. Adopted from that, an algorithm to find the α-optimal route for a schedule is introduced in Algorithm 1. The pre-route π_{pre} is a predefined route from q_v to $s_{v,\Gamma}[0]$, defaulted to $\emptyset$. Function $Find\alpha ShortestPath(s_{v,\Gamma})$ returns the α-shortest path that visits a series of locations in the order set by schedule $s_{v,\Gamma}$. Function $ComputeMeanProbability(\Gamma, \pi)$ returns the mean probability of on-time arrival when requests are served by a predefined route π, denoted by:

$$mean_prob(s_{v,\Gamma}) = \frac{1}{|\Gamma|}\sum_{r\in\Gamma}\Phi\left(\frac{\tau(o_r, d_r) + \Lambda - \mu_{\pi(r)}}{\sigma_{\pi(r)}}\right) \tag{3}$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution.

Algorithm 1 tries to find the best α value that maximizes Eq. (3). However, the optimal route π^* may be a concatenation of the α-optimal paths for each request with different α values, as the optimal substructure property does not hold. We propose a recursive algorithm to explore all possible combination of different α values for the requests in a schedule, as shown in Algorithm 2. Function $GetRequestsInSchedule(s_{v,\Gamma})$ returns the set of requests included in schedule $s_{v,\Gamma}$. Function $CutSchedule(d_r, s_{v,\Gamma})$ breaks the schedule into two sub-schedules based on the drop-off position of r, the reliabilities and routes of which are then computed by Algorithms 1 and 2, respectively. Given a set of feasible schedules $\digamma_v$ for a vehicle, which is generated in step (b) as shown in Fig 2, they can be processed in parallel on-the-fly by Algorithm 2 to find the optimal routes truly maximizing $mean_prob(s_{v,\Gamma})$.

C. Reliable Allocating

At each round, the dispatcher considers the state of the fleet V, a set of requests R, and a set of scored schedules $\digamma_v$ for each vehicle. The goal is to serve as many requests as possible and dispatch the vehicles towards the most reliable trips so that the overall probability of on-time arrival is maximized. This is illustrated in Fig. 1(d) and formulated as the following Integer Linear Program:

$$\underset{x_{v,s},\epsilon_r}{\operatorname{argmax}} \quad \sum_{v\in V}\sum_{s\in\digamma_v} x_{v,s}\cdot best_prob(s) - \sum_{r\in R}\epsilon_r \cdot p_{miss} \tag{4}$$

$$s.t. \quad \sum_{s\in\digamma_v} x_{v,s} = 1, \quad \forall v \in V \tag{5}$$

$$\sum_{v\in V}\sum_{s\in\digamma_v} x_{v,s}\cdot\Theta_s(r) + \epsilon_r = 1, \quad \forall r \in R \tag{6}$$

where $best_prob(s) = \operatorname{argmax}_\pi \ mean_prob(s)$.

A binary variable $x_{v,s} \in \{0,1\}$ is introduced for each schedule in every $\digamma_v$, where $x_{v,s} = 1$ indicates that s is assigned to v. A binary variable $\epsilon_r \in \{0,1\}$ is introduced for each request, where $\epsilon_r = 1$ indicates that r is ignored. An indicator function $\Theta_s(r) \in \{0,1\}$ is introduced to indicate whether $r \notin s$ or $r \in s$. Constraint (5) guarantees that a vehicles is assigned exactly one schedule and constraint (6) guarantees that no requests are double-assigned to two vehicles. After allocating, each vehicle travels on the maximum-reliability route π^* to serve the requests.

V. Profit Optimization

In practice, profit is one of the main concerns of a platform and passengers may accept longer travel times if there is compensation. In this section, we introduce the concepts of request prices, travel costs and late arrival penalties to our proposed scheme to optimize the profit of dispatch.

Maximizing the platform's profit requires assigning high profit schedules to the vehicles. The profit of a schedule is equal to the price sum of included requests minus the expenses of vehicle travel and the compensation due to late

arrivals, defined as:

$$profit(s_{v,\Gamma}) = \sum_{r \in \Gamma} [rev(r) - penalty(r)] - cost(s_{v,\Gamma}) \quad (7)$$

where $rev(r)$ is the revenue of serving r, $penalty(r)$ is the compensation for dropping r off late, and $cost(s_{v,\Gamma})$ is the expense of the service. They are estimated as:

$$rev(r) = \beta \cdot \varrho(o_r, d_r) \quad (8)$$

$$penalty(r) = \gamma \cdot (ES(r) - \tau(o_r, d_r) - \Lambda) \quad (9)$$

$$cost(s_{v,\Gamma}) = \eta \cdot \varrho(\pi) \quad (10)$$

where β is the charge per unit distance, γ is the compensation per unit time, η is the expense per unit distance, $ES(r) = E[t_{\pi(r)} | t_{\pi(r)} > (\tau(o_r, d_r) + \Lambda)]$ is the expected travel time for r when violating the travel delay constraint, and $\varrho(\pi)$ is the travel distance of route π. Using the formulation for the expected shortfall of the Normal distribution in [20], the term $ES(r)$ can be computed as:

$$ES(r) = \mu_{\pi(r)} + \frac{\sigma_{\pi(r)} \cdot \varphi(\frac{\tau(o_r,d_r)+\Lambda-\mu_{\pi(r)}}{\sigma_{\pi(r)}})}{1 - \Phi(\frac{\tau(o_r,d_r)+\Lambda-\mu_{\pi(r)}}{\sigma_{\pi(r)}})} \quad (11)$$

where $\varphi(\cdot)$ is the probability density function of the standard normal distribution.

As $rev(r)$ is normally an upfront fare and fixed regardless the routing plan, and $\tau(o_r, d_r)$ and Λ are not affected by the routing plan, maximzing $profit(s_{v,\Gamma})$ is equivalent to solve:

$$\pi^* = \operatorname*{argmin}_{\pi} \sum_{r \in \Gamma} \gamma \cdot ES(r) + \eta \cdot \mu_\pi \quad (12)$$

Eq. (12) can be solved by enumerating the combinations of α-shortest paths, $\alpha \in A$, similar to Algorithm 2. Each schedule $s \in \mathcal{F}_v$ can find its optimal route that brings the maximum profit and be scored with that profit by solving Eq. (12). With these profit scored schedules, the objective function of vehicle dispatch becomes:

$$\operatorname*{argmax}_{x_{v,s}, \epsilon_r} \quad \sum_{v \in V} \sum_{s \in \mathcal{F}_v} x_{v,s} \cdot best_profit(s) - \sum_{r \in R} \epsilon_r \cdot p_{miss} \quad (13)$$

$$s.t. \quad \text{constraints (5) and (6)}$$

where $best_profit(s) = \operatorname{argmax}_\pi \ profit(s)$.

The objective (13) represents the sum of the expected profit each vehicle would earn and the total number of served requests, which aims to dispatch the vehicles towards the maximum-profit trips.

VI. Experimental Study

In this section, the proposed methods are implemented and evaluated using historical taxi request data from New York City [21], and compared to the state-of-the-art deterministic approach [3]. Since the implementation of [3] is unavailable, we reimplement it and run it on the same machine to ensure a fair comparison.

TABLE I: Parameter settings (defaults in bold).

Parameters	Settings
Fleet Size $\|V\|$	1000, **2000**, 3000
Max Waiting Time Ω (s)	180, **300**, 420
Capacity κ	2, 4, **6**, 8
Number of Requests $\|R\|$	**400k**, 600k, 800k

A. Simulation Details

The evaluations are conducted using request data from the 11th, 18th and 25th of May 2016. These have similar characteristics and are synthesized to three simulation scenarios of varying number of requests: 400k, 600k and 800k. Simulations are run for the hour with peak demand (19:00-20:00). A stochastic model of Manhattan is computed using the method in [8]. The complete road network contains 4,091 nodes and 9,452 directed edges. The travel times for each road segment for each hour of the day are computed using the pickup/drop off times of taxi trips. These are then processed to compute the daily mean and standard deviation for the travel time for each road segment.

Table I summarizes the major experimental control variables. We set the maximum travel delay to be twice the maximum wait time $\Lambda = 2\Omega$. The charge, compensation and expense parameters are set as $\beta = \$2$/km, $\gamma = \$0.02$/second and $\eta = \$1$/km. The dispatch epoch is set as $\Delta T = 30$ seconds as in [3]. The results are the averages of ten experiment runs.

B. Algorithm Comparison

We examine the effectiveness of the following algorithms:

- DVD (Deterministic Vehicle Dispatch): The method presented in [3] that maximizes the service rate without considering travel time uncertainty.
- RVD (Reliability-aware Vehicle Dispatch): The method from Section IV that considers the probability of on-time arrival for each request.
- PVD (Profit-aware vehicle Dispatch): The method from Section V that optimizes the profit of the platform.

The metrics used in this paper include: the service rate, the violation rate (percentage of late arrivals), the profit, the request distance served (related to the revenue from passengers), the vehicle distance travelled (related to the expense of the platform) and the vehicle mean load (average number of passengers in a vehicle).

C. Results

Fig. 3 shows the results of varying fleet sizes. The performance of all the algorithms improves for larger fleet sizes. A larger fleet brings more seats and allows the dispatcher to allocate more mid-size ride-sharing trips to reduce detours for passengers, which reduces the violation rate. By dispatching vehicles to less uncertain routes, RVP achieves lower violation rates than DVD. It also has higher service rates and more profits than DVD. RVD achieves a 0.77% lower violation rate than DVD when there are only 1000 vehicles, and with more flexibility brought in by 3000 vehicles, this reduction is increased to 7.3%. To achieve this significant

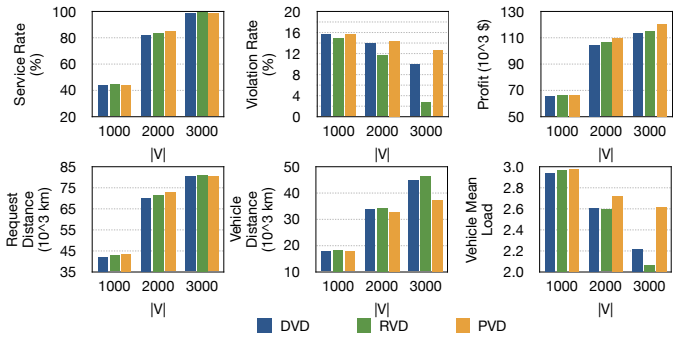

Fig. 3: A comparison of performance metrics for varying fleet sizes ($|R| = 400k, \kappa = 6, \Omega = 300$ s).

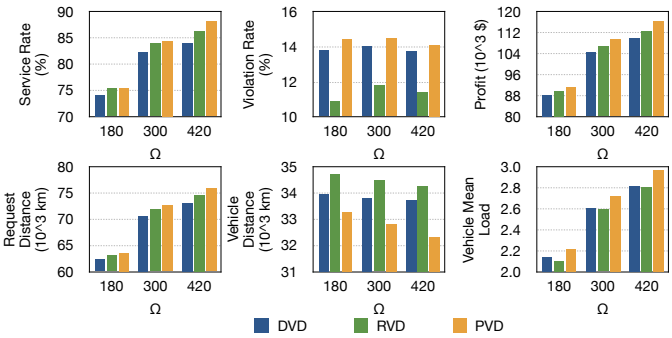

Fig. 4: A comparison of performance metrics for varying values of maximum waiting time constraint ($|R| = 400k, |V| = 2000, \kappa = 6$).

TABLE II: Performance metrics for different methods during the hour with lowest number of requests. ($|R| = 400k, |V| = 2000, \kappa = 6, \Omega = 300$ s)

Metrics	DVD	RVD	PVD
Service Rate (%)	99.57	99.81	99.43
Violation Rate (%)	4.29	0.27	8.26
Profit ($)	7,671.04	7,085.13	9,262.84
Request Distance (km)	7,375.03	7,378.08	7,359.96
Vehicle Distance (km)	7,009.79	7,665.62	5,304.61
Vehicle Mean Load	1.21	1.03	1.62

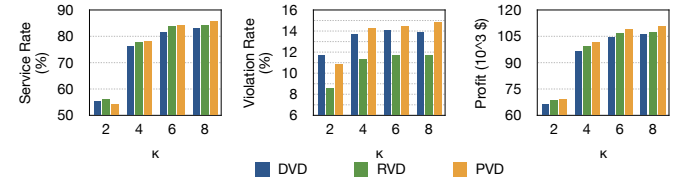

Fig. 5: A comparison of performance metrics for varying vehicle capacities ($|R| = 400k, |V| = 2000, \Omega = 300$ s).

reduction in the violation rate compared to DVD, RVD tends to dispatch small-size ride-sharing trips and has a longer vehicle distance travelled. To make more profit, PVD accepts late arrivals and prefers allocating large-size ride-sharing trips to save on vehicle travel expenses. Large fleet size also increases the increase in profit brought in by PVD, which makes 6.08% more profit than DVD when the fleet size is 3000. The vehicle mean load of PVD is 2.62 when there are 3000 vehicle, this high occupancy rate yields a considerable saving in expense, as vehicle distance travelled is much lower than other two algorithms. When the number of vehicles is sufficiently large, both RVD and PVD can yield more improvements in reliability and profit, respectively, at the cost of profit and reliability, respectively. This is also validated in the results of simulations during 4:00-5:00, when the number of requests is only 6.85% of the peak hour. As shown in Table II, the service rates of all algorithms are very close to 100% due to a sufficient number of vehicles. RVD achieves a close to zero violation rate, but has a 7.64% lower profit than DVD. Similarly, PVD makes a 20.75% higher profit than DVD, with a 3.97% higher violation rate.

Fig. 4 shows the results of varying values of the maximum waiting time constraint. A longer travel delay constraint allows more detours and increases the occupancy rate, thus leading to higher service rates and more profits. The violation rates do not significantly change, except that RVD yields a 0.84% increase when the value of the maximum waiting time constraint increases from 180 s to 300 s. As the constraint increases from 180 s to 420 s, while the reduction in violation rate yielded by RVD over DVD has a small decrease from 2.9% to 2.3%, the increase in service rate increases from 1.19% to 2.21%. PVD utilizes the increase in detour tolerance and achieves higher improvements in both the service rate (from 1.23% to 4.22%) and the profit (from 3.43% to 6.01%).

Fig. 5 shows the results of varying vehicle capacities. All algorithms achieve higher service rates and more profits with higher capacity, as more seats allow more requests to share their trips. The violation rate increases as capacity rises, which may be due to detours caused by ride-sharing. The performance differences among DVD, RVD and PVD are not significantly affected by differences in vehicle capacity, especially when it comes to high-capacity ($\kappa \geq 4$) ride-sharing, where the reductions in violation rates brought in by RVD over DVD are around 2.30($\pm$ 0.05)% and the increases in profit brought by PVD over DVD are around 4.89($\pm$ 0.05)%. PVD has a lower service rate than DVD when capacity is 2, this is because the expense savings are low due to low-capacity ride-sharing and some requests are not profitable, so PVD chooses to ignore them.

Fig. 6 shows the results of varying instance scales. When the number of requests increases, the fleet size is also increased to examine the scalability of the algorithms. The reductions in violation rates brought in by RVD over DVD are 2.30%, 1.98% and 2.13%, respectively. The improvements in profit produced by PVD over DVD are 4.87%, 5.69% and 6.40%, respectively. This indicates that a ride-sharing service can benefit from taking into account the stochastic travel time, regardless of the scale.

We further investigate the effect of the penalty value considered by PVD, as shown in Table III. Although a larger penalty for late arrivals results in a lower profit due to the increase in compensation, it improves all other metrics. In practice, passengers would prefer a low delay constraint, with which PVD yields fewer improvements (as shown in Fig. 4). A larger compensation could increase passengers' tolerance to delays and eventually increase the profit of the platform.

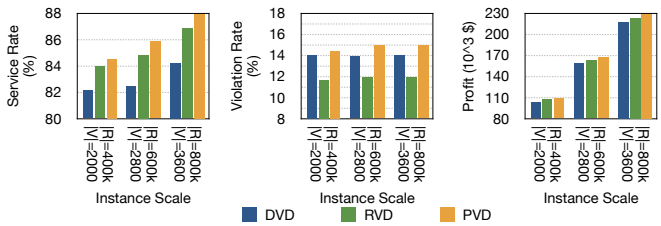

Fig. 6: A comparison of performance metrics for varying instance scales ($\kappa = 6, \Omega = 300$ s).

TABLE III: Performance metrics for varying penalties for PVD. ($|R| = 400k, |V| = 2000, \kappa = 6, \Omega = 300$ s)

Metrics	Penalty		
	0.02	0.06	0.10
Service Rate (%)	84.58	85.34	86.10
Violation Rate (%)	14.50	14.02	12.96
Profit (10^3 $)	109.67	105.19	101.50
Relative Profit Difference With Respect to DVD (%)	4.87	6.03	8.13

D. Discussion

Taking the uncertainty of travel time into consideration significantly improves the performance of vehicle dispatch. Both RVD and PVD achieve more gains over DVD for larger fleet sizes and delay constraints, especially when the number of vehicles is sufficient to handle almost all requests. Also, the performances of RVD and PVD scale well across different vehicle capacities and instance sizes. Although our assumption that edge travel times follow independent Gaussian distributions may limit the accuracy of our model, experiments with different variances, $\{0.5, 1, 2\} \cdot \sigma_e^2$, showed that the improvements yielded by RVD and PVD are stable, suggesting that the proposed approach is robust to different stochastic travel time models.

VII. Conclusion

In this paper, we have presented a travel time uncertainty aware vehicle dispatch scheme for on-demand ride-sharing, a method to allocate the most reliable trips and a method to optimize the profit of vehicle dispatch. Numerical simulations on Manhattan taxi datasets show that, by considering stochastic travel times, the proposed methods improve upon the state-of-the-art deterministic dispatcher in terms of the reliability (up to 7.3% at peak hour), the profit (up to 8.13% at peak hour) and the service rate (up to 4.22% at peak hour). Typically, a 1% improvement is considered significant in an industrial setting [22]. Future work will investigate the trade-off between the reliability and the profit. We also plan to incorporate demand prediction methods to further enhance the performance of on-demand ride-sharing.

References

[1] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[2] Y. Tong, Y. Zeng, Z. Zhou, L. Chen, J. Ye, and K. Xu, "A unified approach to route planning for shared mobility," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1633–1646, 2018.

[3] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.

[4] J. Wen, J. Zhao, and P. Jaillet, "Rebalancing shared mobility-on-demand systems: A reinforcement learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 220–225, Ieee, 2017.

[5] A. Wallar, M. Van Der Zee, J. Alonso-Mora, and D. Rus, "Vehicle rebalancing for mobility-on-demand systems with ride-sharing," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4539–4546, IEEE, 2018.

[6] J. Alonso-Mora, A. Wallar, and D. Rus, "Predictive routing for autonomous mobility-on-demand systems with ride-sharing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3583–3590, IEEE, 2017.

[7] S. Shah, M. Lowalekar, and P. Varakantham, "Neural approximate dynamic programming for on-demand ride-pooling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 507–515, 2020.

[8] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 37, pp. 13290–13294, 2014.

[9] R. Tachet, O. Sagarra, P. Santi, G. Resta, M. Szell, S. H. Strogatz, and C. Ratti, "Scaling law of urban ride sharing," *Scientific reports*, vol. 7, no. 1, pp. 1–6, 2017.

[10] M. Lowalekar, P. Varakantham, and P. Jaillet, "Zac: A zone path construction approach for effective real-time ridesharing," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, pp. 528–538, 2019.

[11] D. Taş, N. Dellaert, T. Van Woensel, and T. De Kok, "Vehicle routing problem with stochastic travel times including soft time windows and service costs," *Computers & Operations Research*, vol. 40, no. 1, pp. 214–224, 2013.

[12] F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau, "The vehicle routing problem with hard time windows and stochastic service times," *EURO Journal on Transportation and Logistics*, vol. 7, no. 3, pp. 223–251, 2018.

[13] J. Long, W. Tan, W. Szeto, and Y. Li, "Ride-sharing with travel time uncertainty," *Transportation Research Part B: Methodological*, vol. 118, pp. 143–171, 2018.

[14] Z. Liu, T. Miwa, W. Zeng, M. G. Bell, and T. Morikawa, "Dynamic shared autonomous taxi system considering on-time arrival reliability," *Transportation Research Part C: Emerging Technologies*, vol. 103, pp. 281–297, 2019.

[15] L. Zheng, L. Chen, and J. Ye, "Order dispatch in price-aware ridesharing," *Proceedings of the VLDB Endowment*, vol. 11, no. 8, pp. 853–865, 2018.

[16] L. Zhang, T. Hu, Y. Min, G. Wu, J. Zhang, P. Feng, P. Gong, and J. Ye, "A taxi order dispatch model based on combinatorial optimization," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 2151–2159, 2017.

[17] S. Lim, H. Balakrishnan, D. Gifford, S. Madden, and D. Rus, "Stochastic motion planning and applications to traffic," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 699–712, 2011.

[18] S. Lim, C. Sommer, E. Nikolova, and D. Rus, "Practical route planning under delay uncertainty: Stochastic shortest path queries," in *Robotics: Science and Systems*, vol. 8, pp. 249–256, 2013.

[19] S. Lim and D. Rus, "Stochastic motion planning with path constraints and application to optimal agent, resource, and route planning," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4814–4821, IEEE, 2012.

[20] M. Norton, V. Khokhlov, and S. Uryasev, "Calculating cvar and bpoe for common probability distributions with application to portfolio optimization and density estimation," *Annals of Operations Research*, pp. 1–35, 2019.

[21] N. Taxi and L. Commission, "Tlc trip record data." `https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page`, June 2019. Accessed: 2020-01-12.

[22] Z. Xu, Z. Li, Q. Guan, D. Zhang, Q. Li, J. Nan, C. Liu, W. Bian, and J. Ye, "Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 905–913, 2018.