2018 IEEE 34th International Conference on Data Engineering

An Efficient Ride-Sharing Framework for Maximizing Shared Routes

(Extended Abstract)

Na Ta[†] Guoliang Li[‡] Tianyu Zhao[‡] Jianhua Feng[‡] Hanchao Ma[§] Zhiguo Gong[§§]

[†]School of Journalism and Communication, Renmin University of China, Beijing, China

[‡]Department of Computer Science, Tsinghua University, Beijing, China

[§]Department of Electronic Engineering and Computer Science, Washington State University, Pullman, USA

[§§]Department of Computer Science, Macau University, Macau, China

tanayun@ruc.edu.cn, {liguoliang, fengjh, zhaoty17}@tsinghua.edu.cn, mahanchao@hotmail.com, fstzgg@umac.mo

***Abstract*—Ride-sharing (RS) has great values in saving energy and alleviating traffic pressure. In this paper, we propose a new ride-sharing model, where each driver requires that the shared route percentage (SRP, the ratio of the shared route's distance to the driver's total traveled distance) exceeds her expected rate (e.g., 0.8) when sharing with a rider. We consider two variants of this problem. The first considers multiple drivers and multiple riders, and aims to compute a set of driver-rider pairs to maximize the overall SRP. We model this problem as the maximum weighted bigraph matching problem. We propose an effective exact algorithm, and an efficient approximate solution with error-bound guarantee. The second considers multiple drivers and a single rider and aims to find the top-k drivers for the rider with the largest SRP. We devise pruning techniques and propose a best-first algorithm to progressively selects drivers with high probability to be in the top-k results.**

I. Introduction

In this paper, we focus on the ride-sharing problem. We consider a new service called *hitch* in Uber and Didi. In the hitch service, there are a large number of part-time drivers, where each driver intends to drive from a source location to a destination and is willing to share the vehicle with a rider. If a driver shares the vehicle with a rider, the driver needs to first drive to the source of the rider from the driver's source, then drive to the destination of the rider from the rider's source (which is called the shared route between the driver and the rider), and finally drive to the destination of the driver from the rider's destination. The shared route percentage (SRP) is the ratio of the shared-route distance to the total route distance (i.e., the sum of the distances of the three parts). Since the driver must drive on the road networks, we consider the road-network distance and we assume that the driver will select the shortest route between two locations. Usually, the driver has a requirement that the SRP value must exceed an expectation value, e.g., 0.8, to guarantee a high sharing utility.

There are two variants of the ride-sharing (`RS`) problem.

(1) `Join-based RS`. There are a group of drivers and a group of riders and it aims to compute the driver-rider pairs that maximize the overall SRP.

(2) `Search-based RS`. There are a group of drivers and a single rider, and it aims to compute the top-k drivers for the rider with the largest SRP values.

Example 1: Figure 1 visualizes drivers d_1 - d_3 and riders r_1 - r_4 on a road network. For rider r_1, d_2 is an *invalid* driver as the SRP that d_2 takes r_1 (0.524) does not meet d_2's requirement(0.6); while d_1 and d_3 are *valid* drivers for r_1. Pair $\langle d_1, r_1\rangle$ denotes that d_1 picks up r_1. For `Join-based RS`, we want to find a set of driver-rider pairs that maximize the overall SRP, such as $\{\langle d_1, r_1\rangle, \langle d_2, r_3\rangle, \langle d_3, r_2\rangle\}$. For `Search-based RS`, given rider r_3, d_2 is the top-1 driver.

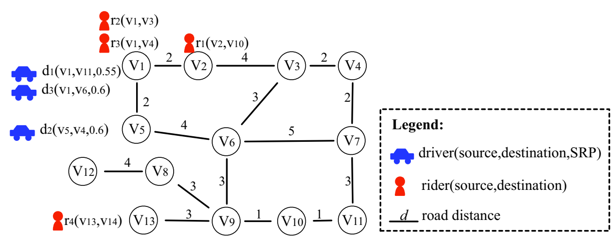


Fig. 1. Running Example.

In this paper, we make the following contributions. (1) We formally define the `Join-based RS` and `Search-based RS` problems and propose an efficient ride-sharing framework. (2) For `Join-based RS`, we devise an exact algorithm to compute the optimal driver-rider pairs. We propose an approximate algorithm to efficiently compute the driver-rider pairs with any given approximate rate. (3) We develop a best-first method for the `Search-based RS` problem to efficiently compute the top-k drivers for a rider. (4) We conduct extensive experiments to demonstrate the high quality and efficiency of our proposed methods.

II. Problem Formulation

A. Preliminaries

We use graph $G\langle V, E\rangle$ to model a road network, where each vertex $v \in V$ denotes a geo-location (e.g., road intersection), and each edge $(u, v) \in E$ is a road segment with a weight (e.g., road distance). A route in G is a connected path, and the route distance is the sum of the distance of each edge on the route. Given two road vertices u, v, we use $\delta(u, v)$ to denote their shortest route distance.

A **rider** $r_j = (r_j^{\triangleright}, r_j^{\triangleleft})$ plans to travel from source location $r_j^{\triangleright}$ to destination location $r_j^{\triangleleft}$ by taking a shared ride with a driver d_i. A **driver** $d_i = (d_i^{\triangleright}, d_i^{\triangleleft}, \pi_i)$ plans to travel from a source location $d_i^{\triangleright}$ to a destination location $d_i^{\triangleleft}$, and is willing to share the ride with a rider r_j if the **shared route percentage** $\pi(d_i, r_j)$ is no less than π_i ($\pi_i \in [0, 1]$), where $\pi(d_i, r_j) = \frac{\delta(r_j^{\triangleright}, r_j^{\triangleleft})}{\delta(d_i^{\triangleright}, r_j^{\triangleright}) + \delta(r_j^{\triangleright}, r_j^{\triangleleft}) + \delta(r_j^{\triangleleft}, d_i^{\triangleleft})}$.

2375-026X/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDE.2018.00255

B. Problem Formulation

Join-based Ride Sharing. Given a drivers set $D = \{d_i\}$ and a riders set $R = \{r_j\}$, we aim to assign each rider r_j to a *valid driver* d_i where $\pi(d_i, r_j) \geq \pi_i$, in order to maximize the overall SRP. $\mathcal{C} = \{(d_i, r_j) | \pi(d_i, r_j) \geq \pi_i\}$ denotes the set of valid driver-rider pairs, $\mathcal{C}_j$ is the set of valid drivers for r_j.

We construct a weighted bigraph $\mathcal{G}(\mathcal{G}_R, \mathcal{G}_D, \mathcal{G}_E)$: $\mathcal{G}_R = R$, $\mathcal{G}_D = D$; if $\pi(d_i, r_j) \geq \pi_i$, an edge is added between d_i and r_j with weight $\pi(d_i, r_j)$. Then we aim to find a subgraph from $\mathcal{G}$ of driver-rider pairs, such that one rider is assigned to at most one driver, and vice versa:

Definition 1 (Matching Plan): A matching plan P_x is a subgraph of $\mathcal{G}$ if no two edges share a common bigraph vertex.

The matching plan with the largest overall weight (i.e., total SRP) is the *optimal* matching plan.

Definition 2 (`Join-based RS`*):* Given drivers set $D = \{d_i\}$, riders set $R = \{r_j\}$, road network G, find the optimal plan P^*: $P^* = \arg\max_{P_x} \sum_{\forall \langle d_i, r_j \rangle \in P_x} \pi(d_i, r_j)$

Search-based Ride Sharing. For any rider r_j, we aim to find the top-k drivers with the largest SRP values.

Definition 3 (`Search-based RS`*):* For rider r_j on road network G, find a k-size valid driver set $\mathcal{C}_j^k = \{d_{i^*}\}$, where
(1) $\mathcal{C}_j^k \subseteq \mathcal{C}_j$, $|\mathcal{C}_j^k| = k$, and
(2) $\forall d_{i^*} \in \mathcal{C}_j^k, \forall d_i \in \mathcal{C}_j \backslash \mathcal{C}_j^k, \pi(d_{i^*}, r_j) \geq \pi(d_i, r_j)$.

III. Algorithm Framework

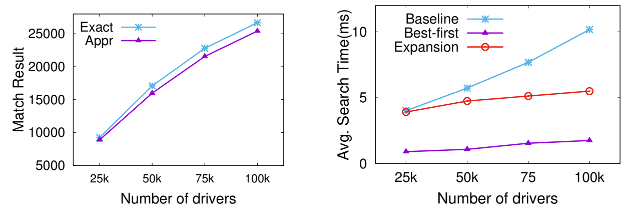
A. Exact Method for Join-Based RS

Invalid Pair Pruning. We construct a candidate set $\tilde{\mathcal{C}} = \{\langle d_i, r_j \rangle | d_i \in D_j^{\triangleright} \cap D_j^{\triangleleft}, \forall r_j \in R\}$, where $D_j^{\triangleright}$ is the set of potentially valid drivers computed using $r_j^{\triangleright}$, and $D_j^{\triangleleft}$ is the set of potentially valid drivers computed using $r_j^{\triangleleft}$. As $|D_j^{\triangleright}| << |D|$ and $|D_j^{\triangleright}| << |D|$, the computational cost is greatly reduced. Set $\tilde{\mathcal{C}}$ is *complete*, i.e., $\mathcal{C} \subseteq \tilde{\mathcal{C}}$.

The Framework for Exact Join-Based RS.
(1) **Candidate Set Generation.** For each rider r_j, we compute $D_j^{\triangleright} = \{d_i | \delta(d_i^{\triangleright}, r_j^{\triangleright}) \leq (\frac{1}{\pi_0} - 1)\delta(r_j^{\triangleright}, r_j^{\triangleleft})\}$ and $D_j^{\triangleleft} = \{d_i | \delta(r_j^{\triangleleft}, d_i^{\triangleleft}) \leq \delta_j^{\pi_0}\}$ [4], where π_0 is the minimum SRP value for all drivers. Thus the candidate driver set for r_j is $\tilde{\mathcal{C}}_j = D_j^{\triangleright} \cap D_j^{\triangleleft}$, and $\tilde{\mathcal{C}} = \cup_{r_j \in R}\{\langle d_i, r_j \rangle | d_i \in \tilde{\mathcal{C}}_j\}$.
(2) **Constructing the bigraph.** For each pair $\langle d_i, r_j \rangle \in \tilde{\mathcal{C}}$, we compute the real SRP value $\pi(d_i, r_j)$. If $\pi(d_i, r_j) \geq \pi_i$, we add an edge between d_i and r_j in the bigraph.
(3) **Computing the optimal matching plan.** The problem is essentially a maximum weighted bigraph matching problem and we use the minimal cost network flow algorithm [1] to compute the optimal matching plan P^*.

B. Approximate Method for Join-based RS

we propose a two-stage approximate method. First, we construct two bigraphs that bounds the accurate answer P^* within range $[\mathtt{LB}(P), \mathtt{UB}(P)]$. Then, we gradually update certain edges in the bound graphs into exact SRP values, so that $\mathtt{LB}(P)$ and $\mathtt{UB}(P)$ can converge to a predefined threshold τ, i.e., $\frac{\mathtt{UB}(P)}{\mathtt{LB}(P)} \leq \tau$. The final $\mathtt{LB}(P)$ value is the bounded approximate answer to P^*, and $\frac{P^*}{\mathtt{LB}(P)} \leq \tau$.



(a) `Join-based RS` Effectiveness. ($\pi_0 = 0.8, \tau = 1.5$) (b) `Search-based RS` Efficiency. ($\pi_0 = 0.7, k = 3$)

Fig. 2. Selected Experiment Results on `Taxi` Dataset.

C. Exact Methods for Search-based RS

Similar to `Join-based RS`, we first prune a large number of drivers that cannot meet the SRP requirements. Then we propose a **best-first** algorithm that progressively selects the drivers with high probability to be in the top-k results and prunes the drivers that cannot be in the top-k results [3].

IV. Experimental Study

Datasets. We use the road network of Beijing with 338,024 vertices and 440,525 edges [2]. We use two ride-sharing datasets, `Taxi` and `UCar`. (1) `Taxi` contains about $200,000$ trajectories of user orders generated by more than 8,000 public taxicabs in one month in Beijing. (2) `UCar` contains $1,267,000$ trajectories of user orders generated by nearly 2,000 private drivers within one week in Beijing. We extract the pick-up and drop-off locations from `Taxi` and `UCar` and randomly assign them to a set of drivers and a set of riders.

Evaluating Join-based RS. We implement the `Exact` and the `Appr` solutions. We evaluate the efficiency, effectiveness, and their trends by varying parameters π_0 and τ (the bound ratio threshold for $\mathtt{UB}(P)/\mathtt{LB}(P)$ in `Appr` iteration). The `Appr` solution achieves high performance in all test cases (Figure 2(a)).

Evaluating Search-based RS. We compare three strategies: (1) `Baseline`: searches all possible candidate drivers, computes all SRP values and selects the top k drivers (2) `Expansion`: expansion-based method that gradually increases the candidate drivers set for a single rider. (3) `Best-first`: only computes k driver-rider shared route percentages in the best scenario. Experiments show that the `Best-first` method is the most efficient and scalable technique for `Search-based RS` (Figure 2(b)).

Acknowledgment

This work was supported by the 973 Program of China (2015CB358700), NSF of China (61632016, 61373024, 61602488, 61422205, 61472198), ARC DP170102726, FDCT/007/2016/AFJ, Key Grant Project on Humanities and Social Sciences of MOE of China (16JJD860008),"Double First-Class" Project, SJC, RUC (17RDXW002).

References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows - theory, algorithms and applications*. Prentice Hall, 1993.
[2] H. Hu, G. Li, Z. Bao, Y. Cui, and J. Feng. Crowdsourcing-based real-time urban traffic speed estimation: From trends to speeds. In *ICDE*, pages 883–894, 2016.
[3] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, and Z. Gong. An efficient ride-sharing framework for maximizing shared route. *IEEE TKDE*, (99):1–15, 2017.
[4] R. Zhong, G. Li, K. Tan, L. Zhou, and Z. Gong. G-tree: An efficient and scalable index for spatial search on road networks. *IEEE TKDE*, 27(8):2175–2189, 2015.