Real-time Personalized Taxi-Sharing

Xiaoyi Duan[†], Cheqing Jin✉[†], Xiaoling Wang[†], Aoying Zhou[†], and Kun Yue[‡]

[†]Institute for Data Science and Engineering, School of Computer Science and Software Engineering, East China Normal University, China
[‡]School of Information Science and Engineering, Yunnan University, Kunming, China
dollyxiaoyi@gmail.com {cqjin, xlwang, ayzhou}@sei.ecnu.edu.cn
kyue@ynu.edu.cn

Abstract. Taxi-sharing is an efficient way to improve the utility of taxis by allowing multiple passengers to share a taxi. It also helps to relieve the traffic jams and air pollution. It is common that different users may have different attitudes towards the taxi-sharing scheduling plan, such as the fee to be paid and the additional time to the destination. However, this property has not been paid enough attention to in the traditional taxi-sharing systems – the traditional focus is how to decrease the travel distance. We study the problem of personalized taxi-sharing in this paper, with the consideration of each passenger's preference in payment, travel time and waiting time. We first define the satisfaction degree of each party involved in the scheduling plan, based on which two goals are defined to evaluate the overall plan, including MaxMin and MaxSum. Subsequently, we devise a two-phase framework to deal with this problem. The statistical information gathered during the offline phase will be used to hasten query processing during the online phase. Experimental reports upon the real dataset illustrate the effectiveness and efficiency of the proposed method.

1 Introduction

The rapid increment of vehicles in large cities brings with some serious social issues, such as traffic jams and air pollution [1–3]. Taxi is an important means of transport because it is convenient for people to go to the destination quickly. However, the benefits of taxi have not been fully exploited since one taxi is allowed to take only one passenger or one group of passengers, leaving some seats unoccupied. Consequently, it is meaningful to devise a taxi-sharing system that allows more than one group of passengers to share one taxi, so as to relieve the effects of traffic jams and air pollution.

Actually, some attempts have been made to fulfill the idea of vehicle sharing. For example, some Web sites (e.g, AApinche) allow uses to publish their travel plan or to find a matched partner from the existing posts. However, it usually takes a long time before users can find their matches [4, 5]. Recently, some works focus on devising a ride-sharing strategy, which allows multiple groups of passengers sharing one vehicle [6, 7]. The T-share system aims at finding a plan to

minimize the total travel distance for each request [8]. Unfortunately, none of the above works consider the customized preference of each user. Our pruning rules fully take account of users' preferences.

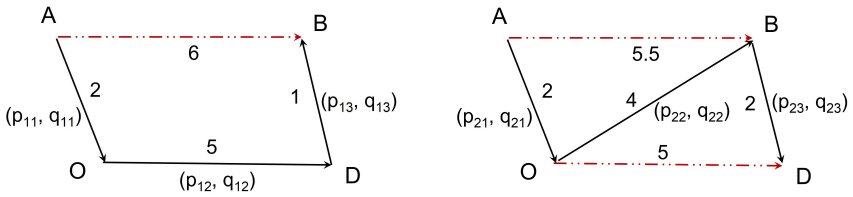
In general, a successful schedule plan should satisfy each party in this plan. Otherwise, this plan may not go into reality. There are three parties in a taxi-sharing plan: driver, taxi rider, and waiting passenger. Taxi rider is the passenger who has already ridden on the taxi, and waiting passenger is the one who is hailing a taxi [9]. Each user has personalized requirement about the satisfaction. Someone is sensitive to the fee, but insensitive to the driving time, while someone else may be insensitive to the fee but wants to go to the destination as soon as possible. Therefore, a customized taxi-sharing plan needs to fully consider users' preferences and then we designed the plan.

However, it is challenging to deal with personalized taxi-sharing due to the following reasons. (i) *Real-time service requirement.* It is critical to send the response to users as soon as possible. However, it is non-trivial to achieve the goal due to the huge amount of taxis and the dynamic properties of taxis. (ii) *Personalized preference.* Similar route is not the only factor to be concerned in taxi sharing, because customers' service preferences are too diverse to compose a sharing plan, such as sharing purpose and requirements to partners. (iii) *Different route matching.* As most users in our system tend to have different origins and destinations, it is not likely to find a taxi rider with the same origin and destination. In order to enhance the sharing chance, we need to relax the matching condition. To handle the aforementioned challenges, we propose a novel framework in this paper, which contains two phases: *offline* and *online*. The goal of the *offline* phase is to compute some statistical information to support online computation. For example, we can estimate the upper and lower bounds of the driving time between two points. During the *online* phase, some pruning rules are devised to hasten query processing.

The contributions are summarized below.

- We formalize the personalized taxi-sharing issue which fully considers users' preferences. We define *satisfaction* degree of a person (driver, waiting passenger, or taxi rider) and two global targets, MaxSum and MaxMin, to measure the quality of the overall plan.
- We deal with the real-time taxi-sharing issue efficiently to support the scenario with huge number of taxis and passengers. To hasten query processing of online phase, we precompute several operators during offline phase and use pruning rules to save significant part of computations.
- We conduct extensive experiments on the real trajectory dataset to validate the effectiveness and efficiency of the proposed method.

The remainder of this paper is organized as follows. Section 2 introduces the preparatory work of data model and problem statement. Section 3 introduces the framework of our new method. Section 4 describes the offline phase that precompute some statistics. Section 5 describes the online phase to process queries efficiently by using pruning rules. Section 6 reports experimental results upon



(a) WPF plan (b) TRF plan

Fig. 1. Taxi-sharing Plans (WPF and TRF)

real data sets. Section 7 reviews related work. A brief conclusion is given in the last section.

2 Preliminaries

Data model: A typical taxi-sharing system involves three parties, including taxi driver (shorten as TD), taxi rider (shorten as TR), and waiting passenger (shorten as WP) [1]. Since TR and WP may have different origins and destinations, taxi sharing plans can be diverse. Assuming a WP plans to go from O to D, and a TR is traveling from A to B, there exist two possible scheduling plans to share this taxi, namely WPF (Waiting Passenger First) and TRF (Taxi Rider First). As illustrated in Figure 1, the route in WPF plan is $AODB$, as the taxi goes to D at first after picking up WP at O, while the route in TRF plan is $AOBD$, as the taxi goes to B at first instead. In this paper, we focus on the scenario that a taxi is shared by at most two groups of passengers, and leave the scenario that multiple groups sharing one taxi as a piece of future work.

Pricing Strategy: The pricing strategy varies a lot in different cities, e.g, the unit price may change by different time and places. Here, we consider a model for the illustration purpose where the per-kilometer price is fixed, denoted as k. The detail of pricing strategy is described below.

Let $\overrightarrow{D}_{WPF} = (\mathsf{rdis}(A,O), \mathsf{rdis}(O,D), \mathsf{rdis}(D,B))$ denote a distance vector in the WPF plan, where $\mathsf{rdis}(l_1, l_2)$ denotes the road network distance between two locations l_1 and l_2. Let $\overrightarrow{p_1} = k \cdot (p_{11}, p_{12}, p_{13})$ and $\overrightarrow{q_1} = k \cdot (q_{11}, q_{12}, q_{13})$ ($\forall i$, $p_{1i}, q_{1i} \in [0,1]$) denote the payment share for WP and TR in the WPF plan respectively (see Figure 1(a)). Then, the payment for WP (denoted as M_{WP}), TR (denoted as M_{TR}), and the income for TD (denoted as M_{TD}) are computed as: $M_{WP} = \overrightarrow{p_1} \cdot \overrightarrow{D}_{WPF}$, $M_{TR} = \overrightarrow{q_1} \cdot \overrightarrow{D}_{WPF}$, and $M_{TD} = M_{WP} + M_{TR}$. To ensure TD always gets more income, the parameters satisfy: $\forall i$, $p_{1i} + q_{1i} \geq 1$.

Symmetrically, let $\overrightarrow{D}_{TRF} = (\mathsf{rdis}(A,O), \mathsf{rdis}(O,B), \mathsf{rdis}(B,D))$ denote the distance vector in the TRF plan. Let $\overrightarrow{p_2} = k \cdot (p_{21}, p_{22}, p_{23})$ and $\overrightarrow{q_2} = k \cdot (q_{21}, q_{22}, q_{23})$ ($\forall i$, $0 \leq p_{2i}, q_{2i} \leq 1$) denote the payment share for WP and TR re-

[1] Note that TR or WP can also be a group of people rather than one person.

Table 1. An example of pricing strategy based on Figure 1

Plan	$\overrightarrow{\mathbf{p}}$	$\overrightarrow{\mathbf{q}}$	M_{WP}	M_{TR}	M_{TD}
WPF	$2 \times (0.5, 0.6, 0.5)$	$2 \times (0.6, 0.6, 0.6)$	9	9.6	18.6
TRF	$2 \times (0.5, 0.6, 1)$	$2 \times (0.6, 0.6, 0)$	10.8	7.2	18

Note: k is 2 RMB/km.

spectively (see Figure 1(b)). Accordingly, $M_{WP} = \overrightarrow{p_2} \cdot \overrightarrow{D}_{TRF}$, $M_{TR} = \overrightarrow{q_2} \cdot \overrightarrow{D}_{TRF}$, and $M_{TD} = M_{WP} + M_{TR}$.

Example 1. Table 1 illustrates a pricing strategy for Figure 1. For example, in the WPF plan, the payments of WP and TR are computed as: $M_{WP} = 2 \times (0.5 \times 2 + 0.6 \times 5 + 0.5 \times 1) = 9$, $M_{TR} = 2 \times (0.6 \times 2 + 0.6 \times 5 + 0.6 \times 1) = 9.6$.

Preference and Satisfaction: The personalized preference of each user (either WP or TR) is defined below.

Definition 1 (Preference). *The preference of a user (either WP or TR) is defined as* (α, β)*, where* α *controls user preference to the time cost, and* β *controls user preference to the payment.* $\alpha \geq 0$*,* $\beta \geq 0$*, and* $\alpha + \beta = 1$*.*

Furthermore, we use $(\alpha_{WP}, \beta_{WP})$ and $(\alpha_{TR}, \beta_{TR})$ to denote the preference of WP and TR respectively. If $\alpha_{WP} > \beta_{WP}$, WP prefers to save more time. If $\alpha_{TR} < \beta_{TR}$, TR prefers to save money, even though the travel time may be extended.

Generally, the "satisfaction" degree of each party (either TR or WP) must reflect the change with/without taxi-sharing.

Definition 2 (Satisfaction). *For passengers (WP or TR), satisfaction is defined as* $S = \alpha \Delta_T + \beta \Delta_P$*, where* Δ_T *denotes the time difference between sharing taxi ans non-sharing taxi, and* Δ_P *denotes the payment difference between sharing and non-sharing. For drivers (TD), satisfation is defined as income difference between sharing taxi and non-sharing taxi.*

Specifically, the satisfaction degrees of WP (S_{WP}), TR (S_{TR}) and TD (S_{TD}) are computed as:

- **WP's satisfaction**: $S_{WP} = \alpha_w \Delta_T + \beta_w \Delta_P$, where (α_w, β_w) denotes WP's preference in time and payment.
- **TR's satisfaction**: $S_{TR} = \alpha_r \Delta_T + \beta_r \Delta_P$, where (α_r, β_r) denotes TR's preference in time and payment.
- **TD's satisfaction**: $S_{TD} = (I - I^{'})$, where I is the real gain in taxi-sharing, and I' is the gain when only carrying with one passenger along the same route.

A user is satisfied with this plan only if *satisfaction* is greater than zero. Note that there exists a bit difference when computing Δ_T and Δ_P between TR and WP. Originally, TR can go to the destination without carrying WP. But in a

Table 2. An Example of MaxSum and MaxMin

Plan	S_{TR}	S_{WP}	S_{TD}	MaxSum	MaxMin
WPF	1.68	12.62	2.8	17.1	1.68
TRF	2.48	11.24	2	15.72	2

taxi-sharing plan, he must pick up WP at first, so that the time is extended. Under such situation, $\Delta_T < 0$. Different from TR, WP has two choices, either sharing a taxi with WP, or just waiting for a new taxi. But it is unclear whether a new taxi will come there quickly or not. We use the average waiting time to estimate the time of the second choice by using historical statistics. Under such a condition, the value of Δ_T is definitely not negative.

The *satisfaction* degrees defined above illustrate how a party treats the scheduling plan. A scheduling plan is valid only when all of three parties feel satisfied with that plan, i.e, $S_{TR} \geq 0 \wedge S_{WP} \geq 0 \wedge S_{TD} \geq 0$. This condition can also be shorten as $S_{TR} \geq 0 \wedge S_{WP} \geq 0$ since $S_{TD} \geq 0$ always holds.

Query definition: The final task is to select a best choice out of all valid scheduling plans. Let's consider two valid plans with satisfactions as (S_{TR}, S_{WP}, S_{TD}) and $(S'_{TR}, S'_{WP}, S'_{TD})$. If $(S_{TR} \geq S'_{TR} \wedge S_{WP} \geq S'_{WP} \wedge S_{TD} \geq S'_{TD})$, the former is better. Otherwise, there exists no clear winner. Hence, we define two queries for this issue, including MaxSum and MaxMin.

Definition 3 (MaxSum query). *Given a set of scheduling plans, return a valid plan with maximal value of the sum of satisfactions.*

Definition 4 (MaxMin query). *Given a set of scheduling plans, return a valid plan with maximal value of the minimal satisfaction among three.*

Example 2. Consider the satisfaction degrees in Table 2. The MaxSum query will return the WPF plan, while the MaxMin query will return the TRF plan.

3 The Framework

A typical taxi-sharing system manages a large number of taxis. The position of each taxi will be reported to the system continuously. When a WP sends a request to the system, the system will select a most suitable taxi for WP immediately.

Figure 2 illustrates the framework, which contains two phases, *offline* phase for statistics and *online* phase for query processing. With the help of statistical information stored in the *offline* phase, the query processing can be significantly improved, avoiding comparing all taxis running in the system.

Offline phase. This phase aims at providing statistical information to support pruning rules. Recall that the satisfaction is actually the tradeoff of the traveling distance and the travel time, but to compute the road network distance and travel time precisely is infeasible. Hence, we attempt to estimate them in the

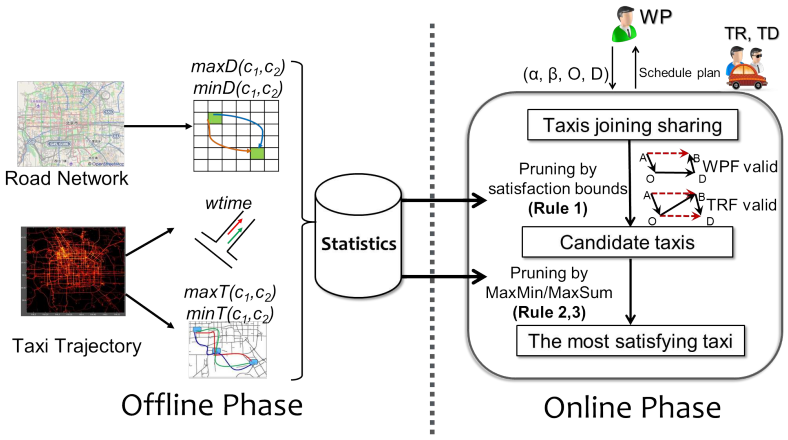


Fig. 2. Framework of Real-time personalized Taxi Sharing

offline manner. The road network helps to compute the distance between two points. Their travel time and waiting time of a road are estimated based on taxi trajectory records. More details will be presented in Section 4.

Online phase. The goal of this phase is to select a taxi most suitable for a request launched by a WP. Given all taxis that are willing to join sharing, we devise several pruning rules so that the candidate taxis to be further evaluated are significantly smaller than the original set of taxis. By using the statistics generated from the offline phase, each pruning rule can be evaluated in $O(1)$ time. In comparison, the cost to evaluate a taxi needs to find the shortest path between two points, which is expensive when the distance between these two points is far. See Section 5 for the details about this phase.

4 Offline phase

During the offline phase, we generate statistical information to support the pruning rules (to be introduced in the next section). The road network is divided into cells with equal sizes. We need to compute the following five operators based on historical trajectories. The first four operators use cells of source point and the destination point as input, while the last one is for a pick-up or drop-off point in the space. Each operator returns a value that is pre-computed and stored in the system.

- $\mathsf{minD}(c, c')$. It returns a value no greater than the minimal distance of the shortest path between two points in two cells, c and c', respectively.
- $\mathsf{maxD}(c, c')$. It returns a value no smaller than the maximal distance of the shortest path between two points in two cells, c and c', respectively.
- $\mathsf{minT}(c, c')$. It estimates the minimal travel time between two arbitrary points in two cells, c and c', respectively.

- $\mathsf{maxT}(c, c')$. It estimates the maximal travel time between two arbitrary points in two cells, c and c', respectively.
- $\mathsf{wtime}(p)$. It returns the average waiting time for the road segment where the pick-up or drop-off point p locates, which is calculated by the average time intervals between two unoccupied taxis passing by.

For any two cells c and c', the shortest path between them must start and end at the border of each cell. Let $P(c)$ and $P(c')$ denote the sets of the points that intersect with roads on the border of cells c and c' respectively. Then, minD is computed as $\min_{p \in P(c), p' \in P(c')} \mathsf{rdis}(p, p')$, where $\mathsf{rdis}(p, p')$ denotes the road network distance of the shortest path between two points p and p'. The common ways to implement $\mathsf{rdis}(p, p')$ includes Dijkstra's algorithm [10] and the A* algorithm [11].

To compute maxD precisely is more complex than minD, since the starting and ending points for the corresponding path may be located inside of two cells, no longer at the borders. Let $d(c, p)$ denote the maximal distance of the shortest path from point p to any point in c. In general, computing $d(c, p)$ is cheap since cell c is set small. In this way, we return $\min_{p \in P(c), p' \in P(c')}(\mathsf{rdis}(p, p') + d(c, p) + d(c', p'))$.

It is infeasible to predict minT or maxT precisely only based on road network. It is affected by some other factors, such as the number of traffic lights in the path. Moreover, it is inaccurate to use the maximal allowed speed to estimate minT, since in most cases the driving speed is lower than the maximal allowed speed. Hence, we attempt to use the historical trajectory database. Let $\mathsf{ttime1}(c, c', tr)$ denote the travel time during trajectory tr leaving c and entering c'. We return $\min_{tr} \mathsf{ttime1}(c, c', tr)$ for the minT operator. For the case where there exists no historical trajectory passing through c and c', we can (i) search for another cell c'', and return $\min_{c'', tr, tr'}(\mathsf{ttime1}(c, c'', tr) + \mathsf{ttime1}(c'', c', tr'))$, or (ii) return the time by using the maximal allowed speed.

We return $\max_{tr} \mathsf{ttime2}(c, c', tr)$ to estimate maxT based on historical trajectories. It records the time when tr first comes in c and finally leaves c'. Compared with $\max_{tr} \mathsf{ttime1}(c, c', tr)$, this new subroutine considers the travel time in each cell. For the case when there exists no historical trajectory passing through c and c', we return: $\min_{c''}(\max_{tr} \mathsf{ttime2}(c, c'', tr) + \max_{tr'} \mathsf{ttime2}(c'', c', tr'))$.

The goal of the last operator, wtime, is to estimate the average time to wait for a vacant taxi. We match the pick-up point p to the road, and then compute the average number of unoccupied taxis passing during a certain period. For example, if 10 unoccupied taxis have gone through the road in one hour, it costs approximately $3 = (\frac{60}{10 \times 2})$ minutes to wait for a vacant taxi in average.

5 Online Phase

The goal of the online phase is to select one taxi to match a WP's request. To enhance the performance, some pruning rules are devised to avoid expensive computation. The following contents are organized as below. We first introduce

the bound analysis for two plans in Section 5.1 and 5.2. Subsequently, three pruning rules, along with the overall algorithm, are described in Section 5.3.

5.1 Bound Analysis of the WPF Plan

We first study the WPF plan (Figure 1(a)). Assume a WP wants to go from O to D, and a TD happens to be at A, taking a TR to B at that time. The route in the WPF plan will be AODB. We analyze the lower and upper bounds of satisfactions for three parties below.

Waiting passenger. WP's satisfaction is defined as: $S_{WP} = \alpha_w \Delta_T + \beta_w \Delta_P$ (Definition 2), where Δ_T represents the time difference between waiting for an unoccupied taxi at O and waiting for the taxi traveling from A to O. The first item is estimated by $\mathsf{wtime}(O)$, while the second item is within $(\mathsf{minT}(A,O), \mathsf{maxT}(A,O))$. Hence, Δ_T is in a range of $(\mathsf{wtime}(O) - \mathsf{maxT}(A,O), \mathsf{wtime} - \mathsf{minT}(A,O))$.

Δ_P is computed as: $\Delta_P = k \cdot \mathsf{rdis}(O,D) - M_{WP}$, where $k \cdot \mathsf{rdis}(O,D)$ denotes the WP's payment from O to D without sharing, and M_{WP} is the payment with taxi-sharing, i.e, $M_{WP} = p_{11}\mathsf{rdis}(A,O) + p_{12}\mathsf{rdis}(O,D) + p_{13}\mathsf{rdis}(D,B)$. Note that WP may also need to pay for the segment DB though he will leave at D according to our pricing strategy. For any two points P and P', $\mathsf{minD}(P,P') \leq \mathsf{rdis}(P,P') \leq \mathsf{maxD}(P,P')$. Hence, the upper bound of the payment is $U_{M_{WP}} = p_{11}\mathsf{maxD}(A,O) + p_{13}\mathsf{maxD}(D,B)$, and the lower bound is $L_{M_{WP}} = p_{11}\mathsf{minD}(A,O) + p_{13}\mathsf{minD}(D,B)$. Accordingly, the upper bound (U^w_{WPF}) and the lower bound (L^w_{WPF}) of a WP's satisfactions are defined below.

$$U^w_{WPF} = \alpha_w \mathsf{wtime}(O) + \beta_w (k - p_{12}) \mathsf{rdis}(O,D) - \alpha_w \mathsf{minT}(A,O) - \beta_w L_{M_{WP}}$$
$$L^w_{WPF} = \alpha_w \mathsf{wtime}(O) + \beta_w (k - p_{12}) \mathsf{rdis}(O,D) - \alpha_w \mathsf{maxT}(A,O) - \beta_w U_{M_{WP}}$$

Taxi rider. TR's satisfaction is defined as: $S_{TR} = \alpha_r \Delta_T + \beta_r \Delta_P$ (Definition 2). Δ_T is the difference of travel time between the original route AB and the new route $AODB$. Hence, the lower bound of Δ_T is $\mathsf{minT}(A,B) - \mathsf{maxT}(A,O) - \mathsf{maxT}(O,D) - \mathsf{maxT}(D,B)$, and the upper bound of Δ_T is $\mathsf{maxT}(A,B) - \mathsf{minT}(A,O) - \mathsf{minT}(O,D) - \mathsf{minT}(D,B)$. $\Delta_P = k \cdot \mathsf{rdis}(A,B) - M_{TR}$, where $k \cdot \mathsf{rdis}(A,B)$ denotes the payment for the segment AB without sharing, and the payment for sharing is $M_{TR} = q_{11}\mathsf{rdis}(A,O) + q_{12}\mathsf{rdis}(O,D) + q_{13}\mathsf{rdis}(D,B)$. Similar to those of WP, the upper and lower payment bounds of TR are computed as $U_{M_{TR}} = q_{11}\mathsf{maxD}(A,O) + q_{12}\mathsf{rdis}(O,D) + q_{13}\mathsf{maxD}(D,B)$, $L_{M_{TR}} = q_{11}\mathsf{minD}(A,O) + q_{12}\mathsf{rdis}(O,D) + q_{13}\mathsf{minD}(D,B)$. Accordingly, the upper bound (U^r_{WPF}) and the lower bound (L^r_{WPF}) of a TR's satisfactions are defined below.

$$\begin{aligned} U^r_{WPF} = {} & \alpha_r \mathsf{maxT}(A,B) + k\beta_r \mathsf{maxD}(A,B) \\ & -\alpha_r[\mathsf{minT}(A,O) + \mathsf{minT}(O,D) + \mathsf{minT}(D,B)] - \beta_r L_{M_{TR}} \qquad (1) \\ L^r_{WPF} = {} & \alpha_r \mathsf{minT}(A,B) + k\beta_r \mathsf{minD}(A,B) \\ & -\alpha_r[\mathsf{maxT}(A,O) + \mathsf{maxT}(O,D) + \mathsf{maxT}(D,B)] - \beta_r U_{M_{TR}} \qquad (2) \end{aligned}$$

Taxi Driver. TD's satisfaction is defined as $I - I'$ (Definition 2). If launching taxi-sharing, the driver's income is the sum of TR's payment and WP's payment,

i.e, $I = M_{TR} + M_{WP}$, and $I' = k \cdot (\mathsf{rdis}(A,O) + \mathsf{rdis}(O,D) + \mathsf{rdis}(D,B))$ when he travels along the same route. Thus, the upper bound (U^d_{WPF}) and the lower bound (L^d_{WPF}) of a driver's satisfactions are defined below.

$$U^d_{WPF} = \lambda_1 \mathsf{maxD}(A,O) + \lambda_2 \mathsf{rdis}(O,D) + \lambda_3 \mathsf{maxD}(D,B) \quad (3)$$
$$L^d_{WPF} = \lambda_1 \mathsf{minD}(A,O) + \lambda_2 \mathsf{rdis}(O,D) + \lambda_3 \mathsf{minD}(D,B) \quad (4)$$

where $\lambda_1 = p_{11} + q_{11} - k$, $\lambda_2 = p_{12} + q_{12} - k$, and $\lambda_3 = p_{13} + q_{13} - k$.

5.2 Bound Analysis of the TRF Plan

Assuming a WP wants to go from O to D, and a TD is at A, taking a TR to B at that time. The route in the TRF plan will be AOBD. We analyze the lower and upper bounds of satisfactions for the three parties.

Waiting passenger. First, Δ_T for WP is the time difference between her original travel time of OD (plus the waiting time $\mathsf{wtime}(O)$) and the new travel time of $AOBD$. Second, Δ_P is the payment difference between $k \cdot \mathsf{rdis}(O,D)$ and $M_{WP} = p_{21}\mathsf{rdis}(A,O) + p_{22}\mathsf{rdis}(O,B) + p_{23}\mathsf{rdis}(D,B)$. The upper and lower bounds of payment are $U_{M_{WP}}$ and $L_{M_{WP}}$. Thus, the upper bound (U^w_{TRF}) and the lower bound (L^w_{TRF}) of a WP's satisfactions are defined below.

$$\begin{aligned} U^w_{TRF} &= \alpha_w[\mathsf{wtime}(O) + \mathsf{maxT}(O,D)] + k\beta_w \mathsf{rdis}(O,D) \\ &\quad -\alpha_w[\mathsf{minT}(A,O) + \mathsf{minT}(O,B) + \mathsf{minT}(B,D)] - \beta_w L_{M_{WP}} \quad (5) \\ L^w_{TRF} &= \alpha_w[\mathsf{wtime}(O) + \mathsf{minT}(O,D)] + k\beta_w \mathsf{rdis}(O,D) \\ &\quad -\alpha_w[\mathsf{maxT}(A,O) + \mathsf{maxT}(O,B) + \mathsf{maxT}(B,D)] - \beta_w U_{M_{WP}} \quad (6) \end{aligned}$$

Taxi rider. The route for TR changes from AB to AOB, so that Δ_T rises. In addition, the payment change Δ_P is computed as: $M_{TR} - k \cdot \mathsf{rdis}(A,B)$, where $M_{TR} = \mathsf{rdis}(A,O)q_{21} - \mathsf{rdis}(O,B)q_{22}$. Let $U_{M_{TR}}$ and $L_{M_{TR}}$ denote the upper and lower bounds of payment. Thus, the upper bound (U^r_{TRF}) and the lower bound (L^r_{TRF}) of a TR's satisfactions are defined below.

$$\begin{aligned} U^r_{TRF} &= \alpha_r \mathsf{maxT}(A,B) + k\beta_r \mathsf{maxD}(A,B) \\ &\quad -\alpha_r(\mathsf{minT}(A,O) + \mathsf{minT}(O,B)) - \beta_r L_{M_{TR}} \quad (7) \\ L^r_{TRF} &= \alpha_r \mathsf{minT}(A,B) + k\beta_r \mathsf{minD}(A,B) \\ &\quad -\alpha_r(\mathsf{maxT}(A,O) + \mathsf{maxT}(O,B)) - \beta_r U_{M_{TR}} \quad (8) \end{aligned}$$

Taxi driver. The route for TD is $AOBD$. Thus, the upper bound (U^d_{TRF}) and the lower bound (L^d_{TRF}) of a TD's satisfactions are defined below.

$$U^d_{TRF} = \lambda_1 \mathsf{maxD}(A,O) + \lambda_2 \mathsf{maxD}(O,B) + \lambda_3 \mathsf{maxD}(B,D) \quad (9)$$
$$L^d_{TRF} = \lambda_1 \mathsf{minD}(A,O) + \lambda_2 \mathsf{minD}(O,B) + \lambda_3 \mathsf{minD}(B,D) \quad (10)$$

where $\lambda_1 = p_{21} + q_{21} - k$, $\lambda_2 = p_{22} + q_{22} - k$, and $\lambda_3 = p_{23} + q_{23} - k$.

Algorithm 1: Schedule(w)

$C \leftarrow \emptyset$, $\tau \leftarrow 0$, $C' \leftarrow \emptyset$;
foreach *registered taxi* r **do**
  $WPFvalid \leftarrow$ **false**; $TRFvalid \leftarrow$ **false**;
  if $\min(U^w_{WPF}, U^r_{WPF}) > 0$ **then**
    $WPFvalid \leftarrow$ **true**;
  if $\min(U^w_{TRF}, U^r_{TRF}) > 0$ **then**
    $TRFvalid \leftarrow$ **true**;
  if $WPFvalid$ **or** $TRFvalid$ **then**
    $C \leftarrow C \cup \{r\}$, $\alpha_1 \leftarrow 0$, $\alpha_2 \leftarrow 0$, $\beta_1 \leftarrow 0$, $\beta_2 \leftarrow 0$;
    if $WPFvalid =$ **true then**
      $\alpha_1 \leftarrow \min(L^w_{WPF}, L^r_{WPF}, L^d_{WPF})$; $\beta_1 \leftarrow L^w_{WPF} + L^r_{WPF} + L^d_{WPF}$;
    if $TRFvalid =$ **true then**
      $\alpha_2 \leftarrow \min(L^w_{TRF}, L^r_{TRF}, L^d_{TRF})$; $\beta_2 \leftarrow L^w_{TRF} + L^r_{TRF} + L^d_{TRF}$;
    if *the query is* MaxMin **and** $\max(\alpha_1, \alpha_2) > \tau$ **then**
      $\tau \leftarrow \max(\alpha_1, \alpha_2)$;
    else if *the query is* MaxSum **and** $\max(\beta_1, \beta_2) > \tau$ **then**
      $\tau \leftarrow \max(\beta_1, \beta_2)$;
foreach *registered taxi* $r \in C$ **do**
  if *the query is* MaxMin **and**
  $(\min(U^w_{WPF}, U^r_{WPF}, U^d_{WPF}) > \tau \quad || \quad \min(U^w_{TRF}, U^r_{TRF}, U^d_{TRF}) > \tau)$ **then**
    $C' \leftarrow C' \cup \{r\}$;
  if *the query is* MaxSum **and**
  $(U^w_{WPF} + U^r_{WPF} + U^d_{WPF} > \tau \quad || \quad U^w_{TRF} + U^r_{TRF} + U^d_{TRF} > \tau)$ **then**
    $C' \leftarrow C' \cup \{r\}$;
foreach *candidate taxi* $r \in C'$ **do**
  Compute the satisfactions of three parties;
return a taxi with the biggest minimal (total) satisfaction to MaxMin (MaxSum);

5.3 Pruning Rules and Query Processing

We then design some pruning rules. In any feasible plan, the satisfaction degrees for all three parties must be positive. Note that TD's satisfaction is always positive under our pricing strategy settings. Hence, the rule is as below.

Pruning rule 1 *One taxi with* $\min(U^w_{WPF}, U^r_{WPF}) < 0$ *or* $\min(U^w_{TRF}, U^r_{TRF}) < 0$ *is improper.*

All improper taxis can be filtered out by using Pruning rule 1. However, the computation cost is still high if there exist a large number of suitable taxis. Therefore, we propose other two pruning rules to filter a significant part of suitable taxis, which respectively correspond to MaxMin and MaxSum queries.

According to the bound analysis, the upper and lower bounds of two target satisfaction values are $\max(U^w_{WPF}, U^r_{WPF}, U^d_{WPF})$ and $\min(L^w_{WPF}, L^r_{WPF}, L^d_{WPF})$ respectively. Hence, let τ denote the maximal lower bound for all taxis. Then, any taxi with the upper bound below τ can be filtered safely. The same filtering process is used for TRF strategy. The pruning rule is defined formally below.

Pruning rule 2 *The* MaxMin *query returns a taxi with the greatest minimal satisfaction among all of three parties. A taxi with* $\min(U^w_{WPF}, U^r_{WPF}, U^d_{WPF}) < \tau$ *or* $\min(U^w_{TRF}, U^r_{TRF}, U^d_{TRF}) < \tau$ *can be pruned safely.*

Pruning rule 3 *The* MaxSum *query returns a taxi with the greatest sum of satisfactions. A taxi with* $\min(U^w_{WPF} + U^r_{WPF} + U^d_{WPF}) < \tau$ *or* $\min(U^w_{TRF} + U^r_{TRF} + U^d_{TRF}) < \tau$ *can be pruned safely.*

Algorithm 1 describes details on selecting the most suitable taxi by using the above pruning rules. At first, the algorithm scans all registered taxis to construct a candidate set C by using Pruning rule 1. In addition, the threshold τ is computed for further pruning (at lines 2-17). Subsequently, it scans all candidate taxis in C to construct C' by using Pruning rules 2 and 3 (at lines 18-22). At last, it scans each taxi in C' to obtain the final result (at lines 23-24). The most proper plan for each kind of query is returned to WP (at lines 25).

6 Experiments

We report some experimental results in this section. The offline phase is conducted on a Hadoop cluster. All codes for online processing are written in Java, and run on a computer with 16GB RAM and Intel Xeon 2.00GHz CPU.

Road Network: We utilize the road network of Beijing, which is in the range of 30,000 meters in width and length. It is partitioned into 30 * 30 cells, each with 1,000 meters in width and length.

Taxi Trajectory: We use a real taxi trajectory set containing 13,000 Beijing taxis over one month (October 2013). The number of occupied/unoccupied taxis varies a lot in time and place. We sample every hour within a day to count all taxis and occupied taxis. About 4,000 taxis in average are occupied during the rush hours, and 90% of occupied taxis move around urban districts, while around 1,000 taxis are occupied at mid-night. We assume that all occupied taxis join taxi-sharing. Therefore, the number of occupied taxis (TR) varies from 1,000 to 4,000 in the experiments. Since the requests are processed in order, the number of passengers does not affect the query time. The parameter setting of the pricing strategy are listed in Table 1.

6.1 Effectiveness

We evaluate the effectiveness of the proposed taxi-sharing method. First, we compare the travel time and payment between taxi-sharing(TS) and non-sharing(NS) methods after randomly selecting requests from the trips in the real dataset. The

number of TRs ranges from 1,000 to 4,000, and the number of WPs is set to 1,000. The preferences of TRs and WPs are generated uniformly in (0,1). Figure 3 illustrates the average time for WP/TR with and without sharing taxi. Figure 4 compares payment of WP/TR and income of TD in TS and NS methods. Note that the travel time for WP in two methods contains the waiting time. When using TS strategy, the travel time for WP is reduced even if WP plans to wait for TR, because WP's waiting time decreases. WP may pay more money in taxi-sharing scenario, the average payment for WPs is less than NS as the amount of TRs rises. Although TR spends more time on traveling, the average payment is significantly reduced. In addition, more occupied taxis make WP and TR both save more time or money. TDs' income rises by more than 15 percents compared to NS in all situations, due to the payment sharing strategy (Table 1). In fact, TD can earn more if any of the parameters in Table 1 rises.

We then evaluate the influence of preference parameters, assuming there are 2,000 TRs. α_{WP} and α_{TR} are generated uniformly in (0,1). We compare the average travel time difference (ΔT), average payment difference (ΔP) from NS to TS for WP/TR, and TD's average income difference (ΔI) from TS to NS, as reported in Figure 5. In Figure 5(a), higher α_{WP} results in more ΔT for WP, which means WP saves more time in TS. In Figure 5(b), lower α_{WP} leads to more ΔP, which indicates WP saves more money. Figure 5(c) illustrates two metrics: (1) TD's average ΔI. For all preference settings, TD can earn more in TS compared to NS. Moreover, biased preferences of passengers lead to more income. (2) Average α_{TR}. If WP and TR have the opposite preferences in time (money), they are more likely to share one taxi. For example, the time-preferred WP is matched with the payment-preferred TR.

6.2 Efficiency

We then evaluate the performance of pruning rules. Figure 6 illustrates the performance of three pruning rules, including WPF+TRF (Rule 1), MaxMin (Rule 2), and MaxSum (Rule 3). Recall that WPF+TRF is the basis of MaxMin and

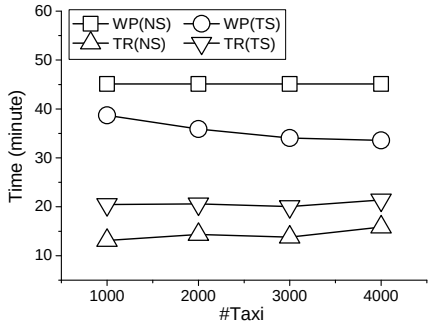


Fig. 3. Time Comparison

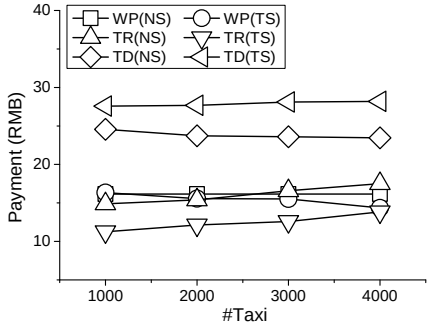
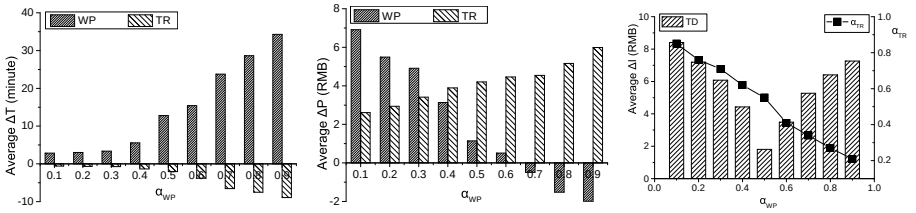


Fig. 4. Payment/Income Comparison



(a) ΔT for WP and TR (b) ΔP for WP and TR (c) Average ΔI for TD

Fig. 5. Parameter Settings Comparison

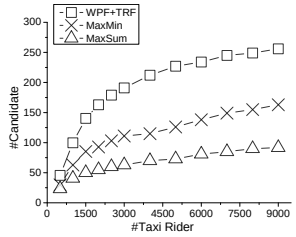
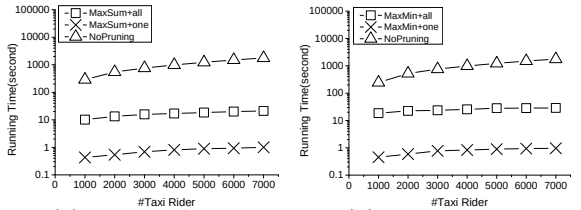


Fig. 6. Pruning effects



(a) For MaxSum (b) For MaxMin

Fig. 7. Efficiency comparison

MaxSum. By WPF+TRF, approximately 97% of taxis are safely filtered, e.g, the number of candidates is around 250 when there are nine thousand taxi riders. After employing MaxMin or MaxSum, 36% or 60% more candidates are removed safely, e.g, only 160 or 95 candidates are left when $|TR| = 9,000$. Moreover, the pruning power will be strengthened when the number of taxi riders increases. For example, when $|TR| = 3,000$, around 2 percents of taxis (60) are left for processing. But when $|TR| = 9,000$, around 1 percent of taxis (90) are left.

The use of pruning rules deeply influences the running time of the proposed methods. Since computing the shortest path for each candidate costs much time, we use two schemes to fulfill this goal. The first is to precisely compute the satisfactions for all candidates (+all). The second is to randomly choose a candidate every time, and try to evaluate whether this one satisfies the requirement or not (+one). If not, we repeat to choose another one. Figure 7(a) and 7(b) show that MaxMin and MaxSum pruning methods both scale significantly better than NoPruning method, because NoPruning method needs to precisely compute each taxi. Note that MaxMin+one and MaxSum+one methods can answer the query in a second.

7 Related Work

The problem of ride-sharing has been studied intensively in recent years. There are several popular methods for ride-sharing, mainly including mobile applications (e.g Uber), and the agencies providing ride-sharing service (e.g Lyft). [12]

proposes the dynamic ride-sharing problem. [13] discusses the dynamic slugging problem with the vehicle-capacity and delay bounded constraints. However, they require drivers' original routes not to be changed and trips to be arranged in advance. Later works propose new ride-sharing problems. The work [14] summarizes different optimization goals for ride-sharing: to minimize system-wide vehicle-miles, to minimize system-wide travel time and to maximize the number of participants. [15] creates a dynamic ride-sharing community service architecture, which combines ITS, ride-sharing and social network. We also develop a passenger matching system [16] to find passengers with similar locations and estimate additional time and payment. [17] proposes a scheme that allows detours and four sharing patterns. Existing ride-sharing methods [12, 13, 15] apply to some of them, but our work contains all of them.

Taxi sharing is a more popular way of ride-sharing in realtime service. Research efforts on analysis of taxi-sharing have been made to address the objective issue by considering different constraints. For example, [18] considers the vehicle capacity constraint, [19] considers the waiting time constraint, and [8] considers time and money constraints. Our work considers all these constraints and the pricing strategy of [8] can be considered as a special case of our pricing method. Other works focus on how to schedule passengers. [20] builds a dynamic and distributed taxi-sharing system which processes every passenger's request on each end of taxi. [19] allows more than two groups of passengers sharing one taxi, but it simply filters out taxis outside the waiting time constraint. [8] aims to obtain global minimal additional incurred travel distance which is an NP-complete problem. So it utilizes a greedy algorithm to find taxi-sharing candidates by predicting taxis' future positions, which is a very uncertain task. To the best of our knowledge, our work first takes into account of different people's customized preferences for taxi-sharing, in purpose of saving time and payment.

8 Conclusion

We have studied the taxi-sharing problem on a parameterized real-time taxi-sharing system for user satisfaction. We develop a method to maximize people's satisfaction with their personalized needs. Experimental results show the efficiency and effectiveness of our method. Future work includes: (1) allowing more than two groups of passengers to share one taxi, (2) creating indexes for passengers' moving positions to speedup online processing, and (3) making estimation of travel time more accurate by using probability distribution function instead of scalar.

9 Acknowledge

Our research is supported by the 973 program of China (No. 2012CB316203), NSFC (U1401256, 61370101, U1501252, 61402180 and 61472345), Shanghai Knowledge Service Platform Project (No. ZF1213), Innovation Program of Shanghai

Municipal Education Commission(14ZZ045), and Natural Science Foundation of Shanghai (No. 14ZR1412600).

References

1. Brunekreef, B., Holgate, S.T.: Air pollution and health. The lancet **360**(9341) (2002) 1233–1242
2. Morency, C.: The ambivalence of ridesharing. Transportation **34**(2) (2007) 239–253
3. Chan, N.D., Shaheen, S.A.: Ridesharing in north america: Past, present, and future. Transport Reviews **32**(1) (2012) 93–112
4. Baldacci, R., Maniezzo, V., Mingozzi, A.: An exact method for the car pooling problem based on lagrangean column generation. Operations Research **52**(3) (2004) 422–439
5. Attanasio, A., Cordeau, J., Ghiani, G., Laporte, G.: Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. Parallel Computing **30**(3) (2004) 377–387
6. Gidofalvi, G., Pedersen, T.B., Risch, T., Zeitler, E.: Highly scalable trip grouping for large-scale collective transportation systems. In: EDBT. (2008) 678–689
7. Ferguson, E.: The rise and fall of the american carpool: 1970–1990. Transportation **24**(4) (1997) 349–376
8. Ma, S., Zheng, Y., Wolfson, O.: T-share: A large-scale dynamic taxi ridesharing service. In: ICDE. (2013) 410–421
9. Song, L., Wang, C., Duan, X., et al: Taxihailer: A situation-specific taxi pick-up points recommendation system. In: DASFAA. (2014) 523–526
10. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**(1) (1959) 269–271
11. Zeng, W., Church, R.L.: Finding shortest paths on real road networks: the case for a*. IJGIS **23**(4) (2009) 531–543
12. Agatz, N., Erera, A., Savelsbergh, M., Wang, X.: Sustainable passenger transportation: Dynamic ride-sharing. Technical report, ERIM Report Series Research in Management (2010)
13. Ma, S., Wolfson, O.: Analysis and evaluation of the slugging form of ridesharing. In: SIGSPATIAL/GIS. (2013) 64–73
14. Agatz, N., Erera, A., Savelsbergh, M., Wang, X.: Optimization for dynamic ride-sharing: A review. EJOR **223**(2) (2012) 295 – 303
15. Fu, Y., Fang, Y., Jiang, C., Cheng, J.: Dynamic ride sharing community service on traffic information grid. In: ICICTA. Volume 2., IEEE (2008) 348–352
16. Duan, X., Jin, C., Wang, X.: POP: A passenger-oriented partners matching system. In: 31st IEEE ICDE Workshops 2015. (2015) 117–118
17. Furuhata, M., Dessouky, M., et al: Ridesharing: The state-of-the-art and future directions. Transportation Research Part B: Methodological **57**(0) (2013) 28–46
18. Tao, C.C.: Dynamic taxi-sharing service using intelligent transportation system technologies. In: 2007 WiCOM. (2007) 3209–3212
19. Huang, Y., Jin, R., Bastani, F., Wang, X.S.: Large scale real-time ridesharing with service guarantee on road networks. CoRR **abs/1302.6666** (2013)
20. d'Orey, P., Fernandes, R., Ferreira, M.: Empirical evaluation of a dynamic and distributed taxi-sharing system. In: ITSC. (Sept 2012) 140–146