HAL
open science

SCRAM: A Sharing Considered Route Assignment Mechanism for Fair Taxi Route Recommendations

Shiyou Qian, Jian Cao, Frédéric Le Mouël, Issam Sahel, Minglu Li

To cite this version:

Shiyou Qian, Jian Cao, Frédéric Le Mouël, Issam Sahel, Minglu Li. SCRAM: A Sharing Considered Route Assignment Mechanism for Fair Taxi Route Recommendations. Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'2015), Aug 2015, Sydney, Australia. 10.1145/2783258.2783261 . hal-01180500

HAL Id: hal-01180500

https://inria.hal.science/hal-01180500v1

Submitted on 28 Jul 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SCRAM: A Sharing Considered Route Assignment Mechanism for Fair Taxi Route Recommendations

Shiyou Qian
Department of Computer Science and Engineering, Shanghai Jiao Tong University
qshiyou@sjtu.edu.cn

Jian Cao *
Department of Computer Science and Engineering, Shanghai Jiao Tong University
cao-jian@sjtu.edu.cn

Frédéric Le Mouël
University of Lyon, INSA-Lyon, CITI-INRIA Lab
frederic.le-mouel@insa-lyon.fr

Issam Sahel
University of Lyon, INSA-Lyon, CITI-INRIA Lab
issam.sahel@insa-lyon.fr

Minglu Li
Department of Computer Science and Engineering, Shanghai Jiao Tong University
mlli@sjtu.edu.cn

ABSTRACT

Recommending routes for a group of competing taxi drivers is almost untouched in most route recommender systems. For this kind of problem, recommendation fairness and driving efficiency are two fundamental aspects. In the paper, we propose SCRAM, a sharing considered route assignment mechanism for fair taxi route recommendations. SCRAM aims to provide recommendation fairness for a group of competing taxi drivers, without sacrificing driving efficiency. By designing a concise route assignment mechanism, SCRAM achieves better recommendation fairness for competing taxis. By considering the sharing of road sections to avoid unnecessary competition, SCRAM is more efficient in terms of driving cost per customer (DCC). We test SCRAM based on a large number of historical taxi trajectories and validate the recommendation fairness and driving efficiency of SCRAM with extensive evaluations. Experimental results show that SCRAM achieves better recommendation fairness and higher driving efficiency than three compared approaches.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database applications-Data Mining

Keywords

Recommender Systems; Assignment Mechanism; Fairness; Taxis

1. INTRODUCTION

Taxi drivers need to find customers when their taxis are vacant. Meanwhile, customers expect to find a taxi as quickly as possible. With the help of route recommender systems, on one hand, taxi drivers can more quickly find customers, increasing their revenues; on the other hand, customers can find a taxi in a shorter time, saving waiting time. Therefore, route recommender systems for taxi drivers are of great social and economic importance because these systems aim at providing personalized and context-aware route recommendations [12, 3].

Recommender systems focused on public transportation service have been studied extensively [6, 16, 11, 15, 17]. The common objective of these systems is to minimize the driving costs (in terms of time or distance) of taxi drivers while maximizing the revenues which are expressed as success probabilities of picking up customers. However, existing recommender systems have several limitations. First, these systems seldom address the route recommendation fairness problem involving a group of competing taxi drivers. Without a fair route assignment mechanism, recommendation fairness is not guaranteed for competing taxi drivers. Second, the sharing of road sections existing in the recommended routes is not considered when providing route recommendations, resulting in poor driving efficiency due to the unnecessary competition of taxi drivers. Third, precisely speaking, most systems, such as [6, 11, 15], are coarse-grained because a route recommended by these systems is just a driving direction in nature, rather than an actual driving route composed of consecutive road sections.

In the paper, we propose SCRAM, a sharing considered route assignment mechanism for fair taxi route recommendations. SCRAM aims to provide fair and efficient route recommendations for a group of competing taxi drivers. The primary objective of SCRAM is to guarantee recommendation fairness for a group of competing taxi drivers. The by-product of SCRAM is the improved driving efficiency of taxi drivers by considering the sharing of road sections existing in the recommended routes. In addition, SCRAM is capable of providing taxi drivers with actual driving routes, rather than rough driving directions.

SCRAM differs from most existing route recommender systems in four aspects. First, SCRAM considers the sharing degree of road sections in the process of computing recommended routes. The consideration of sharing is the basis on which to provide recommendation fairness for competing taxi drivers. Second, the success probability of picking up customers has priority over driving cost in the route evaluation function employed by SCRAM. This principle naturally conforms to the reality that the goal of taxi drivers is to minimize driving cost under the premise of finding customers. Third, the success probabilities of road sections

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783261 .

contained in a route are weighted decreasingly with the distances from the starting intersection to the road sections because potential customers at far road sections are more likely to be picked up by other taxis. Fourth, neither success probability nor driving cost is sufficient to represent the recommendation fairness and driving efficiency of taxi drivers. We present a new metric, called driving cost per customer (DCC) which combines both driving cost and success probability.

We evaluate the recommendation fairness and driving efficiency of SCRAM based on historical taxi trajectories which were collected in Shanghai, China. The trajectories contain the GPS data of more than 4000 taxi drivers operating in a period of 126 days. Experimental results show that SCRAM outperforms its counterparts to a large degree in terms of both recommendation fairness and driving efficiency.

The main contributions of the paper are as follows:

- By identifying three route evaluation principles, we define an evaluation function which complies with the demands of taxi drivers.
- We design a concise route assignment mechanism to guarantee the recommendation fairness of competing taxi drivers and prove the properties of the assignment mechanism by three theorems.
- Based on real trajectory data, extensive experiments are conducted to evaluate the recommendation fairness and driving efficiency of SCRAM.

The rest of the paper is organized as follows. Section 2 discusses related work, highlighting the difference between our proposed approach and the existing ones. Section 3 formulates the fair route recommendation problem. Section 4 defines the route evaluation function which incorporates three evaluation principles. Section 5 presents the route assignment mechanism and the proof of three theorems which theoretically describe the properties of SCRAM. Section 6 illustrates experimental results. We conclude the paper in section 7.

2. RELATED WORK

Route recommendation for taxi drivers has drawn much attention from researchers [6, 16, 11, 8, 14, 15, 17, 4, 18, 19, 9]. Most existing recommender systems are designed based on real world GPS trajectories which were collected through a large number of probing taxis. These recommender systems can be roughly divided into two categories: macroscopic recommender systems and microscopic ones.

For macroscopic recommender systems, such as [6, 16, 8, 15, 19], only driving directions are provided for taxi drivers rather than actual driving routes. Generally, customer locations are extracted from GPS trajectories by these systems and the locations are clustered into multiple representative small areas, which are the recommended driving directions for taxi drivers. For example, in the LCP approach proposed in [6], these small areas are referred to as pick-up points which are learned from the trajectories of high-profit taxi drivers. A taxi driver who requests a route recommendation is provided with a sequence of pick-up points as driving directions.

Compared with macroscopic recommender systems, microscopic ones provide taxi drivers with actual driving routes. T-Finder proposed in [17] uses the same method as LCP [6] to extract the representative small areas from the trajectories, but these small areas are termed as parking places. Instead of just telling taxi drivers where to go, T-Finder goes one step further than LCP by providing taxi drivers with detailed driving routes connecting parking places. SCRAM belongs to the microscopic category. However, SCRAM differs from existing microscopic recommender systems, such as T-Finder, in two aspects. First, SCRAM addresses the route recommendation problem for a group of competing taxi drivers, rather than a single one. We design a concise route assignment mechanism to guarantee the recommendation fairness for taxi drivers. Second, we consider the sharing of road sections occurring in the recommended routes, which exists in reality and is ignored in most recommender systems. The combination of these two aspects results in a fair and efficient route recommender system for competing taxi drivers.

To the best of our knowledge, the problem of recommending routes for a group of competing taxi drivers is seldom studied in the existing route recommender systems. In [6] and [17], the round-robin mechanism is used to assign recommended routes to multiple requesting taxi drivers, without guaranteeing the recommendation fairness. The route recommender system proposed in [11] mainly serves a single novice by providing an optimal driving direction connecting multiple small areas. In essence, the core of recommending routes for multiple competing taxi drivers can be regarded as a load balance problem. As for load balance, recommendation fairness and driving efficiency are two contradicting factors. Therefore, the challenge of designing a load balance policy is to guarantee the recommendation fairness without sacrificing the driving efficiency of competing taxi drivers. In the paper, we propose SCRAM to address this challenge.

3. PROBLEM FORMULATION

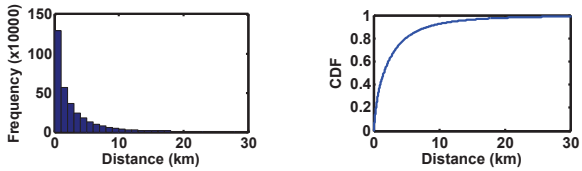
In reality, the road network of a city, such as Shanghai, is characterized by a set of intersections and a set of road sections. Thus, a road network can be represented by a graph $G = (I, R)$, where $I = \{I_1, I_2, ..., I_n\}$ is a finite set of n intersections and $R = \{R_1, R_2, ..., R_m\}$ is a finite set of m road sections. A road section is determined by two intersections, $R_i = (I_j, I_k)$. A road section R_i is associated with five properties, the direction (one-way or bidirectional) $R_i.dir$, the speed constraint $R_i.speed$, the length $R_i.length$, the starting intersection $R_i.s$, and the ending intersection $R_i.e$. Furthermore, by mining the real trajectories generated by a large number of taxis, each road section has a success probability $P(R_i)$ which represents the chance of picking up customers for taxi drivers, $P(R_i) \in [0, 1.0]$ for $1 \leqslant i \leqslant m$. A route is a directed sequence of L road sections, $W_i = \{R_1, R_2, ..., R_L\}$, where $W_i.s = R_1.s$, $W_i.e = R_L.e$ and $R_j.e = R_{j+1}.s$ for $1 \leqslant j < L$ which means that consecutive road sections contained in a route should share an intersection. Let D be a group of N competing taxi drivers that request route recommendations, $D = \{D_1, D_2, ..., D_N\}$. Based on the above definitions and notations, we can formally define the problem of fair taxi route recommendation as:

DEFINITION 1. *The fair taxi route recommendation problem.*

Given: a road network $G = (I, R)$, a probability set $P = \{P(R_1), P(R_2), ..., P(R_m)\}$, a cost matrix T representing the driving cost between two neighboring intersections, a group of N competing taxi drivers $D = \{D_1, D_2, ..., D_N\}$ located near an intersection.

Objective: Finding a set of N routes $W = \{W_1, W_2, ..., W_N\}$ from the candidate routes and assigning a route to each requesting driver (D_i, W_j) while satisfying the following condition.

CONDITION 1. *When assigning routes to drivers, the assignment mechanism must guarantee recommendation fairness for competing taxi drivers from a long-term perspective. The driving cost per customer (DCC) should be a constant for all competing*



(a) (b)

Figure 1: Distribution and CDF of driving distances

taxi drivers. In other words, the standard deviation of DCC should be minimized.

The objective of SCRAM is to address the recommendation fairness problem for a group of competing taxis. If taxis are far away from each other without competition, route recommendations can be provided independently. Thus, in the paper, it is assumed that a group of competing taxi drivers is located near an intersection which is used as the starting point to generate candidate routes. This situation is observable in reality. For example, taxis tend to wait for customers at certain places, such as subway exits or shopping centers, but sometimes the queue is too long, so some taxis are reluctant to continue waiting.

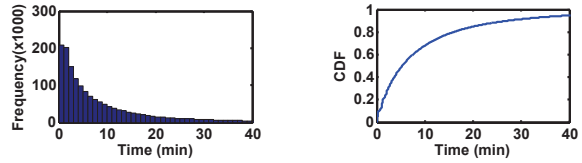
The assumption can be released to a more common case where taxis are scattered in a small area. In this case, the nearest intersection from the center of the area could be selected as the starting point. Since all recommended routes are generated from the starting point, some taxis need to go there to follow the recommended routes. Superficially, this may affect the recommendation fairness of competing taxi drivers. Due to the sharing of road sections existing in the recommended routes, in fact, the impact is negligible. We discuss this case in detail in Section 5.

When recommending driving routes for taxi drivers, the length of routes is usually restricted. Note the length of a route is the number of road sections that are contained in the route. Let λ be the maximum number of road sections that are connected to intersections in a road network. In a general case with no restriction on the length of recommended routes, the number of candidate routes is $\sum_{i=1}^{n} \lambda(\lambda-1)^{i-1}$, where n is the number of intersections appearing in the road network. As proved in [6], the complexity of computing N recommended routes is $O(n!)$. When the length of recommended routes is limited to a constant number L, the number of candidate routes is reduced to $\lambda(\lambda-1)^{L-1}$. For the fair taxi route recommendation problem, a length-constrained case is more realistic, which complies with the traffic patterns extracted from the real trajectories generated by a large number of taxis.

By analyzing the taxi trajectories collected in Shanghai, China, we find that taxi drivers usually travel along a small number of road sections to pick up customers. The distribution and cumulative distribution function (CDF) of driving distances are depicted in Figure 1. As shown in the figure, more than 40% of taxi drivers pick up customers in less than 1 km. This observation is called locality driving behavior of taxi drivers, which is used to determine the length of recommended routes. The distribution and CDF of driving times taken by taxi drivers are shown in Figure 2, which also confirms the locality driving behavior of taxi drivers. It is observed that almost 40% of taxi drivers find a customer in less than 5 minutes. Further details on the trajectory data are given in Section 6.

4. EVALUATION FUNCTION

The way to evaluate routes is critical to selecting recommended routes from the candidate ones. Generally, a route can be measured



(a) (b)

Figure 2: Distribution and CDF of driving times

by its success probability of picking up customers or driving cost. When recommending routes to taxis, success probability and driving cost should be considered simultaneously. The definition of an evaluation function should comply with the demands of taxi drivers.

4.1 Evaluation Principles

4.1.1 Priority Principle

The most important aspect of defining a route evaluation function is to determine the goal of taxi drivers. If taxi drivers solely choose energy saving as their purpose, the best choice for them is just to stay in the same place. However, each taxi driver needs to pay a certain amount of money as a management fee even if he/she does nothing each day. Therefore, the goal of taxi drivers is to minimize driving cost under the premise of finding customers. Naturally, the first principle of evaluation is that success probability should have higher priority than driving cost.

4.1.2 Decaying Principle

The success probabilities of road sections contained in a route should not be weighted equally. The driving distance from the starting point to a road section is crucial for taxi drivers to succeed in finding customers. Generally, taxi drivers more easily find potential customers at near road sections, while potential customers at far road sections are more likely to be picked up by other taxi drivers. Therefore, the second principle of evaluation is that the success probabilities of road sections contained in a route should be weighted decreasingly with the driving distances from the starting intersection to the road sections.

4.1.3 Sharing Principle

Given a road section with a high success probability, if only one taxi cruises on it, the taxi driver is more likely to find a customer. If there are multiple taxis traversing the road section, the chance of picking up customers is substantially decreased. To guarantee recommendation fairness and driving efficiency, taxis should be prevented from all driving towards a route which has high success probability. Thus, the third principle of evaluation is that when multiple routes contain a road section, no matter its place in each route, it is reasonable to share the success probability of the road section.

To effectively evaluate routes, we propose the expected driving cost (EDC) function which incorporates the three evaluation principles mentioned above.

4.2 EDC without Sharing

To simplify the discussion, we illustrate the EDC function via an example. Figure 3 shows a route composed of L road sections, $R_1 \rightarrow R_2 \rightarrow \ldots \rightarrow R_{L-1} \rightarrow R_L$. $P(R_i)$ and $T(R_i)$ are the success probability and driving cost of road section R_i ($1 \leqslant i \leqslant L$), respectively. Let E_i be the event that a taxi driver picks up a customer at road section R_i. When a taxi driver follows this

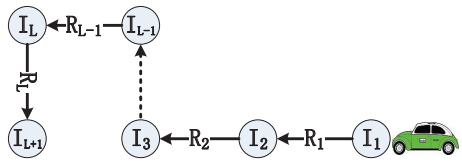


Figure 3: An example of a route

route, he/she may pick up a customer at road section R_i with a probability $P(E_i)$. Let E_∞ be the event that a pick-up event never happens. $\Omega = \{E_i\}_{i\in[1,L]} \cup E_\infty$ defines the universe of events and $P(\Omega) = \sum_{i=1}^{L} P(E_i) + P(E_\infty) = 1$. We define the EDC function as:

$$EDC = \sum_{i=1}^{L} T(E_i)\frac{(2^{P(E_i)}-1)}{i} + T(E_\infty)\frac{(2^{P(E_\infty)}-1)}{L+1}, \quad (1)$$

where the events probabilities are:

$$P(E_i) = \begin{cases} P(R_1) & i = 1 \\ P(R_i)\prod_{j=1}^{i-1}(1-P(R_j)) & i \in [2, L] \\ \prod_{j=1}^{L}(1-P(R_j)) & i = \infty \end{cases} \quad (2)$$

and the driving costs are:

$$T(E_i) = \begin{cases} T(R_1)(1-P(R_1)) & i = 1 \\ \sum_{j=1}^{i-1} T(R_j) + T(R_i)(1-P(R_i)) & i \in [2, L] \\ \sum_{j=1}^{L} T(R_j)/P(R_j) & i = \infty \end{cases} \quad (3)$$

In Equation (1), the exponential processing of $P(E_i)$ realizes the priority principle, which highlights the priority of success probability over driving cost. An illustration is given in Table 1. Routes W_1 and W_2 are composed of two road sections. The first road section of W_1 has a lower success probability and driving cost, while the first road section of W_2 has a higher success probability and driving cost. The second road sections of W_1 and W_2 have the same success probability and driving cost. In this case, according to the priority principle, W_2 should be superior to W_1 in terms of EDC. Due to the exponential processing, the priority of success probability over driving cost is exacerbated with higher success probability values. For example, as listed in Table 1, due to the high success probability of W_4's first road section, even the driving cost of W_4's first road section is four times that of W_3, and W_4 is still better than W_3 in terms of EDC.

The denominators in Equation (1), i and $L+1$, fulfil the decaying principle. For example, routes W_5 and W_6 listed in Table 1 both contain two road sections with the same driving cost. W_5's first road section has a larger success probability, and the second road section has a smaller success probability. On the contrary, W_6's first road section has a smaller success probability, while the second road section has a larger success probability. In this case, due to the decaying principle, W_6 is superior to W_5 in terms of EDC.

4.3 EDC with Sharing

The sharing of road sections occurring in routes makes the evaluation of routes more complicated and costly. According to

	$T(R_1)$	$T(R_2)$	$P(R_1)$	$P(R_2)$	EDC
W_1	2	4	0.2	0.2	6.13
W_2	3	4	0.3	0.2	5.54
W_3	2	4	0.4	0.2	3.90
W_4	8	4	0.7	0.2	3.63
W_5	3	2	0.1	0.2	9.13
W_6	3	2	0.2	0.1	8.04

Table 1: Example routes used to explain the evaluation principles

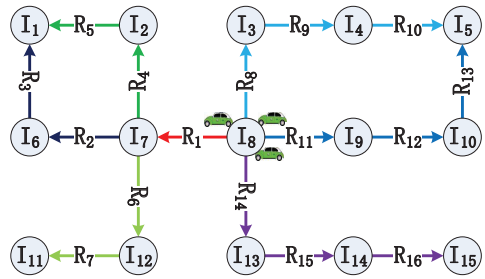


Figure 4: An illustration of routes sharing a common road section

ID	Road Sections
W_1	$R_1 \to R_2 \to R_3$
W_2	$R_1 \to R_4 \to R_5$
W_3	$R_1 \to R_6 \to R_7$
W_4	$R_8 \to R_9 \to R_{10}$
W_5	$R_{11} \to R_{12} \to R_{13}$
W_6	$R_{14} \to R_{15} \to R_{16}$

Table 2: List of candidate routes

the sharing principle, if a road section is contained in multiple routes, the success probability of the road section should be shared by these routes. The problem of implementing the sharing principle is that when a road section is shared by multiple routes, it is non-trivial to compute the recommended routes from the candidate ones since the shared times of the road section is conditional. Therefore, conditional EDC should be computed for routes containing common road sections.

An example is given to illustrate the definition of conditional EDC. As shown in Figure 4, there are six candidate routes listed in Table 2. Of these routes, W_4, W_5 and W_6 are independent routes, while W_1, W_2 and W_3 share a common road section R_1. Three taxi drivers located at intersection I_8 request route recommendations. Therefore, three recommended routes should be selected from the set of candidate routes $\{W_1, ..., W_6\}$ in terms of EDC. Concerning the sharing times of R_1, three situations may appear, which are listed in the following.

1. The sharing principle is applied by simply dividing the success probability of R_1 by the number of routes that contain R_1. In this example, the success probability of R_1 is divided by 3. Due to the division of R_1's success probability, W_4, W_5 and W_6 may be selected as the three optimal routes.
2. R_1 is assumed to be shared by only two routes. Since the success probability of R_1 is just divided by two, the possibility of W_1 and W_2 being optimal is promoted. So, W_1, W_2 and W_5 may be selected as the three optimal routes.
3. If the principle of sharing success probability is not applied in the computation of EDC values, W_1, W_2 and W_3 may be selected as the three optimal routes for recommendation.

However, statically determining the sharing times, such as the above three policies, is not correct, because the above three situations maybe occur simultaneously. The shared times of R_1

should be determined conditionally in a combination that selects three routes from the candidate set. For the example shown in Figure 4, there are $C(6,3) = 20$ combinations to select three routes from the six candidate ones. In each combination, the sharing times of a road section is deterministic. Therefore, conditional EDC should be computed for each route according to the actual shared times of R_1 in the combination. The combination with the minimum summed conditional EDC value is the solution which contains the three optimal routes.

Let $S(R_i, c)$ be the number of routes that share the road section R_i in a combination c, the conditional events probabilities are defined as:

$$P(E_i, c) = \begin{cases} \frac{P(R_1)}{S(R_1,c)} & i = 1 \\ \frac{P(R_i)}{S(R_i,c)} \prod_{j=1}^{i-1} (1 - \frac{P(R_j)}{S(R_j,c)}) & i \in [2, L] \\ \prod_{j=1}^{L} (1 - \frac{P(R_j)}{S(R_j,c)}) & i = \infty \end{cases} \quad (4)$$

The conditional EDC of a route in the combination c is defined as:

$$EDC = \sum_{i=1}^{L} T(E_i) \frac{(2^{P(E_i,c)} - 1)}{i} + T(E_\infty) \frac{(2^{P(E_\infty)} - 1)}{L+1} \quad (5)$$

The division of $P(R_i)$ by $S(R_i, c)$ realizes the sharing principle established for route evaluation, preventing taxis from all driving towards a route with high success probability. The success probability of a road section is an expectation used in the evaluation function. Actually, the arrival time of taxis may affect the sharing of success probabilities, but the arrival order cannot be taken into account in advance. It would be too complicated to model the arrival order in the evaluation function. To simplify the model, the time relations are not considered in the evaluation function.

4.4 Computation

According to the definition of EDC, optimal route computation is a combinatorial optimization problem in nature. Even though the number of candidate routes is small, the number of combinations is very large. Therefore, the computation performance should be considered seriously. We deal with this problem from three aspects. First, the computation of candidate routes can be done offline. Second, to reduce computation cost, it is acceptable to obtain an approximate solution rather than the exact one. Third, the divide-and-conquer idea can be utilized to decrease the computation complexity by dividing a map into multiple overlapping areas.

Actually, the computation of candidate routes can be done offline. The information that is needed to compute candidate routes consists of the road network, the route length and the traffic patterns (success probabilities and driving costs of road sections), which are all available in advance. With the availability of the road network and the route length, all candidate routes can be generated by using the depth first search algorithm which considers the restrictions in the road network, such as one-way streets or no-turn-right intersections. It is not necessary to generate candidate routes for all intersections, just for part of the popular intersections which can be extracted from the trajectories.

Since computing optimal routes is a combinatorial optimization problem in nature, multiple optimization algorithms can be utilized to solve the problem. In the experiments, we use Genetic Algorithm (GA) [7] to compute recommended routes by obtaining an approximate solution. In the initialization stage, N routes are selected from the candidate set, where N is also the number of taxi drivers requesting route recommendations. The EDC function is used as the fitness function to evaluate the solution domain. By iterating a certain number of rounds, GA produces an approximate solution which contains the recommended routes.

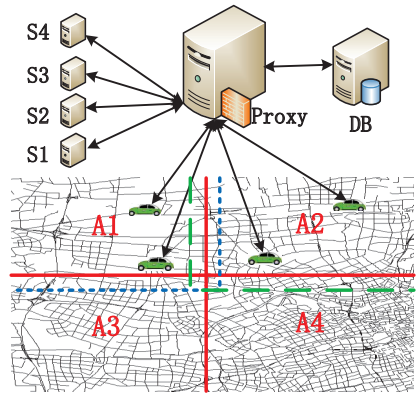


Figure 5: Overlapping division of a map

For big cities, such as New York or Shanghai, there are a large number of intersections and road sections. It is challenging to compute recommended routes by using only one server. This challenge can be addressed by dividing a road network into multiple overlapping areas. An area corresponds to a server which is responsible for processing requests from the area. Since a route may span more than one area, overlapping division is used to deal with this situation. The degree of overlapping increases with route length. An example shown in Figure 5 is given to illustrate the overlapping division. In the example, a map is divided into four areas, denoted by $A1, A2, A3$ and $A4$. The blue and green dashed lines represent the overlapping area of $A1$ and $A2$, respectively. Four servers are deployed to provide the route recommendation service, each serving an area. Through the divide-and-conquer method, the computation performance can be significantly improved, fully satisfying the performance requirement of real applications. The computational resources can be provided by cloud computing platforms, such as EC2.

5. ROUTE ASSIGNMENT MECHANISM

The goal of designing a route assignment mechanism is to propose an assignment function $f : D_i \rightarrow W_j$ that assigns each taxi driver a route. The assignment problem has been addressed largely by mechanism designers [5] [2]. Assignment mechanisms can be classified into two families. Money mechanisms [10] allow the exchange of currency during the assignment and non-money mechanisms [13] do not allow it. The route assignment problem belongs to the family of money mechanisms.

A fair route assignment mechanism for taxi drivers should meet the following conditions. First, it should guarantee recommendation fairness from a long-term perspective. Some taxi drivers may seldom request route recommendations while others may frequently do so. The assignment mechanism should be able to guarantee recommendation fairness for taxi drivers who frequently request route recommendations. Second, the assigned routes cannot be predicted by taxi drivers, even given the set of recommended routes. For example, when a driver is assigned the worst route at the previous stage, it should not be guaranteed that the best route will be assigned to him in the current stage. Therefore, the mechanism should make the assignment decision based on multiple historical assignment records, not just one. Third, the assignment mechanism should prevent violent vibration when consecutively

assigning routes to a taxi driver. By considering all these aspects, we design a fair route assignment mechanism (SCRAM) which works through the following steps:

1. At the beginning of the assignment, the balance of each driver $D_i \in D$ is set to $b_i^0 = 0$.
2. At stage k, the routes are sorted on their EDC values in ascending order and the drivers are sorted on their balances in descending order. Then, the jth driver in the driver list is assigned the jth route in the route list.
3. After the assignment at stage k, the balances are recomputed as $b_i^{k+1} = \frac{\sum_{j=k-s-1}^{k} b_i^j}{s} + c_i^k$, where $c_i^k = V_i^k - m^k$ is the amount charged to driver D_i. m^k is the mean value of the routes' EDC values and V_i^k is the EDC value of the route assigned to driver D_i at stage k.

In SCRAM, the driver with the highest balance receives the best route. This mechanism makes the standard deviation of drivers' balances at each stage reach its minimum. Experimental results also verify that SCRAM is capable of guaranteeing recommendation fairness for a group of competing taxi drivers.

As mentioned in Section 3, a group of competing taxis may be scattered in a small area, and the nearest intersection from the center of the area is selected as the starting point to generate candidate routes. So, some taxis that are not located at the intersection need to get there to follow the recommend routes. This situation may affect the performance of SCRAM, but the impact is negligible due to two facts. First, since the road sections near the starting intersection are originally intensely shared, the gathering of taxis would not greatly exacerbate the competition. Second, in addition to the intra-group competition arising from the taxi drivers who request route recommendations, taxi drivers also face inter-group competition coming from taxi drivers who do not request route recommendations. The sharing principle considers the intra-group competition, while the decaying principle takes into account the inter-group competition. The combination of the two principles well models reality.

5.1 Properties of SCRAM

THEOREM 1. *In SCRAM, the expected balance of drivers at each stage is zero,* $\sum_{i=1}^{N} b_i^k = 0, k \in N^+$.

PROOF. We will prove that the assignment problem is a zero-sum game which means that the sum of drivers' balances b_i^k is 0 for $\forall k \in N^+$. This means that

$$\sum_{i=1}^{N} \frac{b_i^k}{N} = 0, \forall k \in N^+. \tag{6}$$

We can show by induction that for $\forall k \in N^+$, the following proposition is true

$$\sum_{i=1}^{N} b_i^k = 0, \forall k \in N^+. \tag{7}$$

The game starts with zero balance for all drivers, $b_i^0 = 0, i \in [1, N]$, therefore

$$\sum_{i=1}^{N} b_i^0 = 0, \tag{8}$$

so the proposition (7) is true for $k = 0$.

Suppose that the proposition (7) is true for k, then for stage $k+1$

$$\begin{aligned} \sum_{i=1}^{N} b_i^{k+1} &= \sum_{i=1}^{N} \left(\frac{\sum_{j=k-s-1}^{k} b_i^j}{s} + c_i^k \right) \\ &= \sum_{i=1}^{N} \left(\frac{b_i^{k-s-1} + b_i^{k-s} + ... + b_i^k}{s} + V_i^k - m^k \right), \end{aligned} \tag{9}$$

where m^k is the mean of the routes' EDC values,

$$m^k = \sum_{i=1}^{N} \frac{V_i^k}{N}. \tag{10}$$

So

$$\begin{aligned} \sum_{i=1}^{N} b_i^{k+1} &= \sum_{i=1}^{N} \frac{b_i^{k-s-1}}{s} + \sum_{i=1}^{N} \frac{b_i^{k-s}}{s} + ... + \sum_{i=1}^{N} \frac{b_i^k}{s} \\ &\quad + \sum_{i=1}^{N} V_i^k - \sum_{i=1}^{N} m^k \\ &= \sum_{i=1}^{N} \frac{b_i^{k-s-1}}{s} + \sum_{i=1}^{N} \frac{b_i^{k-s}}{s} + ... + \sum_{i=1}^{N} \frac{b_i^k}{s} \\ &\quad + \sum_{i=1}^{N} V_i^k - N m^k \end{aligned} \tag{11}$$

Since $\sum_{i=1}^{N} b_i^{k-s-1} = \sum_{i=1}^{N} b_i^{k-s} = ... = \sum_{i=1}^{N} b_i^k = 0$ and $\sum_{i=1}^{N} V_i^k - N m^k = 0$, so

$$\sum_{i=1}^{N} b_i^{k+1} = 0. \tag{12}$$

Therefore, the proposition (7) is true for $k+1$. By the principle of induction, the proposition (7) is true for $\forall k \in N^+$. □

THEOREM 2. *In SCRAM, the driver with the highest balance receives the best route. This mechanism makes the standard deviation of drivers' balances at each stage reach its minimum,* $min(\sum_{i=1}^{N} \frac{(b_i^k)^2}{N})^{1/2}, \forall k \in N^+$.

PROOF. As proved in Theorem 1, the mean of drivers' balances at each stage is zero. The standard deviation σ of drivers' balances at stage $k+1$ is

$$\sigma = \left(\sum_{i=1}^{N} \frac{(b_i^{k+1})^2}{N} \right)^{1/2}, \tag{13}$$

where $b_i^{k+1} = \frac{\sum_{j=k-s-1}^{k} b_i^j}{s} + c_i^k$. For presentation simplicity, let x_i denote $\frac{\sum_{j=k-s-1}^{k} b_i^j}{s}$ and y_i represent c_i^k at stage k, then

$$\sigma = \left(\sum_{i=1}^{N} \frac{(x_i + y_i)^2}{N} \right)^{1/2} = \left(\sum_{i=1}^{N} \frac{x_i^2 + 2x_i y_i + y_i^2}{N} \right)^{1/2}. \tag{14}$$

As mentioned above, the problem of route assignment is to propose a function $f : D_i \to W_j$ for $1 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant N$. Given the order of drivers, route assignment is to get a permutation of $\{1, 2, ..., N\}$ which realizes the map from drivers to routes. Let S_N be the set of all permutations of $\{1, 2, ..., N\}$ and θ is a permutation in S_N. $\theta(i)$ denotes the ith element in θ. The problem of minimizing σ is equivalent to

$$\theta^* = \arg \min_{\theta \in S_N} \sum_{i=1}^{N} x_i y_{\theta(i)} \tag{15}$$

Without loss of generality, we suppose that x_i is sorted in ascending order and y_i is sorted in descending order. This is enough to find a permutation $\theta \in S_N$ which minimizes

$$\sum_{i=1}^{N} x_i y_{\theta(i)} \tag{16}$$

In the following, the notation $\sum_{i=1}^{N} x_i y_{\theta(i)}$ is simplified by $\sum_N(\theta)$. We will prove by induction that for $N \geqslant 2$ the following formula is true

$$\sum_N (Id) \leqslant \sum_N (\theta), \forall \theta \in S_N, \tag{17}$$

where Id is the identity permutation applied in SCRAM.

For $N = 2$, $S_2 = \{\theta_1, \theta_2\}$ where $\theta_1 = Id$ and θ_2 is defined as $\theta_2(1) = 2$ and $\theta_2(2) = 1$. So

$$\begin{aligned} \textstyle\sum_N(\theta_2) - \sum_N(Id) &= x_1y_2 + x_2y_1 - x_1y_1 - x_2y_2 \\ &= (y_2 - y_1)(x_1 - x_2). \end{aligned} \tag{18}$$

Since x_i is in ascending order and y_i is in descending order, $\sum_N(Id) \leqslant \sum_N(\theta_2)$. Formula (17) is true for $N = 2$.

Suppose that Formula (17) is true for $N > 2$. Let $x, y \in R^{N+1}$, such that x_i is sorted in ascending order and y_i is sorted in descending order. We need to prove

$$\sum_{N+1} (Id) \leqslant \sum_{N+1} (\theta), \forall \theta \in S_{N+1}. \tag{19}$$

For all $\theta \in S_{N+1}$,

$$\sum_{N+1} (\theta) = \sum_{i=1}^{N+1} x_i y_{\theta(i)} = \sum_{i=1}^{N} x_i \bar{y}_{\bar{\theta}(i)} + x_{N+1} y_j, \tag{20}$$

where $j = \theta(N+1)$, $\bar{y} = (y_1, ..., y_{j-1}, y_{j+1}, ..., y_{N+1})$ and $\bar{\theta} \in S_N$ is defined as

$$\bar{\theta}(i) = \begin{cases} \theta(i) & if\ \theta(i) \in \{1, ..., j-1\} \\ \theta(i) - 1 & if\ \theta(i) \in \{j+1, ..., N+1\} \end{cases} \tag{21}$$

Since x_i is in ascending order and $\bar{y}_i$ is in descending order, the induction assumption implies

$$\sum_{N+1} (\theta) \geqslant \sum_{i=1}^{N} x_i \bar{y}_i + x_{N+1} y_j = A_j. \tag{22}$$

Let's compute the difference $A_j - A_{N+1}$ for $j \in \{1, ..., N\}$,

$$\begin{aligned} A_j - A_{N+1} &= \sum_{i=1}^{N} x_i \bar{y}_i + x_{N+1} y_j - \left(\sum_{i=1}^{N} x_i y_i + x_{N+1} y_{N+1} \right) \\ &= \sum_{i=1}^{j-1} x_i y_i + \sum_{i=j}^{N} x_i y_{i+1} + x_{N+1} y_j \\ &\quad - \sum_{i=1}^{N} x_i y_i - x_{N+1} y_{N+1} \\ &= -\sum_{i=j}^{N} x_i y_i + \sum_{i=j}^{N} x_i y_{i+1} + x_{N+1}(y_j - y_{N+1}) \\ &= \sum_{i=j}^{N} x_i (y_{i+1} - y_i) + x_{N+1}(y_j - y_{N+1}) \\ &\geqslant \sum_{i=j}^{N} x_{N+1}(y_{i+1} - y_i) + x_{N+1}(y_j - y_{N+1}) \\ &= x_{N+1}(y_{N+1} - y_j) + x_{N+1}(y_j - y_{N+1}) = 0 \end{aligned} \tag{23}$$

where $A_{N+1} = \sum_{N+1}(Id)$. We have

$$\sum_{N+1} (Id) \leqslant A_j \leqslant \sum_{N+1} (\theta), \forall \theta \in S_{N+1} \tag{24}$$

Therefore, Formula (17) is true for $N + 1$. By the principle of induction, Formula (17) is true for all $N \geqslant 2$. □

THEOREM 3. *In SCRAM, the sum of each driver's balance is converged to a constant C,* $\lim_{N \to \infty} \sum_{k=1}^{N} b_i^k = C$ *for all* $D_i \in D$.

PROOF. As designed in SCRAM, the balance of driver D_i at stage $k+1$ is $b_i^{k+1} = \frac{\sum_{j=k-s-1}^{k} b_i^j}{s} + c_i^k$, where $c_i^k = V_i^k - m^k$. For the sake of simplicity, set s=1. Then $b_i^{k+1} = b_i^k + c_i^k$, we get the series of b_i^N as

$$\begin{aligned} b_i^0 &= 0 \\ b_i^1 &= b_i^0 + c_i^1 = c_i^1 \\ b_i^2 &= b_i^1 + c_i^2 = c_i^1 + c_i^2 \\ & \ldots\ldots \\ b_i^N &= \textstyle\sum_{k=1}^{N} c_i^k = \sum_{k=1}^{N} (V_i^k - m_i^k) \end{aligned} \tag{25}$$

The sum of D_i balance is

$$\sum_{k=1}^{N} b_i^k = \sum_{k=1}^{N} \sum_{j=1}^{k} (V_i^j - m_i^j) \tag{26}$$

When $s = 1$, the priority at stage $k+1$ is determined by the priority at stage k. If the driver D_i gets a higher priority at stage k, then he will receive a relatively lower priority at stage $k+1$. This relationship can be expressed by a function f

$$(V_i^{k+1} - m_i^{k+1}) = f(V_i^k - m_i^k). \tag{27}$$

Therefore, the value of c_i^k is not monotonically increased with the stage, but fluctuant, either positive or negative. So,

$$\lim_{N \to \infty} \sum_{k=1}^{N} b_i^k = C. \tag{28}$$

□

6. SIMULATION RESULTS

We evaluate the recommendation fairness and driving efficiency of SCRAM through simulations which are conducted based on real world trajectories. Two parameters are used in the experiments, where N is the number of competing drivers and L is the length of recommended routes.

6.1 Setting

6.1.1 Road Network

The simulation is based on the road network of Shanghai which contains about 22,420 intersections and 32,919 road sections. The average length of road sections is 0.45 km.

6.1.2 Trajectories

The taxi trajectories were collected in Shanghai, China from approximately 4000 probe-taxis operating over a period of 126 days [1]. A trajectory consists of a sequence of points. Each point contains seven fields: ID, timestamp, longitude, latitude, speed, angle, and status. The meaning of the first six fields is well understood. The last field is the current status of a taxi, "0" indicating vacant and "1" for occupied.

6.1.3 Traffic Patterns

Traffic patterns, such as success probabilities of road sections, are extracted from the trajectories. These patterns are time dependent. A common way to compute time-dependent pattern knowledge is to partition a day into a certain number of fixed time slots (e.g., 30 minutes a slot), as utilized in [6]. In the experiments, we also employ this approach. Given a road section, the number of vacant taxis passing the road section is counted, and the number of taxis that succeed in finding customers at the road section is also summed in each time slot. The two numbers are used to compute the success probability of the road section. Traffic patterns are usually updated periodically (weekly or monthly) with new taxi trajectories. The success probabilities of road sections and the distribution of customers are extracted in the time slot 8:00AM - 8:30AM.

6.2 Compared Methods

SCRAM is compared with three approaches. The first one is RAN which is used as a baseline. In RAN, recommended routes are randomly selected from the candidate set and then routes are randomly assigned to drivers. The second approach is LCP, which was proposed in [6]. The route length of LCP is set to three in the experiments. Please note the route length of LCP is the number of pick-up points, rather than the number of road sections. In the experiments, we utilize the shortest path algorithm to determine the actual route connecting two pick-up points for LCP. The round-robin mechanism is used to assign routes to taxis in LCP. The third approach is a version of SCRAM without considering the sharing of road sections, denoted by SCRAM-W/O. By comparing SCRAM with SCRAM-W/O, we highlight the effect of road section sharing on recommendation fairness and driving efficiency.

To provide a fair comparison, the number of road sections contained in recommended routes should be the same for the four approaches. For RAN, SCRAM-W/O, and SCRAM, the length of recommended routes is equal. Since LCP recommends long-distance routes and uses the round-robin mechanism to assign routes, the number of recommended routes should be determined by the number of road sections contained in the routes of SCRAM. For example, there are 10 taxi drivers requesting route recommendations, and the route length of SCRAM and LCP is 8 and 16, respectively. For SCRAM, 10 routes are recommended and assigned to taxi drivers. For LCP, only 5 routes are selected as the recommended routes, which are assigned to taxi drivers by the round-robin mechanism.

Both LCP and T-Finder [17] extract small popular areas from the trajectories of high-profit taxi drivers. These areas are called pick-up points in LCP, while they are termed as parking places in T-Finder. A route recommended by LCP is simply a sequence of pick-up points without actual routes to connect these points. T-Finder goes one step further by giving the actual route connecting two parking places. The shortest path algorithm is used by T-Finder to generate actual routes. In the experiments, we adopt the same algorithm to generate the actual route connecting two pick-up points for LCP. Therefore, LCP is extremely similar to T-Finder, which is not compared in the experiments to avoid duplication.

6.3 Evaluation Results

When a taxi driver is provided with a recommended route, if there are customers distributed on the route and the taxi driver reaches a customer earlier than other competing taxis by following the route, the taxi driver succeeds in picking up a customer. In our simulations, each experiment is conducted for ten rounds with different starting intersections that are randomly selected from the set of all intersections. In each round, a taxi driver is recommended a route and is marked as to whether the taxi driver picks up a customer or not. If the taxi driver succeeds in picking up a customer, he stops cruising on the route, and the distance from the departing location to the place of the customer is calculated as the driving cost. Otherwise, the distance from the departing location to the end of the recommended route is computed. We use the speed constrained on road sections as the driving speed in the experiments.

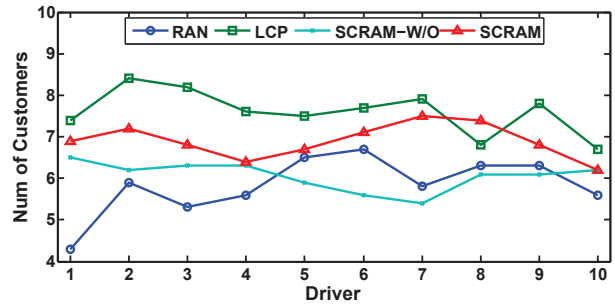


Figure 6: Customers picked up by taxi drivers.

	Min	Max	Avg	Std
RAN	4.30	6.70	5.91	0.72
LCP	6.70	8.40	7.53	0.56
SCRAM-W/O	5.40	6.30	5.96	0.35
SCRAM	6.20	7.50	6.86	0.50

Table 3: Statistics of picked up customers

6.3.1 The Number of Picked up Customers

In the experiment, the number of competing taxi drivers N is ten and the length of recommended routes L is set to $L = 8$. The number of picked up customers is summed for each taxi driver. The results are depicted in Figure 6. Two observations can be made from the figure. First, LCP is able to find more customers than RAN, SCRAM-W/O and SCRAM, by, on average, 27%, 24% and 11%, respectively. This is attributable to the fact that the recommended routes of LCP have longer distance than the ones of the other three approaches. Therefore, the sharing degree of road sections is correspondingly less intense for LCP. Second, although the set of candidate routes is the same for RAN, SCRAM-W/O and SCRAM, SCRAM finds 14% and 12% more customers than RAN and SCRAM-W/O, respectively. The reason is that SCRAM considers the sharing of road sections when computing recommended routes, alleviating unnecessary competition. The minimum, maximum, average and standard deviation are listed in Table 3. It is observed that there is no obvious difference in the standard deviations.

6.3.2 Driving Distance

The setting of the experiment is the same as the previous one. For each taxi driver, the driving distance is summed for ten rounds. The results are plotted in Figure 7. LCP finds the largest number of customers at the highest driving cost. On average, LCP incurs 137%, 134% and 223% more driving cost than RAN, SCRAM-W/O and SCRAM, respectively. SCRAM incurs the least driving cost to find customers and the driving distances of taxi drivers are more stable. The minimum, maximum, average and standard deviation of driving distances are listed in Table 4. It is observed that the standard deviation increases with driving distances. LCP fluctuates more intensely than the other three approaches.

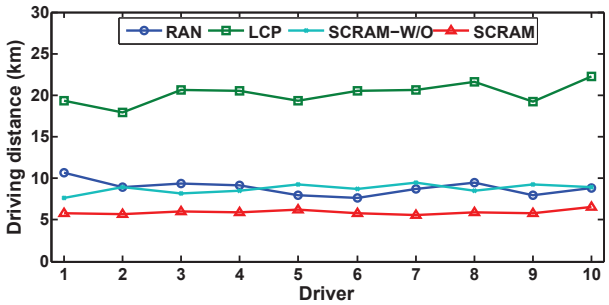


Figure 7: Driving distance of taxi drivers.

	Min	Max	Avg	Std
RAN	7.63	9.50	8.62	0.72
LCP	17.95	22.23	20.48	1.36
SCRAM-W/O	7.67	9.45	8.80	0.55
SCRAM	5.56	6.55	5.99	0.35

Table 4: Statistics of driving distances

6.3.3 *Driving Cost per Customer*

Neither the number of customers nor the driving distance is sufficient to represent the recommendation fairness and driving efficiency of taxi drivers. It is meaningless to consider only one metric. In order to compare the recommendation fairness and driving efficiency of taxi drivers, we propose a new metric called driving cost per customer (DCC), which is defined as:

$$DCC = \frac{driving\ cost}{number\ of\ customers}. \tag{29}$$

DCC represents the average driving cost to find a customer, reflecting the driving efficiency of taxi drivers. Meanwhile, the standard deviation of DCC indicates the recommendation fairness of competing taxi drivers.

The DCC values of the four approaches are shown in Figure 8. On average, SCRAM is 56%, 189% and 55% more efficient than RAN, LCP, and SCRAM-W/O, respectively. The minimum, maximum, average, and standard deviation of DCC are listed in Table 5. It is obvious that the standard deviation of DCC increases with the values of DCC. RAN, LCP and SCRAM-W/O fluctuate intensely. On the contrary, SCRAM is much more stable due to the consideration of road section sharing in the computation of optimal routes and the design of an assignment mechanism to guarantee recommendation fairness. SCRAM achieves 322%, 278% and 77% better recommendation fairness than RAN, LCP and SCRAM-W/O, respectively.

Overall, LCP provides taxis with long-distance routes to traverse, while SCRAM suggests taxis cruise in the local area.

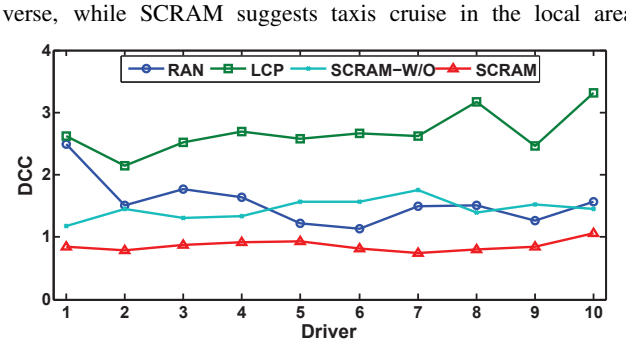


Figure 8: DCC of taxi drivers.

	Min	Max	Avg	Std
RAN	1.14	2.49	1.56	0.38
LCP	2.14	3.32	2.68	0.34
SCRAM-W/O	1.18	1.75	1.45	0.16
SCRAM	0.74	1.06	0.86	0.09

Table 5: Statistics of DCC

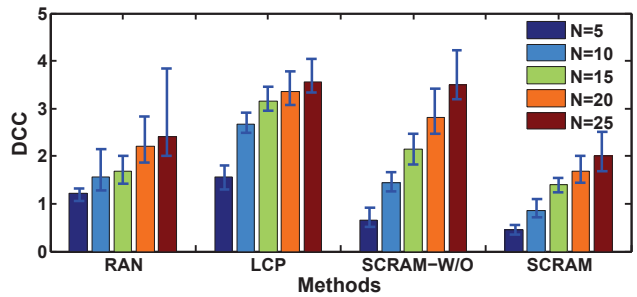


Figure 9: The effect of taxi drivers.

	N=5	N=10	N=15	N=20	N=25
RAN	0.15	0.38	0.31	0.49	0.87
LCP	0.30	0.34	0.45	0.63	0.60
SCRAM-W/O	0.10	0.16	0.43	0.70	0.99
SCRAM	0.04	0.09	0.14	0.29	0.38

Table 6: Standard deviation of DCC

Therefore, the area covered by the routes of SCRAM is smaller than that of LCP. A route recommended by LCP is longer and has more chance of finding customers, compared with SCRAM. However, smaller is not necessarily bad. As demonstrated by the experimental results, SCRAM is fairer and more efficient than LCP in terms of driving cost per customer.

6.4 Scalability

More insights of scalability are presented for the four approaches in this section. The recommendation fairness and driving efficiency of SCRAM are evaluated by changing the number of taxi drivers and the length of recommended routes.

6.4.1 *The Number of Taxi Drivers*

Under the same conditions, increasing the number of taxis aggravates the competition, which correspondingly leads to the growth of DCC. As shown in Figure 9, DCC increases with the number of taxi drivers for the four approaches. For RAN and LCP, the superiority of SCRAM gradually diminishes with the increase of taxis due to the aggravated competition. For example, when $N = 5$, SCRAM is 164% and 240% more efficient than RAN and LCP, respectively. When $N = 25$, SCRAM is only 19% and 77% more efficient than RAN and LCP, respectively. However, the superiority of SCRAM over SCRAM-W/O increases with taxis. For example, when $N = 5$, SCRAM is 44% more efficient than SCRAM-W/O, but when $N = 25$, SCRAM is 74% more efficient than SCRAM-W/O. This demonstrates that SCRAM-W/O is more affected by the number of taxis than other approaches. The standard deviations of DCC are listed in Table 6. Overall, the standard deviation of DCC increases with the number of taxi drivers. When $N = 25$, SCRAM still achieves 127%, 56% and 160% better recommendation fairness than RAN, LCP and SCRAM-W/O, respectively, even under intense competition.

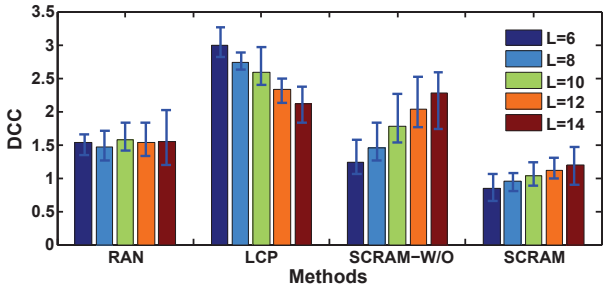


Figure 10: The effect of route length.

	L=6	L=8	L=10	L=12	L=14
RAN	0.15	0.23	0.24	0.25	0.33
LCP	0.34	0.22	0.44	0.29	0.39
SCRAM-W/O	0.14	0.16	0.37	0.36	0.41
SCRAM	0.10	0.09	0.12	0.11	0.20

Table 7: Standard deviation of DCC

6.4.2 The Length of Recommended Routes

The effect of route length on DCC is depicted in Figure 10. According to the locality driving behavior discovered in Figure 1, it is not necessary to test large numbers of road sections in the experiment. For RAN, route length has almost no effect on DCC. The DCC of LCP decreases with route length. As mentioned above, the number of road sections contained in the recommended routes of the four approaches should be equal for the sake of fair comparison. Increasing the route length of SCRAM causes more routes to be selected for the round-robin assignment in LCP, which indirectly alleviates the competition. Increasing the route length of SCRAM indeed results in finding more customers but at the cost of driving longer distances. The increasing growth of driving distance is far larger than the increasing growth of customers, causing an increase of DCC. Without considering the sharing of road sections, the DCC of SCRAM-W/O is more easily affected by the length of routes than SCRAM. The standard deviations of DCC are listed in Table 7. On average, SCRAM achieves almost 100% better recommendation fairness than other three approaches.

7. CONCLUSION

Route recommendation for taxi drivers is of great economic and social importance. On one hand, customers can quickly find taxis, saving waiting time; on the other hand, taxi drivers can find customers in a shorter time by following the recommended routes, increasing their revenues. In this paper, we propose SCRAM, a sharing considered route assignment mechanism for fair taxi route recommendations. In SCRAM, the sharing of road sections existing in the recommended routes is considered in the process of computing optimal routes. Then, recommended routes are assigned to taxi drivers from a long-term perspective to guarantee recommendation fairness. In comparison with three approaches, SCRAM is capable of providing taxi drivers with more efficient routes that have the least DCC. Furthermore, SCRAM achieves better recommendation fairness for competing taxi drivers, which is verified by the standard deviations of DCC.

8. ACKNOWLEDGMENTS

This work is supported by National 863 Program (2015AA01A202). This work is also partially supported by China National Science Foundation (61272438, 61472253, 61420106010), Research Funds of Science and Technology Commission of Shanghai Municipality (14511107702). Frédéric Le Mouël's work is funded by a grant from Rhone-Alpes Region, France.

9. REFERENCES

[1] Suvnet-trace data. http://wirelesslab.sjtu.edu.cn.

[2] A. Abdulkadiroglu and T. Sonmez. School choice: A mechanism design approach. *The American Economic Review*, 93(3):729–747, 2003.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734–749, 2005.

[4] Y. Ding, S. Liu, J. Pu, and L. M. Ni. Hunts: A trajectory recommendation system for effective and efficient hunting of taxi passengers. In *MDM*, pages 107–116. IEEE, 2013.

[5] A. G. Erdman and G. N. Sandor. *Mechanism design: analysis and synthesis (Vol. 1)*. Prentice-Hall, Inc., 1997.

[6] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, and M. Pazzani. An energy-efficient mobile recommender system. In *SIGKDD*, pages 899–908. ACM, 2010.

[7] D. E. Goldberg et al. *Genetic algorithms in search, optimization, and machine learning*, volume 412. Addison-wesley Reading Menlo Park, 1989.

[8] H. Hu, Z. Wu, B. Mao, Y. Zhuang, J. Cao, and J. Pan. Pick-up tree based route recommendation from taxi trajectories. In *Web-Age Information Management*, pages 471–483. Springer, 2012.

[9] J. Lee, I. Shin, and G.-L. Park. Analysis of the passenger pick-up pattern for taxi location recommendation. In *NCM*, pages 199–204. IEEE, 2008.

[10] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140. ACM, 1999.

[11] S. Qian, Y. Zhu, and M. Li. Smart recommendation by mining large-scale gps traces. In *WCNC*, pages 3267–3272. IEEE, 2012.

[12] P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[13] J. Schummer and R. V. Vohra. Mechanism design without money. *Algorithmic Game Theory*, 10:243–299, 2007.

[14] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *SIGKDD*, pages 195–203. ACM, 2012.

[15] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *SIGSPATIAL*, pages 99–108. ACM, 2010.

[16] J. Yuan, Y. Zheng, L. Zhang, X. Xie, and G. Sun. Where to find my next passenger. In *UbiComp*, pages 109–118. ACM, 2011.

[17] N. J. Yuan, Y. Zheng, L. Zhang, and X. Xie. T-finder: A recommender system for finding passengers and vacant taxis. *Knowledge and Data Engineering, IEEE Transactions on*, 25(10):2390–2403, 2013.

[18] D. Zhang, T. He, Y. Liu, and J. A. Stankovic. Callcab: A unified recommendation system for carpooling and regular taxicab services. In *Big Data, International Conference on*, pages 439–447. IEEE, 2013.

[19] M. Zhang, J. Liu, Y. Liu, Z. Hu, and L. Yi. Recommending pick-up points for taxi-drivers based on spatio-temporal clustering. In *CGC)*, pages 67–72. IEEE, 2012.