Recommendation-based Team Formation for On-demand Taxi-calling Platforms

Lingyu Zhang[1], Tianshu Song[2], Yongxin Tong[2], Zimu Zhou[3], Dan Li[1], Wei Ai[4], Lulu Zhang[1]
Guobin Wu[1], Yan Liu[1] and Jieping Ye[1]
[1]AI Labs, Didi Chuxing, Beijing, China, [2]SKLSDE Lab, Beihang University, Beijing, China,
[3]ETH Zurich, Zurich, Switzerland, [4]University of Michigan, Ann Arbor, USA
[1]{zhanglingyu, lynnlidan, zhanglulululu, wuguobin, aliceliuyan, yejieping}@didiglobal.com,
[2]{songts, yxtong}@buaa.edu.cn, [3]zzhou@tik.ee.ethz.ch, [4]aiwei@umich.edu

ABSTRACT

On-demand taxi-calling platforms often ignore the social engagement of individual drivers. The lack of social incentives impairs the work enthusiasms of drivers and will affect the quality of service. In this paper, we propose to form teams among drivers to promote participation. A team consists of a leader and multiple members, which acts as the basis for various group-based incentives such as competition. We define the Recommendation-based Team Formation (RTF) problem to form as many teams as possible while accounting for the choices of drivers. The RTF problem is challenging. It needs both accurate recommendation and coordination among recommendations, since each driver can be in at most one team. To solve the RTF problem, we devise a Recommendation-Matrix-Based Framework (RMBF). It first estimates the acceptance probability of recommendations and then derives a recommendation matrix to maximize the number of formed teams from a global view. We conduct trace-driven simulations using real data covering over 64,000 drivers and deploy our solution on a large on-demand taxi-calling platform for online evaluations. Experimental results show that RMBF outperforms the greedy-based strategy by forming up to 20% and 12.4% teams in trace-driven simulations and online evaluations, and the drivers who form teams and are involved in the competition have more service time, number of finished orders and income.

KEYWORDS

Team Formation, Recommendation, Incentive Mechanism

ACM Reference Format:
Lingyu Zhang[1], Tianshu Song[2], Yongxin Tong[2], Zimu Zhou[3], Dan Li[1], Wei Ai[4], Lulu Zhang[1] and Guobin Wu[1], Yan Liu[1] and Jieping Ye[1]. 2019. Recommendation-based Team Formation for On-demand Taxi-calling Platforms. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM'19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Attracting and motivating drivers is essential for the success of on-demand taxi-calling platforms such as Didi Chuxing[1], Uber[2] and Grab[3]. Many platforms design dynamic pricing strategies [19] to attract drivers. Consequently, many platforms such as Uber managed to increase the number of monthly active drivers from 50,000 in January 2014 to over 150,000 in January 2015 [7].

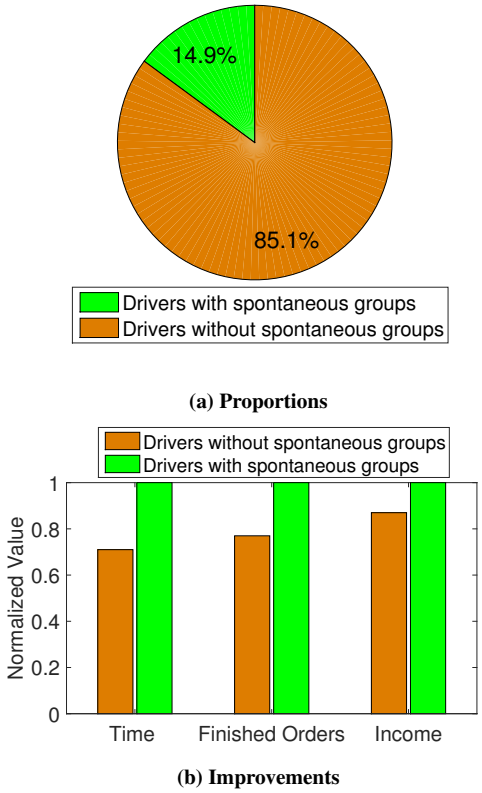
Although momentary incentives have been widely applied in on-demand taxi-calling platforms, social engagement is largely overlooked in these platforms. Among the large numbers of drivers registered, many of them feel unmotivated working as individuals, and decide to quit after a period of time. Some studies find that only 4% of the drivers will remain on Uber after one year[4].

Rather than further designing incentive mechanisms at the *individual* level [12, 24, 27], we propose to team up the drivers, where various *group*-based incentives can be applied. We are inspired by the recent findings that grouping promotes participation in crowd-based applications [1]. To explore the potential of grouping for taxi-calling services, we conduct a survey among DiDi Chuxing drivers. According to the results, 15% drivers formed spontaneous groups (see Fig. 1a). We find that drivers affiliated to at least one self-organized group tend to remain in service for a longer time, complete more taxi orders and earn more income than those not in any group (see Fig. 1b). The results are consistent with the findings in [1].

Inspired by these findings, we propose to form *teams* among taxi drivers. A team is a group of drivers with one *leader* and multiple *members*. It is envisioned to promote collaboration and coordination, and serves as the basis for *group-based* incentive mechanisms such as gamification [22] and competition [4]. For instance, DiDi chuxing has launched a team competition program where various competitions are organized for its drivers to team up and the teams that finish the highest number of taxi orders within a period can earn extra rewards. These group-based incentive mechanisms tend to motivate workers to produce better results and will eventually reduce the cost and increase the profits of crowdsourcing platforms [6].

To realize the full potential of groups in incentive mechanism design, taxi-calling platforms need to devise effective schemes to form as many teams as possible. Self-organized groups are ineffective and only contribute to a small portion of the entire population of drivers (see Fig. 1a). This is because drivers usually have limited

[1]https://en.wikipedia.org/wiki/Didi_Chuxing.
[2]https://en.wikipedia.org/wiki/Uber_(company).
[3]https://en.wikipedia.org/wiki/Grab_(application).
[4]https://www.cnbc.com/2017/04/20/only-4-percent-of-uber-drivers-remain-after-a-year-says-report.html



(a) Proportions

(b) Improvements

Figure 1: Survey results among DiDi Chuxing drivers on self-organized groups and their performance.

information about other drivers and thus can only team up locally in an ad hoc manner. In contrast, the platform has access to diverse information of drivers at a large scale and should be able to optimize the team formation process from a holistic view.

In this paper, we propose a recommendation-based approach to team formation for on-demand taxi-calling platforms. Given a set of drivers registered as either leaders or members, the platform recommends members to leaders in multiple rounds and the leaders can decide whether to accept the recommendation or not. The goal is to maximize the number of teams formed (*i.e.* groups of a leader and a given number of members). We define team formation as a recommendation problem rather than a combinational optimization problem [13] because the recommendation-based approach endows drivers' choices and hence improves their sense of participation.

Our recommendation problem also differs from conventional recommendation systems [16, 18, 20] in that we not only need high accuracy per recommendation but also coordination among recommendations to maximize the number of teams formed (since each member can only join one team). Specifically, we want to maximize the Team Formation Success Number (TFSN), *i.e.* the number of teams, where each team is exactly of the required team size (called a feasible team).

Formally, we define the Recommendation-based Team Formation (RTF) problem and prove that it is NP-hard. To solve the RTF problem, we first propose a greedy algorithm based on mainstream recommendation system designs. We further devise a Recommendation-Matrix-Based Framework (RMBF) to solve the problem from a holistic view. RMBF first estimates the probability for each leader to accept a recommended member. Afterwards it uses a recommendation matrix to optimize the TFSN in each round of recommendation, which takes both accuracy per recommendation and conflict among recommendations into consideration, and derives globally optimized results. We evaluate the performance of RMBF on real data involving over 64,000 drivers collected by DiDi Chuxing, a large on-demand taxi-calling platform in China. Trace-driven simulations show that RMBF-based approaches can form up to 20% more teams than greedy-based recommendation strategies. We also integrate our methods into the group-based competition programs held by the platform. Real-world online experiments show that RMBF can help form up to 12.4% more teams than a greedy strategy.

The main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to propose team formation in on-demand taxi-calling platforms. It serves as a basis to design group-based collaborative or competitive incentive mechanisms for such platforms.
- We define team formation as a recommendation problem and prove it is NP-hard. We then design a generic recommendation-matrix-based framework (RMBF) to solve it approximately.
- We evaluate RMBF on a large-scale on-demand taxi-calling platform using both trace-driven simulations and online evaluations. Experimental results show that RMBF outperforms the greedy-based strategy by forming up to 20% and 12.4% teams in trace-driven simulations and online evaluations.

2 RELATED WORK

This section reviews related work in recommendation systems and team formation in social networks.

2.1 Recommender Systems

Recommendation systems attempt to recommend the most suitable items to users and have been widely applied in entertainment (recommendations for movies, music), e-commerce (recommendations for consumers of products), etc. Commonly used recommendation techniques include collaborative filtering (CF) [18], content based (CB) [16] and knowledge based (KB) [20]. These techniques can be further combined and improved. For example, Yu *et al.* [25] suggest that CB and CF can be combined under a hierarchical Bayesian framework. Shang *et al.* [9] use ANN to generate the personalized recommendation. Xue *et al.* [23] present a technique where individuals are grouped and the unrated items are predicted by use of the users' ratings in a group. Zhang *et al.* [26] develop a Fuzzy-based recommender system which combines user-based and item-based collaborative filtering techniques with fuzzy set techniques to make the personalized recommendation. We refer interested readers to [3] for a comprehensive overview of recommendation systems.

Our work is inspired by the research on recommendation systems and takes a recommendation based approach to team formation.

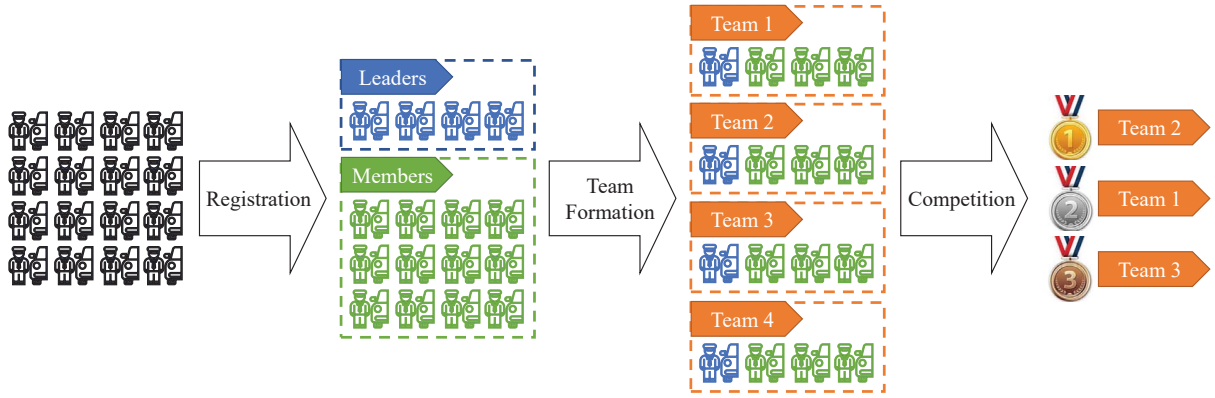


Figure 2: An example of competition-based social mechanism built upon team formation.

However, the proposed recommendation-based team formation problem differs from traditional recommendation systems in two folds. *(i)* The recommendations (*i.e.* members) in our problem are correlated (*i.e.* each member can only be in one team). *(ii)* We not only require accurate recommendation, but also coordination among recommendations. Hence our problem is more challenging.

2.2 Team Formation in Social Networks

Team formation is one important problem in social networks. In this thread of research, teams of experts with different skills are formed to complete tasks requiring multiple skills. The goal is often to find a qualified team with the minimal communication cost, which is defined based on the social graph of the experts [2, 10, 13, 14]. Theodoros *et al.* [13] propose to use the diameter and the sum of the weights of the minimum spanning tree of the team members' social graph as the communication cost. Mehdi *et al.* [10] define the communication cost as the sum of the shortest distances between the members and the leader. Other goals such as workload balance among members have also been considered [2, 14].

Our team formation problem differs from the existing literature in two aspects. *(i)* The teams in our problem are formed as the basis to design social-based incentives rather than to solve complex tasks. Hence the members in our problem are homogeneous. *(ii)* We aim to maximize the number of teams formed, while the goals of above studies are mainly related with the team's communication cost defined based on the social networks.

3 PROBLEM STATEMENT

This section introduces team formation in on-demand taxi-calling platforms (Sec. 3.1) and formally define the recommendation based team formation problem (Sec. 3.2). Finally we analyze the hardness of the problem in Sec. 3.3.

3.1 Preliminaries

To form teams on taxi-calling platforms, each driver first registers as either a leader or a member. Each team consists of one leader and multiple members. We distinguish leaders and members because research in management shows that leadership can enhance the effectiveness of the team [15]. We set only one leader for each team because one leader can make the team formation succinct, compared with no-leader or multi-leader situations. For ease of management one member can join at most one team.

Teams formed on the platform can be used to implement various social incentive mechanisms such as competition [6]. Fig. 2 illustrates an example of competition-based mechanism built upon teams. For fair competition, each team needs to be of equal size. In the rest of this work, we will take competition (and thus equal team sizes) as an example to define our problem and optimization framework. Other social incentives also apply.

- **Registration.** The taxi-calling platform publishes information about team-based competitions including the number of each team. Drivers who are willing to participate in the competitions register as leaders or members.
- **Team Formation.** Registered drivers form teams. It can be achieved by leaders inviting members or the platform recommending members to leaders.
- **Competition.** Feasible teams take part in different competitions for monetary or other rewards.

Our work focuses on the team formation stage and proposes a recommendation-based approach to maximize the number of feasible teams that can be formed.

3.2 Problem Formulation

Denote $L = \{l_1, l_2, ..., l_{|L|}\}$ as the set of drivers who register as leaders and $M = \{m_1, m_2, ..., m_{|M|}\}$ as the set of drivers who register as members. Let S be the required size of each team, which is predefined by the specific mechanisms. A team is *feasible* if its size equals S. Further assume at most N_R rounds are allowed to recommend members to leaders whose teams are not feasible yet. In each round, a certain number of members are recommended to each leader. Denote lck_l as the number of members a leader l still lacks to form a feasible team. Hence the number of recommendation should not exceed lck_l. Leaders decide whether to accept the recommendations or not before the next round. Once a member is accepted, s/he will not be recommended to other leaders in the subsequent rounds. If the same member is accepted by more than one leader, the member

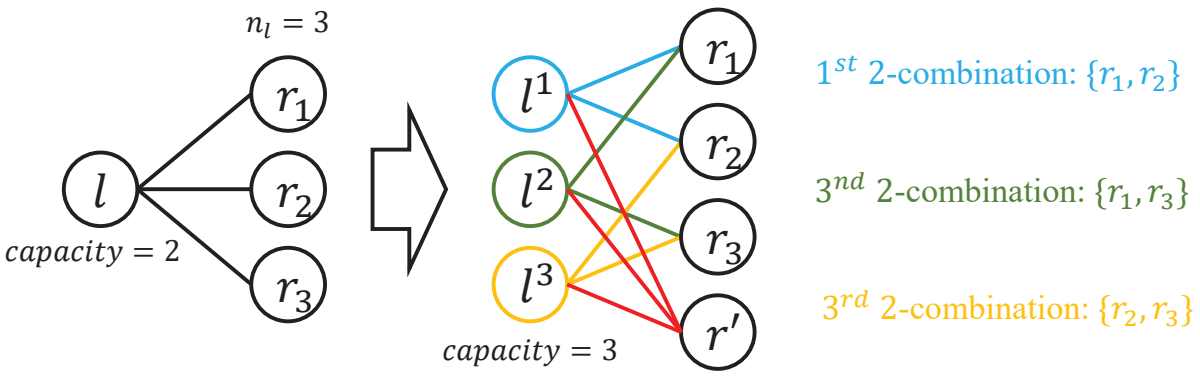


Figure 3: An example of instance transformation.

will be allocated randomly. We define the recommendation-based team formation problem (RTF) below.

DEFINITION 1 (RECOMMENDATION-BASED TEAM FORMATION). *Given sets L and M, a budget N_R on the rounds of recommendations and the size of the feasible team S, the platform recommends drivers in M who do not have a team to leaders in L whose current team size is smaller than S in each round and observes the decisions of the leaders. The goal is to maximize the number of feasible teams after N_R rounds.*

3.3 Hardness Analysis

We analyze the hardness of the RTF problem above by considering a simplified version, where for all $l \in L$ and $m \in M$, we know in advance whether or not l will accept m if m is recommended to l. Such simplified version of the RTF problem can be formulated as a bipartite matching problem called Recommendation with Oracle (RwO) as follows.

DEFINITION 2 (RECOMMENDATION WITH ORACLE). *Give an unweighted bipartite graph $G =< L, R, E >$ where L and R represent the left and right nodes, respectively, and E is the set of edges between L and R. For each $l \in L$, integer $c_l > 1$ is the capacity of l, which is the maximum number of nodes in R that can be matched to it. We say l is exhausted if exactly c_l nodes in R are assigned to it. The problem is to give a matching between L and R to maximize the number of $l \in L$ whose capacity is exhausted.*

We then analyze the hardness of the RwO problem.

LEMMA 1. *The RwO problem is NP-Hard.*

PROOF. We prove by reduction from the set packing problem [11]. In the set packing problem, we are given a list of sets $S_1, S_2, \cdots, S_n$, and the problem is to answer whether there exists p sets among them such that the p sets are mutually disjoint. Given an instance of the set packing problem, we create a left node l_i for each S_i and a right node r_j for each element s_j in $\cup_{p=1}^{n} S_p$. There is an edge between l_i and r_j if $s_j \in S_i$. The capacity c_{l_i} for l_i is set to $|S_i|$. Thus we get an instance of the decision version of RwO problem. As the set packing problem is NP-Complete, the RwO problem is NP-Hard. □

If the capacities of the nodes in L is a constant k (we call this the k-RwO problem), it is still NP-Hard.

LEMMA 2. *The k-RwO problem is NP-Hard.*

PROOF. We prove by reduction from the k-set packing problem [8]. In the k-set packing problem, all the sets $S_1, S_2, \cdots, S_n$ have the same size of k. The following proof is similar to that of Lemma 1 and we omit it here. □

We have shown that the k-set packing problem can be reduced to the k-RwO problem. Next we prove that the k-RwO problem can be reduced to the $(k+1)$-set packing problem.

LEMMA 3. *The k-RwO problem is no harder than the $(k+1)$-set packing problem.*

PROOF. The reduction is as follows. Given an instance of the k-RwO problem, for each $l \in L$, denote n_l as the number of l's neighbor nodes. For the i-th k-combination of these n_l neighbor nodes, we create a node l^i and add an edge between l^i and each of the nodes in the i-th combination. Then we create a dummy node r' for R and add an edge between all the l^i and r'. Now we get an instance of the $(k+1)$-set packing problem. Fig. 3 shows an example of how to conduct the reduction. □

From Lemma. 1 to Lemma. 3 it can be inferred that the hardness of RwO is similar to the set packing problem. Note that unless $P = NP$ the maximum k-set packing problem cannot be efficiently approximated within a factor of $\Omega(\ln k/k)$ [8]. The original RTF problem is even harder than the RwO problem because it does not know whether a leader will accept a recommendation or not. In the next two sections, we propose two solutions to the RTF problem.

4 GREEDY-BASED BASELINE SOLUTION

This section presents a greedy based baseline solution (BL for short) to the RTF problem.

Main Idea. To form as many feasible teams as possible, we first identify the team leader l^* with the smallest number of lacked members (denoted by lck_{l^*}). Ties can be broken arbitrarily. Then lck_{l^*} drivers with the highest probabilities to be accepted are recommended to l^*.

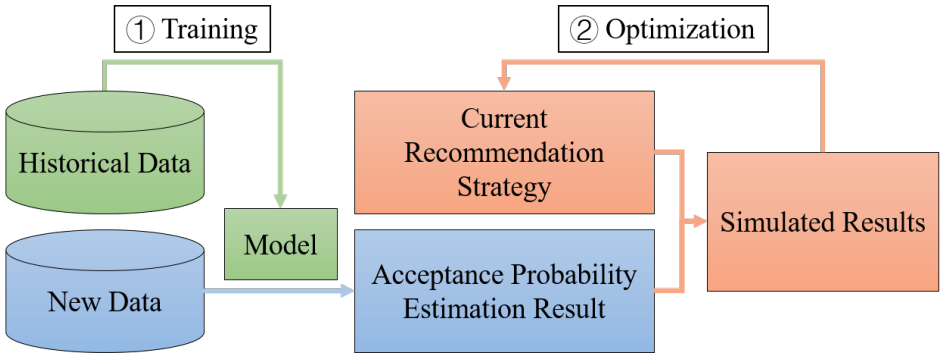


Figure 4: Recommendation-Matrix-Based Framework.

Acceptance Probability Estimation. As with mainstream recommendation systems, BL estimates the probability for each leader to accept certain recommended member from historical data. Similar to other practical recommendation systems, the platform may face the cold-start problem. BL solves the cold-start problem by using the similarity between leaders and members for recommendation before sufficient records on acceptance probability are collected. Specifically, BL use age and hometown information to calculate the similarity between leaders and members, *i.e.*, $Sim(l, m) = \alpha Sim_g(l, m) + (1-\alpha) Sim_d(l, m)$, where $Sim_g(l, m) = 1 - |g_l - g_m|/G$ and $Sim_d(l, m) = 1 - (\mathbf{1}_1(l, m) + \mathbf{1}_2(l, m) + \mathbf{1}_3(l, m))/3$. g_l and g_m are the ages of leader l and member m, respectively. G is the largest age difference among all leader-member pairs. $\mathbf{1}_1(l, m)$, $\mathbf{1}_2(l, m)$ and $\mathbf{1}_3(l, m)$ indicate whether l and m have the same level 1, 2 and 3 addresses, respectively. The similarity metric proves successful in other group-based behaviour analysis [1]. After collecting enough data, a model on acceptance probability can be trained through existing learning methods.

Algorithm 1: Baseline Algorithm

input : L, M
output : A recommendation between L and M
1 **while** $\exists l \in L$, l *is not processed* **do**
2 $l^* \leftarrow l$ with the minumum lck_l;
3 $R_{l^*} \leftarrow$ set of lck_{l^*} members in M who have the highest probability to be accepted and have not been recommended before;
4 Recommend R_{l^*} to l and mark l processed;
5 **return** $\{(l, R_l) | l \in L\}$.

Algorithm Sketch. Algorithm 1 shows the procedure of BL. While there exists a leader l whose set of recommended members is not determined (Line 1), we identify the number of members l lacks in Line 2 and recommend l the members who are not recommended before and have the highest probability to be accepted in Lines 3-4.

Summary. BL essentially takes a greedy strategy, since it preferentially recommends members who are most likely to be accepted by the leaders whose teams are easiest to be formed. Hence its optimization is *local*. This motivates us to design a solution to optimize TFSN from a global view.

5 RECOMMENDATION-MATRIX-BASED FRAMEWORK (RMBF)

In this section, we introduce RMBF, a recommendation-matrix-based framework to solve the RTF problem. The main advantage of RMBF is that it both increases the accuracy of each recommendation and decreases the conflicts among recommendations, *i.e.* one member is accepted by multiple leaders. Therefore RMBF is able to maximize the number of feasible teams from a global view.

RMBF Overview. Fig. 4 shows the workflow of RMBF.

- **Training.** As with BL (Sec. 4) , RMBF first learns a model to estimate the probability for each leader to accept certain recommended member. Then a probability matrix $P_{|M|\times|L|}$ can be derived given the sets of registered members M and leaders L, where $P_{m,l} \in [0, 1]$ indicates the probability of accepting member m by leader l.
- **Optimization.** Using P, RMBF then derives a recommendation matrix $R_{|M|\times|L|}$ which globally optimizes the TFSN in each recommendation round. R is a zero-one matrix where $R_{m,l} = 1$ means m is recommended to l and $R_{m,l} = 0$ means m is not recommended.

Essential in RMBF is the optimization stage. The main challenge is to evaluate the impact of a recommendation matrix R on the number of teams that can be formed. Then an operational objective can be defined upon such evaluation metric and classical optimization methods can be applied to find the optimized recommendations. We propose a novel method to evaluate the impact of a recommendation matrix on team formation as follows.

Metrics to Evaluate Recommendation Matrix. To assess the impact of a recommendation matrix R on the number of teams that can be formed, we derive the expected number of teams formed in a round under R, P and $\{lck_l | l \in L\}$.

We recommend at most lck_l members to l in each round. Thus, if the recommended number is less than lck_l, l will not form a feasible in this round. Thus we only consider the leaders who need exactly lck_l members to be recommended. If a member m is recommended to a leader l, the probability that m is eventually in l's team is

$$Pr(l, m) = \sum_{r'_1=0}^{1} \cdots \sum_{r'_n=0}^{1} \frac{\prod_{i=0}^{n} [P_{l'_i,m}^{r'_i} (1 - P_{l'_i,m})^{1-r'_i}]}{1 + \sum_{i=1}^{n} r'_i} \quad (1)$$

where l'_i represents the leader who also receives the recommendation of member m, and r'_i is the indicator for whether l'_i accepts m. $1 + \sum_{i=1}^{n} r'_i$ is used to rescale the probability if more than one leaders accept m. n is the number of leaders (l excluded) that m is recommended to. r'_0 is always 1, *i.e.* l accepts m and l'_0 is l. Thus we loop i from 0 in the numerator to calculate the probability whether m is accepted by different leaders.

Based on Eq. (1), the probability that l can form a feasible team in this round can be calculated by

$$Pr(l) = \prod_{i=1}^{lck_l} Pr(l, m_i). \tag{2}$$

Then the expected number of feasible teams formed in this round is

$$E(R, P) = \sum_{l \in L} Pr(l). \tag{3}$$

With Eq. (3), now we can find an optimized R given P using existing optimization methods. Note that the major complexity to compute Eq. (3) is the calculation of Eq. (1). It can be computation-prohibitive to calculate Eq. (1) if n, *i.e.*, the number of leaders (l excluded) who m is recommended to, is too large. Fortunately, to ensure the fairness, all the members should have equal chance to be recommended which limits the value of n. Generally, n is approximately the required number of a feasible team S, which can be smaller than 10 in practice.

Summary. RMBF is a generic framework to solve the RTF problem. Central in RMBF is a metric to assess the impact of a recommendation matrix on the expected number of teams that can be formed. As with BL, RMBF needs to calculate the acceptance probability of recommendations. We experiment with multiple representative methods and choose XGBoost for its simplicity and effectiveness (Sec. 6.1). With the proposed metric to evaluate a recommendation matrix, RMBF optimizes $E(R, P)$ in each round of recommendation. Many classical optimization methods can be applied to the optimization. In this work, we choose genetic algorithms as the optimization method via both simulation (Sec. 6.2) and real-world experiments (Sec. 6.3).

6 EVALUATION

This section evaluates the performance of BL and RMBF. Because estimating the acceptance probability of recommendations is the cornerstone of both BL and RMBF, we first evaluate different models for acceptance probability estimation and the effectiveness of different features in Sec. 6.1. As the cost of doing online evaluation is very high, trace-driven simulation (in Sec. 6.2)via the data collected from DiDi Chuxing is necessary, through which we can identify the best candidates for online evaluation. Finally, we conduct online evaluation on the DiDi Chuxing platform in Sec. 6.3 and discuss the effect of our group-based incentive via competition in Sec. 6.4.

6.1 Acceptance Probability Estimation

Since both BL and RMBF need to estimate the acceptance probability of drivers, in this part we explore models suited for acceptance probability estimation.

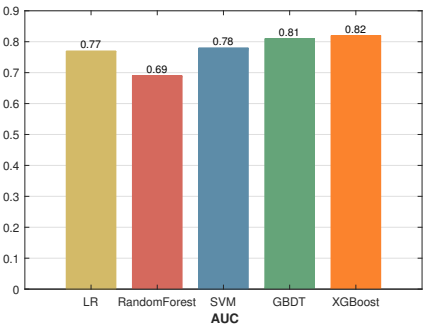


Figure 5: AUC of the compared methods.

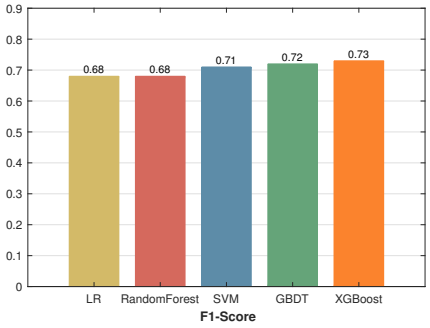


Figure 6: F1-Score of the compared methods.

Setup. We use data collected from group-based competition activities organized by DiDi Chuxing held in October, 2017 in Wuhan and Hangzhou. The dataset contains information of leaders and members, and the historical recommendation records. There are 1,705,374 records in total. Among them 5.24% are recommendations accepted by the leaders. To balance the positive and negative samples, we conduct down-sampling, after which 178,714 samples remain, and the proportion of positive samples is 50%. We use 70% of the samples for training the acceptance estimation model and the remaining 30% for testing.

We use the following features to estimate the acceptance probabilities of leaders:

- **Similarity Features:** the similarity of the ages, and hometowns of the leaders and members;
- **Individual Features:** hometown, gender and age of the leaders, and members;
- **Platform Features:** average online time, average number of finished orders in a day, and average weekly income of the leaders and members.

Table 1 summarizes the features we use for acceptance probability estimation.

To assess the importance of features, we calculate their mutual information (MI). Table 2 shows the results. As is shown, features with the highest MI are *(i)* the similarity of hometown, *(ii)* the similarity of age and *(iii)* the age of the members, which is aligned with previous studies [1].

Table 1: Description of features.

Type	Feature
Similarity Features	The similarity of the ages of the leaders and members
	The similarity of the hometowns of the leaders and members
Individual Features	Hometowns of the leaders
	Gender of the leaders
	Age of the leaders
	Hometowns of the members
	Gender of the members
	Age of the members
Platform Features	The average online time of the leaders in a day
	The average number of finished orders of the leaders in a day
	The average weekly income of the leaders
	The average online time of the members in a day
	The average number of finished orders of the members in a day
	The average weekly income of the members

Table 2: Mutual information of features.

Rank	Feature	MI
1	The similarity of the hometowns of the leaders and members	0.607175279
2	The similarity of the ages of the leaders and members	0.569887252
3	Age of the members	0.553651225
4	Hometowns of the members	0.354052674
5	Hometowns of the leaders	0.062810456
6	Age of the leaders	0.024951617
7	The average weekly income of the members	0.024767756
8	The average online time of the members in a day	0.024659155
9	The average number of finished orders of the members in a day	0.017719414
10	The average online time of the leaders in a day	0.013553273
11	The average weekly income of the leaders	0.009437473
12	The average number of finished orders of the leaders in a day	0.001606935
13	Gender of the leaders	0.001203234
14	Gender of the members	0.001123594

We compare five models for acceptance probability estimation: Logistic Regression (LR), RandomForest, SVM, Gradient Boosted Decision Tree (GBDT) and XGBoost.

Results. Fig. 5 and Fig. 6 show the AUC and F1-Score of the five models using the three categories of features on acceptance probability estimation. XGBoost achieves the highest AUC and F1-Score and will be used for BL and RMBF hereafter.

6.2 Offline Evaluation on Team Formation

This part evaluates the performance using trace-driven simulation with data collected from DiDi Chuxing.

Setup. We collect the data from six team-based competitions organized by DiDi Chuxing, held in November, 2017 in Changshan and Shenzhen, two cities in China. Fig. 7 shows the statistics of the collected data sets. Specifically, it shows the number of drivers who registered as leaders and members in each competitions and the ratios between the number of members and leaders. The team size of all the six competitions are seven. In others words, each feasible team requires one leader and six members. There are over 41,400 drivers who participated in the six competitions. On average, there are about 5,963 drivers who registered as member and 943 who registered as leader, and the average ratio is about 6.32.

Compared Algorithms. We use the best model (*i.e.* XGBoost) to obtain the acceptance probabilities for each round of recommendation in BL and RMBF. Since RMBF is a generic framework where different optimization methods can be applied, we compare the following optimization methods: Genetic Algorithm (GA) [5], Simulated Annealing (SA) [21] and Hill Climbing Algorithm (HC) [17]. The implement details of the above algorithms are as follows.

- **GA**. We first generate population (namely many recommendation matrices) randomly. Each matrix of the population is viewed as an individual. Then we perform crossover which chooses rows from two individuals randomly. Next, we perform mutation where we randomly choose a row of an individual and exchange a '0' and a '1'. Last, we keep the individuals with the highest fitness according to Eq. (1), Eq. (2) and Eq. (3) (selection step). Note that all the individuals will

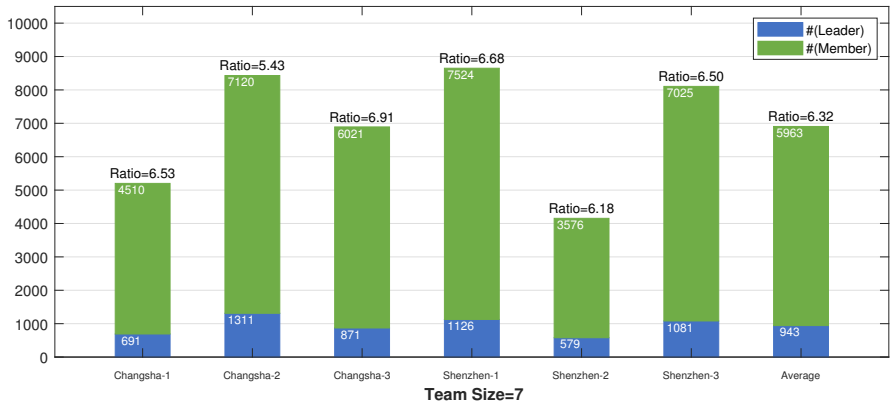
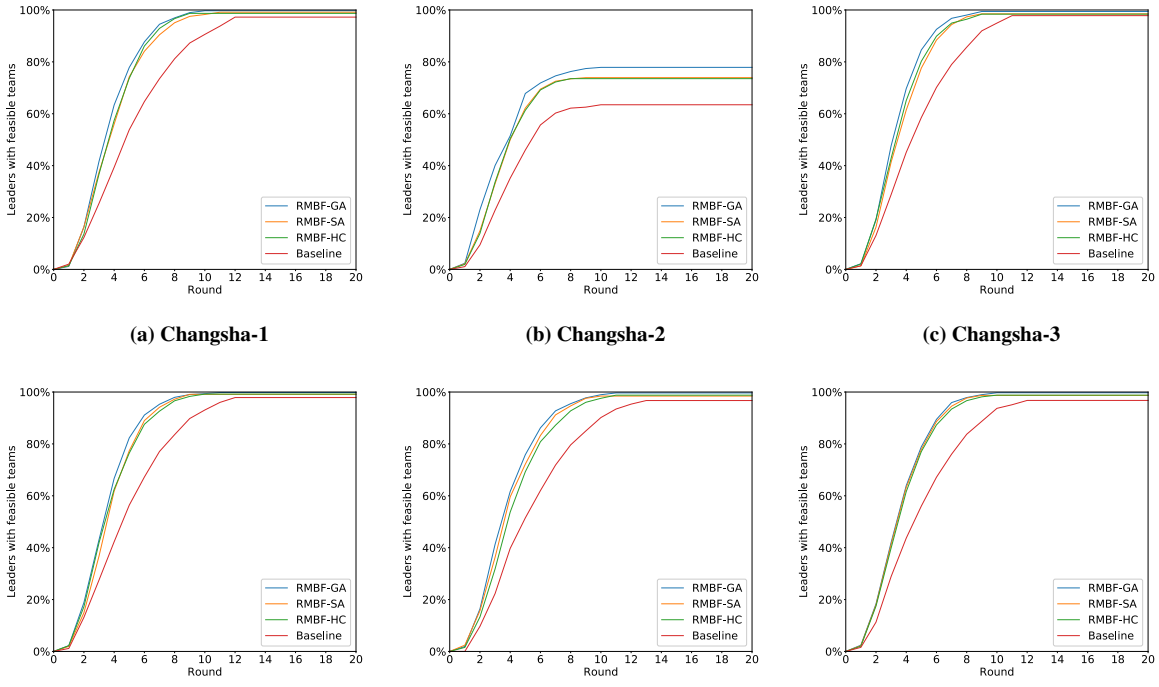


Figure 7: Statistics of dataset for offline evaluation.



(a) Changsha-1 **(b) Changsha-2** **(c) Changsha-3**

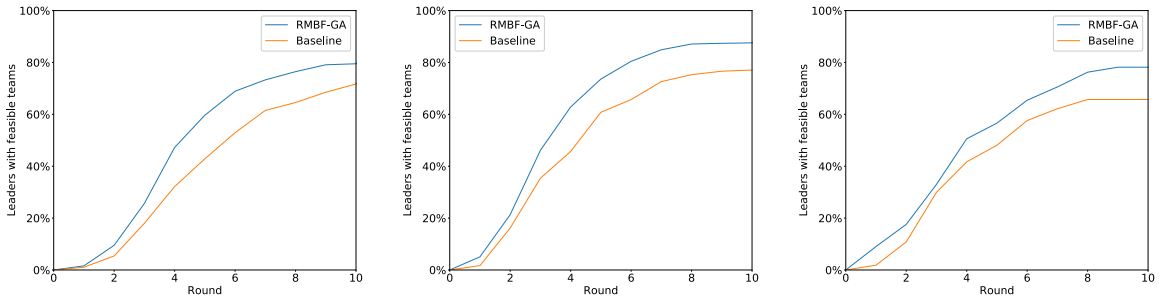
(d) Shenzhen-1 **(e) Shenzhen-2** **(f) Shenzhen-3**

Figure 8: Results of offline evaluation.

satisfy the constraint that at most lck_l members are recommended to leader l. The crossover, mutation and selection steps are performed iteratively until convergence or the iteration number exceeds the threshold.

- **HC**. We first generate the recommendation matrix randomly. Then with a probability we change '0' to '1' and '1' to '0' to get a new recommendation matrix. We use Eq. (1), Eq. (2) and Eq. (3) to evaluate the original matrix and the new one. The better one is remained and a new matrix based on the remained one is generated with a probability. The above steps are performed iteratively until convergence or the iteration number exceeds the threshold.
- **SA**. The implement of SA is similar with that of HC. The difference is that each time a new matrix is generated, with a probability we remain the matrix which is not the better one.

After obtaining the recommendations, we simulate the decisions of leaders to accept or reject the recommendations according to the acceptance probabilities P.



(a) Changsha-4 **(b) Shenzhen-4** **(c) Wuhan-1**

Figure 9: Results of online evaluation.

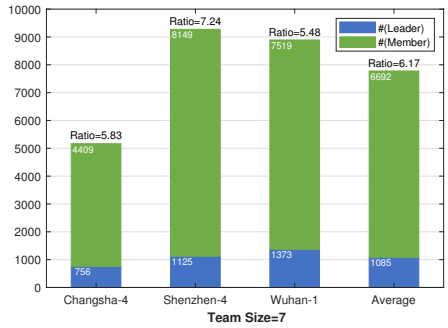


Figure 10: Statistics of dataset for online evaluation.

Results. Fig. 8 plots the percentage of leaders with feasible teams with the increase of recommendation rounds. We plots the results of the first 20 rounds because *(i)* all the algorithms saturate in the first 20 rounds; and *(ii)* in practice we may have very limited budget for recommendation *e.g.* only 10 rounds in the group-based competitions held by DiDi Chuxing (see Sec. 6.3) and thus 20 rounds is enough for simulation. We observe that except Fig. 8b, after 20 rounds all the algorithms can help form feasible teams for all the leaders. This is because during the simulation, the decisions of the leaders are based on the estimated probability. Accordingly, given sufficient chances (rounds) to recommend members, we can always form feasible teams for all the leaders. In Fig. 8b, all the algorithms fail to form teams for every leader. The reason is that the ratio between the members and leaders for the Changsha-3 dataset is lower than 6, while a feasible team needs 7 drivers in total (see Fig. 7). Therefore in Changsha-3 dataset, some drivers are bound to fail in team formation due to insufficient members.

Among the algorithms, the RMBF-based algorithms (namely RMBF-GA, RMBF-SA and RMBF-HC) all perform better than BL. Particularly, given the same rounds of recommendation, RMBF-GA is able to form up to 20% more teams than BL. This is reasonable as compared with BL, the RMBF-based algorithms can optimize recommendations from a global view. RMBF-GA is better than RMBF-SA and RMBF-HC, indicating that genetic algorithm based optimization method may be more suitable for our problem.

6.3 Online Evaluation on Team Formation

In this part, we integrate our methods into the group-based competition activities held by DiDi Chuxing platform to evaluate the performance on team formation in the wild.

Setup. We integrate BL and RMBF-GA into DiDi Chuxing as the member recommendation algorithms for three team-based competitions held in December, 2017 in Changshan, Shenzhen and Wuhan, three cities in China. We did not compare RMBF-HC and RMBF-SA due to the high cost to organize competitions for online evaluation. The statics of the three competitions are shown in Fig. 10. Similar with that in Fig. 7, Fig. 10 also shows the number of drivers who registered as leaders and members in each competitions and the ratios between the number of members and leaders. The team size of all the three competitions are seven. In others words, each feasible team requires one leader and six members. There are over 23,300 drivers who participated in the six competitions. On average, there are about 6,692 drivers who registered as member and 1,085 who registered as leader, and the average ratio is about 6.17. Upon receiving the recommendations by our methods, real-world leaders registered for the competitions decide whether to accept them or not.

Results. Fig. 9 shows the results of team formation. Note that limited by the cost of online evaluation, only ten rounds of recommendation are conducted. Overall RMBF-GA performs better than BL in all the three competitions, which is aligned with the trace-driven simulation results. After 10 rounds of recommendation, RMBF-GA manages to form 7.8% to 12.4% more teams than BL. The only exception is Shenzhen-4 (Fig. 9b), where RMBF-GA and BL performs similarly. This may be because in this dataset, the ratio between the drivers registered as members and leaders is high (7.24 in Fig. 10), which gives BL more choices when conducting its greedy strategy.

Comparing the results of online evaluation and offline simulation for the first 10 rounds, the results of online evaluation are worse by about 10%. Specifically, in the simulations, if the ratio of the members and the leaders is larger than 6, after 10 rounds almost all the leaders can form teams using RMBF-based algorithms (see Fig. 8 except Fig. 8b). In the online experiments, 88.8% leaders form teams (see Fig. 9b). This is because during the offline simulation we simulate whether the leaders will accept the recommended members according to the acceptance probability matrix P, while in the online

evaluation the response of the leaders may differ from P. This result indicates that it will increase the number of teams formed by further improving the accuracy of acceptance probability estimation.

6.4 Discussion

In the above subsections we focus mainly on the effectiveness of our recommendations methods for team formation. However, the ultimate goal is to stimulate the drivers through the group-based incentive which is via competition in our paper.

During the online evaluation conducted in Changsha, Shenzhen and Wuhan in December 2017, we also record other data generated the competition to evaluate the effectiveness of our group-based incentive via competition among different teams. We find that on average, after participating in the competition and forming teams, the service time, number of finished orders and income of drivers have improved by 33.8%, 34.4% and 27.4%, respectively. This result indicates that the group-based incentives are promising and can be a constructive supplement for other incentives at the individual level.

7 CONCLUSION

In this paper, we propose a Recommendation-based Team Formation (RTF) problem to promote participation in on-demand taxi-calling platforms. The aim is to form as many teams as possible while accounting for the choices of drivers. We prove that the RTF problem is NP-hard. To solve the RTF problem, we first propose a greedy-based recommendation strategy. Then we design a Recommendation-Matrix-Based Framework (RMBF). It first estimates the acceptance probability of recommendations and then derives a recommendation matrix to maximize the number of formed teams from a global view. We evaluate the performance of the proposed solutions via trace-driven simulations using real data covering over 64,000 drivers as well as real-world online evaluations on a large on-demand taxi-calling platform. Experimental results show that RMBF can form up to 20% and 12.4% teams in trace-driven simulations and online evaluations than the greedy-based strategy. We also find that after participating in the competition and forming teams, the service time, number of finished orders and income of drivers have improved obliviously, indicating the effectiveness of group-based incentive. We envision our work as a generic and effective building block to design various group-based incentive mechanisms for current and future on-demand transportation services.

8 ACKNOWLEDGMENTS

We are grateful to anonymous reviewers for their constructive comments. Yongxin Tong's work is partially supported by the National Science Foundation of China (NSFC) under Grant No. 61822201, U1811463 and 71531001, Science and Technology Major Project of Beijing under Grant No. Z171100005117001 and Didi Gaia Collborative Research Funds for Young Scholars. Yongxin Tong is the corresponding author of this work.

REFERENCES

[1] Wei Ai, Roy Chen, Yan Chen, Qiaozhu Mei, and Webb Phillips. 2016. Recommending Teams Promotes Prosocial Lending in Online Microfinance. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)* 113, 52 (2016), 14944–14948.

[2] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. 2012. Online Team Formation in Social Networks. In *The 21st International Conference on World Wide Web (WWW)*. ACM, 839–848.

[3] Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender Systems Survey. *Knowledge-based Systems* 46 (2013), 109–132.

[4] Gary Bornstein, Uri Gneezy, and Rosmarie Nagel. 2002. The Effect of Intergroup Competition on Group Coordination: An experimental study. *Games and Economic Behavior* 41, 1 (2002), 1–25.

[5] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[6] Oluwaseyi Feyisetan and Elena Simperl. 2017. Social Incentives in Paid Collaborative Crowdsourcing. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 6 (2017), 73.

[7] Jonathan V Hall and Alan B Krueger. 2018. An Analysis of the Labor Market for Uber's Driver-Partners in the United States. *ILR Review* 71, 3 (2018), 705–732.

[8] Elad Hazan, Shmuel Safra, and Oded Schwartz. 2006. On the Complexity of Approximating k-Set Packing. *Computational Complexity* 15, 1 (2006), 20–39.

[9] Shang H Hsu, Ming-Hui Wen, Hsin-Chieh Lin, Chun-Chia Lee, and Chia-Hoang Lee. 2007. AIMED- A Personalized TV Recommendation System. In *Proceedings of the 5th European Conference on Interactive TV and Video (EuroITV)*. Springer, 166–174.

[10] Mehdi Kargar and Aijun An. 2011. Discovering Top-k Teams of Experts with/without a Leader in Social Networks. In *The 20th ACM International Conference on Information and Knowledge Management (CIKM)*. ACM, 985–994.

[11] Richard M Karp. 1972. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*. Springer, 85–103.

[12] Jintao Ke, Hai Yang, Xinwei Li, Hai Wang, and Jieping Ye. 2019. Pricing and Matching Frictions in On-demand Ride-Splitting Markets. *Available at SSRN 3357362* (2019).

[13] Theodoros Lappas, Kun Liu, and Evimaria Terzi. 2009. Finding a Team of Experts in Social Networks. In *The 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 467–476.

[14] Anirban Majumder, Samik Datta, and KVM Naidu. 2012. Capacitated Team Formation Problem on Social Networks. In *The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 1005–1013.

[15] Frederick P Morgeson, D Scott DeRue, and Elizabeth P Karam. 2010. Leadership in Teams: A Functional Approach to Understanding Leadership Structures and Processes. *Journal of Management* 36, 1 (2010), 5–39.

[16] Michael J Pazzani and Daniel Billsus. 2007. Content-based Recommendation Systems. In *The Adaptive Web*. Springer, 325–341.

[17] Stuart J Russell and Peter Norvig. 2016. *Artificial Intelligence: A Modern Approach*. Prentice Hall.

[18] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative Filtering Recommender Systems. In *The adaptive web*. Springer, 291–324.

[19] Yongxin Tong, Libin Wang, Zimu Zhou, Lei Chen, Bowen Du, and Jieping Ye. 2018. Dynamic Pricing in Spatial Crowdsourcing: A Matching-Based Approach. In *Proceedings of the 2018 International Conference on Management of Data (SIGMOD 2018)*. 773–788.

[20] Shari Trewin. 2000. Knowledge-based Recommender Systems. *Encyclopedia of Library and Information Science* 69, Supplement 32 (2000), 180.

[21] Peter JM Van Laarhoven and Emile HL Aarts. 1987. Simulated Annealing. In *Simulated Annealing: Theory and Applications*. Springer, 7–15.

[22] Niko Vegt, Valentijn Visch, Huib de Ridder, and Arnold Vermeeren. 2015. Designing Gamification to Guide Competitive and Cooperative Behavior in Teamwork. In *Gamification in Education and Business*. Springer, 513–533.

[23] Gui-Rong Xue, Chenxi Lin, Qiang Yang, WenSi Xi, Hua-Jun Zeng, Yong Yu, and Zheng Chen. 2005. Scalable Collaborative Filtering Using Cluster-based Smoothing. In *The 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 114–121.

[24] Hai Yang, Chaoyi Shao, Hai Wang, and Jieping Ye. 2018. Integrated Reward Scheme and Surge Pricing in a Ridesourcing Market. *Available at SSRN 3198081*.

[25] Kai Yu, Volker Tresp, and Shipeng Yu. 2004. A Nonparametric Hierarchical Bayesian Framework for Information Filtering. In *The 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. ACM, 353–360.

[26] Zui Zhang, Hua Lin, Kun Liu, Dianshuang Wu, Guangquan Zhang, and Jie Lu. 2013. A Hybrid Fuzzy-based Personalized Recommender System for Telecom Products/Services. *Information Sciences* 235 (2013), 117–129.

[27] Libin Zheng, Lei Chen, and Jieping Ye. 2018. Order Dispatch in Price-aware Ridesharing. In *Proceedings of the VLDB Endowment (VLDB)*. 853–865.