Preference-Aware Task Assignment in On-Demand Taxi Dispatching: An Online Stable Matching Approach

Boming Zhao,[1] Pan Xu,[2] Yexuan Shi,[1] Yongxin Tong,[1] Zimu Zhou,[3] Yuxiang Zeng[4]
[1]BDBC, SKLSDE Lab, Beihang University, China
[2]University of Maryland, College Park, USA
[3]ETH Zurich, Zurich, Switzerland
[4]The Hong Kong University of Science and Technology, Hong Kong SAR, China
[1]{tongzhbo,skyxuan,yxtong}@buaa.edu.cn, [2]panxu@cs.umd.edu, [3]zzhou@tik.ee.ethz.ch, [4]yzengal@cse.ust.hk

Abstract

A central issue in on-demand taxi dispatching platforms is task assignment, which designs matching policies among dynamically arrived drivers (workers) and passengers (tasks). Previous matching policies maximize the profit of the platform without considering the preferences of workers and tasks (*e.g.,* workers may prefer high-rewarding tasks while tasks may prefer nearby workers). Such ignorance of preferences impairs user experience and will decrease the profit of the platform in the long run. To address this problem, we propose *preference-aware* task assignment using online stable matching. Specifically, we define a new model, *Online Stable Matching under Known Identical Independent Distributions* (OSM-KIID). It not only maximizes the expected total profits (OBJ-1), but also tries to satisfy the preferences among workers and tasks by minimizing the expected total number of blocking pairs (OBJ-2). The model also features a practical arrival assumption validated on real-world dataset. Furthermore, we present a linear program based online algorithm LP-ALG, which achieves an online ratio of at least $1-1/e$ on OBJ-1 and has at most $0.6\cdot|E|$ blocking pairs expectedly, where $|E|$ is the total number of edges in the compatible graph. We also show that a natural Greedy can have an arbitrarily bad performance on OBJ-1 while maintaining around $0.5\cdot|E|$ blocking pairs. Evaluations on both synthetic and real datasets confirm our theoretical analysis and demonstrate that LP-ALG strictly dominates all the baselines on both objectives when tasks notably outnumber workers.

1 Introduction

On-demand taxi dispatching has gained global popularity as an economic, efficient and sustainable alternative to urban transportation (Zhang et al. 2017; Dickerson et al. 2018a; Tong et al. 2016a; 2016b; Xu et al. 2018). One essential functionality of on-demand taxi dispatching platforms such as Uber and DiDi is to assign tasks (passengers) to workers (drivers) dynamically. Task assignment in on-demand taxi dispatching is commonly modeled as online bipartite matching. Each worker/task has a specific location attribute and is regarded as a vertex in the bipartite graph. Tasks arrive sequentially at random, and have to be either rejected or matched with an available worker shortly upon arrival. A worker matched to a task needs to finish a trip predetermined by the task. The platform earns profit from the finished tasks and aims to design matching policies that optimize certain objectives, *e.g.,* maximizing the expected total profit (Tong et al. 2016b; Zhang et al. 2017).

To continently make profits, it is important for taxi dispatching platforms to account for human factors such as preferences of workers and tasks. Prior work has integrated the preferences of *either* workers *or* tasks into the optimization objectives. For example, some researchers propose to minimize the sum of distances between the origin of each task and the matched worker over all matches (Bei and Zhang 2018; Tong et al. 2016a; Bansal et al. 2014). This way the overall waiting time of all passengers is reduced. Others maximize the total utility obtained through all successful matches, where the utility is defined to depict the workers' preference on payment (Tong et al. 2016b; Dickerson et al. 2018a). Despite these pioneer studies, the preference of only *one side* (workers *or* tasks) is considered. We argue that the matching policies should reflect the preferences of *both sides* (workers *and* tasks), which we will illustrate via the following example.

Image during the rush hours of Monday, Alice requested a taxi on Uber to take her from home to office for a short-distance trip. At the same time, Bob appeared on Uber as driver and he happened to be close to Alice's home. To minimize the waiting time of Alice, Uber should match Alice and Bob. However, this might hurt Bob's interest. This is because during rush hours passengers far outnumber drivers. Thus Bob preferred to wait for requests of long-distance rides to earn more profit. A question arises whether Uber should reject Alice or assign her to Bob. This is a common example in on-demand taxi dispatching platforms, where the preferences of workers and tasks may differ or even conflict with each other. That is, a passenger may prefer to be picked up immediately by a driver nearby while the driver may prefer to wait for long-distance rides. *How can we design matching policies to reconcile the preferences of both the workers and tasks such that they are satisfied to a best degree?*

To assign tasks such that the preferences of both workers and tasks are considered while the profit of the platform is maximized, we propose a new online stable matching model. Assume a compatible bipartite graph $G=(U,V,E)$, where U and V represent the respective set of offline workers and

online tasks. There is an edge $f = (u, v)$ if worker u is capable of performing task v (*i.e.,* the distance between them is below a given threshold). At the beginning of the online phase, all U are there and vertices from V come stochastically. Upon each arrival of an online task v, an *immediate and irrevocable* decision is required: either reject v, or match v with an available neighbor u. Each v is associated with a profit $w_v > 0$, which the platform can get if v is matched; each edge $f = (u, v)$ has a distance $\mathbf{d}(u, v) > 0$. For each u, we say u prefers v over v' if $w_v > w_{v'}$, *i.e.,* drivers prefer high-rewarding rides. Similarly, we say v prefers u over u' if $\mathbf{d}(u, v) < \mathbf{d}(u', v)$, *i.e.,* passengers prefer drivers nearby. For a given matching $\mathcal{M}$ over G, we define an edge f as a *blocking pair* or *blocking edge* iff (1) $f = (u, v) \in E, f \notin \mathcal{M}$ and (2) u prefers v over $u_{\mathcal{M}}$ and v prefers u over $v_{\mathcal{M}}$, where $u_{\mathcal{M}}$ and $v_{\mathcal{M}}$ are the respective vertices matched to u and v in $\mathcal{M}$[1]. Our goal is to design an online matching policy such that the following two objectives are optimized. Let $\mathcal{M}$ be the (random) matching obtained.

- Maximization of the expected total profit over all completed tasks (**OBJ-1**): $\max \mathbb{E}[\sum_{v \in V_{\mathcal{M}}} w_v]$, where $V_{\mathcal{M}}$ is the set of tasks matched in $\mathcal{M}$.
- Minimization of the expected number of blocking pairs (**OBJ-2**): $\min \mathbb{E}[\mathrm{BP}(\mathcal{M})]$, where $\mathrm{BP}(\mathcal{M})$ denotes the number of blocking pairs in $\mathcal{M}$.

Note that the number of blocking pairs quantifies the degree of dissatisfaction about their preferences from both workers and tasks. A matching policy that yields a smaller number of blocking pairs addresses the preferences among workers and tasks better. We also consider a more practical arrival assumption, called *known identical independent distributions* (KIID), where each round online tasks arrive following an identical independent distribution, which is assumed *known* to the algorithm beforehand. This is mainly due to the fact that we can often learn the arrival patterns of online tasks via mining massive historical data (Yao et al. 2018; Li et al. 2018; Wang, Fu, and Ye 2018). KIID has already been used to capture the arrival pattern of online tasks in various crowdsourcing applications (Dickerson et al. 2018b; Singer and Mittal 2013; Singla and Krause 2013). We call our new model *Online Stable Matching under Known Identical Independent distributions* (OSM-KIID).

Competitive Ratio. Competitive ratio is a commonly-used metric to evaluate the performance of online algorithms. Consider an online maximization problem for example. Let $\mathrm{ALG}(\mathcal{I}) = \mathbb{E}_{I \sim \mathcal{I}}[\mathrm{ALG}(I)]$ denote the expected performance of ALG on an input $\mathcal{I}$, where the expectation is taken over the random arrival sequence I. Let $\mathrm{OPT}(\mathcal{I}) = \mathbb{E}[\mathrm{OPT}(I)]$ denote the expected *offline optimal*, where $\mathrm{OPT}(I)$ refers to the optimal value after we observe the full arrival sequence I. Then, competitive ratio is defined as $\min_{\mathcal{I}} \frac{\mathrm{ALG}(\mathcal{I})}{\mathrm{OPT}(\mathcal{I})}$. It is a common technique to use an LP to upper bound the $\mathrm{OPT}(\mathcal{I})$ (called the benchmark LP) and hence get a valid lower bound on the target competitive ratio. In our paper, we conduct online competitive ratio analysis only on the first objective. Specifically, for a given (random) offline instance I, offline optimal $\mathrm{OPT}(I)$ is defined as the maximum profit (OBJ-1) among all possible *stable* matchings on I.

1.1 Contributions

Our contributions can be summarized in three aspects.

First, we propose a new online stable matching model under KIID (OSM-KIID) to address the preference-aware task assignment problem in on-demand taxi dispatching applications. OSM-KIID distinguishes itself from existing online stable marriage problems as follows.

- Our model considers two objectives, *i.e.,* maximizing the total profit and minimizing the number of blocking pairs. Prior studies only optimize the latter one (Khuller, Mitchell, and Vazirani 1994; Lee 1999; Miyazaki 2014).
- Our model allows partial preference list with ties for each worker/task and admits a practical arrival assumption, *i.e.,* KIID, which is supported by the real dataset.

Second, we present elegant theoretical analysis for our model. We first construct a linear program (LP (1)), which is proved a valid upper bound for the expected maximum profit on the offline optimal *stable* matching. Then we propose an effective LP-based online algorithm LP-ALG with provable performances on both objectives. Let $|E|$ be the number of edges in the compatible graph.

Theorem 1. LP-ALG *achieves an online ratio at least* $1 - 1/e$*, while the expected number of blocking pairs is at most* $|E| * \frac{(e^2+e-1)^2}{4(e-1)e^3} \sim 0.6 * |E|$.

Note that the online ratio is evaluated on OBJ-1 against offline optimal stable matching. We also examine carefully a natural heuristic, Greedy, which simply assigns each online arriving task to a nearest available worker (breaking ties in an arbitrary way). Unfortunately, we identify a class of instances on which Greedy can have an arbitrarily bad performance with respect to OBJ-1.

Theorem 2. *For any* $\epsilon > 0$*, there exists an instance such that Greedy achieves an online ratio at most* ϵ *with an expected number of blocking pairs of at least* $0.5*|E|+o(|E|)$.

Comparing the above two theorems, LP-ALG can significantly beat Greedy in terms of OBJ-1 in the worst case, while lose slightly on OBJ-2. Additionally, we show that the number of blocking pairs in LP-ALG is at most 8 times of that in the optimal non-adaptive[2] algorithm which matches LP-ALG on OBJ-1. We prove that by characterizing a Pareto line for the optimal non-adaptive online algorithms achieving a constant online ratio for OBJ-1.

Third, we test our model and algorithm on a real dataset collected from a large on-demand taxi dispatching platform. We present intensive analysis of the real dataset, which

[1]Here assume in the case when u is not matched in $\mathcal{M}$, $u_{\mathcal{M}}$ is a dummy node which has a profit of 0. Similarly when v is not matched in $\mathcal{M}$, $v_{\mathcal{M}}$ is a dummy node such that $\mathbf{d}(v_{\mathcal{M}}, v) = \infty$.

[2]An algorithm is called *non-adaptive* if its strategy does not respond to the random outcomes observed during previous rounds. Our LP-based LP-ALG belongs to the class of non-adaptive.

validates our KIID arrival assumption. Also, we propose several natural baselines, such as Greedy and Greedy-RT (threshold-based online policies), and test our LP-based online algorithm against them on both synthetic and real datasets. Our experimental results confirm our theoretical analysis, and show our LP-based algorithm can strictly dominate all those baselines (on both objectives) when online taxi requests outnumber taxi drivers *e.g.,* during rush hours.

1.2 Other Related Work

Our online stable matching can be viewed as an online version of the stable marriage problem, which has a combined flavor from online matching (*i.e.,* maximization of the total profits over all matches) and stable marriage (*i.e.,* minimization the number of blocking pairs). We survey related work along these two lines as follows.

Online Matching. Online (bipartite) matching problems are primarily motivated by Internet advertising. In the basic version, we are given a bipartite graph $G = (U, V, E)$ where U and V represent the respective set of offline advertisers and online keywords or impressions. Upon each arrival of a keyword v, a central clearinghouse must make an instant and irrevocable decision to either reject v or assign v to one of its neighbor u and obtain a profit w_e for the match $e = (u, v)$. The goal is to design an efficient online matching algorithm such that the expected total weight (profit) of all matches is maximized. Following the seminal work of (Karp, Vazirani, and Vazirani 1990), there has been a large body of research on related variants (Mehta 2013) and other applications like crowdsourcing (Zeng et al. 2018). Particularly, online matching has been widely applied in on-demand taxi dispatching for online task assignment to maximize the number of matches (Zhang et al. 2017; Tong et al. 2017b; Kazemi and Shahabi 2012).

Stable Marriage. Ever since its introduction in the seminal 1962 paper (Gale and Shapley 1962), the stable marriage problem has been used in many applications, including resource allocation (Lee 1999), hospital-doctor matching markets (Deng, Panigrahi, and Waggoner 2017), SDN controller assignment (Wang et al. 2016), crowdsourcing (Xia and Muthukrishnan 2017) etc. We refer readers to (Gusfield and Irving 1989; Manlove 2013) for a more comprehensive survey. Relevant to our settings is online stable marriage (Khuller, Mitchell, and Vazirani 1994; Lee 1999; Miyazaki 2014; Huzhang et al. 2017; Xia and Muthukrishnan 2017). Huzhang *et al.* try to find a stable arrangement for house-roommate matching to maximize the social welfare between roommates (Huzhang et al. 2017). Xia *et al.* adopt a global uniform pricing strategy to seek a stable matching in crowdsourcing (Xia and Muthukrishnan 2017).

2 Main Model

Assume a bipartite graph $G = (U, V, E)$ where U and V represent the set of workers (offline) and task types[3] (online) respectively. We have a finite time (known) horizon T and for each time $t \in [T] \doteq \{1, 2, \cdots, T\}$, a vertex v will be sampled (we also say v arrives) from a known distribution $\{p_v\}$ such that $\sum_{v \in V} p_v = 1$. Note that the sampling process is independent and identical across the T rounds. We assume integral arrival rates[4] for all tasks (*i.e.,* $p_v * T$ are all integers), and without loss of generality (WLOG) we further assume that $p_v = 1/T$ by creating $p_v * T$ copies for each v. In this case, we have $T = |V|$. Upon the arrival of each online task v, an *immediate and irrevocable* decision is required: either reject v, or match v with an available neighbor u. Each v is associated with a profit $w_v > 0$ which we can get if v is matched; each edge $f = (u, v)$ has a distance $\mathbf{d}(u, v) > 0$. For each u, we say u prefers v over v' if $w_v > w_{v'}$. Similarly, we say v prefers u over u' if $\mathbf{d}(u, v) < \mathbf{d}(u', v)$. For a given matching $\mathcal{M}$ over G, we define an edge f as *blocking pair* or *blocking edge* iff (1) $f = (u, v) \in E, f \notin \mathcal{M}$ and (2) u prefers v over $u_{\mathcal{M}}$ and v prefers u over $v_{\mathcal{M}}$, where $u_{\mathcal{M}}$ and $v_{\mathcal{M}}$ are the respective vertices matched to u and v in $\mathcal{M}$ ($u_{\mathcal{M}}$ and $v_{\mathcal{M}}$ are dummy nodes if u, v are not matched, where $u_{\mathcal{M}}$ has zero profit and $v_{\mathcal{M}}$ has infinity distance from v). Our goal is to design an online matching policy such that the two objectives on the resultant matching $\mathcal{M}$ is optimized (see the definition in the introduction).

In our paper, we conduct online competitive ratio analysis only on OBJ-1. Notably, our offline optimal is defined as the expected maximum profit (OBJ-1) on the offline optimal *stable* matching. For a general bipartite graph when each vertex can have an arbitrary preference list, a stable matching may even not exist (Iwama et al. 1999). However, in our case, there always exists a stable matching (*i.e.,* a matching with no blocking pair) in any given offline instance. That is mainly because the preference order of each worker toward its neighboring tasks inherits from the same order of their profits. See the lemma below and we defer the proof to our full paper[5].

Lemma 1. *For any given offline instance of* OSM-KIID*, there always exists a stable matching with no blocking pair.*

3 An LP-based Algorithm

Our benchmark LP is as follows. Recall that our offline optimal is defined as the expected profit (evaluated on the first objective) on the offline optimal stable matching. Note that each v may have multiple copies in an offline arrival sequence. For two edges sharing a vertex u, say $f_1 = (u, v_1), f_2 = (u, v_2)$, we say f_1 *dominates* f_2 iff $w_{v_1} > w_{v_2}$. Similarly for two edges sharing a vertex of v, say $f_1 = (u_1, v), f_2 = (u_2, v)$, we say f_1 dominates f_2 iff v strictly prefer u compared to u'. For each given edge $f = (u, v) \in E$, let S_f^L and S_f^R be the sets of edges incident to u and v respectively which is dominated by f. Consider any given offline optimal stable matching algorithm. For each edge $f = (u, v) \in E$, let x_f be the probability that

[3] A task type refers to a certain class of tasks who share certain attributes such as locations (details in Section 6).

[4] "Integral arrival rate" is a common assumption in Online Matching Models under KIID, see *e.g.,* (Haeupler, Mirrokni, and Zadimoghaddam 2011; Feldman et al. 2009). In this case, we can simply assume each online vertex is uniformly sampled each time.

[5] https://drive.google.com/file/d/1drIYxOsCKhQyrdnpioO7EwmMi0EYcTP8/view?usp=sharing

we match u with v (or one copy of v if multiple arrivals). For each u, let y_u be the probability that u is not matched, and for each v let y_v be expected number of unmatched v. For each worker u (task v), let E_u (E_v) be the set of neighboring edges incident to u (v). Consider the below LP program:

$$\max \sum_v w_v \sum_{f \in E_v} x_f \tag{1}$$

$$\text{s.t.} \sum_{f \in E_v} x_f + y_v = 1 \qquad \forall v \in V \tag{2}$$

$$\sum_{f \in E_u} x_f + y_u = 1 \qquad \forall u \in U \tag{3}$$

$$\sum_{f' \in S_f^L \cup S_f^R} x_{f'} + y_u + y_v + x_f \le 1 + \tfrac{1}{e} \quad \forall f = (u,v) \in E \tag{4}$$

$$0 \le x_f, y_u, y_v \le 1 \qquad \forall f \in E, u \in U, v \in V \tag{5}$$

Now we explain constraints in LP (1). Consider constraint (2) first. Notice that $\sum_{f \in E_v} x_f$ denotes the expected number of matched copies of v while y_v is the expected number of unmatched copies of v. Thus the sum of two should be equal to the expected arrivals of v, which is 1. For constraint (3), the first term $\sum_{f \in E_u} x_f$ is the probability that u is matched in the offline optimal while y_u is the probability that u is not matched. Thus the sum should be 1. The highlight point is constraint (4), which is inspired from the LP program for the offline stable matching (Teo and Sethuraman 1997). We add detailed justification in the proof of Lemma 2.

Lemma 2. *The optimal value to* LP (1) *is a valid upper bound for the expected profit on the offline optimal stable matching.*

Proof. We show that in any offline stable matching, $\{x_f, y_u, y_v\}$ should be feasible to LP (1). Hence LP (1) will be a valid upper bound for the expected offline optimal.

Constraints (2) (3) and (5) follow directly from the definitions of x_f, y_u and y_v. Here we focus on the justification of constraint (4). Consider a given $f = (u, v)$, a fixed (random) instance I and a stable matching $\mathcal{M}$ on I. Let X_v be the number of copies of v in I, Y_u indicate if u is unmatched, Y_v be the number of unmatched copies of v, Z^L indicate if some $f \in S_f^L$ is added in $\mathcal{M}$, and Z^R be the number of edges in S_f^R added into $\mathcal{M}$. Let X_f indicate if u gets matched with one copy of v. Split our discussion into the following cases:

- $X_v = 0$. v never comes. In this case $Y_v = Z_v = X_f = 0$ and thus $Z^L + Z^R + Y_u + Y_v + X_f = Z_u + Y_u \le 1$.
- $X_v = K \ge 1$, *i.e.,* v has K arrivals/copies in I. If $X_f = 1$, then $Y_u = Z^L = 0$ and $Z^R + Y_v \le K - 1$. Thus $Z^L + Z^R + Y_u + Y_v + X_f \le K$. If $X_f = 0$ and $Y_u + Z^L = 1$ (u either unmatched or matched to some v' worse than v), then $Y_v + Z^R = 0$ (none of copies v should be unmatched or matched to u' worse than u). In this case $Z^L + Z^R + Y_u + Y_v + X_f \le 1$. If $X_f = 0$ and $Y_u + Z^L = 0$ (u matched with some $v' \ne v$ but with $w_{v'} \ge w_v$), then $Z^R + Y_v \le K$. Summarizing all the above cases, we have $Z^L + Z^R + Y_u + Y_v + X_f \le K$.

Let $\Lambda = Z^L + Z^R + Y_u + Y_v + X_f$. The above analysis shows that if $X_v = 0$ which occurs with probability $1/e$, $\Lambda \le 1$; if $X_v \ge 1$, then $\Lambda \le X_v$. Therefore $\mathbb{E}[\Lambda] \le 1/e + \mathbb{E}[X_v] = 1 + 1/e$. Note that $\mathbb{E}[\Lambda] = \mathbb{E}[Z^L + Z^R + Y_u + Y_v + X_f] = \sum_{f' \in S_f^L \cup S_f^R} x_{f'} + y_u + y_v + x_f$. Thus constraint (4) is justified. □

Algorithm 1: A simple LP-based non-adaptive algorithm: LP-ALG

1 Suppose v arrives at time t.
2 Sample a neighbor u with probability $x^*_{u,v}$. If u is available, then assign v to u; otherwise, reject it.

Now we present an LP-based non-adaptive algorithm. Let $\{x^*_{u,v}\}$ be an optimal solution to LP (1).

Next, we prove the first main Theorem 1.

Proof. Focus on a given $f^* = (u, v) \in E$. Let X_{f^*} be the expected number of blocking edges contributed by f^* (note that when v has two copies, say v and v', (u, v) and (u, v') can be both blocking edges simultaneously). Let $\mathcal{M}$ be the random matching returned by LP-ALG. Suppose A_u is the event that u is either unmatched or u is matched with some v' such that $w_{v'} < w_v$ in $\mathcal{M}$. The complement event of A_u, denoted by $\neg A_u$, can be interpreted as u is matched with some v' such that $w_{v'} \ge w_v$. Let $\beta_u = \sum_{f \in E_u, f \notin S_{f^*}^L} x^*_f$ and $x^*_u = \sum_{f \in E_u} x^*_f$. Recall that $T = |V| = n$. Observe that

$$\Pr[\neg A_u] = \sum_{t=1}^{n} \frac{\beta_u}{n} \Big(1 - \frac{x^*_u}{n}\Big)^{t-1} \tag{6}$$

$$= \beta_u \frac{1 - \exp(-x^*_u)}{x^*_u} \ge \beta_u (1 - 1/e) \tag{7}$$

Inequality (7) is because $x^*_u = 1 - y^*_u \le 1$. Let $\alpha_u = \sum_{f \in S_{f^*}^L} x_f + y_u$. Thus $\beta_u = 1 - \alpha_u$. Consequently

$$\Pr[A_u] = 1 - \Pr[\neg A_u] \le 1 - (1 - \alpha_u)(1 - 1/e)$$

Assume A_u happens. Let $\mathcal{C}(v)$ be the set of copies of v (including v) which arrived. For each $v' \in \mathcal{C}(v)$ which is not matched in $\mathcal{M}$, we assume it is matched to a dummy node, say ω, which assumes to be worse (*i.e.,* farther) than any neighbor of v. For each $v' \in \mathcal{C}(v)$, if v' is matched to some u' such that u' is worse than u, then we claim (u, v') is a blocking edge. Thus X_{f^*} is equal to the number of edges in $f \in M$, denoted by H_v, such that f has one end in $\mathcal{C}(v)$ while the other end is some u' which is strictly worse than u (note that u' can be ω). Thus,

$$\mathbb{E}[X_{f^*}] = \Pr[A_u]\mathbb{E}[H_v | A_u] \le \Big(1 - (1 - \alpha_u)(1 - 1/e)\Big)\mathbb{E}[H_v | A_u]$$

Notice that conditioning on A_u occurs, each round an edge $f = (u', v)$ with u' being strictly worse than u arrives with probability at most $\Big(\sum_{f \in S_{f^*}^R} x_f + y_v\Big)/(n - \sum_{f \in E_u} x_f) \le \Big(\sum_{f \in S_{f^*}^R} x_f + y_v\Big)/(n-1) \doteq \alpha_v/(n-1)$. This implies that $\mathbb{E}[H_v | A_u] \le \alpha_v \frac{n}{n-1} \sim \alpha_v$. Therefore,

$$\mathbb{E}[X_{f^*}] \le \Big(1 - (1 - \alpha_u)(1 - 1/e)\Big)\alpha_v \tag{8}$$

From Constraints in LP (1), we see that (1) $\alpha_u = \sum_{f \in S^L_{f^*}} x_f + y_u \le 1$ due to Constraint (3); (2) $\alpha_v = \sum_{f \in S^R_{f^*}} x_f + y_v \le 1$ due to Constraint (2); (3) $\alpha_u + \alpha_v \le 1 + 1/e$ due to Constraint (4). Subject to all these three constraints, the right hand term in Equation (8) will get the maximum value of $\frac{(e^2+e-1)^2}{4(e-1)e^3} \sim 0.6$. By linearity of expectation, we get our claim for the expected number of blocking edges.

As for the online ratio, observe that for each given edge $f^* = (u, v) \in E$, it will be added into the online matching in LP-ALG with probability at least $\frac{x^*_{f^*}}{n} \sum_{t=1}^{n} \left(1 - \frac{x^*_u}{n}\right) \ge x^*_{f^*}(1 - 1/e)$. By linearity of expectation, we get our claim for the online ratio. □

4 A Natural Baseline: Greedy

Algorithm 2: A Natural Baseline: Greedy

1 Suppose v arrives at time t.
2 Reject v if all its neighbors are matched; otherwise assign v to one nearest available neighbor u (breaking ties in an arbitrary way).

Now we start to prove Theorem 2.

Proof. Consider a star graph where $U = \{u\}$, $V = \{v_j | j \in [n]\}$ and $T = n \ge 4/\epsilon$. Suppose we use j to denote v_j and edge $f_j = (u, v_j)$ when the context is clear. Suppose $w_j \doteq w_{v_j}$ and let $w_1 = 1$ and $w_j = \epsilon/(j * n)$ for all $2 \le j \le n$.

Let A and B are the respective profit and expected number of blocking edges in Greedy. Observe that $A = \frac{\sum_j w_j}{n} \le \frac{1+\epsilon}{n}$. Observe that whenever v_1 comes at least once, offline optimal stable matching will include (u, v_1) and thus we claim that offline optimal stable matching has an expected profit OFF-OPT $\ge (1 - 1/e)$. Thus the final online ratio is at most $A/$ OFF-OPT $\le \frac{1+\epsilon}{n(1-1/e)} \le \epsilon$.

Now we compute the value B. Focus on the first round $t = 1$. If v_j comes with $j \ge 2$, then B will be equal to the expected number of arrivals of $\{v_\ell | \ell < j\}$ during the next $n - 1$ rounds. Thus $B = \sum_{j=2}^{n} \frac{1}{n} \frac{(j-1)(n-1)}{n} = \frac{(n-1)^2}{2n}$. Since $|E| = n$, we get our claim. □

5 Hardness Results

The model in our paper has two objectives which complicate the hardness analysis. To simplify it, we focus only on those non-adaptive algorithms which achieves a constant competitive ratio on OBJ-1. Consider a two-dimensional line $\mathcal{P}$ where (x, y) axis represent respectively the online competitive ratio and the expected number of blocking pairs. We say an online algorithm ALG can never beat the line $\mathcal{P}$ (called Pareto line), if it is impossible that ALG can achieve an online ratio $x \in [0, 1]$ and an expected number of blocking pairs y such that (x, y) falls strictly below the line $\mathcal{P}$. In other words, any point $(x, y) \in \mathcal{P}$ can be viewed as an upper bound of the optimal performance among all possible non-adaptive algorithms: no non-adaptive algorithm can achieve a performance, which is strictly better than (x, y) on one objective while at least as good as (x, y) on the other.

We characterize a non-adaptive algorithm as $\{\mathbf{x}_v | v \in V\}$ where each $\mathbf{x}_v \in [0, 1]^{N_v}$ (N_v is the size of $\mathcal{N}_v$, the set of neighbors of v) such that $\sum_{u \in \mathcal{N}_v} x_{v,u} \le 1$. A non-adaptive algorithm NADAP parameterized with $\{\mathbf{x}_v\}$ will sample a neighbor $u \in \mathcal{N}_v$ with probability $x_{u,v}$, and assign v to u if it is available. Obviously it includes our LP-based algorithm as a special case.

Theorem 3. *No non-adaptive algorithm achieveing a constant competitive ratio could beat the Pareto line* $\left(\frac{1}{1-1/e}\frac{1-\exp(-\Delta)}{\Delta}, |E| * \exp(-\Delta)\right)$ *with* $\Delta \ge 1$*, where the first and second elements are the respective online ratio and expected number of blocking pairs, and* $|E|$ *is the number of edges in the compatible graph.*

Consider a star graph where $U = \{u\}$, $V = \{v_j | j \in [n]\}$ and $T = n$. We use j to denote v_j and edge $f_j = (u, v_j)$ when the context is clear. Let $w_j \doteq w_{v_j}$ and $w_1 > w_2 > \cdots > w_n$. In our case each v has only one neighbor. Thus each non-adaptive algorithm can be simply captured by a single vector $\mathbf{y} \in [0, 1]^n$, where y_j indicates the probability that we match v_j with u when v_j comes and u is available. Observe that in any optimal non-adaptive online algorithm, we are sure $y_1 = 1$. This is because whenever v_1 comes and we match (u, v_1), we will get the largest possible profit of w_1 and the smallest number of blocking pairs, which is 0.

Consider a given non-adaptive algorithm NADAP(Δ), which is captured by $\mathbf{y}$ with $y_1 = 1$ and $\Delta = \sum_j y_j \ge 1$. Let CR($\Delta$) and BP($\Delta$) be the respective competitive ratio and the expected number of block pairs for NADAP(Δ).

Lemma 3.

$$\mathrm{CR}(\Delta) \le \frac{1}{1-1/e}\frac{1}{\Delta}\left(1 - (1 - \frac{\Delta}{n})^n\right)$$

Proof. Let us focus on the probability that $f_1 = (u, v_1)$ is added. Observe that

$$\Pr[f_1 \text{ is added}] = \sum_{t=1}^{n} \frac{1}{n}\left(1 - \frac{\Delta}{n}\right)^{t-1} = \frac{1}{\Delta}\left(1 - \left(1 - \frac{\Delta}{n}\right)^n\right)$$

Let OFF-OPT be the expected offline stable optimal on the star graph instance. Thus, we have that

$$\text{OFF-OPT} \ge (1 - 1/e)w_1 + (1/e)(1 - 1/e)w_2$$

Consider the extreme case when $w_1 \gg w_2$, *e.g.,* $w_1 = 1, w_2 = \epsilon$. Then we can simply focus on the profit of NADAP(Δ) by adding f_1 and ignore all the rest. The final ratio will be almost $\frac{\Pr[f_1 \text{ is added}]}{\text{OFF-OPT}}$. Thus we get our claim. □

From Lemma 3, we see that for any $\Delta = \omega(1)$ as $n \to \infty$ (*i.e.,* $\Delta \to \infty$ as $n \to \infty$), we have that $\mathrm{CR}(\Delta) = o(1)$. In other words, if we want CR(Δ) to get a constant online ratio, we have to require Δ is a constant as well.

Lemma 4.

$$\mathrm{BP}(\Delta) \ge \left(1 - \frac{\Delta}{n}\right)^n n$$

Proof. With probability $\left(1-\frac{\Delta}{n}\right)^n$, we will end up with u being not matched. In this case, the number of blocking pairs will be $T = n$. □

Summarizing the results in Lemmas 3 and 4 yields that of Theorem 3. From Theorem 1, our LP-ALG achieves a ratio of $1-1/e$ while maintains the number of blocking pairs at most $0.6 * |E|$. Applying Theorem 3, we conclude that for any non-adaptive algorithm achieving a ratio of at least $1-1/e$, it will have at least $|E|*\exp(-\Delta) \sim |E|*0.1$ blocking pairs in the worse case, where $\Delta \sim 2.23$ is the unique solution to the equation $(1-\exp(-\Delta))/\Delta = (1-1/e)^2$. Thus we claim that the number of blocking pairs in LP-ALG is at most 6 times of that in the optimal non-adaptive algorithm which matches LP-ALG on the first objective.

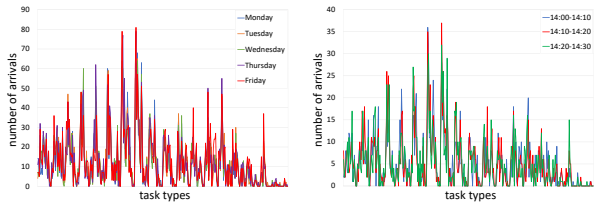
6 Experiment

6.1 Experimental Setup

Dataset. The real dataset is collected through the GAIA initiative[6], hosted by Didi Chuxing[7]. The dataset includes the trip records of passengers and the trajectory records of taxis in Chengdu, China for the month of November, 2016. For each trip record, we have its starting coordinates (*i.e.,* latitude and longitude), destination coordinates and the time at which the trip was initiated. For each taxi, we have several trajectory records, each of them contains the coordinates of this taxi during certain time intervals.

Inspired by (Tong et al. 2018), we experiment with the records during the peak-hour period 14:00-14:30, with longitude and latitude ranging from $(104.04, 104.13)$ and $(30.65, 30.73)$ respectively. We discretize the map of this range into a grid of 324 cells. Each cell has a unit size of 500 meters. For each cell $k \in \{1, 2, \ldots, 324\}$ and trip length $\ell \in \{1, 2, \ldots, 10\}$, we create a task type $v(k, \ell)$, which denotes the group of taxi requests with starting location from the cell k and trip length ℓ km. We then extract 4360 tasks whose lengths are between 1km to 10km from the records.[8] This way we create $|V| = 3240$ task types. Notice that the dataset includes only the successfully completed trips. To mimic the imbalance in the numbers of drivers and passengers, we uniform sample $|U| = m$ drivers (workers) from the whole records, where m ranges from 500 to 1500. For each driver u, we set its location as the place it appears for the first time in the record. For each task v of type (k, ℓ), we set its profit $w_v = \ell$ and set its starting location as the center of cell k. Each driver u and task v have a restricted circular range with radius of r_u and r_v such that we create an edge (u, v) iff $\mathbf{d}(u, v) \leq \min(r_u, r_v)$, where $\mathbf{d}(u, v)$ refers to Euclidean distance between u and v. We set $r_u = r_v = 500$ meters.

Justification of the KIID Assumption. First, we plot the average arrivals of each task type on each weekday (*i.e.,* Monday, Tuesday, $\cdots$, Friday) over the four weeks in November 2016. The result is shown in Figure 1a. The arrival distribution over different task types shares a nearly same pattern over the five days. This suggests the arrival distribution of different task types can be viewed as independent and identical over each weekday. Second, we plot the average arrivals of each task type on each 10-min time interval during the peak-hour period 14:00-14:30 over the 30 days of November, see Figure 1b. Again, it shows that the arrival distribution among different task types shares a similar pattern over all intervals. Similar results are observed in the heatmaps of arrivals among all task types in each case. More details can be found in our full paper[9].



(a) (b)

Figure 1: Arrival distribution of task types on the real dataset collected from DiDi Chuxing in Chengdu, China: (a) average arrivals of task types during 14:00-14:30 each weekday over the 4 weeks in Nov. 2016; (b) average arrivals of task types during each of the three 10-min intervals of 14:00-14:30 over the 30 days of Nov. 2016.

LP-based Algorithm and Baselines. We compare our LP-based non-adaptive algorithm LP-ALG with two baselines: Greedy as shown in Section 4 and Greedy-RT. The details are as follows. Let $\{x^*_{u,v}\}$ be an optimal solution to the benchmark LP (1). Assume some request v arrives.

- LP-ALG: Sample a neighbor u with probability $x^*_{u,v}$. If u is available, and then assign v to u; otherwise, reject it.
- Greedy (Khuller, Mitchell, and Vazirani 1994): Reject v if none of its neighbors is available; otherwise assign v to one nearest available neighbor u (breaking ties in an arbitrary way).
- Greedy-RT (Ting and Xiang 2015; Tong et al. 2016b): At the very beginning of online phase, we sample a threshold τ_u and τ_v uniformly from two respective pre-calculated intervals for each driver u and task v. During the online phase when some request v arrives, assign v to some available u if u and v satisfy each threshold, *i.e.,* $w_v > \tau_u$ and $\mathbf{d}(u, v) < \tau_v$ (breaking ties in an arbitrary way).

Among the above three algorithms, both Greedy and Greedy-RT are *adaptive* in the sense that their strategies respond to the outcome of previous strategies and arrivals.

Methodology. We use the first 20 days in November for training and propose five different approaches to learning the arrival distribution of different task types each round.

[6]http://gaia.didichuxing.com

[7]http://www.didichuxing.com

[8]The extracted tasks account for 90% of the total.

[9]https://drive.google.com/file/d/1drIYxOsCKhQyrdnpioO7EwmMi0EYcTP8/view?usp=sharing

They are Historical Average (use the average as the prediction), Linear Regression (use a linear regression model with the number of tasks during the 5 most recent periods), Neural Network (a neural network with 2 hidden layers and the features are the number of tasks during the 5 most recent periods and the type ID), Gradient Boosted Regression Tree (use non-parametric regression) and the state-of-the-art approach LinUOTD (Tong et al. 2017a). The results show that LinUOTD outperforms the other methods in the final prediction. Thus, we use α_v to represent the arrival rate in the predicted result of LinUOTD. We round each α_v to a near integral $n_v \in \{0, 1, 2, \cdots\}$, and remove all those v with $n_v = 0$. For each type v, we create n_v copies of type v and in this way we have $T = \sum_v n_v$ task types and each type will have a unit expected arrival over T rounds (*i.e.,* each $p_v = 1/T$).

On the real dataset, we test the instances with $|U| = m = \{500, 750, 1000, 1250, 1500\}$. For each instance, we run LP-ALG and the two baselines over the 10 remaining days and take the average as the final performance. The competitive ratio is taken as the ratio of the average profits obtained to the benchmark LP value. The blocking pair ratio is computed as the ratio of the average number of blocking pairs to the number of edges in the compatible graph.

On the synthetic dataset, we test the same set of instances. For each instance, however, we run the three algorithms each for 1000 independent trials. During each trial, we sample an online task uniformly each time. We take the average over those 1000 trials as the final performance.

The LP program is solved via the Glop Linear Solver on a PC with Intel(R) Core(TM) i7-7700HQ 2.80GHz processor and 16GB Memory. All the LP programs can be solved within four minutes.

6.2 Results

Figures 2 and 3 show the results on the real and synthetic dataset, respectively.

LP-based algorithm LP-ALG always achieves a smaller number of blocking pairs than the two baselines. This is mainly due to the fact that LP-ALG is guided by the offline optimal strategy which is restricted to a *stable* matching on each offline instance. The constraint (4) in LP (1) exactly captures this point. Thus, we can interpret that LP-ALG first gives priority to small number of blocking pairs and then tries to maximize OBJ-1. Note that Greedy always tries to assign an arrival task v to a nearest available worker in each step, which in some sense helps reduce the number of blocking pairs incident to v. However, Greedy can only achieve that goal locally while LP-ALG can make it in a globally-optimized way by solving a well-designed LP (*i.e.,* the benchmark LP). Observe that the blocking pair ratio achieved by LP-ALG is always below 0.1, which is far below 0.6 as analyzed in the worst case in Theorem 1.

As for the competitive ratio, LP-ALG first dominates the two baselines and then gets close to Greedy, as the number of drivers increases. Yet the gap between Greedy and LP-ALG is still notable when there are relatively sufficient workers. The ratio achieved by LP-ALG keeps slightly above 0.63, which is consistent with our prediction of $1 - 1/e$ as in Theorem 1. This suggests that our competitive ratio analysis for LP-ALG is nearly tight and the worst case is close to real scenarios. Comparably, the worst case where Greedy achieves a very bad ratio as shown in the proof of Theorem 2 seems far away from real scenarios.

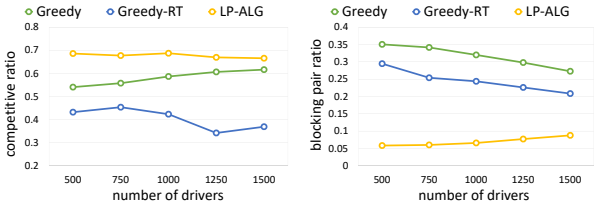


Figure 2: Experimental results on the real dataset.

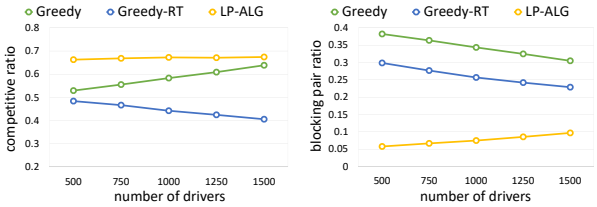


Figure 3: Experimental results on the synthetic dataset.

Figures 2 and 3 show that the advantages of high competitive ratio and small number of blocking pairs both diminish as the number of drivers increases. This is expected since the more (offline) drivers, the less need for a globally-optimized strategy like LP-ALG. Our results imply that our LP-based algorithm will be particularly effective during the peak hours when online tasks outnumber the drivers.

7 Conclusion

In the paper, we proposed an online stable matching model under KIID (OSM-KIID) to address the preference-aware task assignment problem in on-demand taxi dispatching applications. The model is featured by two objectives: maximizing the total profit and minimizing the overall dissatisfaction about preferences among workers and tasks. We constructed an LP, which proves to be a valid upper bound on the expected maximum profit on the offline optimal stable matching. We further propose an LP-based online algorithm LP-ALG, which achieves an online ratio of at least $1 - 1/e$ on the first objective and maintains at most $0.6 * |E|$ blocking pairs. Experimental results confirm our theoretical analysis on LP-ALG, and show that LP-ALG can beat natural baselines such as Greedy when tasks outnumber the drivers.

Acknowledgements

Yexuan Shi and Yongxin Tong's works are partially supported by National Grand Fundamental Research 973 Program of China under Grant 2015CB358700, the Science and Technology Major Project of Beijing under Grant No. Z171100005117001, National Science Foundation of China

(NSFC) under Grant No. 71531001 and Didi Gaia Collborative Research Funds for Young Scholars. Pan Xu's work is partially supported by NSF Awards CNS 1010789 and CCF 1422569. Pan Xu and Zimu Zhou are the corresponding authors in this paper.

References

Bansal, N.; Buchbinder, N.; Gupta, A.; and Naor, J. 2014. A randomized $o(log^2 k)$-competitive algorithm for metric bipartite matching. *Algorithmica* 68(2):390–403.

Bei, X., and Zhang, S. 2018. Algorithms for trip-vehicle assignment in ride-sharing. In *AAAI*, 3–9.

Deng, Y.; Panigrahi, D.; and Waggoner, B. 2017. The complexity of stable matchings under substitutable preferences. In *AAAI*, 480–486.

Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2018a. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *AAAI*, 1007–1014.

Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2018b. Assigning tasks to workers based on historical data: Online task assignment with two-sided arrivals. In *AAMAS*, 318–326.

Feldman, J.; Mehta, A.; Mirrokni, V. S.; and Muthukrishnan, S. 2009. Online stochastic matching: Beating 1-1/e. In *FOCS*, 117–126.

Gale, D., and Shapley, L. S. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69(1):9–15.

Gusfield, D., and Irving, R. W. 1989. *The Stable marriage problem - structure and algorithms*. Foundations of computing series. MIT Press.

Haeupler, B.; Mirrokni, V. S.; and Zadimoghaddam, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*, 170–181.

Huzhang, G.; Huang, X.; Zhang, S.; and Bei, X. 2017. Online roommate allocation problem. In *IJCAI*, 235–241.

Iwama, K.; Miyazaki, S.; Morita, Y.; and Manlove, D. 1999. Stable marriage with incomplete lists and ties. In *ICALP*, 443–452.

Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *STOC*, 352–358.

Kazemi, L., and Shahabi, C. 2012. Geocrowd: enabling query answering with spatial crowdsourcing. In *GIS*, 189–198.

Khuller, S.; Mitchell, S. G.; and Vazirani, V. V. 1994. On-line algorithms for weighted bipartite matching and stable marriages. *Theor. Comput. Sci.* 127(2):255–267.

Lee, H. 1999. Online stable matching as a means of allocating distributed resources. *Journal of systems architecture* 45(15):1345–1355.

Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; and Liu, Y. 2018. Multi-task representation learning for travel time estimation. In *KDD*, 1695–1704.

Manlove, D. F. 2013. *Algorithmics of Matching Under Preferences*, volume 2 of *Series on Theoretical Computer Science*.

Mehta, A. 2013. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science* 8(4):265–368.

Miyazaki, S. 2014. On the advice complexity of online bipartite matching and online stable marriage. *Information Processing Letters* 114(12):714–717.

Singer, Y., and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. In *WWW*, 1157–1166.

Singla, A., and Krause, A. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *WWW*, 1167–1178.

Teo, C.-P., and Sethuraman, J. 1997. LP based approach to optimal stable matchings. In *SODA*, 710–719.

Ting, H. F., and Xiang, X. 2015. Near optimal algorithms for online maximum edge-weighted b -matching and two-sided vertex-weighted b -matching. *Theor. Comput. Sci.* 607(P2):247–256.

Tong, Y.; She, J.; Ding, B.; Chen, L.; Wo, T.; and Xu, K. 2016a. Online minimum matching in real-time spatial data: experiments and analysis. *PVLDB* 9(12):1053–1064.

Tong, Y.; She, J.; Ding, B.; Wang, L.; and Chen, L. 2016b. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*, 49–60.

Tong, Y.; Chen, Y.; Zhou, Z.; Chen, L.; Wang, J.; Yang, Q.; Ye, J.; and Lv, W. 2017a. The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *KDD*, 1653–1662.

Tong, Y.; Wang, L.; Zhou, Z.; Ding, B.; Chen, L.; Ye, J.; and Xu, K. 2017b. Flexible online task assignment in real-time spatial data. *PVLDB* 10(11):1334–1345.

Tong, Y.; Zeng, Y.; Zhou, Z.; Chen, L.; Ye, J.; and Xu, K. 2018. A unified approach to route planning for shared mobility. *PVLDB* 11(11):1633–1646.

Wang, T.; Liu, F.; Guo, J.; and Xu, H. 2016. Dynamic SDN controller assignment in data center networks: Stable matching with transfers. In *INFOCOM*, 1–9.

Wang, Z.; Fu, K.; and Ye, J. 2018. Learning to estimate the travel time. In *KDD*, 858–866.

Xia, C., and Muthukrishnan, S. 2017. Revenue-maximizing stable pricing in online labor markets. In *HCOMP*, 216–225.

Xu, Z.; Li, Z.; Guan, Q.; Zhang, D.; Li, Q.; Nan, J.; Liu, C.; Bian, W.; and Ye, J. 2018. Large-scale order dispatch in on-demand ride-hailing platforms: A learning and planning approach. In *KDD*, 905–913.

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI*, 2588–2595.

Zeng, Y.; Tong, Y.; Chen, L.; and Zhou, Z. 2018. Latency-oriented task completion via spatial crowdsourcing. In *ICDE*, 317–328.

Zhang, L.; Hu, T.; Min, Y.; Wu, G.; Zhang, J.; Feng, P.; Gong, P.; and Ye, J. 2017. A taxi order dispatch model based on combinatorial optimization. In *KDD*, 2151–2159.