A Unified Approach to Online Matching with Conflict-Aware Constraints

Pan Xu[1], **Yexuan Shi**[2], **Hao Cheng**[2], **John Dickerson**[1], **Karthik Abinav Sankararaman**[1], **Aravind Srinivasan**[1], **Yongxin Tong**[2], **Leonidas Tsepenekas**[1]
[1]University of Maryland, College Park, USA
[2]BDBC, SKLSDE Lab and IRC, Beihang University, China
[1]{panxu,john,kabinav,srin,ltsepene}@cs.umd.edu, [2]{skyxuan,haocheng,yxtong}@buaa.edu.cn

Abstract

Online bipartite matching and allocation models are widely used to analyze and design markets such as Internet advertising, online labor, and crowdsourcing. Traditionally, vertices on one side of the market are fixed and known a priori, while vertices on the other side arrive online and are matched by a central agent to the offline side. The issue of possible *conflicts* among offline agents emerges in various real scenarios when we need to match each online agent with a set of offline agents. For example, in event-based social networks (*e.g.,* Meetup), offline events conflict for some users since they will be unable to attend mutually-distant events at proximate times; in advertising markets, two competing firms may prefer not to be shown to one user simultaneously; and in online recommendation systems (*e.g.,* Amazon Books), books of the same type "conflict" with each other in some sense due to the diversity requirement for each online buyer.

The conflict nature inherent among certain offline agents raises significant challenges in both modeling and online algorithm design. In this paper, we propose a unifying model, generalizing the conflict models proposed in (She *et al.*, TKDE 2016) and (Chen *et al.*, TKDE 16). Our model can capture not only a broad class of conflict constraints on the offline side (which is even allowed to be sensitive to *each* online agent), but also allows a general arrival pattern for the online side (which is allowed to change over the online phase). We propose an efficient linear programming (LP) based online algorithm and prove theoretically that it has nearly-optimal online performance. Additionally, we propose two LP-based heuristics and test them against two natural baselines on both real and synthetic datasets. Our LP-based heuristics experimentally dominate the baseline algorithms, aligning with our theoretical predictions and supporting our unified approach.

1 Introduction

Online bipartite matching problems are primarily motivated by the Internet advertising business. In the basic setting, we have a set of offline advertisers and online keywords (impressions). Each time a keyword v arrives online, a central clearinghouse (such as a search engine or other matchmaking platform) must make an instant and irrevocable decision to either reject v or assign v to an advertiser showing interest toward v: the clearinghouse will obtain a profit $w_{u,v}$ as a result. Each advertiser has a unit capacity (can be matched only once) and we aim to design an efficient online 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 (see also the survey by Mehta (2013)).

Apart from Internet advertising, online matching and its variants have wide applications in other domains such as crowdsourcing marketplaces (Assadi, Hsu, and Jabbari 2015; Ho and Vaughan 2012) (*e.g.,* Amazon Mechanical Turk, offline tasks *vs.* online workers), online recommendation systems (Ahmed, Dickerson, and Fuge 2017; Chen et al. 2016b; Sha, Wu, and Niu 2016; Qin and Zhu 2013) (*e.g.,* Amazon Books, offline products *vs.* online buyers) and ridesharing platforms (Tong et al. 2016a; Dickerson et al. 2018a) (*e.g.,* Uber and Lyft, offline drivers *vs.* online requests). In many real applications, we need to match each online agent with multiple offline agents simultaneously. A generic model is as follows: we have a set of offline agents U and online agents V; each time an online agent v arrives, we need to immediately and irrevocably decide if we should reject v or assign v to a set $S \subseteq \mathcal{N}_v$ ($\mathcal{N}_v$ refers to the set of offline agents interesting to v). A natural goal is to design an online matching policy such that the total weight among all matches is maximized subject to certain kinds of capacity constraints, *e.g.,* each u can be matched at most c_u times while each v can be matched with a set $|S| \leq c_v$, where c_u and c_v are given as input. In many real applications, there are inherent conflicts among offline agents by nature, which causes significant algorithmic challenges. Consider the following motivating examples.

Event-Participant Arrangement in Social Networks. Li et al. (2014) introduced an interesting event-participant matching problem arising in Event-Based Social Networks (EBSNs) such as Meetup and Plancast (Liu et al. 2012). In this case, the two sets are offline events and online users. Each time a user arrives, we need to match her to some events she shows interest in. Naturally, there are potential conflicts among a set of events S with respect to a user if she cannot attend all events in S on time due to travel/time constraints.

Online Recommendations in E-Commerce. Chen et al. (2016a) studied online recommendations in E-Commerce (*e.g.,* Amazon, Netflix) via online matching models where the two sets are the offline products/items (*e.g.,* books, music, movies) and online users/buyers. Each time an online user comes, the system needs to generate a set of recommendations she has potential interest in. Items naturally conflict with each other in the sense that we often need to avoid recommending multiple items of the same type to each buyer due to diversity concerns (Chen et al. 2016a).

Task Assignment in Spatial Crowdsourcing Markets. Tong *et al.* (2016b; 2016a; 2017) considered task assignment problems in spatial crowdsourcing markets where the two sides are the respective set of offline tasks and online workers. In this context, each worker has to travel to the location of a task to complete it and the traveling time dominates the total completion time (*e.g.,* online food delivery services such as Uber Eats). In this case, two offline tasks may conflict for a worker v if she cannot complete the two tasks before the respective deadlines.

Conflicting relations among offline agents can be very complicated, causing significant challenges in the modeling. Here are two examples. She et al. (2016) characterized the conflict among events in EBSNs as a collection C of conflicting pairs of events. They assume that no user can be assigned to any pair of conflicting events in C. Chen et al. (2016a) modeled the conflict in a similar but more soft way: each time an online agent can be assigned a set of offline agents S, where the number of conflicting pairs from S in C should be below a given threshold. Both papers are tightly related and address a common use case—inter-agent conflict—found in many applications; yet, as discussed below, both approaches cannot model the full complexity of many real-world settings.

Both approaches assume that the conflict nature among offline agents applies *uniformly* to all online agents. This assumption greatly limits the applications. Consider the scenario of event-participant arrangement in EBSNs for example. It is possible that two events conflict for some user v but not for v' due to the different schedules of v and v' or different relative distances from v and v' to the two events. The same issue exists in the spatial crowdsourcing markets as well. Another drawback of the state-of-the-art models (She et al. 2016; Chen et al. 2016a) is that they can only capture pairwise conflict relations: what if potential conflicts exist only among at least three offline agents? Consider the scenario of online recommendations and suppose we define a set of offline items as conflicting if it includes more than two items of a single type (thus, we see that no pair of conflicting offline agents exists). Motivated by the wide-ranging applications of those two recent conflict-aware models, in this paper, we propose a *unifying* model that can help overcome the above two limitations. Next, we briefly discuss our approach (1.1), our contributions (1.2), and additional related work (1.3).

1.1 Preliminaries & Brief Overview of our Model

Collection of Admissible Sets. We assume that for each online agent v, it has a specific collection $\mathcal{A}_v$ of admissible subsets of offline agents. Thus, each time v comes online, we need to make an instant and irrevocable decision, either to reject v or assign v a subset $S \in \mathcal{A}_v$. We assume $\mathcal{A}_v$ is given as part of input, which can be obtained by integrating all information from v with that of offline agents. For example, consider the event-participant arrangement. For each user v, $\mathcal{A}_v$ (the collection of feasible sets of events) can be computed after considering her capacity (c_v), the distance from v to $\mathcal{N}_v$ (the set of events interesting to v), schedules of v and events in $\mathcal{N}_v$, and the travel options available to v. In real applications, c_v captures the "patience" of the online user v toward the offline agents, *e.g.,* the number of recommendations to receive from the system and that of events to attend in EBSNs, which is typically very small.[1] The property of small c_v values greatly limits the size of $\mathcal{A}_v$; we will leverage this for model flexibility and algorithmic tractability.

Known Adversarial Distributions (KAD)**.** Assume that we have T online rounds and a set of online agent types V (both T and V are known as input). For each round $t \in [T]$ [2], an online agent v is sampled from V with probability $p_{v,t}$ such that $\sum_{v \in V} p_{v,t} \leq 1$ for all t. Note that the sampling distributions $\{p_{v,t}\}$ are independent and allowed to change over time. Current online matching models attacking conflict issues (She et al. 2016; Chen et al. 2016a) both assume adversarial arrival (*i.e.,* the full arrival sequence is completely unknown), which we believe is too conservative. In fact, in many real scenarios, we can predict the arrival patterns of online agents via exploiting historical data (Li et al. 2018; Wang, Fu, and Ye 2018; Yao et al. 2018; Rzeszotarski and Kittur 2011). Note that KAD is a generalization of another well-studied online arrival assumption, Known Identical Independent Distributions (KIID), where the arrival distributions are assumed the same throughout the online phase (Dickerson et al. 2018b; Singer and Mittal 2013; Singla and Krause 2013).

In addition to the above two features, we assume that each offline agent u has an integral capacity c_u, the maximum number of times u can be matched. This is consistent with most current literature. We call our new model *Online Matching with Conflict-Aware Constraints under Known Adversarial Distributions* (OM-CC-KAD).

Competitive Ratio. Let $\mathrm{ALG}(\mathcal{I}, \mathcal{D})$ denote the expected value obtained by an online algorithm ALG on an input $\mathcal{I}$ and arrival distribution $\mathcal{D}$. Let $\mathrm{OPT}(\mathcal{I})$ denote the expected *offline optimal*, which refers to the optimal solution when we are allowed to make choices after observing the entire sequence of online arrivals. Then, the competitive ratio is defined as $\min_{\mathcal{I},\mathcal{D}} \frac{\mathrm{ALG}(\mathcal{I},\mathcal{D})}{\mathrm{OPT}(\mathcal{I})}$. A common technique is to use a *benchmark* LP to upper bound $\mathrm{OPT}(\mathcal{I})$, and hence get a valid lower bound on the target competitive ratio.

1.2 Our Contributions

First, we propose a unifying model OM-CC-KAD, which can capture very general conflicting relations among offline agents (which can even be sensitive to each online agent) and also general arrival distributions of online agents (which is allowed to change throughout the online phase).

Second, we show how this model can be *cleanly* analyzed under a theoretical framework. We first construct a linear program (LP henceforth) LP (1) which we show is a valid

[1]Direct evidence can be seen in Table 1: the median number of events attended by users is 2 within three months.

[2]Throughout this paper, we use $[N]$ to denote the set $\{1, 2, \ldots, N\}$, for any integer N.

upper-bound on the expected offline optimal (note that the latter is hard to characterize). Next, we propose an efficient LP-based online algorithm that achieves a competitive ratio of at least $\frac{1}{e(\Delta+1)}$, where Δ is the maximum size of any admissible set. We give a *tight* online analysis and show further that it has a *nearly optimal* performance among all possible LP-based online algorithms by presenting relevant hardness results.

Theorem 1. LP (1) *is a valid benchmark for* OM-CC-KAD. *There exists an online algorithm based on* LP (1)*, which achieves an online competitive ratio of at least* $\frac{1}{\Delta+1}(1-\frac{1}{\Delta+1})^{\Delta} \geq \frac{1}{e(\Delta+1)}$. *The online competitive raito analysis is tight.*

Theorem 2. *No online algorithm can achieve a ratio better than* $\frac{1}{\Delta-1+1/\Delta}$ *for* OM-CC-KAD*, if using* LP (1) *as the benchmark.*

To prove Theorem 1, we need to *lower bound* $\Pr[\bigwedge_u \mathsf{SF}_u]$ for a certain family of *negatively correlated* events $\{\mathsf{SF}_u\}$ (an *upper bound* follows directly from the FKG inequality (Fortuin, Kasteleyn, and Ginibre 1971)). For the simple case of unit capacities, we decompose $(\bigwedge_u \mathsf{SF}_u)$ into an intersection of independent events and hence obtain an exact value for $\Pr[\wedge_u \mathsf{SF}_u]$. For the case of general integral capacities, we propose a shadow auxiliary arrival process which is equivalent to the original arrival process but much simpler to analyze. We then try to identify the worst-case scenario on this shadow process where $\Pr[\bigwedge_u \mathsf{SF}_u]$ is minimized. Note that our final ratio result has completely removed the dependency on the maximum weight over all possible online assignments, which appears in the online competitive-ratio results of both She et al. (2016) and Chen et al. (2016a). This is the exact theoretical evidence showing the advantage of shifting from adversarial arrivals to KAD by exploiting historical data—which can be done in many real-world settings.

Third, we propose two LP-based heuristics and test them against two natural baselines, Greedy and Uniform Sampling, on both real (Meetup, (Liu et al. 2012)) and synthetic data. Our experimental results confirm our theoretical predictions about the LP-based heuristics and also show that they can achieve not only significantly better average performances (*effectiveness*) than the two baselines but also maintain relative low variances (*robustness*).

1.3 Additional Related Work

Online matching models and related variants have been intensively studied during the last decade under different arrival assumptions, including adversarial arrivals, random arrival order and known distributions such as KIID. See the survey (Mehta 2013). The KAD arrival assumption, which is a generalization of KIID, has attracted an increasing number of theoretical and empirical investigations (Alaei, Hajiaghayi, and Liaghat 2012; Alaei, Hajiaghayi, and Liaghat 2013; Dickerson et al. 2018a; Xu et al. 2017).

There are several interesting works which have studied other kinds of constraints imposed on offline agents in different online matching models. Xu et al. (2017) introduced the multiple-budgeted assignment problem arising in crowdsourcing markets, where each assignment of a task to a worker will consume multiple offline resources. Wang and Wong (2016) introduced the Matroid Online Bipartite Matching model, where the set of all matched offline agents is required to be independent in a given matroid. Kell and Panigrahi (2016) considered Online Budgeted Allocation with General Budgets, where offline agents have multiple tiers of budget constraints forming a laminar structure. There are several interesting matching problems with constraints with applications in the AI and game theory community, which studied the stability and strategic issues, see, *e.g.,* (Kawase and Iwasaki 2018; Goto et al. 2016; Kurata et al. 2017; Aziz et al. 2018).

2 Main Model

In this section, we present a formal statement of our main model. Suppose we have a bipartite graph $G=(U,V)$ where U and V represent the offline and online agents respectively. (Since we will not use single edges of G explicitly, we just refer to G as (U,V).) 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 or comes) from a known distribution $\{p_{v,t}\}$ such that $\sum_{v\in V} p_{v,t} \leq 1$. Thus, with probability $1-\sum_{v\in V} p_{v,t}$, no vertex from V will come. Note that the sampling process is independent across different rounds. For each v, it has an admissible collection of subsets of its neighbors $\mathcal{A}_v \subseteq 2^U$ [3] such that each $S \in \mathcal{A}_v$ has cardinality of at most Δ. Each time a vertex v arrives, we need to make an immediate and irrevocable decision: either to reject v or assign v an admissible set $S \in \mathcal{A}_v$. Assume that each u has a capacity $c_u \in \mathbb{Z}_+$, *i.e.,* u can be included in at most c_u different admissible sets. Each assignment S to v is associated with a non-negative weight/profit $w_v(S)$ and our goal is to design an online assignment policy such that the expected total weight of all assignments made is maximized. Note that $w_v(S)$ can be different for distinct v even for the same S.

For an assignment $f=(v,S,t)$ (assignment of $S \in \mathcal{A}_v$ to v at t), let $x_f = x_{v,S,t}$ be the probability that f is made in any offline optimal algorithm. We use LP (1) as our benchmark LP.

$$\max \sum_{t\in[T]} \sum_{v\in V} \sum_{S\in\mathcal{A}_v} w_v(S) x_{v,S,t} \tag{1}$$

$$\textstyle\sum_{S\in\mathcal{A}_v} x_{v,S,t} \leq p_{v,t} \qquad \forall v\in V, t\in[T] \tag{2}$$

$$\textstyle\sum_{t\in[T]} \sum_{v\in V} \sum_{S\in\mathcal{A}_v : S\ni u} x_{v,S,t} \leq c_u \qquad \forall u\in U \tag{3}$$

$$0 \leq x_{v,S,t} \leq 1 \qquad \forall v\in V, S\in\mathcal{A}_v, t\in[T] \tag{4}$$

This LP can be intuitively interpreted as follows. Constraint (2) says that for any given t and v, the probability that we assign v an admissible set should be no more than probability that v arrives at time t. Constraint (3) means that the expected number of times that u is assigned throughout the T rounds should be no more than its capacity c_u. The last constraint (4) is due to the fact that all $\{x_{v,S,t}\}$ are probability values and hence should lie in the interval $[0,1]$. The above analysis suggests that any offline optimal solution

[3] The collection $\mathcal{A}_v$ is usually closed under taking subsets, but we do not require it here.

$\{x_{v,S,t}\}$ should be feasible for the above LP. Therefore, we have Lemma 1, which claims that the optimal solution of this LP is an upper bound on the expected offline optimal value.

Lemma 1. *The optimal value to* LP (1) *is a valid upper bound for the offline optimal algorithm.*

The whole proof mainly consists of justification that any offline optimal strategy $\{x_{v,S,t}\}$ should be feasible for LP (1) and thus the optimal LP value offers a valid upper bound for the offline optimal value. We omit the proof here.

3 LP-based Online Sampling Algorithms

In this section, we present a class of parameterized LP-based online algorithms. Suppose $\{x^*_{v,S,t}\}$ is an optimal solution to LP (1). The main idea behind SAMP (described in Algorithm 1) is as follows. Suppose a vertex v arrives at time t: we sample an admissible set S from $\mathcal{A}_v$ with probability $\alpha x^*_{v,S,t}/p_{v,t}$, where $\alpha \in (0,1]$ is a parameter to be optimized in the analysis. We say a vertex u is *safe* at (the beginning of) time t iff u has at least one remaining capacity at t. Consequently, we say an assignment $f = (v, S, t)$ is *safe* (or S is safe at t) iff each vertex $u \in S$ is safe at t.

Algorithm 1: An LP-Based Sampling Algorithm (SAMP(α))

1. Suppose at time $t \in [T]$, the vertex v arrives.
2. Sample an admissible set S from $\mathcal{A}_v$ with probability $\alpha x^*_{v,S,t}/p_{v,t}$.
3. If S is safe at time t (*i.e.,* each vertex in S has at least one capacity), then assign S to v; otherwise, reject v.

The second step of Algorithm 1 is well defined since we have $\sum_{S\in\mathcal{A}_v} \alpha x^*_{v,S,t}/p_{v,t} \leq 1$ due to constraint (2) in our benchmark LP.[4]

A warm-up analysis for SAMP. Now we present a warm-up analysis for the case SAMP(1/2) and show that it achieves a ratio at least $\frac{1}{4\Delta}$.

Theorem 3. *By choosing* $\alpha = \frac{1}{2\Delta}$, SAMP($\alpha$) *achieves an online competitive ratio of at least* $\frac{1}{4\Delta}$.

The main idea is to prove that each assignment $f = (v, S, t)$ is *successfully* made with probability at least $\frac{x^*_f}{4\Delta}$. Thus by linearity of expectation and Lemma 1, we get our conclusion.

Proof. Without losing of generality (WLOG) assume that $t = T$ and consider a given assignment $f^* = (v^*, S^*, T)$. For each $u \in S^*$, let A_u be the (random) number of copies of u exhausted during the previous $T - 1$ rounds. Suppose $\mathcal{H}_u = \{f = (v, S, t)| t < T, v \in V, S \in \mathcal{A}_v, u \in S\}$, the set of assignments during the first $T-1$ rounds where u is involved. For each assignment $f = (v, S, t)$, let X_f indicate if f is *successfully* made. Thus we see that $A_u = \sum_{f\in\mathcal{H}_u} X_f$. Notice that for each $f = (v, S, t)$, we have $\mathbb{E}[X_f] \leq \frac{\alpha x^*_f}{p_{v,t}} \cdot p_{v,t} = \alpha x^*_f$, which is due to the fact that f is successfully made only if v comes at t and f is sampled. Therefore,

$$\begin{aligned}\mathbb{E}[A_u] &= \mathbb{E}\Big[\textstyle\sum_{f\in\mathcal{H}_u} X_f\Big] = \sum_{t<T,v\in V}\sum_{S:S\in\mathcal{A}_v,S\ni u}\mathbb{E}[X_{v,S,t}] \\ &\leq \textstyle\sum_{t<T,v\in V}\sum_{S:u\in S\in\mathcal{A}_v}\alpha x^*_{v,S,t} \\ &\leq \alpha c_u \quad \text{due to constraint (3)}\end{aligned}$$

Thus by Markov's inequality, $\Pr[A_u \geq c_u] \leq \alpha$, which implies that $\Pr[A_u \leq c_u - 1] \geq 1 - \alpha$. Let SF_{f^*} be the event that f^* is safe, *i.e.,* all vertices in S^* has at least one capacity left at (the beginning of) T. By applying the union bound and our assumption that $|S^*| \leq \Delta$, we have

$$\Pr[\mathsf{SF}_{f^*}] = \Pr[\wedge_{u\in S^*}(A_u \leq c_u - 1)] \geq 1 - \Delta\alpha \qquad (5)$$

Therefore, f^* will be *successfully* made with probability at least $\frac{\alpha x^*_{f^*}}{p_{v^*,T}} \cdot p_{v^*,T} \cdot (1 - \Delta\alpha) = \alpha x^*_{f^*}(1 - \Delta\alpha)$. By setting $\alpha = \frac{1}{2\Delta}$, we get that f^* is made with probability at least $\frac{x^*_{f^*}}{4\Delta}$. Thus we get our claim. □

A tight analysis for SAMP. Consider first the following example which motivates us to obtain the tight analysis.

Example 1. *Consider the graph* $G = (U, V)$*, where* $U = \{u_i | i = 1, 2\}$ *and* $V = \{v_j | j = 1, 2, 3\}$*. For ease of notation, we directly use i and j to denote* u_i *and* v_j *respectively. Set* $T = 2$ *and the arrival distributions are as follows: when* $t = 1$*,* $j = 1, 2$ *arrives with equal probability* 1/2 *and when* $t = 2$*,* $j = 3$ *will arrive with probability* 1*. Each* $\mathcal{A}_j$ *includes only one single set* S_j *with* $1 \leq j \leq 3$*, where* $S_1 = (u_1)$*,* $S_2 = (u_2)$ *and* $S_3 = (u_1, u_2)$ *with* $\Delta = 2$*. Each* u_i *has a unit capacity* $c_i = 1$ *for* $i = 1, 2$*.*

Assume all assignments have a unit weight. Let $f_1 = (v_1, S_1, t = 1)$*,* $f_2 = (v_2, S_2, t = 1)$ *and* $f_3 = (v_3, S_3, t = 2)$*. Consider such an optimal solution to* LP-(1)*:* $x^*_{f_1} = x^*_{f_2} = x^*_{f_3} = 1/2$*. Let us analyze the assignment* f_3 *when* v_3 *comes at* $t = 2$ *by running* SAMP(α)*. Observe that* f_3 *is safe iff both* u_1 *and* u_2 *are safe (*i.e., *both should have one capacity left at* $t = 2$*). According to* SAMP(α)*, at* $t = 1$ *we will choose* f_1 *with probability* $\frac{\alpha x^*_{f_1}}{p_{v_1,t=1}} = \alpha$ *when* v_1 *comes at* $t = 1$*; similarly we will choose* f_2 *with probability* α *when* v_2 *comes at* $t = 1$*. Notice that at* $t = 2$*,* u_1 *and* u_2 *each is safe with probability* $1 - \alpha/2$*, while both* u_1 *and* u_2 *are safe with probability* $1 - \alpha$*.*

Remarks. (1) The key inequality (5) in the proof of Theorem 3 obtained via Markov's inequality and the union bound is not tight. Consider f_3 in the above example, $\Pr[\mathsf{SF}_{f_3}] = 1 - \alpha > 1 - \Delta\alpha$. (2) Unfortunately, the events that different u are safe are not *positively* correlated as desired (in which case we can replace the RHS of inequality (5) with the product of probabilities of all u is safe). Let SF_i be the event that u_i is safe at $t = 2$ in the above example for $i = 1, 2$. We have shown that $\Pr[\mathsf{SF}_1 \wedge \mathsf{SF}_2] = 1 - \alpha < \Pr[\mathsf{SF}_1] \cdot \Pr[\mathsf{SF}_2]$.

Next, we prove Theorem 1. Due to space limitations, we present only the proof for the simple case of unit capacities and show that it is the exact case when SAMP achieves

[4] In other words, with probability $1 - \sum_{S\in\mathcal{A}_v} \alpha x^*_{v,S,t}/p_{v,t}$, we will do nothing and directly reject vertex v.

the worst performance. The full proof of the general case is highly technically involved, which we defer to the supplementary materials. The high-level idea for the case of unit capacities is as follows: we try to decompose the event that a given assignment $f = (v^*, S^*, t^*)$ being safe along the dimension of online time-steps $t = 1, 2, \cdots, t^* - 1$ instead of each $u \in S^*$ is safe.

Proof. WLOG consider the case $t = T$ and a given assignment $f^* = (v^*, S^*, T)$. For each $t < T$, let $\mathcal{F}_t = \{f = (v, S, t) : v \in V, S \in \mathcal{A}_v, |S \cap S^*| \geq 1\}$ be the set of assignments at t in which the admissible set has at least one element in S^*. Let SF_t be the event that f^* is safe at (the beginning of) t. Thus we have $\Pr[\mathsf{SF}_1] = 1$ and $\Pr[\mathsf{SF}_{t+1}|\mathsf{SF}_t] = 1 - \sum_{f \in \mathcal{F}_t} \alpha x_f^*$. Therefore,

$$\Pr[\mathsf{SF}_T] = \Pr[\wedge_{t<T} \mathsf{SF}_t] = \prod_{t=1}^{T-1} \Pr[\mathsf{SF}_{t+1}|\mathsf{SF}_t] \quad (6)$$

$$= \prod_{t=1}^{T-1} \left(1 - \sum_{f \in \mathcal{F}_t} \alpha x_f^*\right) \quad (7)$$

Here are two useful observations. First, for each $t < T$,

$$\sum_{f \in \mathcal{F}_t} \alpha x_f^* \leq \sum_{v \in V} \sum_{S \in \mathcal{A}_v} \alpha x_{v,S,t}^*$$
$$\leq \sum_{v \in V} \alpha p_{v,t} \leq \alpha \quad \text{(due to constraint (2))}$$

Second,

$$\sum_{t<T, f \in \mathcal{F}_t} \alpha x_f^* \leq \sum_{u \in S^*} \left(\sum_{t<T, v \in V, S \in \mathcal{A}_v, S \ni u} \alpha x_{v,S,t}^*\right)$$
$$\leq \sum_{u \in S^*} \alpha \leq \alpha\Delta \quad \text{(due to constraint (3))}$$

The first observation says that each term after the minus symbol in the product (7) is at most α while the second indicates the sum of all those terms (after the minus) over $t \in T$ is at most $\alpha\Delta$. Therefore, we claim that the product (7) has a minimum value of $(1-\alpha)^\Delta$. This implies that f^* will be *successfully* made with probability at least $\alpha x_{f^*}^* \cdot (1-\alpha)^\Delta$. By choosing $\alpha = 1/(\Delta + 1)$ and using linearity of expectation, we prove our claim. □

The example below shows that the above analysis is tight.

Example 2 (Tight Example)**.** *Consider the graph $G = (U, V)$ where $U = \{u_i | i \in [m]\}, V = \{v_j | j \in [n]\}$ with $m = \Delta, n = T = \Delta + 1$. For ease of notation, let us use i and j to denote u_i and v_j respectively for $i \in [m], j \in [n]$. For each $t \in [T]$, $p_{j,t} = 1$ iff $j = t$ and 0 otherwise (hence at time t, vertex $j = t$ arrives surely). For each vertex v_j, its corresponding admissible collect $\mathcal{A}_j$ includes only one single subset S_j, where $S_j = (u_i, i = j)$ for each $j \in [n-1]$ and $S_n = U$ and thus the largest size of $\{S_j | j \in [n]\}$ is equal to Δ. Set $c_i = 1$ for all $i \in [m]$ (all u_i have a unit capacity).*

For each $j \in [n]$, let $f_j = (v_j, S_j, t = j)$ be the only possible assignment associated with v_j when it comes at $t = j$. Set $x_j = x_{f_j}$ and consider such an optimal solution to LP (1)*: $x_j^* = 1 - \epsilon$ for each $j \leq \Delta$ and $x_{\Delta+1}^* = \epsilon$ (we can always make such a feasible solution to be optimal by arranging a proper weight vector).*

Now consider the assignment $f_n = (v_n, S_n, t = T)$ when the last vertex v_n comes at $t = T$. Let us compute the probability $\Pr[\mathsf{SF}_{f_n}]$ *that f_n is safe in* SAMP(α)*. Observe that assignment f_n will be safe iff none of $f_j, j < n$ is successfully made before. Notice that at each time $t < T$,* SAMP(α) *will successfully make the assignment $f_{j=t}$ with probability $\frac{\alpha x_j^*}{p_j} \cdot p_j = \alpha(1-\epsilon)$. This implies that* $\Pr[\mathsf{SF}_{f_n}] = (1-\alpha(1-\epsilon))^\Delta$*, which essentially matches the lower bound we compute for* $\Pr[\mathsf{SF}_{f_n}]$ *in the proof of Theorem 1.*

4 Hardness Results

In this section, we prove Theorem 2. The big idea is to reduce the classical k-uniform hypergraph matching to a special case of OM-CC-KAD with $\Delta = k$.

Proof of Theorem 2. Consider a given k-uniform hypergraph $\mathcal{H} = (\mathcal{U}, \mathcal{E})$, where $\mathcal{U}$ and $\mathcal{E}$ are the respective set of vertices and hyperedges with $|\mathcal{U}| = m$ and $|\mathcal{E}| = n$. Thus we have that each hyperedge has a cardinality of k. A hypermatching on $\mathcal{H}$ is a set of hyperedges which is mutually disjoint. Based on $\mathcal{H}$, we create an instance $\mathcal{I}(\mathcal{H})$ of OM-CC-KAD as follows. Set the offline and online sets of vertices as $\mathcal{U}$ and $\mathcal{E}$ respectively. Let $\mathcal{E} = \{f_j | j \in [n]\}$ and we view each hyperedge as an online vertex in our case. Set $T = n$ and for each $t \in [T]$, $p_{f_j,t} = 1$ iff $j = t$ and 0 otherwise. In other words, at time t, $f_{j=t}$ comes surely and no other vertex will come. For each f_j, it has a single admissible set $S_j = f_j \subseteq \mathcal{U}$ (recall that f_j is a hyperedge, *i.e.,* a subset of $\mathcal{U}$). Thus in our case $\Delta = \max_j |S_j| = k$. Assume all admissible sets have a unit weight and set $c_u = 1$ for all $u \in \mathcal{U}$.

Let LP$(\mathcal{H})$ be the standard relaxed LP for $\mathcal{H}$ (we use LP$(\mathcal{H})$ to denote the optimal LP value as well). Similarly, assume LP$(\mathcal{I}(\mathcal{H}))$ as the benchmark LP (1) for $\mathcal{I}(\mathcal{H})$. We can verify that LP$(\mathcal{H})$ is essentially the same as LP$(\mathcal{I}(\mathcal{H}))$ and thus the optimal values are the same. Note that any feasible set of assignments $\{f_j\}$ collected by any online algorithm surely form a hypermatching over $\mathcal{H}$. Thus, we claim that OPT$(\mathcal{I}(\mathcal{H})) \leq$ OPT$(\mathcal{H})$, where the former is the optimal online performance while the latter is the maximum size of matching over $\mathcal{H}$. From (Füredi, Kahn, and Seymour 1993) we see that there does exist some $\mathcal{H}^*$ such that $\frac{\text{OPT}(\mathcal{H}^*)}{\text{LP}(\mathcal{H}^*)} \leq \frac{1}{k-1+1/k}$. Thus we claim that $\frac{\text{OPT}(\mathcal{I}(\mathcal{H}^*))}{\text{LP}(\mathcal{I}(\mathcal{H}^*))} \leq \frac{\text{OPT}(\mathcal{H}^*)}{\text{LP}(\mathcal{H}^*)} \leq \frac{1}{k-1+1/k}$. □

5 Experiments

In this section, we present our experimental results. We test SAMP against several natural heuristic baselines on both the Meetup dataset (Liu et al. 2012) and synthetic datasets. The details of our datasets and experimental setup are as follows.

Meetup dataset and experimental setup. We use the Meetup dataset from (Liu et al. 2012) collected between Oct. 2011 to Jan. 2012. Meetup[5] is a social platform which facilitates pairing users with their target events. In this dataset, there are three entities: users, events and groups. Every user and event is associated with a geographic coordinate. Users and groups are associated with tags which show the corresponding features. Every event inherits the tags of the group hosting the event. In our experiment, we focus on the city of San Francisco, and extract users and events with longitude and latitude in ranges $(-122.3475, -122.5201)$ and

[5] https://www.meetup.com/

(37.7112, 37.8153) respectively. Table 1 presents the statistics of the extracted data. The number of events attended by users and that of users admitted by events both follow Pareto distributions (visualized in the supplementary materials).

# users	13492
# events	13596
Avg. # events per user	6.02
Median # events per user	2
Avg. # users per event	5.97
Median # users per event	3

Table 1: Statistics of the Meetup dataset (Liu et al. 2012) in San Francisco from Oct. 2011 to Jan. 2012.

In our experimental setup, we randomly sample 50 events (U) and 100 users (V) from the extracted data. We compute the weight/profit of each assignment similar to (She et al. 2016). First, we manually cluster tags with similar meanings and select the 20 most popular tags. Then, we represent each user u and event v as a canonical binary vector of dimension 20 over the selected tags (denote by $\mathbf{l}_u$ and $\mathbf{l}_v$ respectively). For each pair of event u and user v, we define the similarity between u and v as $w(u,v) \doteq 1 - \frac{||\mathbf{l}_u - \mathbf{l}_v||}{\sqrt{20}}$, where $||\mathbf{l}_u - \mathbf{l}_v||$ denotes the Euclidean distance and the denominator $\sqrt{20}$ is to ensure that $w(u,v) \in [0,1]$. For each subset $S \subseteq U$, we define $w_v(S) = \sum_{u \in S} w(u,v)$. For each event u and user v, let $\widehat{c}_u$ and $\widehat{c}_v$ be the respective number of users that u admitted and number of events that v attended. We set capacity $c_u = \widehat{c}_u$ and choose $c_v \in \{\lceil \frac{1}{3}\widehat{c}_v \rceil, \lceil \frac{2}{3}\widehat{c}_v \rceil, \widehat{c}_v\}$ (since the dataset is collected over 3 months we offset user's capacity accordingly). The dataset does not include, explicitly, the conflict information over events and we generate the collection of admissible sets $\mathcal{A}_v$ for each user v as follows. We set $N_v = 2c_v$ and $A_v = 2N_v$. Select N_v nearest events to v from U as the neighbors of v (denoted by $\mathcal{N}_v$). We sample c_v events from $\mathcal{N}_v$ uniformly with replacement and set an admissible set S as the set of all distinct events sampled. We repeat this process until we get $\min(A_v, 2^{N_v} - 1)$ different admissible sets. Set $T = 1000$ and for each time $t \in [T]$, we generate a random vector $\{p_{v,t}\}$ such that each $p_{v,t}$ takes a uniform value from $[0,1]$ and $\sum_v p_{v,t} = 1$.

Synthetic datasets. We set $|U| = 50$ and $|V| = 100$, *i.e.,* 50 events and 100 user types. For each user v, we generate its neighbor $\mathcal{N}_v$ by independently sampling each u with probability p, where p is a given parameter. Let $N_v = |\mathcal{N}_v|$; we set $c_v = \max(5, \frac{N_v}{2})$ and $A_v = 2N_v$. After obtaining $\mathcal{N}_v, c_v$ and A_v, we generate the collection of admissible sets $\mathcal{A}_v$ for each user v similar to that in the real dataset. Thus each $S \in \mathcal{A}_v$ has size at most c_v and $\mathcal{A}_v$ has at most A_v sets. Let C_U be an upper bound of the capacity for all events and for each event v, we sample a value from $[C_U]$ according to the distribution learned from the extracted dataset (with appropriate scaling such that the total sum is 1). For each pair (u,v), we sample a uniform value $w(u,v)$ in $[0,1]$ and set $w_v(S) = \sum_{u \in S} w(u,v)$ for each $S \in \mathcal{A}_v$. The generation of arrival distributions of $\{p_{v,t}\}$ is the same as that in the real dataset. In our experimental setup, we have three parameters (C_U, T, p), where T is the number of online rounds. We generate a set of instances by varying one single parameter over a range and fixing all other parameter on the default value. Table 2 summarizes the details.

Factors	Setting
C_U	$6, 8, \mathbf{10}, 12, 14, 16, 18, 20$
T	$100, 200, 300, 400, \mathbf{500}, 600, 700, 800, 900, 1000$
p	$0.05, \mathbf{0.1}, 0.15, 0.2, 0.25$

Table 2: Synthetic dataset, the default settings are marked as bold.

LP-based heuristics and baselines. We propose two LP-based heuristics—SAMP(1) and SAMP(0.8)—and test these two against two other baselines, namely Greedy and Uniform Sampling. The details of these algorithms are as follows. Consider a user v arriving at time t.

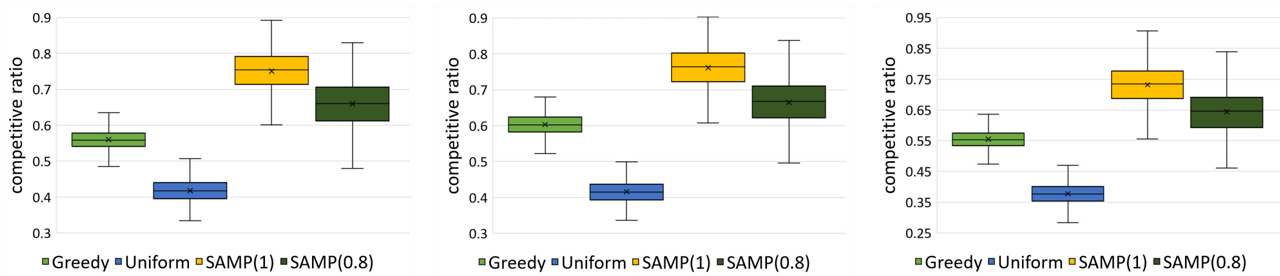
- SAMP(1): Samples an admissible set $S \in \mathcal{A}_v$ with probability $x^*_{v,S,t}/p_{v,t}$ and assigns S to v if S is safe.
- SAMP(0.8): Samples an admissible set $S \in \mathcal{A}_v$ with probability $0.8x^*_{v,S,t}/p_{v,t}$ and assigns S to v if S is safe.
- Greedy. Assigns v a safe admissible $S \in \mathcal{A}_v$ which has the maximum weight $w_v(S)$ among all safe choices at time t (breaking ties arbitrarily).
- Uniform. Samples a set $S \in \mathcal{A}_v$ *uniformly* and assigns S to v if S is safe.

Note that among the above four algorithms, only Greedy is *adaptive* in the sense that its strategy responds to the outcome of previous strategies. We test the algorithm SAMP(α) as suggested by Theorem 1 with $\alpha = \frac{1}{1+\Delta}$ where $\Delta = \max_v c_v$ is the largest possible size over all possible admissible sets. On practical instances, we find SAMP($\frac{1}{1+\Delta}$) is too conservative and works optimally only for the theoretical worst case as shown in Example 2. As shown in Figures 1 and 2, heuristics of SAMP(α), with larger α values, (*e.g.,* 1 and 0.8 here) work much better in both real and synthetic datasets.

Methodology. We solve all LPs via the Glop Linear Solver[6] on commodity hardware: an Intel Core i7-7700 (2.80 GHz) machine with 16GB of main memory. For most of the instances on both real and synthetic datasets, the LPs can be solved within 30 seconds. For each generated instance (synthetic and real), we run each of the four algorithms for 1000 independent trials. For the real case, we output the maximum, third quartile, median, first quartile, minimum (the four lines from top to down) and average (the cross) among the 1000 trials in a box plot. For the synthetic case, we simply take the average as the final performance. We compute the ratio of the performance of each algorithm to the optimal value of the corresponding LP as the final competitive ratio. The results of the four algorithms on the different settings of the real and synthetic datasets are summarized respectively in Figures 1 and 2.

Discussion. Overall, our LP-based heuristics work much better than the two baselines. We observe that SAMP(1) > SAMP(0.8) > Greedy > Uniform overall. Notably, SAMP(1) can beat the rest two baselines by a non-trivial

[6] https://developers.google.com/optimization/lp/glop



(a) $c_v = \frac{1}{3}\widehat{c_v}$ (b) $c_v = \frac{2}{3}\widehat{c_v}$ (c) $c_v = \widehat{c_v}$

Figure 1: Competitive ratios achieved by our two LP-based methods, SAMP(1) and SAMP(0.8), and the two baseline methods, Greedy and Uniform, on the real-world Meetup dataset (Liu et al. 2012).

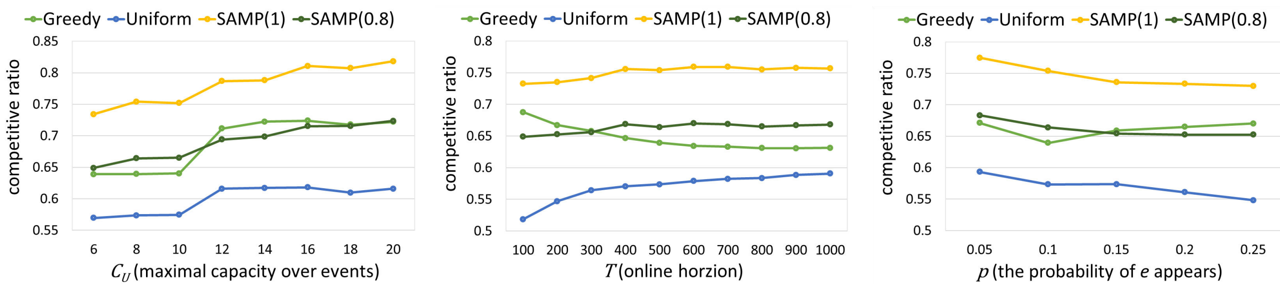


Figure 2: Competitive ratios achieved by our two LP-based methods, SAMP(1) and SAMP(0.8), as well as two baseline methods, Greedy and Uniform, on a synthetic dataset.

ratio gap of at least 0.1. The three plots in Figure 1 show that LP-based heuristics achieve not only significantly better average performances (*effectiveness*) but also maintain relative low variances (*robustness*). In fact, as shown in the three plots, the worst performance of SAMP(1) (*i.e.,* the minimum value) among the 1000 trials almost matches the median value of Greedy, and the median performance of SAMP(1) can noticeably beat the best performance of Greedy.

The first plot in Figure 2 shows that as the upper bound of capacity on the event side gets larger, the competitive ratios of the LP-based heuristics and Greedy both increase similarly, and the gap stays around 0.1. The first part of the observation is also consistent with our theoretical analysis for SAMP, which suggests that the performance of SAMP achieves the worst when all $c_u = 1$ and gets better when each c_u gets larger. The second plot in Figure 2 shows that as T gets larger, the performances of our LP-based heuristics get better while that of Greedy gets worse. This highlights the power of global-arrangement in the online matching policy design, which is possessed exclusively by the LP-based heuristics. As T gets larger, we expect to have more arrivals of online agents and thus have a more critical need to globally optimize the online matching strategy by integrating all external information such as the arrival distributions of online agents, capacities on the offline agents, and different structures of admissible sets for online agents. LP-based heuristics achieve it by solving a well-designed LP (*i.e.,* the benchmark LP), while Greedy and Uniform are both incapable of addressing that. This third plot in Figure 2 is consistent with our theoretical prediction: when p gets larger (so do N_v and c_v, according to the experimental setup), $\Delta = \max_v c_v$ also gets larger, thus SAMP expects worse performance by Theorem 1.

6 Conclusion

In this paper, we proposed a unifying model of online matching with offline conflict-aware constraints, a setting that arises in many practical applications where interactions or constraints exist between agents or items on the same side of the market. Our model can capture very general conflict constraints over the offline agents; it also allows the arrival distributions of online agents to change over the online phase. We proposed an LP-based online algorithm with provable performance.

Acknowledgements

The research of Karthik Sankararaman and Pan Xu was supported in part by NSF Awards CNS 1010789 and CCF 1422569. Aravind Srinivasan's research was supported in part by NSF Awards CNS-1010789, CCF-1422569 and CCF-1749864, and by research awards from Adobe, Inc. Yexuan Shi, Hao Cheng, and Yongxin Tong are partially supported by the National Science Foundation of China (NSFC) under Grant No. 61822201 and 61532004, National Grand Fundamental Research 973 Program of China under Grant 2015CB358700, the Science and Technology Major Project of Beijing under Grant No. Z171100005117001 and Didi Gaia Collaborative Research Funds for Young Scholars. Yongxin Tong is the corresponding author in this paper. The authors would like to thank the anonymous reviewers for their helpful feedback.

References

Ahmed, F.; Dickerson, J. P.; and Fuge, M. 2017. Diverse weighted bipartite b-matching. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 35–41.

Alaei, S.; Hajiaghayi, M.; and Liaghat, V. 2012. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, 18–35.

Alaei, S.; Hajiaghayi, M.; and Liaghat, V. 2013. The online stochastic generalized assignment problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer. 11–25.

Assadi, S.; Hsu, J.; and Jabbari, S. 2015. Online assignment of heterogeneous tasks in crowdsourcing markets. *CoRR* abs/1508.03593.

Aziz, H.; Chen, J.; Gaspers, S.; and Sun, Z. 2018. Stability and pareto optimality in refugee allocation matchings. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 964–972.

Chen, C.; Zheng, L.; Srinivasan, V.; Thomo, A.; Wu, K.; and Sukow, A. 2016a. Conflict-aware weighted bipartite b-matching and its application to e-commerce. *IEEE Transactions on Knowledge and Data Engineering* 28(6):1475–1488.

Chen, X.; Ma, W.; Simchi-Levi, D.; and Xin, L. 2016b. Dynamic recommendation at checkout under inventory constraint. *SSRN:https://ssrn.com/abstract=2853093*.

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 *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 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 *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 318–326.

Fortuin, C. M.; Kasteleyn, P. W.; and Ginibre, J. 1971. *Communications in Mathematical Physics* (2).

Füredi, Z.; Kahn, J.; and Seymour, P. D. 1993. On the fractional matching polytope of a hypergraph. *Combinatorica* 13(2).

Goto, M.; Iwasaki, A.; Kawasaki, Y.; Kurata, R.; Yasuda, Y.; and Yokoo, M. 2016. Strategyproof matching with regional minimum and maximum quotas. *Artif. Intell.* 235:40–57.

Ho, C.-J., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 45–51.

Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, 352–358.

Kawase, Y., and Iwasaki, A. 2018. Approximately stable matchings with budget constraints. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 1113–1120.

Kell, N., and Panigrahi, D. 2016. Online budgeted allocation with general budgets. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 419–436. ACM.

Kurata, R.; Hamada, N.; Iwasaki, A.; and Yokoo, M. 2017. Controlled school choice with soft bounds and overlapping types. *J. Artif. Intell. Res.* 58:153–184.

Li, K.; Lu, W.; Bhagat, S.; Lakshmanan, L. V.; and Yu, C. 2014. On social event organization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1206–1215. ACM.

Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; and Liu, Y. 2018. Multi-task representation learning for travel time estimation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1695–1704.

Liu, X.; He, Q.; Tian, Y.; Lee, W.-C.; McPherson, J.; and Han, J. 2012. Event-based social networks: Linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1032–1040.

Mehta, A. 2013. Online matching and ad allocation. *Found. Trends Theor. Comput. Sci.* 8(4).

Qin, L., and Zhu, X. 2013. Promoting diversity in recommendation by entropy regularizer. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2698–2704.

Rzeszotarski, J. M., and Kittur, A. 2011. Instrumenting the crowd: Using implicit behavioral measures to predict task performance. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, 13–22.

Sha, C.; Wu, X.; and Niu, J. 2016. A framework for recommending relevant and diverse items. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 3868–3874.

She, J.; Tong, Y.; Chen, L.; and Cao, C. C. 2016. Conflict-aware event-participant arrangement and its variant for online setting. *IEEE Transactions on Knowledge and Data Engineering* 28(9):2281–2295.

Singer, Y., and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22Nd International Conference on World Wide Web*, 1157–1166.

Singla, A., and Krause, A. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22Nd International Conference on World Wide Web*, 1167–1178.

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. *Proc. VLDB Endow.* 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 *32nd IEEE International Conference on Data Engineering*, 49–60.

Tong, Y.; Wang, L.; Zhou, Z.; Ding, B.; Chen, L.; Ye, J.; and Xu, K. 2017. Flexible online task assignment in real-time spatial data. *Proc. VLDB Endow.* 10(11).

Wang, Y., and Wong, S. C.-w. 2016. Matroid online bipartite matching and vertex cover. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, 437–454. ACM.

Wang, Z.; Fu, K.; and Ye, J. 2018. Learning to estimate the travel time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 858–866.

Xu, P.; Srinivasan, A.; Sarpatwar, K. K.; and Wu, K.-L. 2017. Budgeted online assignment in crowdsourcing markets: Theory and practice. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 1763–1765.

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 *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.