Online Matching with Controllable Rewards and Arrival Probabilities

Yuya Hikima[1] , **Yasunori Akagi**[1] , **Naoki Marumo**[2] and **Hideaki Kim**[1]
[1]NTT Human Informatics Labratories, NTT Corporation,
[2]NTT Communication Science Laboratories, NTT Corporation
{yuuya.hikima.ys, yasunori.akagi.cu}@hco.ntt.co.jp, naoki.marumo@ntt.com,
hideaki.kin.cn@hco.ntt.co.jp

Abstract

Online bipartite matching has attracted much attention due to its importance in various applications such as advertising, ride-sharing, and crowdsourcing. In most online matching problems, the rewards and node arrival probabilities are given in advance and are not controllable. However, many real-world matching services require them to be controllable and the decision-maker faces a non-trivial problem of optimizing them. In this study, we formulate a new optimization problem, *Online Matching with Controllable Rewards and Arrival probabilities* (OM-CRA), to simultaneously determine not only the matching strategy but also the rewards and arrival probabilities. Even though our problem is more complex than the existing ones, we propose a fast $1/2$-approximation algorithm for OM-CRA. The proposed approach transforms OM-CRA to a saddle-point problem by approximating the objective function, and then solves it by the Primal-Dual Hybrid Gradient (PDHG) method with acceleration through the use of the problem structure. In simulations on real data from crowdsourcing and ride-sharing platforms, we show that the proposed algorithm can find solutions with high total rewards in practical times.

1 Introduction

Online bipartite matching [Mehta, 2012; Feldman *et al.*, 2009] has attracted much attention because of its many applications, such as advertising [Mehta, 2012], ride-sharing [Dickerson *et al.*, 2018], and crowdsourcing [Ho and Vaughan, 2012]. These applications require platformers to find efficient online ways of allocating their limited resources among customers or participants, which significantly impacts their business profits. Formally, the problem of allocation is defined on a bipartite graph $G = (U, V; E)$: U is the fixed node set known in advance, and nodes in V arrive at each time in a probabilistic manner; a reward is obtained when an arrival node in V is matched with a node in U; the goal is to find a matching strategy that maximizes the total rewards.[1]

Existing studies assume that rewards and arrival probabilities are given a priori as fixed constants [Mehta, 2012; Dickerson *et al.*, 2018], in which case only the matching strategy needs to be optimized. However, platformers can usually control rewards, which affect arrival probabilities. This triggers the non-trivial problem of optimizing the rewards as well as the matching strategy for greater profits, due to the trade-off between rewards and arrival probabilities. For example, in a ride-sharing platform, higher fares (rewards) increase the profit obtained with each match, but cause lower participation (arrival) probabilities of requesters for taxis assignment, which might degrade the total profits obtained even if the matching strategy is optimal. Although important and appearing in many applications (described in Section 3.2), the problem has not, to the best of our knowledge, been tackled directly in online matching studies.

In this study, we propose the novel problem of simultaneously optimizing the matching strategy and the trade-off between rewards and arrival probabilities to maximize business profits. We call the problem *Online Matching with Controllable Rewards and Arrival probabilities* (OM-CRA). OM-CRA solutions realize profitable online matching under an appropriate balance of rewards and arrival probabilities. In the case of a ride-sharing platform, business profits are expected to increase in the following scenario: for areas with few taxis, a fare (reward) per one match is prioritized over the arrival probabilities of requesters; for areas with many taxis, the arrival probabilities are increased with lower fares to stimulate many matches.

Although OM-CRA is an important problem for applications, OM-CRA has two difficulties to solve: (i) we need to simultaneously optimize the matching strategy and the trade-off between the rewards and the arrival probabilities (finding just the optimal matching strategy is hard [Manshadi *et al.*, 2012]); (ii) it takes a lot of time to compute the objective value because it is the expected total rewards for the random order of participants with exponential realizations.

In this paper, we develop a fast 1/2-approximation algorithm for OM-CRA. First, based on recent studies on online matching [Dickerson *et al.*, 2018; Alaei *et al.*, 2012], we approximate the objective value by the optimal value of a linear optimization problem. Using this approximation reduces

[1]Codes/details of our experiments and the proof of Lemma 2 can be found in https://github.com/Yuya-Hikima/IJCAI2022-Online-Matching-with-Controllable-Rewards-and-Arrival-Probabilities.

OM-CRA to a non-convex continuous optimization problem whose objective value can be evaluated easily. Then, we reduce the problem to a convex-concave saddle-point problem [Benzi *et al.*, 2005] under mild assumptions. Many important functions typically used in the applications satisfy these assumptions. We then solve the convex-concave saddle-point problem by the Primal-Dual Hybrid Gradient (PDHG) method [Chambolle and Pock, 2011; Goldstein *et al.*, 2015]. It is a powerful optimization method for saddle-point problems; an optimal solution is attained by alternately updating the primal and the dual variables. By utilizing the problem structure, we can accelerate the update of the primal variables in PDHG, which allows us to find a solution quickly even for large-scale problems.

We conducted simulation experiments on real data from a crowdsourcing platform and a ride-sharing platform. The results show that the proposed algorithm outputs solutions with higher total rewards than baselines in practical time.

Notation. Bold lowercase symbols (e.g., $\boldsymbol{x}, \boldsymbol{y}$) denote vectors, and $\|\boldsymbol{x}\|$ denotes the Euclidean norm of vector $\boldsymbol{x}$. Capitalized writing forms (e.g., $\mathcal{A}$) represent linear maps, while $\mathcal{A}^\top$ denotes the transpose map for the linear map $\mathcal{A}$. The inner product of the vectors $\boldsymbol{x}, \boldsymbol{y}$ is denoted by $\boldsymbol{x}^\top \boldsymbol{y}$. For $\boldsymbol{x} \in \mathbb{R}^n$, we denote $(\boldsymbol{x})_+ := (\max\{0, x_i\})_{i=1,\dots,n}$.

2 Related Works

2.1 Online Bipartite Matching

In the research field of online matching, the following four problem settings have been the focus of most studies (overviewed by [Mehta, 2012]): (a) *Known Identical Independent Distributions* (KIID), where each node v has the same arrival probability throughout the entire period [Feldman *et al.*, 2009; Haeupler *et al.*, 2011]; (b) *Unknown Identical Independent Distributions* (Unknown IID), where each node v has a fixed but unknown arrival probability [Devanur *et al.*, 2011]; (c) *Adversarial order*, where an adversary can determine the arrival order of all nodes v [Karp *et al.*, 1990; Sun *et al.*, 2017]; (d) *Random order*, where all v arrive in a random permutation order [Mahdian and Yan, 2011]. In addition, recently, generalizations of KIID have been reported; *Known Adversarial Distribution* (KAD) assumes that each arrival node v has a different arrival probability at each time [Alaei *et al.*, 2012]. *Online Matching with Reusable Resources under Known Adversarial Distributions* (OM-RR-KAD) is a generalization of KAD, where fixed node u rejoins the system after a certain period after the previous assignment [Dickerson *et al.*, 2018].

In this paper, we further extend OM-RR-KAD and propose a new problem called *Online Matching with Controllable Rewards and Arrival probabilities* (OM-CRA), which optimizes not only the matching strategy but also the trade-off between the rewards and the arrival probabilities under the same setting as OM-RR-KAD. The reason for focusing on the OM-RR-KAD setting is that it is suitable for various applications. For example, in a ride-sharing platform, taxis (resources) can be reused after a certain period of being allocated to requesters. In addition, the arrival probability of requester v, defined by origin/destination areas, varies with time of day. Our problem can deal with these applications since it is an expansion of OM-RR-KAD.

2.2 Optimization problems with Decision-Dependent Noise

Since the uncertainty (i.e., arrival probabilities of nodes) in our problem depends on the decision variables, our optimization problem is classified as a optimization problem with decision-dependent noise [Hellemo *et al.*, 2018]. This category of problems has been addressed for several application areas, such as matching platforms [Hikima *et al.*, 2021; Tong *et al.*, 2018], crowdsourcing systems [Wang *et al.*, 2018], single item markets [Babaioff *et al.*, 2015], and transportation investment [Peeta *et al.*, 2010].

Among these studies, [Hikima *et al.*, 2021; Tong *et al.*, 2018] have tackled matching problems with decision-dependent noise, similar to ours. [Hikima *et al.*, 2021] optimizes some variables, which affect rewards and node existence probabilities, to maximize the expected profit from batch-type matching. In the context of the ride-sharing platform, [Tong *et al.*, 2018] determines the fare for passengers in each area to control passenger participation rates and maximize the expected profits from taxi-requester matching. The difference between these existing studies and ours is whether or not the timeline factor is considered. Existing studies consider only one-time (batch) matching and seek a myopic solution at a certain point in the time horizon. In contrast, our study considers online matching situations and seeks an optimal solution from a long-term perspective.

A problem setting similar to ours is found in [Babaioff *et al.*, 2015]. That work performs price optimization for single item markets. Here, the reward for selling a product and the participant's purchase probability can be controlled by the price. It is similar to our study because we control rewards and arrival probabilities in online matching situations. However, that study did not consider the compatibility of resources and participants, while our study considers it. This allows us to deal with crowdsourcing platforms with skill compatibility between tasks and workers, and ride-sharing platforms with distance compatibility between taxis and requesters.

Optimization Methods. For the optimization problem with decision-dependent noise, search methods such as Bayesian optimization [Brochu *et al.*, 2010] and random search [Bergstra and Bengio, 2012] can be applied. However, for our problem, they do not provide good solutions in practical time for the following reasons: (i) the feasible region of the optimization problem is too large to be adequately explored; (ii) it takes a lot of time to compute the exact objective value or its approximation with high accuracy. In contrast, our method can output a $1/2$-approximation solution in practical time.

3 Problem Formulation

3.1 Matching Procedure

Notation. In this paper, we consider allocating resources to participants who arrive one by one at each time $t \in T :=$

$\{1, 2, \ldots, t_{\max}\}$.[2] The set of resources $U \coloneqq \{u_1, \ldots, u_m\}$, the set of participants $V \coloneqq \{v_1, \ldots, v_n\}$, and bipartite graph $G = (U, V; E)$ are given. Each edge $(u, v) \in E$ represents that resource u can be matched to participant v. All $u \in U$ and $v \in V$ are incident to at least one $e \in E$. For $e = (u, v) \in E$ and $t \in T$, if resource u is matched to participant v at time t, the platformer receives reward $w_{et} \in \mathbb{R}$ and the participant v pays the platformer reward $x_{vt} \in \mathbb{R}$. Thus, the platformer receives a total of $w_{et} + x_{vt}$ rewards. For $e = (u, v) \in E$ and $t \in T$, the constant $c_{et} \in \{1, 2, \ldots, t_{\max}\}$ denotes the time duration that must pass for resource u to become available again when u is matched to v at time t, i.e., u will be available at time $t + c_{et}$ if $e = (u, v)$ is taken at time t. The case $c_{et} = |T|$ for all $t \in T$ and $e \in E$ corresponds to the problem where the resources are non-reusable. Here, w_{et} and c_{et} are given constants while x_{vt} is a variable controlled by the platformer. We also emphasize that w_{et} and x_{vt} can take a negative value.

System Procedure. In the above setting, we consider the following system procedure: (I) The platformer determines the value of x_{vt} for each $v \in V$ and $t \in T$. Then, repeat the following (II) and (III) for each $t \in T$. (II) Participant v appears and accepts the reword/cost with probability $p_{vt}(x_{vt})$ or any of the participant $v \in V$ appears but rejects the reword/cost with probability $1 - \sum_{v \in V} p_{vt}(x_{vt})$. Here, $p_{vt} : \mathbb{R} \to (0, r_{vt})$ is a given monotonically decreasing function and $r_{vt} \in \mathbb{R}_{>0}$ is a given constant such that $\sum_{v \in V} r_{vt} = 1$. (III) If participant $v \in V$ arrives, the platformer chooses either (a) to assign one resource u from U to the participant and get the reward of $w_{et} + x_{vt}$, where $e = (u, v)$, or (b) not to assign any resource. If the platformer chooses (a), allocated resource u is removed from graph G for c_{et} periods.

Platformer's Goal. The goal of the platformer is to maximize the total rewards by deciding x_{vt} and matching strategy appropriately. Here, too large x_{vt} leads to a decrease in the total reward due to a decrease in the arrival probabilities of participants, while too small x_{vt} also leads to a decrease in the total reward due to a decrease in the reward for each match. The platformer needs to make decisions while considering such trade-offs.

3.2 Applications

Crowdsourcing Platform. In the crowdsourcing platform, the platformer needs to assign their tasks to arriving workers in real-time [Ho and Vaughan, 2012]. We start with the set of tasks U and the set of worker groups V, where worker groups are differentiated by skills and attributes. The platformer can determine wage $x_{vt} (\leq 0)$ for each worker group $v \in V$ and each time $t \in T$. Here, x_{vt} is negative because it is the price the platformer pays for workers. Then, at each time $t \in T$, a worker in group $v \in V$ arrives with probability $p_{vt}(x_{vt}) = r_{vt} s_{vt}(x_{vt})$; worker v appears with probability r_{vt} and accepts the wage x_{vt} with probability $s_{vt}(x_{vt})$. The crowdsourcing platformer assigns task $u \in U$ to the accepting worker v and takes the rewards $w_{et} + x_{vt}$, where $e = (u, v)$. Here, $w_{et} (\geq 0)$ is the reward paid by the task-holder to the platformer when task u is solved by worker v. Reward w_{et} is set based on the skills and performance of each worker group. Usually, $w_{e1} = w_{e2} = \cdots = w_{e|T|}$. Once the task u is solved, there is no need to solve it again, so node u is permanently removed after it is assigned. Therefore, $c_{et} = |T|$ for all $e \in E$ and $t \in T$.

Ride-sharing Platform. In the ride-sharing platform, the platformer needs to assign taxis to arriving requesters in real-time [Dickerson *et al.*, 2018]. There are multiple taxis, U, and multiple requester groups, V, where each group $v \in V$ is defined by the origin/destination areas of the request. We consider every taxi $u \in U$ starts and ends at the same location (docking position) for all trips. On receiving a request, the taxi leaves from its docking position to the pick-up point, executes the trip, and returns to its docking position. The platformer can determine fare $x_{vt} (\geq 0)$ for each group $v \in V$ and each time $t \in T$. Then, at each time $t \in T$, a requester in group $v \in V$ arrives with probability $p_{vt}(x_{vt}) = r_{vt} s_{vt}(x_{vt})$; requester v appears with probability r_{vt} and accepts the fare x_{vt} with probability $s_{vt}(x_{vt})$. The platformer assigns taxi $u \in U$ to the accepting requester v and takes the reward $w_{et} + x_{vt}$, where $e = (u, v)$ and $w_{et} (\leq 0)$ is the total cost (e.g., gasoline cost) of allocating taxi u to the requester v at time t. If taxi u is assigned to some v at time t, it becomes unavailable for c_{et} rounds.

Other Applications. Our matching procedure is applicable to the following applications in some situations: Optimization of worker wages and task allocation in crowd-sensing [Pu *et al.*, 2017]; Optimization of parking fees and parking lot allocation in city parking [Meir *et al.*, 2013]; Optimization of usage fees and allocation of virtual machines in cloud computing [Du *et al.*, 2019].

3.3 Optimization Problem

We consider an optimization problem to maximize the total rewards of the platformer in the matching procedure in Section 3.1. We propose *Online Matching with Controllable Rewards and Arrival probabilities* (OM-CRA):

$$\text{(OM-CRA)} \qquad \max_{\boldsymbol{x} \in \mathbb{R}^{V \times T}, \pi \in \Pi} \mathbb{E}_{\boldsymbol{\xi} \sim D(\boldsymbol{x})}[f(\pi, \boldsymbol{x}, \boldsymbol{\xi})].$$

Here, $\boldsymbol{\xi} \in \{V \cup \{\perp\}\}^T$ is a random variable, where $\xi_t = v$ means that participant v arrives at time t, and $\xi_t = \perp$ means no participants arrive at time t. $D(\boldsymbol{x})$ is a probability distribution for $\boldsymbol{\xi} \in \{V \cup \{\perp\}\}^T$; The probability mass function is $\Pr(\boldsymbol{\xi} \mid \boldsymbol{x}) = \prod_{t \in T} \Pr(\xi_t \mid \boldsymbol{x})$, where $\Pr(\xi_t = v \mid \boldsymbol{x}) = p_{vt}(x_{vt})$ and $\Pr(\xi_t = \perp \mid \boldsymbol{x}) = 1 - \sum_{v \in V} p_{vt}(x_{vt})$. Variable π represents a matching strategy in (II) and (III) of the matching procedure in Section 3.1 and Π is the set of all strategies. A matching strategy specifies, upon arrival of vertex $v \in V$, whether to match it, and if so, which $u \in U$ to match it to. Function $f(\pi, \boldsymbol{x}, \boldsymbol{\xi})$ is the expected total reward obtained from performing the matching procedure given $(\pi, \boldsymbol{x}, \boldsymbol{\xi})$.

3.4 Assumption on p_{vt}

We assume the following throughout this paper.

[2] The variable t does not represent an absolute time, but is used to indicate the relative order of appearance of participants.

Assumption 1. *The function $p_{vt} : \mathbb{R} \to (0, r_{vt})$ is monotonically decreasing, differentiable, and bijective. Moreover, $-p'_{vt}(x)/p_{vt}(x)$ is monotonically non-decreasing with respect to x.*

Assumption 1 is not so restrictive; many functions used in real applications satisfy Assumption 1. For example, $p_{vt}(x) = r_{vt}(1 - F(x))$ satisfies Assumption 1 when $F(x)$ is a Gauss error function or a logistic function.[3]

4 Proposed Method

Finding the optimal solution for OM-CRA is difficult for two main reasons: (i) We need to simultaneously optimize the matching strategy π and variable $\boldsymbol{x}$ (even finding just the optimal matching strategy is hard [Manshadi *et al.*, 2012]). (ii) It takes a large amount of time to compute the exact objective value or its approximation with high accuracy. This is because the objective value is the expected total rewards for the random variable $\boldsymbol{\xi}$ with $|V+1|^{|T|}$ realizations. Our solution is to propose a $1/2$-approximation algorithm for OM-CRA.

4.1 Approximation of Objective Function

We consider the approximation of the objective function of OM-CRA. First, we introduce the following problem for a given $\boldsymbol{x}$, and denote the optimal value by $\hat{f}(\boldsymbol{x})$:

$$\begin{aligned} \max_{\boldsymbol{z}\in[0,1]^{E\times T}} \quad & \sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})z_{et} \qquad (1)\\ \text{s.t.} \quad & \sum_{e\in\delta(v)} z_{et} \le p_{vt}(x_{vt}), \quad \forall v\in V,\ \forall t\in T,\\ & \sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'\le c_{et'}} z_{et'} \le 1, \quad \forall u\in U,\ \forall t\in T, \end{aligned}$$

where $\delta(v)$ denotes the set of edges incident to node v.

Here, for given $\boldsymbol{x}$, we consider the following matching strategy $\pi^{\mathrm{ADAP}(1/2)}(\boldsymbol{x})$ [Dickerson *et al.*, 2018]:

1. Let $\boldsymbol{z}^*(\boldsymbol{x})$ be the optimal solution of the problem (1).
2. When vertex v arrives at time t, choose an edge, $e = (u,v) \in E_{vt}$, with probability $\frac{z^*_{et}(\boldsymbol{x})}{p_{vt}(x_{vt})}\frac{1}{2\beta_{et}}$ or reject v with probability $1 - \sum_{e\in E_{vt}} \frac{z^*_{et}(\boldsymbol{x})}{p_{vt}(x_{vt})}\frac{1}{2\beta_{et}}$,

where $E_{vt} := \{(u,v) \in E \mid \text{resource } u \text{ is available at time } t\}$; β_{et} is the probability that edge e is available at time t. We can estimate β_{et} for each $e \in E$ and $t \in T$ with arbitrarily small error by simulating the matching strategy up to $t-1$ [Dickerson *et al.*, 2018].

Then, the following theorem holds by the results of [Dickerson *et al.*, 2018; Alaei *et al.*, 2012].

[3]When $F(x)$ is a twice-differentiable distribution function and $-G'(x)/G(x)$ is monotonically non-decreasing for $G(x) := 1 - F(x)$, F is called a *Monotone Hazard Rate* function [Barlow *et al.*, 1963]. This type of function is frequently used to model the relationship between the offered prices and the acceptance probabilities in the dynamic pricing literature [Babaioff *et al.*, 2015; Tong *et al.*, 2018].

Theorem 1. *For any $\boldsymbol{x}$, the following holds:*

$$\begin{aligned} \frac{1}{2}\hat{f}(\boldsymbol{x}) &\le \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi^{\mathrm{ADAP}(1/2)}(\boldsymbol{x}),\boldsymbol{x},\boldsymbol{\xi})]\\ &\le \max_{\pi\in\Pi}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})] \le \hat{f}(\boldsymbol{x}). \end{aligned}$$

We obtain the following problem by replacing $\max_{\pi\in\Pi}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})]$ with $\hat{f}(\boldsymbol{x})$ for OM-CRA:

$$\begin{aligned} \text{(PA)} \quad \max_{\substack{\boldsymbol{x}\in\mathbb{R}^{V\times T},\\ \boldsymbol{z}\in[0,1]^{E\times T}}} \quad & \sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})z_{et}\\ \text{s.t.} \quad & \sum_{e\in\delta(v)} z_{et} \le p_{vt}(x_{vt}), \quad \forall v\in V,\ \forall t\in T,\\ & \sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'\le c_{et'}} z_{et'} \le 1, \quad \forall u\in U,\ \forall t\in T. \end{aligned}$$

Here, the following lemma and theorem hold:

Lemma 2. *(PA) has an optimal solution under Assumption 1.*

Theorem 3. *Suppose that Assumption 1 holds. Let $\hat{\boldsymbol{x}}$ be an optimal solution for (PA). Then, $(\hat{\boldsymbol{x}}, \pi^{\mathrm{ADAP}(1/2)}(\hat{\boldsymbol{x}}))$ is a $1/2$-approximation solution for OM-CRA.*

From Theorem 3, solving (PA) yields a $1/2$-approximation solution to OM-CRA.

4.2 Reduce (PA) to Convex Optimization Problem

Although (PA) is a non-convex optimization problem with non-convex functions $p_{vt}(x_{vt})$, we can reduce (PA) to a convex optimization problem under Assumption 1. First, we consider the following problem, which is obtained by eliminating $\boldsymbol{x}$ from (PA) by using $x_{vt} := p^{-1}_{vt}(\sum_{e\in\delta(v)} z_{et})$:

$$\begin{aligned} &\text{(CP)}\\ &\min_{\boldsymbol{z}\in[0,1]^{E\times T}} \sum_{t\in T}\Big(\sum_{v\in V} -p^{-1}_{vt}\Big(\sum_{e\in\delta(v)} z_{et}\Big)\sum_{e\in\delta(v)} z_{et} - \sum_{e\in E} w_{et}z_{et}\Big)\\ &\quad \text{s.t.} \quad \sum_{e\in\delta(v)} z_{et} \in S_{vt}, \quad \forall v\in V,\ \forall t\in T,\\ &\qquad\qquad \sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'\le c_{et'}} z_{et'} \le 1, \quad \forall u\in U,\ \forall t\in T. \end{aligned}$$

Here, S_{vt} is the range of function p_{vt}. Then, we show the following proposition.

Proposition 4. *Let Assumption 1 hold. Let the optimal solution of (CP) be $\boldsymbol{z}^*$ and $x^*_{vt} := p^{-1}_{vt}(\sum_{e\in\delta(v)} z^*_{et})$ for all $v \in V$ and $t \in T$. Then, $(\boldsymbol{x}^*, \boldsymbol{z}^*)$ is an optimal solution for (PA).*

From Proposition 4, we can obtain an optimal solution for (PA) by solving (CP). Moreover, the following lemma holds by the results of [Hikima *et al.*, 2021].

Lemma 5. *Let Assumption 1 hold. Then, $-p^{-1}_{vt}(z)z$ is convex with respect to z for all $v \in V$ and $t \in T$. Moreover, the objective function of (CP) is convex.*

From Lemma 5, (CP) is a convex optimization problem.

4.3 PDHG Method for (CP)

We can solve problem (CP) efficiently by applying the Primal-Dual Hybrid Gradient (PDHG) method [Chambolle and Pock, 2011; Goldstein *et al.*, 2015]. The PDHG method was developed for convex-concave saddle-point problems and approaches the optimal solution by alternately updating primal and dual variables. In this section, we reduce (CP) to a convex-concave saddle-point problem and solve the problem by the PDHG method.

Here, (CP) can be rewritten as the following problem:

$$(\mathrm{CP}') \min_{\boldsymbol{z}\in[0,1]^{E\times T}} \sum_{t\in T}\sum_{v\in V} f_{vt}\big(\mathbf{1}^\top \boldsymbol{z}_{vt}\big) - \boldsymbol{w}^\top \boldsymbol{z}, \text{ s.t. } \mathcal{A}(\boldsymbol{z}) \le \mathbf{1},$$

$$\text{where} \quad f_{vt}(z) := \begin{cases} -p_{vt}^{-1}(z)z, & \text{if } z \in S_{vt}, \\ \infty, & \text{otherwise.} \end{cases}$$

Symbol $\mathbf{1}$ denotes the all-one vector. Variable $\boldsymbol{z}_{vt}$ denotes $\boldsymbol{z}_{vt} := (z_{et})_{e\in\delta(v)} \in \mathbb{R}^{\delta(v)}$. Vector $\boldsymbol{w}$ denotes $\boldsymbol{w} := (w_{et})_{(e,t)\in E\times T} \in \mathbb{R}^{E\times T}$. $\mathcal{A} \in \mathbb{R}^{E\times T} \to \mathbb{R}^{U\times T}$ is a linear map according to the second constraint of problem (CP), that is, $\mathcal{A}(\boldsymbol{z}) = (\sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'\le c_{et'}} z_{et'})_{(u,t)\in U\times T}$.

The problem (CP') is equivalent to the following saddle-point problem:

$$\min_{\boldsymbol{z}\in[0,1]^{E\times T}} \max_{\boldsymbol{\lambda}\in\mathbb{R}^{U\times T}_{\ge 0}} \mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda}), \quad \text{where} \tag{2}$$

$$\mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda}) = \sum_{v\in V}\sum_{t\in T} f_{vt}\left(\mathbf{1}^\top \boldsymbol{z}_{vt}\right) - \boldsymbol{w}^\top\boldsymbol{z} + \boldsymbol{\lambda}^\top\left(\mathcal{A}(\boldsymbol{z}) - \mathbf{1}\right). \tag{3}$$

The function $\mathcal{L}$ is called the Lagrange function. Since this problem is convex-concave from Lemma 5 and the definition of f_{vt}, we can solve it by the PDHG method.

In each iteration k of PDHG, the following updates are performed:

$$\begin{cases} \boldsymbol{z}^{k+1} := \displaystyle\arg\min_{\boldsymbol{z}\in[0,1]^{E\times T}} \mathcal{L}(\boldsymbol{z},\boldsymbol{\lambda}^k) + \frac{1}{2\tau_k}\|\boldsymbol{z}-\boldsymbol{z}^k\|^2, & (4) \\ \boldsymbol{\lambda}^{k+1} := \displaystyle\arg\max_{\boldsymbol{\lambda}\in\mathbb{R}^{U\times T}_{\ge 0}} \mathcal{L}(\boldsymbol{z}^{k+1},\boldsymbol{\lambda}) - \frac{1}{2\sigma_k}\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^k\|^2. & (5) \end{cases}$$

Here, $\tau_k, \sigma_k > 0$ are step-size parameters. Eq. (5) can be solved explicitly as $\boldsymbol{\lambda}^{k+1} = (\boldsymbol{\lambda}^k + \sigma_k(\mathcal{A}(\boldsymbol{z}^{k+1}) - \mathbf{1}))_+$. Although (4) requires solving a $(|E||T|)$-dimensional optimization problem, we can solve it quickly by utilizing the problem structure.

The optimization problem in (4) can be decomposed into the following $|\delta(v)|$-dimensional optimization problems for $v \in V$ and $t \in T$:

$$\min_{\boldsymbol{z}_{vt}\in[0,1]^{\delta(v)}} f_{vt}(\mathbf{1}^\top\boldsymbol{z}_{vt}) + \frac{1}{2\tau_k}\|\boldsymbol{z}_{vt} - \boldsymbol{z}_{vt}^k - \tau_k(\boldsymbol{w}_{vt} - \mathcal{A}_{vt}^\top(\boldsymbol{\lambda}^k))\|^2 \tag{6}$$

where $\mathcal{A}_{vt} : \mathbb{R}^{\delta(v)} \to \mathbb{R}^{U\times T}$ is the linear map such that

$$(\mathcal{A}_{vt}(\boldsymbol{z}_{vt}))_{(u,t')} = \begin{cases} z_{et}, & \text{if } e = (u,v),\ 0 \le t'-t \le c_{et} \\ 0, & \text{otherwise} \end{cases}$$

The vector $\boldsymbol{w}_{vt}$ denotes $\boldsymbol{w}_{vt} := (w_{et})_{e\in\delta(v)} \in \mathbb{R}^{\delta(v)}$.

Moreover, the following proposition holds for problem (6).

Proposition 6. *Let $\boldsymbol{a} := \boldsymbol{z}_{vt}^k + \tau_k(\boldsymbol{w}_{vt} - \mathcal{A}_{vt}^\top(\boldsymbol{\lambda}^k))$. We consider the following equation with respect to s:*

$$\sum_{i=1}^{n}\left(a_i - \tau_k f'_{vt}(s)\right) = s. \tag{7}$$

Then, the left-hand side of Eq. (7) *is monotonically decreasing, and* (7) *has a unique solution, s^*. In addition, $\boldsymbol{z}_{vt} = (\boldsymbol{a} - \tau_k f'_{vt}(s^*)\mathbf{1})$ is the optimal solution for the problem* (6).

Algorithm 1 Adaptive PDHG for (CP')

Input: $\boldsymbol{z}^0$, $\boldsymbol{\lambda}^0$, τ_0, σ_0, α_0, η, and c
1: $\tau \leftarrow \tau_0, \sigma \leftarrow \sigma_0, \alpha \leftarrow \alpha_0$
2: **for** $k = 0, 1, \dots$ **:**
3:   **for** $v \in V, t \in T$ **:**
4:     $s^* \leftarrow$ (solution to eq. (7))
5:     $\boldsymbol{z}_{vt}^{k+1} \leftarrow \big(\boldsymbol{z}_{vt}^k - \tau\mathcal{A}_{vt}^\top(\boldsymbol{\lambda}^k) - \tau f'_{vt}(s^*)\mathbf{1}\big)_+$
6:   $\boldsymbol{\lambda}^{k+1} \leftarrow (\boldsymbol{\lambda}^k + \sigma(2\mathcal{A}(\boldsymbol{z}^{k+1}) - \mathcal{A}(\boldsymbol{z}^{k+1}) - \mathbf{1}))_+$
7:   **if** $\frac{c}{2\tau}\|\boldsymbol{z}^{k+1} - \boldsymbol{z}^k\|^2 + \frac{c}{2\sigma}\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\|^2 \le 2(\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k)^\top A(\boldsymbol{z}^{k+1} - \boldsymbol{z}^k)$ **:**
8:     $\tau \leftarrow \frac{\tau}{2}, \sigma \leftarrow \frac{\sigma}{2}$
9:   $\boldsymbol{p} \leftarrow \frac{1}{\tau}(\boldsymbol{z}^k - \boldsymbol{z}^{k+1}) - A^\top(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1})$
10:   $\boldsymbol{d} \leftarrow \frac{1}{\sigma}(\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k+1}) + A(\boldsymbol{z}^k - \boldsymbol{z}^{k+1})$
11:   **if** $2\|\boldsymbol{p}\| < \|\boldsymbol{d}\|$ **:**
12:     $\tau \leftarrow \tau(1-\alpha), \sigma \leftarrow \frac{\sigma}{1-\alpha}, \alpha \leftarrow \alpha\eta$
13:   **else if** $\|\boldsymbol{p}\| > 2\|\boldsymbol{d}\|$ **:**
14:     $\tau \leftarrow \frac{\tau}{1-\alpha}, \sigma \leftarrow \sigma(1-\alpha), \alpha \leftarrow \alpha\eta$

Because the left-hand side of (7) is monotonically decreasing and the right-hand side is monotonically increasing, the bisection method can quickly find s^*. By Proposition 6, the solution for the problem (6) can be easily calculated from s^*.

We now propose Algorithm 1. Algorithm 1 incorporates the decomposition of the problem (6) and the results of Proposition 6 into the method of [Goldstein *et al.*, 2015]. In this algorithm, lines 3–6 update the primal and dual variables by (4) and (5), and lines 7–14 adjusts the step-size, τ and σ.

This algorithm is a specialized version of the method of [Goldstein *et al.*, 2015]; the updates of the primal variables are accelerated by using problem decomposition (6) and Proposition 6.

5 Experiments

We conduct experiments to show that the following hold: (i) our algorithm outputs solutions with higher total rewards than the baselines in each application; (ii) our algorithm outputs the solution in practical time. We performed simulation experiments using real data from crowdsourcing and ride-sharing platforms. Experiments were run on a computer with Xeon Platinum 8168 (4 x 2.7GHz), 1TB of memory, CentOS 7.6. The program codes were implemented in Python 3.6.3. Codes and more details of our experiments can be found in our repository provided in the footnote on the first page.

5.1 Baselines

We compared our method to the following six baselines.

CU-A. We set $\boldsymbol{x}$ by Capped-UCB [Babaioff *et al.*, 2015], which is the pricing strategy for limited supply in a single item market. This method determines the price while estimating the average participation probability for all demanders; here we take it to be a given function. Specifically, for all v and t, let $x_{vt} := \arg\max_x\{(x+\hat{w})\min(|U||T|/\hat{c}, \hat{p}(x))\}$, where $\hat{w}$ and $\hat{c}$ are the averages of w_{et} and c_{et} for all e and t, respectively, and $\hat{p}(x) = \sum_{t\in T}\sum_{v\in V} p_{vt}(x)$. Here,

$(\|U\|, \|V\|, \|T\|)$	Proposed		CU-A		CU-G		BO-A		BO-G		RS-A		RS-G	
	ETR	time	ETR	time	ETR	time	ETR	time	ETR	time	ETR	time	ETR	time
(100,100,100)	**5.68**	628	2.66	84	4.72	0.00129	1.53	1401	2.72	1015	1.55	1057	2.86	1001
(50,100,100)	**3.99**	460	2.00	49	3.43	0.00140	1.18	1211	1.94	1015	1.17	1036	2.05	1001
(150,100,100)	**6.01**	814	2.70	122	5.03	0.00133	1.60	1604	2.92	1018	1.60	1077	3.08	1001
(100,50,100)	**4.79**	330	2.39	48	4.18	0.00111	1.48	1205	2.40	1015	1.46	1039	2.61	1001
(100,150,100)	**5.39**	927	2.44	123	4.50	0.00147	1.44	1617	2.55	1018	1.43	1063	2.72	1001
(100,100,50)	**2.83**	319	1.10	41	2.10	0.00111	0.75	1192	1.39	1009	0.75	1028	1.49	1001
(100,100,150)	**7.49**	924	3.58	130	6.38	0.00143	2.15	1615	3.70	1023	2.16	1075	3.90	1001

Table 1: Results of real dataset simulation for a crowdsourcing platform. The *time* column of each method indicates the computation time (in seconds). The best value of ETR for each experiment is in bold. Each result is the average of 10 experiments.

date	Proposed		CU-A		CU-G		BO-A		BO-G		RS-A		RS-G	
	ETR	time	ETR	time	ETR	time	ETR	time	ETR	time	ETR	time	ETR	time
1/20	**603**	128	202	237	225	0.0113	62	2208	132	1029	60	1175	138	1001
1/24	**875**	126	435	232	482	0.0117	164	2212	235	1032	162	1134	243	1001
2/24	**932**	135	560	230	624	0.0114	173	2194	255	1034	172	1156	264	1001
2/27	**1107**	136	729	235	802	0.0114	252	2217	335	1039	250	1168	344	1002
3/16	**1081**	129	637	231	702	0.0116	248	2228	339	1023	245	1167	348	1001
3/19	**1104**	127	724	241	798	0.0118	245	2237	333	1027	241	1170	344	1001

Table 2: Results of real dataset simulation for a ride-sharing platform. The *time* column of each method indicates the computation time (in seconds). The best value of ETR for each experiment is in bold. Each result is the average of 10 experiments.

$|U||T|/\hat{c}$ approximates the number of times the resources can be used over the entire period, based on the average length of periods that resources are unavailable. If $c_{et} = |T|$ for all $(e, t) \in E \times T$, i.e., the resource is not reusable, then $|U||T|/\hat{c} = |U|$. Function $\hat{p}(x)$ represents the average total number of arriving requesters for x. Then, we adopt the $\frac{1}{2}$-approximation strategy [Dickerson *et al.*, 2018].

CU-G. CU-G adopts a greedy strategy instead of the $\frac{1}{2}$-approximation strategy for CU-A. The greedy strategy takes the edge $\hat{e}(v, t) = \arg\max_e\{w_{et} + x_{vt} \mid e = (u, v) \in E, u \text{ is available}\}$ when the node v arrives at time t.

BO-A. We apply Bayesian optimization [Brochu *et al.*, 2010] to search $\boldsymbol{x}$ while adopting the $\frac{1}{2}$-approximation strategy [Dickerson *et al.*, 2018] as the matching strategy. BO-A first evaluates five random points of $\boldsymbol{x}$. Then, after running the Bayesian optimization for 1000 seconds, it outputs the solution with the highest objective value among the evaluated points. The set to search for $\boldsymbol{x}$ is $[0, 1]^{V \times T}$ in crowd-sourcing platform experiments and $[0, 50]^{V \times T}$ in ride-sharing platform experiments.

BO-G. BO-G adopts a greedy strategy instead of the $\frac{1}{2}$-approximation strategy of BO-A.

RS-A. We apply the random search to variable $\boldsymbol{x}$ while adopting the $\frac{1}{2}$-approximation strategy [Dickerson *et al.*, 2018] as the matching strategy. In each iteration, search points are generated from the set $[0, 1]^{V \times T}$ in crowd-sourcing platform experiments and $[0, 50]^{V \times T}$ in ride-sharing platform experiments. RS-A outputs the solution with the highest objective value among the points evaluated in 1000 seconds.

RS-G. RS-G adopts the greedy strategy instead of the $\frac{1}{2}$-approximation strategy of RS-A.

5.2 Metric

We use *Expected Total Rewards* (ETR), which is the average of obtained total rewards in 10^3 simulations, as the metric.

5.3 Experiments of Crowd Sourcing Platform

We conduct experiments for a crowdsourcing platform whose problem setting is described in Section 3.2.

Data Set and Parameter Setup. We used an open crowdsourcing dataset [Buckley *et al.*, 2010]. The dataset contains records of workers' judgments on the task of checking the relevance of a given topic and a web page. We set the inputs U, V, E, w_{et}, and p_{vt} from the data for each experiment, based on [Hikima *et al.*, 2021; Ho and Vaughan, 2012].

Experimental Results. Table 1 shows the results of the simulation experiments with different parameter values. The proposed method outperforms all baselines in terms of ETR. This is because the baselines are not able to find proper $\boldsymbol{x}$; (i) **CU-A** and **CU-G** propose an average price for all $v \in V$, which is inappropriate for many $v \in V$; they offer the same price to workers with different skill levels. (ii) **BO-A**, **BO-G**, **RS-A**, and **RS-G** cannot fully explore $\boldsymbol{x}$ since (a) it takes a large amount of time to evaluate the objective value and (b) the dimension of $\boldsymbol{x}$ is large. Moreover, the computation time of the proposed method is short enough for practical use.

5.4 Experiments of Ride-sharing Platform

We conduct experiments on a ride-sharing platform whose setting is described in Section 3.2.

Data Set and Parameter Setup. We used ride data of yellow taxis in Manhattan in New York[4]. We perform sim-

[4]https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

ulations using data from 10:00 to 20:00 on weekends and weekdays for randomly chosen weeks from January to March 2019. We set the inputs U, V, E, w_{et}, and p_{vt} from the data for each experiment, based on [Dickerson *et al.*, 2018; Hikima *et al.*, 2021].

Experimental Results. Table 2 shows the results of the simulation experiments with different dates. Regardless of the date, the proposed method outperforms all baselines in terms of ETR. Moreover, compared to the results of crowdsourcing experiments, the differences between the proposed method and the baselines are larger. This occurs for the following reasons; (i) **CU-A** and **CU-G** output further inappropriate $\boldsymbol{x}$ when the resource is reusable. Capped-UCB, which is used in CU-G and CU-A, is a method to determine $\boldsymbol{x}$ based on *the number of times the resources can be used*, N_R. When the resources are not reusable, N_R is equal to the total number of resources. However, when the resources are reusable, N_R depends on the input and the matching strategy, so we cannot calculate N_R exactly and must approximate it. The inaccuracy of the approximation deteriorates the quality of the output of Capped-UCB. (ii) **BO-A, BO-G, RS-A**, and **RS-G** become more inefficient because $|V|$ is larger than that of crowdsourcing, and then the dimension of $\boldsymbol{x}$ becomes relatively large. In contrast, the proposed method can stably determine appropriate prices even in the large-scale problem with reusable resources. In addition, the proposed method outputs solutions in practical time.

6 Conclusion

We formulated a novel optimization problem, OM-CRA, to simultaneously determine the matching strategy and the trade-off between rewards and arrival probabilities. It is useful in obtaining high profits in various applications. We proposed a fast $1/2$-approximation algorithm for OM-CRA. Simulation experiments on real data from two applications, crowdsourcing and ride-sharing platforms, confirmed the effectiveness of the proposed algorithm.

7 Proof

7.1 Proof of Theorem 1

First, based on [Alaei *et al.*, 2012, Lemma 3.1], we show that $\max_{\pi\in\Pi}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})] \le \hat{f}(\boldsymbol{x})$. For each $\boldsymbol{\xi}$, we consider the following problem:

$$\begin{aligned}(\mathrm{P}(\boldsymbol{\xi}))\quad \max_{\boldsymbol{z}\in\{0,1\}^{E\times T}} \quad & \sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})z_{et}\\ \text{s.t.}\quad & \sum_{e\in\delta(v)} z_{et}\le \xi_{vt},\quad \forall v\in V,\ \forall t\in T,\\ & \sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'<c_{et'}} z_{et'}\le 1, \forall u\in U,\ \forall t\in T,\end{aligned}$$

where ξ_{vt} is a constant that is 1 if $\xi_t = v$ and 0 otherwise. In other words, $\xi_{vt}=1$ if v appears at time t and 0 otherwise. Here, let $\boldsymbol{z}^*(\boldsymbol{\xi})$ be the optimal solution for $(\mathrm{P}(\boldsymbol{\xi}))$. Since the optimal value of $\mathrm{P}(\boldsymbol{\xi})$ is the maximum profit from performing the matching procedure for $\boldsymbol{\xi}$,

$$\max_{\pi\in\Pi} f(\pi,\boldsymbol{x},\boldsymbol{\xi}) \le \sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})z^*_{et}(\boldsymbol{\xi}).$$

Therefore, the following holds:

$$\begin{aligned}&\max_{\pi\in\Pi}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})] \le \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[\max_{\pi\in\Pi} f(\pi,\boldsymbol{x},\boldsymbol{\xi})]\\ &\le \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[\sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})z^*_{et}(\boldsymbol{\xi})]\\ &= \sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[z^*_{et}(\boldsymbol{\xi})]. \end{aligned}\tag{8}$$

From the first constraints of $(\mathrm{P}(\boldsymbol{\xi}))$, $\sum_{e\in\delta(v)} z^*_{et}(\boldsymbol{\xi}) \le \xi_{vt}$ for all $v\in V$, $t\in T$. Since it holds for all $\boldsymbol{\xi}$, the following holds for all $v\in V$ and $t\in T$:

$$\begin{aligned}&\sum_{e\in\delta(v)}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[z^*_{et}(\boldsymbol{\xi})] = \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[\sum_{e\in\delta(v)} z^*_{et}(\boldsymbol{\xi})]\\ &\le \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[\xi_{vt}] = p_{vt}(x_{vt}).\end{aligned}\tag{9}$$

From the second constraints of $(\mathrm{P}(\boldsymbol{\xi}))$, we have $\sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'<c_{et'}} z^*_{et'}(\boldsymbol{\xi}) \le 1$ for all $u\in U$, $t\in T$. Since it holds for all $\boldsymbol{\xi}$, the following holds for all $u\in U$ and $t\in T$:

$$\begin{aligned}&\sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'<c_{et'}}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[z^*_{et'}(\boldsymbol{\xi})]\\ &= \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}\left[\sum_{e\in\delta(u)}\sum_{t':\,0\le t-t'<c_{et'}} z^*_{et'}(\boldsymbol{\xi})\right] \le 1.\end{aligned}\tag{10}$$

Since $z^*_{et}(\boldsymbol{\xi})\in\{0,1\}$ for all $e\in E$, $t\in T$, and $\boldsymbol{\xi}$, the following holds for all $e\in E$, $t\in T$, and $\boldsymbol{\xi}$:

$$\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[z^*_{et}(\boldsymbol{\xi})]\in[0,1].\tag{11}$$

Then, from (9), (10) and (11), $\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[\boldsymbol{z}^*(\boldsymbol{\xi})]$ is a feasible solution for problem (1). Therefore, $\sum_{t\in T}\sum_{e=(u,v)\in E}(x_{vt}+w_{et})\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[z^*_{et}(\boldsymbol{\xi})] \le \hat{f}(\boldsymbol{x})$. Then, from (8), $\max_{\pi\in\Pi}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})]\le\hat{f}(\boldsymbol{x})$.

We show $\frac{1}{2}\hat{f}(\boldsymbol{x}) \le \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi^{\mathrm{ADAP}(1/2)}(\boldsymbol{x}),\boldsymbol{x},\boldsymbol{\xi})]$, based on [Dickerson *et al.*, 2018, Section 3]. About LP(1) of [Dickerson *et al.*, 2018], let $p_{vt} := p_{vt}(x_{vt})$, $w_e := x_{vt}+w_{et}$, and let $\Pr[C_e > t-t']$ be 1 if $c_{et'} > t-t'$ and 0 otherwise. Then, LP(1) of [Dickerson *et al.*, 2018] corresponds to problem (1) in our paper. Therefore, from [Dickerson *et al.*, 2018, Lemma 2, Section 3], $\frac{1}{2}\hat{f}(\boldsymbol{x}) \le \mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi^{\mathrm{ADAP}(1/2)}(\boldsymbol{x}),\boldsymbol{x},\boldsymbol{\xi})]$. Here, although w_e does not have t subscript in LP(1) of [Dickerson *et al.*, 2018], Lemma 2 of [Dickerson *et al.*, 2018] holds. □

7.2 Proof of Lemma 2

The proof can be found in the supplementary material in our repository provided in the footnote on the first page.

7.3 Proof of Theorem 3

From Theorem 1, $\mathbb{E}_{\boldsymbol{\xi}\sim D(\hat{\boldsymbol{x}})}[f(\pi^{\mathrm{ADAP}}(\hat{\boldsymbol{x}})),\hat{\boldsymbol{x}},\boldsymbol{\xi})] \ge \frac{1}{2}\hat{f}(\hat{\boldsymbol{x}})$. Since problem (PA) is equivalent to $\max_{\boldsymbol{x}\in\mathbb{R}^{V\times T}}\hat{f}(\boldsymbol{x})$, for any $\boldsymbol{x}\in\mathbb{R}^{V\times T}$, $\frac{1}{2}\hat{f}(\hat{\boldsymbol{x}}) \ge \frac{1}{2}\hat{f}(\boldsymbol{x})$. Here, for any $(\boldsymbol{x},\pi)\in\mathbb{R}^{V\times T}\times\Pi$, Theorem 1 shows $\frac{1}{2}\hat{f}(\boldsymbol{x}) \ge \frac{1}{2}\max_{\pi'\in\Pi}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi',\boldsymbol{x},\boldsymbol{\xi})] \ge \frac{1}{2}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})]$. Then, we have $\mathbb{E}_{\boldsymbol{\xi}\sim D(\hat{\boldsymbol{x}})}[f(\pi^{\mathrm{ADAP}(1/2)}(\hat{\boldsymbol{x}})),\hat{\boldsymbol{x}},\boldsymbol{\xi})] \ge \frac{1}{2}\mathbb{E}_{\boldsymbol{\xi}\sim D(\boldsymbol{x})}[f(\pi,\boldsymbol{x},\boldsymbol{\xi})]$ for $(\boldsymbol{x},\pi)\in\mathbb{R}^{V\times T}\times\Pi$. This means $(\hat{\boldsymbol{x}},\pi^{\mathrm{ADAP}(1/2)}(\hat{\boldsymbol{x}}))$ is $\frac{1}{2}$-approximation solution for OM-CRA. □

7.4 Proof of Proposition 4

Let $(\hat{\boldsymbol{x}},\ \hat{\boldsymbol{z}})$ be an optimal solution for (PA). First, we show $\sum_{e\in\delta(v)} \hat{z}_{et} \neq 0$ for all $v \in V$ and $t \in T$. We assume there are $\grave{v} \in V$ and $\grave{t} \in T$ satisfying $\sum_{e\in\delta(\grave{v})} \hat{z}_{e\grave{t}} = 0$ to obtain a contradiction. We pick an arbitrary vertex $\grave{e} \in \delta(\grave{v})$. Then, there exists x_M satisfying $x_M + w_{\grave{e}\grave{t}} > \sum_{(e=(u,v),t)\in E\times T\setminus(\grave{e},\grave{t})}\{\hat{x}_{vt} + w_{et}\}$. Here, we let $\epsilon := p_{\grave{v}\grave{t}}(x_M)$ and show that replacing $\hat{x}_{\grave{v}\grave{t}}$ with x_M, $\hat{z}_{\grave{e}\grave{t}}$ $(= 0)$ with ϵ, and $\hat{z}_{et}$ with $\max\{0, \hat{z}_{et} - \epsilon\}$ for all $(e, t) \in E \times T \setminus (\grave{e}, \grave{t})$ increases the objective value of (PA) from $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ without impairing feasibility. The objective value of (PA) increases because $(x_M + w_{\grave{e}\grave{t}})\epsilon - \sum_{(e=(u,v),t)\in E\times T\setminus(\grave{e},\grave{t}):\hat{x}_{vt}+w_{et}\geq 0}(\hat{x}_{vt} + w_{et}) \min\{\hat{z}_{et}, \epsilon\} \geq (x_M + w_{\grave{e}\grave{t}} - \sum_{(e=(u,v),t)\in E\times T\setminus(\grave{e},\grave{t}):\hat{x}_{vt}+w_{et}\geq 0}(\hat{x}_{vt} + w_{et}))\epsilon > 0$. Here, for all $e = (u, v) \in E$ and $t \in T$, if $\hat{x}_{vt} + w_{et} < 0$, then $\hat{z}_{et} = 0$; if $\hat{z}_{et} > 0$ and $(\hat{x}_{vt} + w_{et}) < 0$, we can increase the objective value of (PA) without impairing feasibility by setting $\hat{z}_{et} := 0$ and this contradicts the assumption that $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ is the optimal solution of (PA). The first constraint of (PA) is satisfied because $\epsilon \leq p_{\grave{v}\grave{t}}(x_M)$ and $\sum_{e\in\delta(v)} \max\{0, \hat{z}_{et} - \epsilon\} \leq \sum_{e\in\delta(v)} \hat{z}_{et} \leq p_{vt}(\hat{x}_{vt})$ for all $(v, t) \in \{V \times T\} \setminus \{(\grave{v}, \grave{t})\}$. Here, let $\grave{u} \in U$ be the node incident to $\grave{e}$ and $J(\grave{e}, \grave{t}) := \{t' \in T \mid 0 \leq t' - \grave{t} < c_{\grave{e}\grave{t}}\}$. Then, the second constraint holds for all $(u, t) \in \{U\times T\}\setminus\{\grave{u}\}\times J(\grave{e}, \grave{t})$ since the left-hand side of the inequality only decreases. For all $(u, t) \in \{\grave{u}\} \times J(\grave{e}, \grave{t})$, the second constraint holds because $\sum_{(e,t)\in\{(e,t)\in E\times T|e\in\delta(u),0\leq t-t'<c_{et'}\}\setminus(\grave{e},\grave{t})} \max\{0, \hat{z}_{et'} - \epsilon\} + \epsilon \leq 1$. Therefore, replacing $\hat{x}_{\grave{v}\grave{t}}$ with x_M, $\hat{z}_{\grave{e}\grave{t}}$ $(= 0)$ with ϵ, and $\hat{z}_{et}$ with $\max\{0, \hat{z}_{et} - \epsilon\}$ for all $(e, t) \in E \times T \setminus (\grave{e}, \grave{t})$ increases the objective value of (PA) from $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ without impairing feasibility. This contradicts the optimality of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ for (PA). Then, $\sum_{e\in\delta(v)} \hat{z}_{et} \neq 0$ for all $v \in V$, $t \in T$.

Then, we show that there exists an optimal solution $(\tilde{\boldsymbol{x}},\ \tilde{\boldsymbol{z}})$ for (PA) that satisfies $p_{vt}(\tilde{x}_{vt}) = \sum_{e\in\delta(v)} \tilde{z}_{et}$ for all $v \in V$, $t \in T$. Let $(\hat{\boldsymbol{x}},\ \hat{\boldsymbol{z}})$ be an optimal solution for (PA). Then, from constraints of (PA), $\sum_{e\in\delta(v)} \hat{z}_{et} \leq p_{vt}(\hat{x}_{vt})$ for all $v \in V$, $t \in T$. Suppose that there exists (v, t) satisfying $\sum_{e\in\delta(v)} \hat{z}_{et} < p_{vt}(\hat{x}_{vt})$ and let $\Omega := \{(v, t) \mid \sum_{e\in\delta(v)} \hat{z}_{et} < p_{vt}(\hat{x}_{vt})\}$. Since $\sum_{e\in\delta(v)} \hat{z}_{et} > 0$ and $\lim_{x\to\infty} p_{vt}(x) = 0$ from Assumption 1, there exists a positive scalar d_{vt} that satisfies $\sum_{e\in\delta(v)} \hat{z}_{et} = p_{vt}(\hat{x}_{vt} + d_{vt})$ for all $(v, t) \in \Omega$. Let $\tilde{x}_{vt}$ be $\hat{x}_{vt} + d_{vt}$ for all $(v, t) \in \Omega$, and $\tilde{x}_{vt}$ be $\hat{x}_{vt}$ for all $(v, t) \notin \Omega$, and $\tilde{\boldsymbol{z}}$ be $\hat{\boldsymbol{z}}$. Then, $(\tilde{\boldsymbol{x}},\ \tilde{\boldsymbol{z}})$ is an optimal solution for (PA) because it is a feasible solution and the objective value is greater than or equal to the optimal value from $\hat{\boldsymbol{z}} \geq 0$. Therefore, in (PA), there exists an optimal solution $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{z}})$ satisfying $\sum_{e\in\delta(v)} \tilde{z}_{et} = p_{vt}(\tilde{x}_{vt})$ for all $v \in V$ and $t \in T$. Therefore, (PA) with the constraint "$x_{vt} = p_{vt}^{-1}(\sum_{e\in\delta(v)} z_{et})$ for all $v \in V, t \in T$", that is, (CP) is equivalent to (PA). Then, $(\boldsymbol{x}^*, \boldsymbol{z}^*)$ is an optimal solution for (PA). □

7.5 Proof of Lemma 5

Since $-p'_{vt}(x)/p_{vt}(x)$ is monotonically non-decreasing from Assumption 1, $-p_{vt}(x)/p'_{vt}(x)$ is monotonically non-increasing. Then, $-x - p_{vt}(x)/p'_{vt}(x)$ is monotonically decreasing because $-x$ is monotonically decreasing.

For all $(v, t) \in V \times T$ and given $z \in S_{vt}$, there exists only one x that satisfies $p_{vt}(x) = z$ from Assumption 1. Then, the following equality holds:

$$\begin{aligned}\left(-p_{vt}^{-1}(z)z\right)' &= -p_{vt}^{-1}(z) - \left(p_{vt}^{-1}(z)\right)' z \\ &= -x - p_{vt}(x)/p'_{vt}(x) \end{aligned} \tag{12}$$

Here, we consider $z^1 \in S_{vt}$ and $z^2 \in S_{vt}$ satisfying $z^1 < z^2$. Let x^1 satisfy $p_{vt}(x^1) = z^1$ and x^2 satisfy $p_{vt}(x^2) = z^2$. Then, $x^1 > x^2$ since p_{vt} is monotonically decreasing from Assumption 1. Since (12) holds and $-x - p_{vt}(x)/p'_{vt}(x)$ is monotonically decreasing, $(-p_{vt}^{-1}(z^1)z^1)' = -x^1 - p_{vt}(x^1)/p'_{vt}(x^1) < -x^2 - p_{vt}(x^2)/p'_{vt}(x^2) = (-p_{vt}^{-1}(z^2)z^2)'$. Since $(-p_{vt}^{-1}(z)z)'$ is monotonically increasing, $(-p_{vt}^{-1}(z)z)$ is convex. Moreover, the objective function of (CP) is convex since $-p_{vt}^{-1}\big(\sum_{e\in\delta(v)} z_{et}\big) \sum_{e\in\delta(v)} z_{et}$ is convex with respect to $\boldsymbol{z}$ for all $v \in V$ and $t \in T$ from [Boyd and Vandenberghe, 2004, Section 3.2.2]. □

7.6 Proof of Proposition 6

From Lemma 5, f'_{vt} is monotonically increasing and the left-hand side of equation (7), $\sum_{i\in\delta(v)}(a_i - \tau_k f'_{vt}(s))$, is monotonically decreasing. In addition, $\lim_{s\to 0} f'_{vt}(s) = \lim_{x\to\infty} -x - p_{vt}(x)/p'_{vt}(x) = -\infty$ and $\lim_{s\to r_{vt}} f'_{vt}(s) = \lim_{x\to -\infty} -x - p_{vt}(x)/p'_{vt}(x) = \infty$ since $p_{vt}(x)/p'_{vt}(x)$ is monotonically non-decreasing from Assumption 1. Therefore, $\lim_{s\to 0}\sum_{i\in\delta(v)}(a_i - \tau_k f'_{vt}(s)) = \infty$ and $\lim_{s\to r_{vt}}\sum_{i\in\delta(v)}(a_i - \tau_k f'_{vt}(s)) = -\infty$. Hence, equation (7) has a unique solution $s^* \in (0, r_{vt})$.

Next we show $\boldsymbol{z}_{vt} = \boldsymbol{a} - \tau_k f'_{vt}(s^*)\mathbf{1}$ is the optimal solution for the problem (6). Let $\boldsymbol{y} := \boldsymbol{z}_{vt}$. Problem (6) is equivalent to the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{y}} \ & f_{vt}(s) + \frac{1}{2\tau_k}\|\boldsymbol{y} - \boldsymbol{a}\|^2 \\ \text{s.t.} \ & \mathbf{1}^\top \boldsymbol{y} = s. \end{aligned} \tag{13}$$

Then, we can define its Lagrangian function as follows:

$$L(s, \boldsymbol{y}, \lambda, \boldsymbol{\mu}) := f_{vt}(s) + \tfrac{1}{2\tau_k}\|\boldsymbol{y} - \boldsymbol{a}\|^2 + \lambda\cdot(\mathbf{1}^\top\boldsymbol{y} - s).$$

The KKT conditions are $f'_{vt}(s) - \lambda = 0$, $\frac{y_i - a_i}{\tau_k} + \lambda = 0$ for all $i \in \delta(v)$, and $\mathbf{1}^\top\boldsymbol{y} - s = 0$. Here, let $\lambda^* := f'_{vt}(s^*)$ and $y_i^* := a_i - \tau_k f'_{vt}(s^*)$. Then, $(s^*, \lambda^*, \boldsymbol{y}^*)$ satisfy the KKT conditions. Since problem (13) satisfies the Slater's constraint qualification, $\boldsymbol{y}^*$ is an optimal solution for problem (13) from [Boyd and Vandenberghe, 2004, Section 5.2.3]. Then, $\boldsymbol{z}_{vt}$ satisfying $\boldsymbol{z}_{vt} = \boldsymbol{y}^* = \boldsymbol{a} - \tau_k f'_{vt}(s^*)\mathbf{1}$ is an optimal solution for problem (6).

References

[Alaei *et al.*, 2012] Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online prophet-inequality matching with applications to ad allocation. In *EC*, pages 18–35, 2012.

[Babaioff *et al.*, 2015] Moshe Babaioff, Shaddin Dughmi, Robert Kleinberg, and Aleksandrs Slivkins. Dynamic pricing with limited supply. *ACM Transactions on Economics and Computation*, 3(1):1–26, 2015.

[Barlow *et al.*, 1963] Richard E Barlow, Albert W Marshall, and Frank Proschan. Properties of probability distributions with monotone hazard rate. *The Annals of Mathematical Statistics*, 34(2):375–389, 1963.

[Benzi *et al.*, 2005] Michele Benzi, Gene H Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta numerica*, 14:1–137, 2005.

[Bergstra and Bengio, 2012] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[Brochu *et al.*, 2010] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[Buckley *et al.*, 2010] Chris Buckley, Matthew Lease, and Mark D. Smucker. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *TREC*, 2010.

[Chambolle and Pock, 2011] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[Devanur *et al.*, 2011] Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *EC*, pages 29–38, 2011.

[Dickerson *et al.*, 2018] John Dickerson, Karthik Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *AAAI*, pages 1007–1014, 2018.

[Du *et al.*, 2019] Bingqian Du, Chuan Wu, and Zhiyi Huang. Learning resource allocation and pricing for cloud profit maximization. In *AAAI*, pages 7570–7577, 2019.

[Feldman *et al.*, 2009] Jon Feldman, Aranyak Mehta, Vahab Mirrokni, and Shan Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *FOCS*, pages 117–126, 2009.

[Goldstein *et al.*, 2015] Tom Goldstein, Min Li, and Xiaoming Yuan. Adaptive primal-dual splitting methods for statistical learning and image processing. In *NeurIPS*, pages 2089–2097, 2015.

[Haeupler *et al.*, 2011] Bernhard Haeupler, Vahab S Mirrokni, and Morteza Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *Wine*, pages 170–181, 2011.

[Hellemo *et al.*, 2018] Lars Hellemo, Paul I Barton, and Asgeir Tomasgard. Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15(3):369–395, 2018.

[Hikima *et al.*, 2021] Yuya Hikima, Yasunori Akagi, Hideaki Kim, Masahiro Kohjima, Takeshi Kurashima, and Hiroyuki Toda. Integrated optimization of bipartite matching and its stochastic behavior: New formulation and approximation algorithm via min-cost flow optimization. In *AAAI*, pages 3796–3805, 2021.

[Ho and Vaughan, 2012] Chien-Ju Ho and Jennifer Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, pages 45–51, 2012.

[Karp *et al.*, 1990] Richard M Karp, Umesh V Vazirani, and Vijay V Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358, 1990.

[Mahdian and Yan, 2011] Mohammad Mahdian and Qiqi Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing lps. In *STOC*, pages 597–606, 2011.

[Manshadi *et al.*, 2012] Vahideh H Manshadi, Shayan Oveis Gharan, and Amin Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

[Mehta, 2012] Aranyak Mehta. Online matching and ad allocation. *Theoretical Computer Science*, 8(4):265–368, 2012.

[Meir *et al.*, 2013] Reshef Meir, Yiling Chen, and Michal Feldman. Efficient parking allocation as online bipartite matching with posted prices. In *AAMAS*, pages 303–310, 2013.

[Peeta *et al.*, 2010] Srinivas Peeta, F Sibel Salman, Dilek Gunnec, and Kannan Viswanath. Pre-disaster investment decisions for strengthening a highway network. *Computers & Operations Research*, 37(10):1708–1719, 2010.

[Pu *et al.*, 2017] Lingjun Pu, Xu Chen, Jingdong Xu, and Xiaoming Fu. Crowd foraging: A qos-oriented self-organized mobile crowdsourcing framework over opportunistic networks. *IEEE Journal on Selected Areas in Communications*, 35(4):848–862, 2017.

[Sun *et al.*, 2017] Xiaoming Sun, Jia Zhang, and Jialin Zhang. Near optimal algorithms for online weighted bipartite matching in adversary model. *Journal of Combinatorial Optimization*, 34(3):689–705, 2017.

[Tong *et al.*, 2018] Yongxin Tong, Libin Wang, Zimu Zhou, Lei Chen, Bowen Du, and Jieping Ye. Dynamic pricing in spatial crowdsourcing: A matching-based approach. In *SIGMOD*, pages 773–788, 2018.

[Wang *et al.*, 2018] Wanyuan Wang, Bo An, and Yichuan Jiang. Optimal spot-checking for improving evaluation accuracy of peer grading systems. In *AAAI*, pages 833–840, 2018.