Integrated Optimization of Bipartite Matching and Its Stochastic Behavior: New Formulation and Approximation Algorithm via Min-cost Flow Optimization

Yuya Hikima,[1] Yasunori Akagi,[1] Hideaki Kim,[1] Masahiro Kohjima,[1] Takeshi Kurashima,[1] Hiroyuki Toda[1]

[1]NTT Service Evolution Laboratories, NTT Corporation,
1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847, Japan
{yuuya.hikima.ys, yasunori.akagi.cu, hideaki.kin.cn, masahiro.kohjima.ev, takeshi.kurashima.uf, hiroyuki.toda.xb}@hco.ntt.co.jp

Abstract

The research field of stochastic matching has yielded many developments for various applications. In most stochastic matching problems, the probability distributions inherent in the nodes and edges are set a priori, and are not controllable. However, many matching services have options, which we call *control variables*, that affect the probability distributions and thus what constitutes an optimum matching. Although several methods for optimizing the values of the control variables have been developed, their optimization in consideration of the matching problem is still in its infancy. In this paper, we formulate an optimization problem for determining the values of the control variables so as to maximize the expected value of matching weights. Since this problem involves hard to evaluate objective values and is non-convex, we construct an approximation algorithm via a minimum-cost flow algorithm that can find 3-approximation solutions rapidly. Simulations on real data from a ride-hailing platform and a crowd-sourcing market show that the proposed method can find solutions with high profits of the service provider in practical time.

1. Introduction

Bipartite graph matching has received a great deal of attention as a fundamental discrete optimization problem that has a wide ranging list of applications such as matching of workers to jobs, residents to hospitals, and jobs to machines in cloud computing (Ahuja, Magnanti, and Orlin 1993). Among the variants of bipartite graph matching, stochastic matching techniques have been recently developed because they handle the probabilistic uncertainty of instances; that is, the optimum matching is determined under a set of probability distributions associated with the nodes and edges. In the kidney exchange (Chen et al. 2009), for example, the expected number of patient-donor matching has been maximized through the exchangeable probabilities between patients and donors. In internet advertising (Mehta 2012), advertisements are allocated optimally to website visitors based on the probability distribution over types of visitors, which is estimated from past traffic data from websites.

In most stochastic matching problems, the probability distributions inherent in the nodes and edges are assumed to be given a priori and so are not controllable. However, many of the matching services often have options, which we call *control variables*, that affect the probability distributions and the resulting optimum matching. Finding the control variables that maximize the business profit is one of the central topics for the service providers. For example, a crowd-sourcing servicer can manage worker participation rates for a task by increasing or decreasing the wage (Horton and Chilton 2010), where a small wage causes a lack of participants to assign tasks, while a large one might harm the business profit. Therefore, we tackle the problem of optimizing the control variables with the consideration of the resulting bipartite graph matching, given the effect of control variables on the probabilistic uncertainty in the graph.

In this paper, we focus on a problem setting where the existence of nodes on one side of the bipartite graph follows a probability distribution of which shape is determined by control variables. This problem setting appears in various applications such as taxi-requester matching in taxi dispatching and worker-task matching in crowd-sourcing. If the given graph is a complete bipartite graph with equal edge weights, we can use an existing method (Babaioff et al. 2015), which is proposed as a pricing scheme for single item markets. It optimizes control variables so as to adjust the expected number of the stochastic nodes to the number of nodes on the other side. Because the method requires strong assumptions on graph structures, the applicability is limited. A more sophisticated optimization method, along with its approximation ratio, has recently been proposed by (Tong et al. 2018). It optimizes the control variables so as to maximize the expected weight of the resulting matching. However, this method has limited application because the setting of the edge weights is restricted. Also, the approximation ratio of this method gets worse as the number of nodes on one side increases. This could lead to poor performance in large-scale settings.

To address the limitations, we propose a new optimization problem, which is called *Integrated Stochastic Problem for Control variables and Bipartite matching (ISPCB). ISPCB* is the problem of determining the values of the control variables to maximize the expected value of weights of bipartite matching, and can be categorized as a two-stage stochastic optimization problem in stochastic programming. Since *ISPCB* doesn't require any assumptions on graph structures, it can be applied to various applications: in crowd-sourcing markets, we can allocate a set of tasks to workers at smaller costs by setting an appropriate wage for each worker; in ride-hailing service, we can maximize the profit by setting an appropriate fare for each taxi request.

Since *ISPCB* is difficult and finding a global optimal solution is hard, we propose a fast approximation algorithm; it is guaranteed to output solutions with high objective values. *ISPCB* has two difficulties: It is difficult to evaluate the objective value since the objective function deals with an exponential number of weighted bipartite matching problems; It is a non-convex problem because of the non-convexity of probability functions. In order to overcome these difficulties, we approximate the objective function and reduce the problem to a nonlinear minimum-cost flow problem. Then, we show that the problem is a convex min-cost flow problem under the assumption that the probability function is the *monotone hazard rate function* (Barlow, Marshall, and Proschan 1963). This is a reasonable assumption because the *monotone hazard rate function* includes commonly used distributions such as normal and exponential distributions. Convex min-cost flow problems are known to be solved efficiently by several algorithms, and we use the capacity scaling algorithm (Ahuja, Magnanti, and Orlin 1993; Végh 2016) to solve the convex min-cost flow problem. The proposed algorithm can output 3-approximation solutions for *ISPCB*.

We conduct simulations on real data from a ride-hailing platform and a crowd-sourcing market. The results show that the proposed method outputs more profitable solutions than other methods in practical time.

The contributions of this study are threefold.

1. We formulate a new optimization problem, *ISPCB* that can determine the values of control variables that maximize the expected weights of bipartite matching. It is suitable for various applications with control variables that impact the uncertainty of bipartite graphs.
2. We propose a fast approximation algorithm via a min-cost flow algorithm that can output 3-approximation solutions for *ISPCB*.
3. We conduct simulation experiments on real data to confirm the effectiveness of the formulation of *ISPCB* and the proposed algorithm for *ISPCB* in two applications: a ride-hailing platform and a crowd-sourcing market.

2. Related Works

2.1 Matching Problem with Uncertainty

Many studies have examined matching problems under uncertainty, such as stochastic probing matching (Bansal et al. 2010; Chen et al. 2009; Blum et al. 2015; Adamczyk 2011) and online stochastic matching (Karp, Vazirani, and Vazirani 1990; Feldman et al. 2009; Mehta 2012; Aggarwal et al. 2011). The main difference between these works and ours is the decision variables: they attempt to find optimum matching given probabilities of nodes or edges, while we optimize control variables that demonstrate probabilistic effects on the matching problem.

Most recently, in the context of ride-hailing platform strategies, several studies tackle problems similar to ours. For example, (Tong et al. 2018) determines the fare for passengers in each area to control passenger's participation probability and maximize the expected profit from requester-taxi matching. (Chen et al. 2019) proposes a bandit-based method for deciding the fare for each passenger so as to maximize the expected profit from requester-taxi matching. Although their works are similar to ours, there are two distinct differences: (i) Our problem formulation is general and applicable to various real-world domains such as web-based crowd-sourcing and spatial crowd-sourcing, while previous works are specific to the spatial crowd-sourcing domain. (ii) Our algorithm is guaranteed to output a 3-approximation solution, while previous algorithms do not offer constant approximation ratios.

2.2 Stochastic Programming

Our problem can be seen as the two-stage stochastic optimization problem in stochastic programming (Shapiro, Dentcheva, and Ruszczyński 2014; Birge and Louveaux 2011), which is a research area of optimization for problems with uncertainty. A number of existing studies have been conducted on the two-stage stochastic optimization problem since it is essential in many applications. The L-shaped method (Van Slyke and Wets 1969; Gassmann 1990) and stochastic decomposition (Higle and Sen 1991, 2013) are typical solutions for which theoretical convergence has been proved. However, these methods assume that random variables are independent of the decision variables, and so cannot be applied to our problem where random variables are affected by decision variables.

Among the solutions for optimization problems with random variables dependent on decision variables, global metamodel optimization (Scott, Frazier, and Powell 2011; Brochu, Cora, and De Freitas 2010) can be applied to our problem. However, it is computationally expensive and is undesirable because the applications associated with our problem demand quick solutions.

3. Problem Formulation

3.1 Matching Procedure

In this paper, we consider the following procedure to decide one-to-one matching between multiple resources and multiple participants in a matching platform. The set of participants U, the set of resources V, and a weighted bipartite graph $G = (U, V, E)$ are given; when $(u, v) \in E$ holds, resource v can be matched to participant u. We denote the weight of edge $(u, v) \in E$ by w_{uv}. The platformer can determine the value of control variable x_u for each participant $u \in U$. After the decision of x_u for all $u \in U$, each participant chooses whether or not to continue to participate in the subsequent process. This choice is assumed to be stochastic; participant u will continue to participate with probability $p_u(x_u)$ and quit with probability $1 - p_u(x_u)$. If participant u quits, the corresponding node and its connected edges are removed from graph G. Let $G' = (U', V, E')$ be the

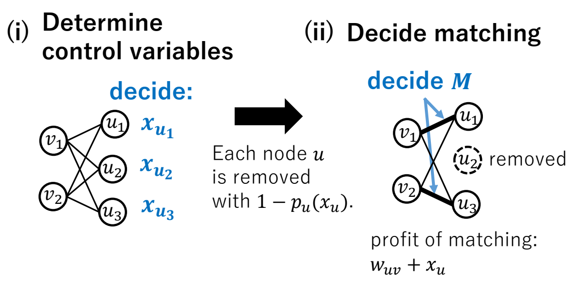
(i) Determine control variables
decide:
x_{u_1}
x_{u_2}
x_{u_3}
Each node u is removed with $1 - p_u(x_u)$.
(ii) Decide matching
decide M
removed
profit of matching: $w_{uv} + x_u$

Figure 1: Matching procedure

graph after above removing procedure. The platformer decides matching $M \subseteq E'$ (i.e., an set of edges without common vertices in E') and gets the profit of w_{uv} and x_u for each $(u, v) \in M$. This procedure is illustrated in Fig. 1. Note that w_{uv} (or x_u) can take a negative value, and in this case the platformer would suffer a loss of $|w_{uv}|$ (or $|x_u|$).

We introduce several real world problems to elucidate the above procedure.

(i) Ride-hailing platform

In the ride-hailing platform, the platformer needs to decide one-to-one matching between multiple requesters and multiple taxis in real time. We divide the time horizon into multiple time steps and consider that taxi dispatch is to be determined at each time step. There are multiple requesters U and taxis V in a two-dimensional space at each time. Let $w_{uv}(\le 0)$ be the total cost of allocating taxi $v \in V$ to requester $u \in U$, including the cost of gasoline, opportunity costs, and other cost factors. The platformer can determine price $x_u(\ge 0)$ for each requester $u \in U$. Then, each requester u accepts $(a_u = 1)$ the price with probability $p_u(x_u)$ or rejects $(a_u = 0)$ the price with probability $1 - p_u(x_u)$. Let the accepting requesters be U' and the combinations of requesters U' and taxis V be E'. The platformer takes the matching $M \subseteq E'$ and gets the profit of $w_{uv} + x_u$ for each $(u, v) \in M$.

(ii) Crowd-sourcing market

In the crowd-sourcing market, the platformer needs to decide one-to-one matching between multiple workers and multiple tasks in real time. We divide the time horizon into multiple time steps and consider that worker-task matching is determined at each time step. There are multiple workers U and tasks V at each time. Let $w_{uv}(\ge 0)$ be the reward paid by the requester of tasks to the platformer when task v is solved by worker u. Reward w_{uv} is calculated based on the skills and performance of each worker. The platformer can determine a wage $x_u(\le 0)$ for each worker $u \in U$. Here, x_u is negative because it is the price the platformer pays for worker u. Then, each worker u accepts the wage, $(a_u = 1)$, with probability $p_u(x_u)$ or rejects the wage, $(a_u = 0)$, with probability $1 - p_u(x_u)$. Let the accepting workers be U' and the combinations of workers U' and tasks V be E'. The crowd-sourcing platformer takes the matching $M \subseteq E'$ and profit $w_{uv} + x_u$ for each $(u, v) \in M$.

3.2 Optimization Problem

We consider an optimization problem to maximize the profit of the platformer in the above procedure. When $a_u \ \forall u \in U$, which are the participant's decisions, are fixed, the problem to decide optimal matching is the following classic weighted bipartite matching problem:

$$\begin{aligned}
(\mathrm{P_{sub}}) \quad & \max_{\boldsymbol{z}} \ \sum\nolimits_{(u,v)\in E} (w_{uv} + x_u) \cdot z_{uv} \\
\text{s.t.} \ & \sum\nolimits_{v\in\delta(u)} z_{uv} \le a_u \quad \forall u \in U \\
& \sum\nolimits_{u\in\delta(v)} z_{uv} \le 1 \quad \forall v \in V \\
& z_{uv} \in \{0,1\} \quad \forall (u,v) \in E,
\end{aligned}$$

where $\delta(\xi)$ is the set of nodes adjacent to the node ξ.

Here, $z_{uv} \in \{0, 1\}$ indicates whether $(u, v) \in E$ are matched $(z_{uv} = 1)$ or not $(z_{uv} = 0)$. The first constraint is that only accepting nodes u can be matched to one v. The second constraint is that each node v can be matched to one u.

Because a_u is a binary random variable generated according to probability $p_u(x_u)$, we focus on the maximization problem of the expected profit of the platformer under control variable $\boldsymbol{x}$, which we call *Integrated Stochastic Problem for Control variables and Bipartite matching (ISPCB)*:

$$(ISPCB) \quad \max_{\boldsymbol{x}\in\mathbb{R}^{|U|}} \ \mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}\big[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})\big],$$

where $f(\boldsymbol{x}, \boldsymbol{z}) = \sum_{(u,v)\in E}(w_{uv}+x_u)\cdot z_{uv}$, which is the objective function of $(\mathrm{P_{sub}})$, and $Z(\boldsymbol{a})$ is the feasible region of $(\mathrm{P_{sub}})$. $D(\boldsymbol{x})$ is a probability distribution for $\boldsymbol{a} \in \{0, 1\}^{|U|}$; The probability mass function of $D(\boldsymbol{x})$ can be calculated by $\Pr(\boldsymbol{a} \mid \boldsymbol{x}) = \prod_{u\in U} \left\{ p_u(x_u)^{a_u} (1 - p_u(x_u))^{(1-a_u)} \right\}$.

Finding the optimum solution for (*ISPCB*) is difficult for two reasons: (i) Random variable $\boldsymbol{a}$ takes $2^{|U|}$ values in $\{0, 1\}^{|U|}$, so it is necessary to solve $2^{|U|}$ weighted bipartite matching problems $(\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z}))$ to calculate the objective value exactly. (ii) The objective function is non-convex since the probability function $p_u(x_u)$ is non-convex.

3.3 Assumption on p_u

We assume the following throughout this paper.

Assumption 1 Probability function $p_u(x)$ is continuous, monotonically decreasing, and bijective in the domain. The domain of $p_u(x)$ is a connected set. There is x satisfying $p_u(x) = 0$ or $\lim_{x\to\infty} p_u(x) = 0$. Moreover, $1 - p_u(x)$ is the *monotone hazard rate distribution (MHR)* (Barlow, Marshall, and Proschan 1963), that is, $-p'_u(x)/p_u(x)$ is monotonically non-decreasing.

Assumption 1 is mild and complementary cumulative distribution functions of common distributions such as normal and exponential distributions satisfy Assumption 1 (Tong et al. 2018; Barlow, Marshall, and Proschan 1963).

The following lemma can be easily derived.

Lemma 1 When probability function $1 - p_u(x)$ is MHR, $x + p_u(x)/p'_u(x)$ is monotonically non-decreasing.

proof. See Section 7.1. □

We will use this lemma to construct an efficient optimization algorithm for (*ISPCB*) in Section 4.

4. Proposed Method

In this section, we propose an approximation algorithm for (*ISPCB*). First, in Section 4.1, we provide a preliminary description of min-cost flow problems, which is referred in Section 4.3 and 4.4. Then, in Sections 4.2-4.4, we propose an approximation algorithm for (*ISPCB*). In Section 4.2, we approximate the objective function of (*ISPCB*), and propose an optimization problem (PA) whose optimal solution is a 3-approximation solution for (*ISPCB*). In Section 4.3, we show that (PA) can be reduced to a convex min-cost flow problem (FP) under Assumption 1. Then, in Section 4.4, we use the capacity scaling algorithm, which is a solution for convex min-cost flow problems, to solve (FP).

4.1 Min-cost Flow Problem

We give a preliminary description of min-cost flow problems. Let $\hat{G} = (\hat{V}, \hat{E})$ be a directed graph with a cost function $c_{ij} : \mathbb{R} \to \mathbb{R}$ and a capacity $\ell_{ij} \in \mathbb{R}_{\geq 0}$ associated with each edge $(i, j) \in \hat{E}$. Each node $i \in \hat{V}$ has a value, $b_i \in \mathbb{R}$, which is called the supply of the node when $b_i > 0$, or the demand of the node when $b_i < 0$. Given the above, the min-cost flow problem can be written as follows:

$$\begin{aligned} \text{(MCF)} \quad \min_{\boldsymbol{z}} \quad & \sum_{(i,j)\in\hat{E}} c_{ij}(z_{ij}) \\ \text{s.t.} \quad & \sum_{j:(i,j)\in\hat{E}} z_{ij} - \sum_{j:(j,i)\in\hat{E}} z_{ji} = b_i \ \ \forall i \in \hat{V} \\ & 0 \leq z_{ij} \leq \ell_{ij} \ \ \forall (i,j) \in \hat{E}. \end{aligned}$$

When no assumptions are placed on the cost function c_{ij}, then min-cost flow problems are generally NP-hard and difficult to solve. However, if the cost functions are convex, several methods can solve min-cost flow problems efficiently (Kiraly and Kovacs 2012; Ahuja, Magnanti, and Orlin 1993). In Section 4.4, we use the capacity scaling algorithm, which is one of those methods, to find an approximation solution of (*ISPCB*).

4.2 Approximation of Objective Function

To propose an approximation algorithm, we consider the approximation of the objective function of (*ISPCB*), that is, $\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})]$. First, we introduce the following problem for a given $\boldsymbol{x}$:

$$\begin{aligned} \max_{\boldsymbol{z}} \quad & \textstyle\sum_{(u,v)\in E} (x_u + w_{uv}) \cdot z_{uv} \qquad (1) \\ s.t. \quad & \textstyle\sum_{v\in\delta(u)} z_{uv} \leq p_u(x_u) \ \ \forall u \in U \\ & \textstyle\sum_{u\in\delta(v)} z_{uv} \leq 1 \ \ \forall v \in V \\ & 0 \leq z_{uv} \leq 1 \ \ \forall (u,v) \in E. \end{aligned}$$

Here, let $\hat{f}(\boldsymbol{x})$ be the optimal value of the above problem.

Then, the following theorem holds.

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

$$1/3 \cdot \hat{f}(\boldsymbol{x}) \leq \mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})] \leq \hat{f}(\boldsymbol{x})$$

proof. See Section 7.2. □

We obtain the following optimization problem by replacing $\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})]$ with $\hat{f}(\boldsymbol{x})$ for (*ISPCB*):

$$\begin{aligned} \text{(PA)} \quad \max_{\boldsymbol{x},\boldsymbol{z}} \quad & \textstyle\sum_{(u,v)\in E} (x_u + w_{uv}) \cdot z_{uv} \\ s.t. \quad & \textstyle\sum_{v\in\delta(u)} z_{uv} \leq p_u(x_u) \ \ \forall u \in U \\ & \textstyle\sum_{u\in\delta(v)} z_{uv} \leq 1 \ \ \forall v \in V \\ & 0 \leq z_{uv} \leq 1 \ \ \forall (u,v) \in E \\ & \boldsymbol{x} \in \mathbb{R}^{|U|}. \end{aligned}$$

Here, let $\bar{\boldsymbol{x}}$ and $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ be optimal solutions for (*ISPCB*) and (PA), respectively. Then, from Theorem 1, we obtain $1/3 \cdot \mathbb{E}_{\boldsymbol{a}\sim D(\bar{\boldsymbol{x}})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\bar{\boldsymbol{x}}, \boldsymbol{z})] \leq 1/3 \cdot \hat{f}(\bar{\boldsymbol{x}}) \leq 1/3 \cdot \hat{f}(\hat{\boldsymbol{x}}) \leq \mathbb{E}_{\boldsymbol{a}\sim D(\hat{\boldsymbol{x}})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\hat{\boldsymbol{x}}, \boldsymbol{z})]$, where the second inequality holds because $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in R^{|U|}} \{\hat{f}(\boldsymbol{x})\}$ from the definition. Therefore, the solution of (PA) is a 3-approximation solution for (*ISPCB*).

4.3 Reduce (PA) to Convex Min-cost Flow Problem

(PA) is a non-convex problem with non-convex functions $p_u(x_u)$. We reduce (PA) to a convex min-cost flow problem under Assumption 1. First, we give the following problem:

$$\begin{aligned} (\text{PA}') \quad \max_{\boldsymbol{z}} \quad & \sum_{u\in U} p_u^{-1}\Big(\sum_{v\in\delta(u)} z_{uv}\Big) \sum_{v\in\delta(u)} z_{uv} + \sum_{(u,v)\in E} w_{uv} z_{uv} \\ s.t. \quad & \sum_{v\in\delta(u)} z_{uv} \in S_u \ \ \forall u \in U \\ & \sum_{u\in\delta(v)} z_{uv} \leq 1 \ \ \forall v \in V \\ & 0 \leq z_{uv} \leq 1 \ \ \forall (u,v) \in E, \end{aligned}$$

where S_u is the range of function p_u. This optimization problem is (PA) in which the decision variable $\boldsymbol{x}$ is eliminated by the substitution $x_u := p_u^{-1}(\sum_{v\in\delta(u)} z_{uv})$.

Then, we show the following theorem.

Theorem 2 Suppose that Assumption 1 holds. Let an optimal solution of (PA') be $\boldsymbol{z}^*$ and $x_u^* := p_u^{-1}(\sum_{v\in\delta(u)} z_{uv}^*)$ for all u. Then, $(\boldsymbol{x}^*, \boldsymbol{z}^*)$ is an optimal solution for (PA).

proof. See Section 7.3. □

From Theorem 2, we can solve (PA) by solving (PA'). Here, we prepare new subscripts s and t. Let $z_{su} := \sum_{v\in\delta(u)} z_{uv}$ for all u and $z_{vt} := \sum_{u\in\delta(v)} z_{uv}$ for all v. Let z_{st} be a slack variable and $n := \min\{|U|, |V|\}$: Then,

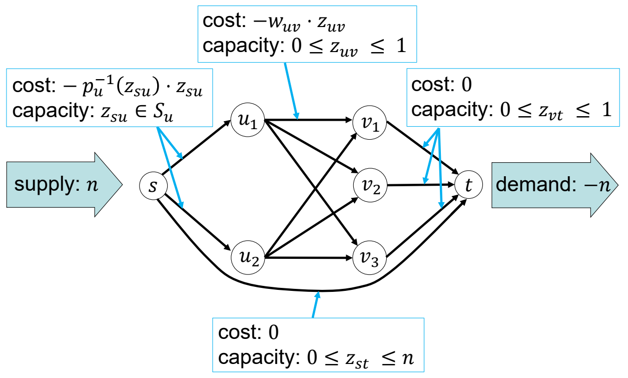
cost: $-w_{uv} \cdot z_{uv}$
capacity: $0 \le z_{uv} \le 1$
cost: $-p_u^{-1}(z_{su}) \cdot z_{su}$
capacity: $z_{su} \in S_u$
cost: 0
capacity: $0 \le z_{vt} \le 1$
supply: n
demand: $-n$
s
u_1
u_2
v_1
v_2
v_3
t
cost: 0
capacity: $0 \le z_{st} \le n$

Figure 2: Illustration of (FP)

(PA′) can be written as follows.

$$\text{(FP)} \quad \min_{\boldsymbol{z}} \sum_{u \in U} -p_u^{-1}(z_{su}) \cdot z_{su} - \sum_{(u,v) \in E} w_{uv} \cdot z_{uv} \tag{2}$$

$$\text{s.t.} \sum_{u \in U} z_{su} + z_{st} = n, \sum_{v \in V} z_{vt} + z_{st} = n \tag{3}$$

$$z_{su} - \sum_{v \in \delta(u)} z_{uv} = 0 \quad \forall u \in U \tag{4}$$

$$\sum_{u \in \delta(v)} z_{uv} - z_{vt} = 0 \quad \forall v \in V \tag{5}$$

$$z_{su} \in S_u \quad \forall u \in U \tag{6}$$

$$0 \le z_{vt} \le 1 \quad \forall v \in V \tag{7}$$

$$0 \le z_{uv} \le 1 \quad \forall (u,v) \in E \tag{8}$$

$$0 \le z_{st} \le n. \tag{9}$$

This is a nonlinear min-cost flow problem for the graph with $U \cup V \cup \{s,t\}$ as nodes.

We explain (FP) in the context of the min-cost flow problem (See Fig. 2). First, (2) represents the cost function for the flow amount of each edge. $-p_u^{-1}(z_{su}) \cdot z_{su}$ is the cost function for each edge of $\{(s,u) \mid u \in U\}$ and $w_{uv} \cdot z_{uv}$ is the cost function for each edge of $\{(u,v) \mid u \in U, v \in V\}$. The cost for z_{vt} and z_{st} is 0 because these variables are not in the objective function. Second, (3), (4) and (5) represent the demand/supply for each node. From these equations, the supply at the node s is n, the demand at the node t is $-n$, and the demands/supplies at other nodes are 0. Therefore, the problem is to find a way of sending n amount of flow from node s to node t through the network. Third, (6), (7), (8) and (9) represent the capacity of flow amount for each edge. We can view (FP) as a min-cost flow problem as shown in Fig. 2.

Then, we show the following theorem.

Theorem 3 When Assumption 1 holds, the cost function for all edges in min-cost flow problem (FP) is convex. In other words, (FP) is a convex min-cost flow problem.

proof. See Section 7.4. □

4.4 Solution for (FP) via Capacity Scaling Algorithm

From Theorem 3, we can use the capacity scaling algorithm (Ahuja, Magnanti, and Orlin 1993; Végh 2016) to solve (FP) as described in Section 4.1. The capacity scaling algorithm explained in (Ahuja, Magnanti, and Orlin 1993) solves the convex min-cost flow problems that restrict feasible $\boldsymbol{z}$ to integer values. We can use this method and obtain an optimal solution in a continuous domain to any desired degree of accuracy by the following procedure: (i) Substitute $\epsilon \cdot y_{ij}$ for each z_{ij}, where ϵ is a sufficiently small value and $y_{ij} \in \mathbb{Z}^{|E|}$; (ii) Find an integer optimal solution $\boldsymbol{y}^*$ of the transformed problem; (iii) Let $z^*_{ij} := \epsilon \cdot y^*_{ij}$. Then, $\boldsymbol{z}^*$is an optimal solution of the original problem with a degree of accuracy of ϵ.

We describe the time complexity when we use the capacity scaling algorithm to solve (FP). For a standard convex min-cost flow problem (MCF) in Section 4.1, Theorem 14.1 of (Ahuja, Magnanti, and Orlin 1993) shows that the capacity scaling algorithm finds an integer optimal solution in $O(|\hat{E}| \cdot \log(\hat{B}) \cdot \hat{S})$ time. Here, $\hat{B} := \max_{i \in \hat{V}} |b_i|$, and $\hat{S}$ is the time complexity for solving the shortest path problem in graph $\hat{G}$ with non-negative edge costs. Dijkstra algorithm with binary heap can solve shortest path problems in $O(|\hat{E}| \cdot \log |\hat{V}|)$ time, so the total time complexity is $O(|\hat{E}|^2 \cdot \log(\hat{B}) \cdot \log |\hat{V}|)$. In addition, when we use this algorithm to find an optimal solution of a convex min-cost flow problem in a continuous domain to a degree of accuracy of ϵ, the total time complexity is $O(|\hat{E}|^2 \cdot \log(\hat{B}/\epsilon) \cdot \log |\hat{V}|)$ because substituting $\epsilon \cdot y_{ij}$ into z_{ij} causes $\hat{B}$ to be multiplied by $1/\epsilon$. From the above, for our problem (FP) with graph $G = (U, V, E)$, the capacity scaling algorithm can find an optimal solution to a degree of accuracy of ϵ in $O(|E|^2 \cdot \log(n/\epsilon) \cdot \log(|U| + |V|))$ time. Note that $n = \min\{|U|, |V|\}$.

5. Experiment

We conduct experiments to show that the followings hold:

- Proposed method outputs a more profitable solution than the other methods in each application.
- Proposed method outputs the solution in practical time.

We performed simulation experiments using real data from a ride-hailing platform and a crowd-sourcing market.

Experiments were run on a computer with Xeon Platinum 8168 of 4 x 2.7GHz, 1TB of memory, running CentOS 7.6. The program codes were implemented in Python.

5.1 Ride-hailing Platform

We conduct experiments in the ride-haling platform whose matching procedure is described in Section 3.1.

Data sets and parameter setup We used ride data gathered in New York[1]. We use yellow taxi data and green taxi data of Manhattan, Queens, Bronx, and Brooklyn. Each record consists of pick-up area, pick-up time, drop-off area, drop-off time, trip distance, total amount charged to passengers. We perform simulations using the data from October 6 and 10, 2019, which are holidays and weekdays in a randomly chosen week. In each day, requester-taxi matching situations are constructed every 5 minutes from 10:00 to

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

Region, Time interval	date	n (requesters)		m (taxis)	
		MEAN	SD	MEAN	SD
Manhattan,	10/6	80.0	11.1	76.1	13.1
30 seconds	10/10	100.1	17.4	96.6	21.6
Queens,	10/6	92.3	22.7	88.3	27.0
300 seconds	10/10	95.4	23.1	89.8	28.1
Bronx,	10/6	5.3	2.8	5.3	2.7
300 seconds	10/10	6.2	2.6	6.2	2.7
Brooklyn,	10/6	26.5	5.7	26.4	5.6
300 seconds	10/10	27.2	5.3	26.7	7.3

Table 1: Summary of the real dataset in ride-hailing platform.

20:00, that is, 120 (10 hours $\times$ 12 times) situations are simulated. As the length of one time step, t_s, which is required to generate the situation, we adopt 30 seconds for Manhattan and 300 seconds for the regions other than Manhattan because of the differences in the amount of data in each region. The features of the dataset used are summarized in Table 1.

We recreate the situations from the data of each region, and set the inputs U, V, w_{uv}, and p_u at each time.
(i) U: We extract taxi request data that has a pick-up time within t_s seconds from the target minute as the requester set U. We set the origin/destination points for each requester $u \in U$ to be the points obtained by adding Gaussian noise to the center point of the pick-up/drop-off area. This is because the taxi ride data records only the areas of the pick-up/drop-off.
(ii) V: We assume that it is possible to dispatch taxis that have completed a request within the past t_s seconds from the target minute. Thus, the data with the corresponding drop-off time is extracted as the taxi set V. The location of each taxi $v \in V$ is set by adding Gaussian noise to the center point of drop-off area as with U.
(iii) w_{uv}: Let $w_{uv} = -18.0 \cdot \tau_{uv}$ when $\tau_{uv} \leq 0.1$, with $w_{uv} = -\infty$ otherwise. Here, τ_{uv} is the time required for taxi v to fulfill request u, and is calculated from the destination and origin of requester u and the location of taxi v. To reflect the real-world constraint that a requester cannot be matched with a taxi that is more than a certain distance (e.g. 5km) away, we set $w_{uv} = -\infty$ when $\tau_{uv} > 0.1$. Parameter 18.0 means taxi driver's opportunity cost, which is based on taxi driver's income.
(iv) p_u: We define p_u in two ways. First, we consider the following piecewise linear function:

$$p_u^{PL}(x) := \begin{cases} 1 & (x < q_u) \\ -\frac{1}{(\alpha-1)\cdot q_u} \cdot x + \frac{\alpha}{\alpha-1} & (q_u \leq x \leq \alpha \cdot q_u) \\ 0 & (x > \alpha \cdot q_u), \end{cases}$$

where α and q_u are constant scalars. We set $\alpha := 1.5$ and let q_u be the actually paid amount for each request u in the data set. When using this model, we restrict the prices that can be set by the proposed method to $[q_u, \alpha \cdot q_u]$ in order to satisfy Assumption 1.

Second, we consider the following sigmoid model:

$$p_u^{Sig}(x) := 1 - \frac{1}{1 + e^{-(x-\beta\cdot q_u)/(\gamma\cdot|q_u|)}},$$

where, β and γ are constants. We set $\beta = 1.3, \gamma = 0.3\sqrt{3}/\pi$. The same q_u as in $p_u^{PL}(x)$ is used.

Metric To measure the expected benefits yielded by each approach to the platformer, we define ER $:= \frac{1}{N}\sum_{k=1}^{N} \max_{\boldsymbol{z}\in\boldsymbol{Z}(\boldsymbol{a}^k)} f(\boldsymbol{x}, \boldsymbol{z})$, where $(\boldsymbol{a}^k)_{k=1}^{N}$ is a set of independent, identically distributed realizations of $\boldsymbol{a}$, which are acceptance results. This metric is the approximated expected revenue of the platformer. We use it to calculate the expected profit obtained at each time. We set $N := 10^2$.

Compared methods We compared the proposed method with two state-of-the-art methods.
MAPS (Tong et al. 2018): It is an approximation algorithm for area basis pricing for taxi service. In applying MAPS to our problem, we have modified the followings: (a) we divided the requesters in our problem into groups of areas and defined the acceptance probability for each area by taking the average of the individual acceptance probability functions within the area; (b) We changed the τ_{uv} (in **Data sets and parameter setup**) to τ_u, which is the time required to fulfill request u. Then, we obtain prices by MAPS.
LinUCB (Li et al. 2010): It is a generic contextual bandit algorithm, which is adopted by (Chen et al. 2019). As arms of the method, we use pricing factors $\{0.6, 0.8, 1.0, 1.2, 1.4\}$ which are multiples of the base price. The base price for each request is calculated according to the trip distance and the average price per unit distance for all requests in the period July to September, 2019. As features for learning, we use (pick-up areas, drop-off areas, hours, trip distance). The initial parameter θ_μ for each arm $\mu \in \{0.6, 0.8, 1.0, 1.2, 1.4\}$ is learned through 11040 processes using the request data in the period July to September, 2019.

Experimental results Table 2 shows the results of the experiments. Regardless of the region and the form of the probability function p_u, the proposed method outperforms all compared methods in ER. The differences between the proposed method and all compared methods in ER are significant (two-sided t-test: $p < 0.0001$) in all experiments. In terms of computational time, the proposed algorithm solves the problem fast enough for practical use, although the existing method, MAPS, requires less computational time.

5.2 Crowd Sourcing Market

We conduct experiments in a crowd-sourcing market whose matching procedure is described in Section 3.1.

Data set and parameter setup We used an open crowd-sourcing dataset (Buckley, Lease, and Smucker 2010). The data set contains records of worker's judgments on the task of checking the relevance of a given topic and a web page. Each record has elements of (topic ID, worker ID, document ID, judgment). Judgment is divided into five categories: highly relevant, relevant, non-relevant, unknown, broken link. Here, broken link indicates that the web page cannot be viewed. This data set is summarized in Table 3. We use this data to replicate the worker-task matching situations and conduct multiple experiments.

We set the inputs U, V, w_{uv}, and p_u from the data for each experiment.

Region,	Proposed		MAPS		LinUCB	
p_u	ER	time	ER	time	ER	time
Manhat-	**1074**	5.185	769	.098	566	9.275
tan, PL	**1456**	8.697	1012	.121	756	11.554
Manhat-	**982**	6.617	733	.125	600	9.411
tan, Sig	**1304**	11.127	967	.147	799	11.641
Queens,	**2175**	6.068	384	.055	784	11.765
PL	**2443**	6.896	441	.060	1061	11.992
Queens,	**1954**	5.468	380	.081	1080	11.382
Sig	**2229**	7.787	449	.091	1315	11.535
Bronx,	**68**	.008	24	.004	27	.184
PL	**94**	.010	37	.005	39	.219
Bronx,	**64**	.010	25	.005	37	.189
Sig	**87**	.013	38	.007	50	.232
Brookl-	**337**	.237	183	.022	142	2.480
yn, PL	**398**	.266	206	.028	169	2.440
Brookl-	**304**	.273	176	.029	162	2.455
yn, Sig	**358**	.356	196	.037	191	2.515

Table 2: Results of ride-hailing platform simulations. For each region and p_u, the first (second) row shows the results using the data from October 6 (October 10). The *time* column of each method indicates the computational time (in seconds). Each result represents the average of 120 dispatch runs. The best value for each dataset in ER is in bold. In all experiments, the differences between the proposed method and all compared methods in ER are significant (two-sided t-test: $p < 0.0001$).

all data	topic ID	document ID	worker ID	judgments
98453	100	19902	766	5

Table 3: Summary of the real dataset in crowd sourcing market.

(i) U: Each worker in the data is assumed to be active with a probability of ϕ with U being the set of active workers.
(ii) V: Each task in the data is assumed to appear with a probability of ψ with V being the set of tasks created.
(iii) w_{uv}: We decide the correct judgment for each task in the data by majority vote. Let ϕ^u_s be the percentage of correct answers of worker u for topic s. In our experiment, we assume that ϕ^u_s are known a priori, and $w_{uv} := \phi^u_{s(v)}$ for each $(u,\ v)$. Here, $s(v)$ means the topic of task v. This setup is based on a scheme that determines the value of solving a task according to the skill of the worker. For topics that worker u has never solved, we set the percentage of correct answers of worker u to be that for the whole of the tasks that the worker u has solved.
(iv) p_u: We define p_u in two ways as with Section 5.1. First, we consider the following piecewise linear function:

$$p^{PL}_u(x) := \begin{cases} 1 & (x < \alpha \cdot q_u) \\ \frac{1}{(\alpha-1)\cdot q_u} \cdot x - \frac{1}{\alpha-1} & (\alpha \cdot q_u \leq x \leq q_u) \\ 0 & (x > q_u). \end{cases}$$

We set $\alpha := 1.5$. Since the data does not contain information on the amount paid to the worker, q_u is generated from a uniform distribution of $[-0.4, -0.1]$ for each worker u. As with Section 5.1, we restrict the wages that can be set by the proposed method to $[\alpha \cdot q_u, q_u]$.

Second, we consider the following sigmoid model:

$$p^{Sig}_u(x) := 1 - \frac{1}{1 + e^{-(x-\beta\cdot q_u)/(\gamma\cdot|q_u|)}},$$

where β and γ are constants. We set $\beta = 1.25, \gamma = 0.25/\pi$. The same setting of q_u as in $p^{PL}_u(x)$ is used.

We conduct experiments with various values of parameters (ϕ, ψ) and form of p_u.

Metric We run 10^3 simulations for each setting. Then, we let the average of the resulting profits of 10^3 simulations be the metric, *ER*. It is an approximation of the expected platformer's benefits.

Compared methods We compared the proposed method to two exiting methods.
Myerson Reserve Price (MRP) (Myerson 1981): MRP is proposed as the optimal price in a single item market when there is enough supply and there is no differentiation among buyers. We consider multiple types of tasks to be one type of task, and all multiple workers to be of average ability. Then, we use this price. Specifically, let $x_u := \text{argmax}_x\{(x + \hat{w}) \cdot p(x)\}$ for all u, where $p(x)$ is average of acceptance probability for all workers, and $\hat{w}$ is the average of the correct rate for all combinations of workers and tasks.
Capped UCB (Babaioff et al. 2015): It is the pricing strategy created to tackle the problem of limited supply in single item market when there is no differentiation among buyers. To use this strategy, we consider multiple types of tasks to be one type of task, and all multiple workers to be of average ability. This method determines the price while estimating $p(x)$, which is acceptance probability for all workers; here we take it to be a given function. Specifically, for all u, let $x_u := \text{argmax}_x\{(x+\hat{w})\cdot\min(|V|, |U|\cdot p(x))\}$, where $p(x)$ and $\hat{w}$ are the same as those defined for MRP.

Experimental results Table 4 shows the results of the simulation experiments with different parameter values. Regardless of the problem parameters and the form of the probability function p_u, the proposed method outperforms all baselines in terms of ER. Moreover, the differences between the proposed method and all compared methods are significant (two-sided t-test: $p < 0.0001$) in all experiments. In addition, the computational time of the proposed method is short enough for practical use, although the compared methods requires less computational time.

5.3 Discussion

From the experimental results and theoretical results, there are pros and cons between our algorithm and compared methods. Our algorithm provides the 3-approximation guarantee, and can significantly increase the profit compared to existing methods. In contrast, MAPS, MRP, and Capped UCB can solve problems quickly, and our method is inferior to those in terms of computational time.

However, there is room for improvement in the computation time of the proposed algorithm. For example, we can

ϕ, ψ	p_u	Proposed		MRP		Capped UCB	
		ER	time	ER	time	ER	time
.1, .0005	PL	**18.8**	6.122	13.4	.009	13.2	.009
	Sig	**18.4**	22.465	12.6	.019	14.0	.019
.1, .001	PL	**20.9**	19.142	13.6	.011	13.6	.011
	Sig	**20.3**	55.682	13.5	.019	13.5	.020
.05, .0005	PL	**10.5**	2.586	6.9	.009	6.9	.010
	Sig	**10.3**	4.778	6.9	.012	6.9	.012
.05, .001	PL	**10.4**	2.563	6.8	.009	6.8	.010
	Sig	**10.2**	12.113	6.9	.013	6.9	.014

Table 4: Results of real dataset simulation. Each result represents the average of 10^3 simulation runs. The *time* column of each method indicates the computational time (in seconds). The best value for each dataset is in bold. In all experiments, the differences between the proposed method and all compared methods in ER are significant (two-sided t-test: $p < 0.0001$).

apply the method of (Végh 2016), which is the state of the art method for convex min-cost flow problems and is expected to reduce the computational complexity. In addition, the computational time of the proposed algorithm can be reduced by dividing the large-scale instances into groups of middle-scale ones (e.g. decreasing the time step or subdividing the region) in exchange for a slight reduction in the objective value.

6. Conclusion

We formulate an optimization problem, *ISPCB*, to determine the values of control variables that maximize the expected value of the weights of bipartite matching. It is suitable for various applications with control variables that affect the probabilities of nodes in a bipartite graph. Moreover, we proposed a fast approximation algorithm that can output 3-approximation solutions for *ISPCB*. Simulations using real data from two applications (ride-hailing platform and crowd-sourcing market) confirmed the effectiveness of our method, the formulation of the *ISPCB* and the proposed approximation algorithm.

Future work includes showing the effectiveness of the proposed method by applying it to actual services, speeding up the proposed method, proving a tight approximation ratio of the proposed method, and developing a method that allows Assumption 1 to be relaxed.

7. Proof

7.1 Proof of Lemma 1

Since $-p'_u(x)/p_u(x)$ is monotonically non-decreasing, $p_u(x)/p'_u(x)$ is monotonically non-decreasing. Then, $x + p_u(x)/p'_u(x)$ is monotonically non-decreasing because x is monotonically increasing. □

7.2 Proof of Theorem 1

First, we introduce Problem A and show Lemma A, which is used in the proof of Theorem 1.

Problem A. We consider an undirected bipartite graph $G = (U, V, E)$ where each node $u \in U$ has probability value $p_u(x_u)$. Here, $p_u : \mathbb{R} \to \mathbb{R}$ and $x_u \in \mathbb{R}$ are given. Suppose steps (i) and (ii) are repeated until U or V becomes empty: (i) Choose $u \in U$ and $v \in V$ and try to match them, which is called *probing*. The *probing* succeeds with probability $p_u(x_u)$ and the *probing* fails with probability $1-p_u(x_u)$; (ii) If the *probing* succeeds, u and v are removed from U and V, respectively, with the benefit of $(x_u + w_{uv})$. If the *probing* fails, no profit is made and u is removed from U. Here, *what is the most profitable probing strategy?*

Lemma A. Let $E[\text{ALG}]$ be the expected profit obtained by the ROUND-COLOR-PROBE algorithm (Bansal et al. 2010) for Problem A. Then, $E[\text{ALG}] \geq 1/3 \cdot \hat{f}(\boldsymbol{x})$.

proof. For the problem (LP1) defined in (Bansal et al. 2010), let $V^{LP1} := U \cup V$, $E^{LP1} := E$, $w_{uv}^{LP1} := (x_u + w_{uv})$ for all $(u, v) \in E^{LP1}$, $t_u^{LP1} := 1$ for all $u \in U^{LP1}$, $t_v^{LP1} := \infty$ for all $v \in V^{LP1}$. Under these settings, let L^* is an optimal value of (LP1). Then, $E[\text{ALG}] \geq 1/3 \cdot L^*$ from (Bansal et al. 2010, Theorem 10). Since (LP1) is equivalent to (1) under the settings, $E[\text{ALG}] \geq 1/3 \cdot \hat{f}(\boldsymbol{x})$. □

Then, we give proof of Theorem 1. First, we show $\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})] \leq \hat{f}(\boldsymbol{x})$. Let $\boldsymbol{Z}'(\boldsymbol{a})$ be the feasible region of ($\mathrm{P}_{\mathrm{sub}}$) where $z_{uv} \in \{0, 1\}$ is changed to $0 \leq z_{uv} \leq 1$. Then, the following equality holds from (Korte and Vygen 2005, Theorem 5.12, Theorem 11.2):

$$\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z}) = \max_{\boldsymbol{z}\in \boldsymbol{Z}'(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z}) \tag{10}$$

For any given $\boldsymbol{x}$, $\max_{\boldsymbol{z}\in \boldsymbol{Z}'(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})$ is a linear programming problem and can be written as follows:

$$\begin{aligned} (\mathrm{LP_p}) \quad & \max_{\boldsymbol{z}} \quad \boldsymbol{h}_1^\top \boldsymbol{z} \\ & \quad \text{s.t.} \quad \boldsymbol{H}_1\boldsymbol{z} \leq \boldsymbol{H}_2\boldsymbol{a} + \boldsymbol{h}_2, \ \ 0 \leq \boldsymbol{z}, \end{aligned}$$

where $\boldsymbol{h}_1$, $\boldsymbol{h}_2$, $\boldsymbol{H}_1$, $\boldsymbol{H}_2$ are constant vectors and constant matrices. Then, the dual problem is as follows:

$$\begin{aligned} (\mathrm{LP_d}) \quad & \min_{\boldsymbol{y}} \quad (\boldsymbol{H}_2\boldsymbol{a} + \boldsymbol{h}_2)^\top \boldsymbol{y} \\ & \quad \text{s.t.} \quad \boldsymbol{H}_1^\top \boldsymbol{y} \geq \boldsymbol{h}_1, \ \ 0 \leq \boldsymbol{y}. \end{aligned}$$

Since the dual problem has the same optimal value as the primal problem in linear programming, the optimal value of $(\mathrm{LP_d})$ is equal to $(\mathrm{LP_p})$, that is, $\max_{\boldsymbol{z}\in \boldsymbol{Z}'(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})$. Let $\boldsymbol{Y}$ be the feasible region of $(\mathrm{LP_d})$. Since $(\boldsymbol{H}_2\boldsymbol{a}+\boldsymbol{h}_2)^\top \boldsymbol{y}$ is concave in $\boldsymbol{a}$ for any $\boldsymbol{y} \in \boldsymbol{Y}$, we obtain $\min_{\boldsymbol{y}\in \boldsymbol{Y}} (\boldsymbol{H}_2\boldsymbol{a}+\boldsymbol{h}_2)^\top \boldsymbol{y}$, that is, $\max_{\boldsymbol{z}\in \boldsymbol{Z}'(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})$ is concave in $\boldsymbol{a}$ from (Boyd and Vandenberghe 2004, Section 3.2.3). Then, the following holds from Jensen's inequality:

$$\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}'(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})] \leq \max_{\boldsymbol{z}\in \boldsymbol{Z}'(\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\boldsymbol{a}])} f(\boldsymbol{x}, \boldsymbol{z}). \tag{11}$$

Then, since (10) and (11) hold and $\hat{f}(\boldsymbol{x}) = \max_{\boldsymbol{z}\in \boldsymbol{Z}'(\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\boldsymbol{a}])} f(\boldsymbol{x}, \boldsymbol{z})$ from definition, the following inequality holds:

$$\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x}, \boldsymbol{z})] \leq \hat{f}(\boldsymbol{x}). \tag{12}$$

We show $1/3 \cdot \hat{f}(\boldsymbol{x}) \leq \mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x},\boldsymbol{z})]$. From Lemma A, if $\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x},\boldsymbol{z})] \geq E[\text{ALG}]$, we get $\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x},\boldsymbol{z})] \geq 1/3 \cdot \hat{f}(\boldsymbol{x})$. In Problem A, for each $u \in U$, whether a matching involving u succeeds or fails is determined regardless of other nodes $v \in V$ and the order of *probings*. For a trial ℓ of Problem A, let $\boldsymbol{a}^\ell \in \{0,1\}^{|U|}$ be the variables that represent whether the matching is successful or not if each $u \in U$ is chosen. Here, $a^\ell_u = 1$ in the case of success, and $a^\ell_u = 0$ in the case of failure. Then, we can let $\boldsymbol{z}^\dagger(\boldsymbol{a}^\ell)$ be the matching done by the ROUND-COLOR-PROBE algorithm in trial ℓ. Here, $z^\dagger_{uv}(\boldsymbol{a}^\ell) = 1$ indicates that u and v are matched, and $z^\dagger_{uv}(\boldsymbol{a}^\ell) = 0$ indicates that they are not matched. From the definition, $\boldsymbol{z}^\dagger(\boldsymbol{a}^\ell)$ is included in $\boldsymbol{Z}(\boldsymbol{a}^\ell)$, that is, the feasible region of (P_{sub}) with $\boldsymbol{a} = \boldsymbol{a}^\ell$. The profit obtained by the ROUND-COLOR-PROBE algorithm in trial ℓ is $\sum_{uv}(p_u + w_{uv}) \cdot z^\dagger_{uv}(\boldsymbol{a}^\ell)$, that is, $f(\boldsymbol{x}, \boldsymbol{z}^\dagger(\boldsymbol{a}^\ell))$. Since $\max_{\boldsymbol{z}\in\boldsymbol{Z}(\boldsymbol{a}^\ell)} f(\boldsymbol{x},\boldsymbol{z}) \geq f(\boldsymbol{x}, \boldsymbol{z}^\dagger(\boldsymbol{a}^\ell))$ for any $\boldsymbol{a}^\ell$,

$$\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x},\boldsymbol{z})] \geq \mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[f(\boldsymbol{x},\boldsymbol{z}^\dagger(\boldsymbol{a}))] = E[\text{ALG}]. \tag{13}$$

From Lemma A and (13),

$$\mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x},\boldsymbol{z})] \geq 1/3 \cdot \hat{f}(\boldsymbol{x}). \tag{14}$$

Then, from (12) and (14), it yields that

$$1/3 \cdot \hat{f}(\boldsymbol{x}) \leq \mathbb{E}_{\boldsymbol{a}\sim D(\boldsymbol{x})}[\max_{\boldsymbol{z}\in \boldsymbol{Z}(\boldsymbol{a})} f(\boldsymbol{x},\boldsymbol{z})] \leq \hat{f}(\boldsymbol{x}). \quad \square$$

7.3 Proof of Theorem 2

Let $(\hat{\boldsymbol{x}},\ \hat{\boldsymbol{z}})$ be an optimal solution for (PA). First, we show $\sum_{v\in\delta(u)} \hat{z}_{uv} \in S_u$ for all $u \in U$. For each $u \in U$, we consider two cases according to whether $\sum_{v\in\delta(u)} \hat{z}_{uv} > 0$ or $\sum_{v\in\delta(u)} \hat{z}_{uv} = 0$.

(i) When $\sum_{v\in\delta(u)} \hat{z}_{uv} > 0$, there exists a real number $c \in S_u$ which satisfies $0 \leq c < \sum_{v\in\delta(u)} \hat{z}_{uv}$, because $\lim_{x\to\infty} p_u(x) = 0$ or there is a real number x satisfying $p_u(x) = 0$ from Assumption 1. Moreover, $\sum_{v\in\delta(u)} \hat{z}_{uv} \leq d$ from the first constraints of (PA), where $d := p_u(\hat{x}_u)$. The set S_u is a connected set because it is an image of a connected set by a continuous function p_u. Because $c < \sum_{v\in\delta(u)} \hat{z}_{uv} \leq d$ for $c, d \in S_u$ and S_u is a connected set, we get $\sum_{v\in\delta(u)} \hat{z}_{uv} \in S_u$.

(ii) When $\sum_{v\in\delta(u)} \hat{z}_{uv} = 0$, we show $0 \in S_u$, which yields $\sum_{v\in\delta(u)} \hat{z}_{uv} \in S_u$. We assume $0 \notin S_u$ to obtain a contradiction. We pick an arbitrary vertex $\grave{v} \in \delta(u)$. Note that we can assume $\delta(u) \neq \emptyset$ for all $u \in U$ without loss of generality, because removing $\{u \mid \delta(u) = \emptyset\}$ from U has no effect on the optimization problem. Here, $\sum_{\grave{u}\in\delta(\grave{v})} \hat{z}_{\grave{u}\grave{v}} \leq 1$ from constraints of (PA). We consider the following two cases.

(ii-a) When $\sum_{\grave{u}\in\delta(\grave{v})} \hat{z}_{\grave{u}\grave{v}} < 1$, we pick an number $x_M \in \{x \mid x + w_{u\grave{v}} > 0\}$. Since $0 \notin S_u$, there exists ϵ satisfying $0 < \epsilon \leq p_u(x_M)$ and $\sum_{\grave{u}\in\delta(\grave{v})} \hat{z}_{\grave{u}\grave{v}} + \epsilon \leq 1$. Replacing $\hat{x}_u$ with x_M and $\hat{z}_{u\grave{v}} (= 0)$ with ϵ increases the objective value for (PA) because $(x^M + w_{u\grave{v}}) \cdot \epsilon > 0$ and this modification does not impair feasibility. This contradicts the optimality of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ for (PA).

(ii-b) When $\sum_{\grave{u}\in\delta(\grave{v})} \hat{z}_{\grave{u}\grave{v}} = 1$, there exists $\bar{u}\ (\neq u) \in U$ satisfying $z_{\bar{u}\grave{v}} > 0$ because $\sum_{v\in\delta(u)} \hat{z}_{uv} = 0$. Since $0 \notin S_u$, it yileds that $\lim_{x\to\infty} p_u(x) = 0$ from Assumption 1. Then, there exists x_M satisfying $x_M + w_{u\grave{v}} > \hat{x}_{\bar{u}} + w_{\bar{u}\grave{v}}$ and $p_u(x_M) = \epsilon < z_{\bar{u}\grave{v}}$. Replacing $\hat{x}_u$ with x_M, $\hat{z}_{u\grave{v}} (= 0)$ with ϵ, and $\hat{z}_{\bar{u}\grave{v}}$ with $\hat{z}_{\bar{u}\grave{v}} - \epsilon$ increases the objective value for (PA) because $(x_M + w_{u\grave{v}}) \cdot \epsilon - (\hat{x}_{\bar{u}} + w_{\bar{u}\grave{v}}) \cdot \epsilon > 0$ and this modification does not impair feasibility. This contradicts the optimality of $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{z}})$ for (PA).

From (i) and (ii), we get $\sum_{v\in\delta(u)} \hat{z}_{uv} \in S_u$ for all $u \in U$.

Next, we show that there exists an optimal solution $(\tilde{\boldsymbol{x}},\ \tilde{\boldsymbol{z}})$ for (PA) that satisfies $p_u(\tilde{x}_u) = \sum_{v\in\delta(u)} \tilde{z}_{uv}$ for all u. Let $(\hat{\boldsymbol{x}},\ \hat{\boldsymbol{z}})$ be an optimal solution for (PA). Then, from constraints of (PA), $\sum_{v\in\delta(u)} \hat{z}_{uv} \leq p_u(\hat{x}_u)$ for all u. Suppose that there exists u satisfying $\sum_{v\in\delta(u)} \hat{z}_{uv} < p_u(\hat{x}_u)$ and let $\hat{U} := \{u \mid \sum_{v\in\delta(u)} \hat{z}_{uv} < p_u(\hat{x}_u)\}$. Since $\sum_{v\in\delta(u)} \hat{z}_{uv} \in S_u$ and p_u is monotonically decreasing, there exists a positive scalar d_u that satisfies $\sum_{v\in\delta(u)} \hat{z}_{uv} = p_u(\hat{x}_u + d_u)$ for all $u \in \hat{U}$. Let $\tilde{x}_u$ be $\hat{x}_u + d_u$ for all $u \in \hat{U}$, $\tilde{x}_u$ be $\hat{x}_u$ for all $u \notin \hat{U}$, 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 $\sum_{v\in\delta(u)} \tilde{z}_{uv} = p_u(\tilde{x}_u)$ for all $u \in U$.

Hence, we can set the first inequality constraints of (PA) as $\sum_{v\in\delta(u)} z_{uv} = p_u(x_u)$ for all $u \in U$. Then, $x_u = p_u^{-1}\big(\sum_{v\in\delta(u)} z_{uv}\big)$ since p_u is a bijective function from Assumption 1. By substituting this equality into variable $\boldsymbol{x}$ for (PA), we obtain (PA'). Therefore, when $\boldsymbol{z}^*$ is an optimal solution for (PA') and $x^*_u = p_u^{-1}(\sum_{v\in\delta(u)} z^*_{uv})$ for all $u \in U$, $(\boldsymbol{x}^*, \boldsymbol{z}^*)$ is an optimal solution for (PA). $\square$

7.4 Proof of Theorem 3

The theorem holds if $p_u^{-1}(z_{su}) \cdot z_{su}$ is concave for each u. For arbitrary $u \in U$, there exists only one y_u that satisfies $z_{su} = p_u(y_u)$ from Assumption 1 when $z_{su} \in S_u$. Then, the following equality holds:

$$\begin{aligned}\big(p_u^{-1}(z_{su}) \cdot z_{su}\big)' &= p_u^{-1}(z_{su}) + z_{su} \cdot \big(p_u^{-1}(z_{su})\big)' \\ &= y_u + p_u(y_u)/p'_u(y_u)\end{aligned} \tag{15}$$

Here, we consider z^1_{su} and z^2_{su} satisfying $z^1_{su} \leq z^2_{su}$. Let y^1_u satisfy $z^1_{su} = p_u(y^1_u)$ and y^2_u satisfy $z^2_{su} = p_u(y^2_u)$. Then, $y^1_u \geq y^2_u$ since p_u is monotonically decreasing from Assumption 1. From (15) and Lemma 1, $\big(p_u^{-1}(z^1_{su}) \cdot z^1_{su}\big)' \geq \big(p_u^{-1}(z^2_{su}) \cdot z^2_{su}\big)'$. Because $\big(p_u^{-1}(z_{su}) \cdot z_{su}\big)'$ is monotonically decreasing, $p_u^{-1}(z_{su}) \cdot z_{su}$ is concave. It holds for all $u \in U$. $\square$

References

Adamczyk, M. 2011. Improved analysis of the greedy algorithm for stochastic matching. *Information Processing Letters* 111(15): 731–737.

Aggarwal, G.; Goel, G.; Karande, C.; and Mehta, A. 2011. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *SODA*, 1253–1264.

Ahuja, R. K.; Magnanti, T. L.; and Orlin, J. B. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall.

Babaioff, M.; Dughmi, S.; Kleinberg, R.; and Slivkins, A. 2015. Dynamic Pricing with Limited Supply. *ACM Transactions on Economics and Computation* 3(1): 1–26.

Bansal, N.; Gupta, A.; Li, J.; Mestre, J.; Nagarajan, V.; and Rudra, A. 2010. When LP Is the Cure for Your Matching Woes: Improved Bounds for Stochastic Matchings. *Algorithmica* 63(4):218–229.

Barlow, R. E.; Marshall, A. W.; and Proschan, F. 1963. Properties of probability distributions with monotone hazard rate. *The Annals of Mathematical Statistics* 34(2): 375–389.

Birge, J. R.; and Louveaux, F. 2011. *Introduction to stochastic programming*. Springer Science & Business Media.

Blum, A.; Dickerson, J. P.; Haghtalab, N.; Procaccia, A. D.; Sandholm, T.; and Sharma, A. 2015. Ignorance is almost bliss: Near-optimal stochastic matching with few queries. In *EC*, 325–342.

Boyd, S.; and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge University Press.

Brochu, E.; Cora, V. M.; and De Freitas, N. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599* .

Buckley, C.; Lease, M.; and Smucker, M. D. 2010. Overview of the TREC 2010 Relevance Feedback Track (Notebook). In *TREC*.

Chen, H.; Jiao, Y.; Qin, Z.; Tang, X.; Li, H.; An, B.; Zhu, H.; and Ye, J. 2019. InBEDE: Integrating Contextual Bandit with TD Learning for Joint Pricing and Dispatch of Ride-Hailing Platforms. In *ICDM*, 61–70.

Chen, N.; Immorlica, N.; Karlin, A. R.; Mahdian, M.; and Rudra, A. 2009. Approximating matches made in heaven. In *ICALP*, 266–278.

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

Gassmann, H. I. 1990. MSLiP: A computer code for the multistage stochastic linear programming problem. *Mathematical Programming* 47(1-3): 407–423.

Higle, J. L.; and Sen, S. 1991. Stochastic decomposition: An algorithm for two-stage linear programs with recourse. *Mathematics of operations research* 16(3): 650–669.

Higle, J. L.; and Sen, S. 2013. *Stochastic decomposition: a statistical method for large scale stochastic linear programming*. Springer Science & Business Media.

Horton, J. J.; and Chilton, L. B. 2010. The labor economics of paid crowdsourcing. In *EC*, 209–218.

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

Kiraly, Z.; and Kovacs, P. 2012. Efficient implementations of minimum-cost flow algorithms. *Acta Univ. Sapientiae* 4(1): 67–118.

Korte, B.; and Vygen, J. 2005. *Combinatorial Optimization: Theory and Algorithms, Third Edition*. Springer Publishing Company.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 661–670.

Mehta, A. 2012. Online Matching and Ad Allocation. *Theoretical Computer Science* 8(4): 265–368.

Myerson, R. B. 1981. Optimal auction design. *Mathematics of operations research* 6(1): 58–73.

Scott, W.; Frazier, P.; and Powell, W. 2011. The correlated knowledge gradient for simulation optimization of continuous parameters using gaussian process regression. *SIAM Journal on Optimization* 21(3): 996–1026.

Shapiro, A.; Dentcheva, D.; and Ruszczyński, A. 2014. *Lectures on stochastic programming: modeling and theory*. Society for Industrial and Applied Mathematics.

Tong, Y.; Wang, L.; Zhou, Z.; Chen, L.; Du, B.; and Ye, J. 2018. Dynamic Pricing in Spatial Crowdsourcing: A Matching-Based Approach. In *SIGMOD*, 773–788.

Van Slyke, R. M.; and Wets, R. 1969. L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics* 17(4): 638–663.

Végh, L. A. 2016. A strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives. *SIAM Journal on Computing* 45(5): 1729–1761.