

# An Improved Approximation Algorithm for Wage Determination and Online Task Allocation in Crowd-Sourcing

Yuya Hikima, Yasunori Akagi, Hideaki Kim, Taichi Asami

NTT Human Informatics Laboratories, NTT Corporation,  
1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847, Japan  
yuya.hikima@gmail.com, {yasunori.akagi.cu, hideaki.kin.cn, taichi.asami.ka}@hco.ntt.co.jp

## Abstract

Crowd-sourcing has attracted much attention due to its growing importance to society, and numerous studies have been conducted on task allocation and wage determination. Recent works have focused on optimizing task allocation and workers' wages, simultaneously. However, existing methods do not provide good solutions for real-world crowd-sourcing platforms due to the low approximation ratio or myopic problem settings. We tackle an optimization problem for wage determination and online task allocation in crowd-sourcing and propose a fast  $1 - 1/\sqrt{k+3}$ -approximation algorithm, where  $k$  is the minimum of tasks' budgets (numbers of possible assignments). This approximation ratio is greater than or equal to the existing method. The proposed method reduces the tackled problem to a non-convex multi-period continuous optimization problem by approximating the objective function. Then, the method transforms the reduced problem into a minimum convex cost flow problem, which is a well-known combinatorial optimization problem, and solves it by the capacity scaling algorithm. Synthetic experiments and simulation experiments using real crowd-sourcing data show that the proposed method solves the problem faster and outputs higher objective values than existing methods.

## 1. Introduction

Crowd-sourcing platforms have attained an important position in current society by offering task holders the means to undertake enormous tasks while providing workers with mass employment opportunities. Crowd-sourcing platforms offer a wide variety of tasks, such as checking the relevance of a topic to a web page (Buckley, Lease, and Smucker 2010) or translating a paragraph from German into English (Ho and Vaughan 2012), and the workers have different skills, such as familiarity with a specific topic or proficiency in a certain language (Fan et al. 2015). The task holders want to solve their set of tasks within a certain period of time at small payments (Singer and Mittal 2013).<sup>1</sup>

Under the above situation, many studies have been conducted on (a) worker-task matching (Ho and Vaughan 2012;

Zhao et al. 2019) and (b) wage determination for each worker (Singer and Mittal 2013; Mao et al. 2013). For goal (a), proper worker-task matching directly impacts the profits of task holders due to the diversity of tasks and workers; the quality of a translation task's outcome for a given language varies with the matched worker's level of familiarity with the language. For goal (b), in order to solve a set of tasks within a certain period of time, it is necessary to offer appropriate wages to workers to encourage them to undertake the task; if the deadline is close, it is necessary to offer high wages to encourage many workers to join in the task.

Recently, joint optimization of task assignment and wage determination has been tackled (Hikima et al. 2022, 2021). This is because offering different wages to each worker considering the assignment leads to more efficient matching, which reduces the task remaining and improves the quality of task outcomes. For example, a platformer can reduce a surplus of difficult tasks by attracting highly skilled workers with higher wages. The quality of certain tasks' outcomes can be improved by raising wages for suitable workers and allocating the tasks to them.

Although the joint optimization of task assignment and wage determination is important, existing methods cannot discover profitable solutions in practical crowd-sourcing operations. (Hikima et al. 2022) tackles an optimization problem for online matching and price (including wage) optimization so as to obtain larger profits for platformers. The problem can deal with reusable/non-reusable resources, including not only tasks in crowd-sourcing but also taxis in ride-sharing platforms. However, in compensation for being applicable to a wide range of resource types, the method of (Hikima et al. 2022) has a low approximation ratio ( $1/2$ -approximation ratio). Moreover, it empirically outputs solutions with low objective values for problems with non-reusable resources, such as tasks in crowd-sourcing platforms (see Section 2.1). (Hikima et al. 2021) also jointly optimizes price (including wage) and matching to obtain greater profits in matching platforms. However, because the optimization is performed separately for each period split from a time horizon, the platformer's profits are reduced in the long run.

To obtain profitable solutions in practical crowd-sourcing operations, we propose a new approximation algorithm for joint optimization of wages and online task assignment in

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

<sup>1</sup>Codes of our experiments, the proof of Lemma 1, and related work on online matching can be found in <https://github.com/Yuya-Hikima/AAAI2023-An-Improved-Approximation-Algorithm-for-Wage-Determination-and-Online-Task-Allocation>.

crowd-sourcing; the proposed approximation algorithm is fast, has a high approximation ratio, and empirically outputs solutions with high objective values. The tackled optimization problem is the problem of dealing with non-reusable resources, derived by fitting the optimization problem of (Hikima et al. 2022) to the application of crowd-sourcing. The problem, as in (Hikima et al. 2022), has the following difficulties: (i) we need to optimize not only wages but also the matching strategy (finding just the optimal matching strategy is hard (Manshadi, Gharan, and Saberi 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 workers arrivals. This paper proposes a high-performance approximation algorithm under specialized conditions where the resource is not reusable.

Our approximation algorithm finds an approximation solution by reducing our problem to a minimum convex cost flow problem; it is a well-known combinatorial optimization problem and has efficient existing methods. Our algorithm consists of the following steps. First, based on the result of (Alaei, Hajiaghayi, and Liaghat 2012), we approximate the objective value of our problem by the optimal value of a linear optimization problem. Using this approximation reduces our problem to a continuous optimization problem whose objective value can be evaluated easily. However, the reduced continuous optimization problem faces two difficulties; (I) it is non-convex because the worker’s acceptance probability function for the offered wage is non-convex; (II) it has a multi-period structure due to the matching policy needed for each time step when a worker arrives. To resolve (I), we reduce the non-convex problem to a convex one under mild assumptions on the worker’s acceptance probability function. Many important functions satisfy these assumptions. To deal with (II), we transform the multi-period problem into a single-period problem by the fact that the problem has the same convex objective function and the same constraints at each period. The transformed optimization problem can be reduced to the minimum convex cost flow problem, which is a well-known combinatorial optimization problem. By using the efficient algorithms available for the minimum convex cost flow problem, such as the capacity scaling algorithm (Ahuja, Magnanti, and Orlin 1993; Végh 2016), we can solve the transformed optimization problem at a low computation cost.

Our proposed method can output solutions (a) with higher objective values and (b) in shorter computation times than the existing method of (Hikima et al. 2022). For (a), our method finds  $(1 - 1/\sqrt{k+3})$ -approximation solutions, where  $k$  is the minimum value among all tasks’ budgets (numbers of possible assignments). Here, the approximation ratio becomes large in practice since  $k$  is assumed to be large in crowd-sourcing applications. For example, (Ho and Vaughan 2012) assumes that  $k$  is about 100, while (Zhang, Ma, and Sugiyama 2015) assumes that  $k$  is about 120-150. In such cases, the proposed method can obtain an approximation solution with approximation ratios of 0.9 or better, much higher than the  $1/2$ -approximation ratio achieved by (Hikima et al. 2022). For (b), the computational time, our method only solves a minimum convex cost flow problem

with a small dimensional decision vector. This allows us to quickly obtain a solution by efficient methods developed in the field of discrete optimization. Conversely, the existing method of (Hikima et al. 2022) applies the primal-dual hybrid gradient method directly to their reduced multi-period optimization problem with high dimension decision variables. This causes the computational time to be long since the method needs to update high dimension decision variables at each iteration.

We conduct synthetic experiments and simulation experiments using real data. The results show that the proposed method outputs solutions with higher objective values than existing methods more than 100 times faster.

## 2. Related Works

### 2.1 Wage Determination and Task Allocation in Crowd-Sourcing Operation

**Wage Determination.** Research on wage determination in crowd-sourcing can be divided into two main categories: (i) *wage posting*, where wages are determined by the platformer (Xiao et al. 2020; Mao et al. 2013), and (ii) *wage bargaining*, where wages are determined through worker bidding (Horton and Zeckhauser 2010; Singer and Mittal 2013; Wen et al. 2014). We focus on (i) *wage posting* because it is used in many platforms for the following reasons: it is easy for workers to understand the wage determination system and is also preferred by risk-averse workers (since wages are fixed in advance). (Xiao et al. 2020; Mao et al. 2013) perform price optimization/prediction in system (i) as in our study, but do not determine task assignments. In contrast, we simultaneously determine wages and task assignments to obtain greater profits.

**Task Allocation.** Research on task allocation in crowd-sourcing can be divided into two main categories: (a) *on-line task allocation* (Ho and Vaughan 2012; Ho, Jabbari, and Vaughan 2013), where each worker arrives one by one, and the platformer must allocate a task irreversibly; (b) *batch task allocation* (Zhao et al. 2019; Tran-Thanh et al. 2014), where the platformer decides the task allocation for multiple workers at a certain time instance or through multiple time steps. Our study focuses on category (a). This system has the advantage that workers do not have to wait because they can be assigned as soon as they arrive. (Ho and Vaughan 2012; Ho, Jabbari, and Vaughan 2013) optimize task allocation in system (a) as in our study, but do not optimize wages. We, however, optimize not only online task assignments but also workers’ wages to obtain greater profits.

**Joint Determination of Wage and Task Allocation.** Optimization focused on wages (or prices) and on task allocation have been addressed in several studies (Hikima et al. 2022, 2021). First, (Hikima et al. 2022) addresses the on-line matching problem with controllable rewards and arrival probabilities for reusable/non-reusable resources. We address almost the same problem as theirs since controlling worker wages in crowd-sourcing changes the arrival probabilities of workers and the rewards, which corresponds to the situation of (Hikima et al. 2022). The difference in problem

setting between that study and ours is our inability to handle reusable resources. While our study deals only with non-reusable resources (because tasks in crowd-sourcing need to be solved only once), (Hikima et al. 2022) can deal with reusable/non-reusable resources, which broadens the application regime (see (Hikima et al. 2022, Section 3.2)). However, despite its versatility, the existing method outputs solutions with low objective values for problems with non-reusable resources. The reason is that the matching strategy (Dickerson et al. 2018) adopted by the existing method does not allocate more than half of the resources to guarantee its approximation rate. Specifically, the probability that each edge  $e$  is available ( $\beta_{et}$  in the paper) must be maintained to at least  $1/2$  (see (Dickerson et al. 2018, Section 2)), which makes the expected value of each remaining resource at the end at least  $1/2$ . Therefore, in crowd-sourcing applications with non-reusable resources (tasks), the existing method will output a low-profit solution. Conversely, our method obtains higher objective values by conserving resources appropriately by adopting the matching strategy of (Alaei, Hajiaghayi, and Liaghat 2012), where each resource is conserved only if the reward from the allocation is less than the reduction in future rewards due to a decrease in the resource’s budget. Second, (Hikima et al. 2021) maximizes the platformer’s profit by jointly optimizing price (including wage) and resource-user matching in matching platforms to get greater profits within single periods split from a time horizon. However, because the optimization is performed separately for each period, the platformer’s profits are reduced in the long run. In contrast, our method optimizes wages and allocations throughout the time series and outputs profitable solutions in the long run.

In addition, (Minder et al. 2012; Anari, Goel, and Nikzad 2014; Zhao, Li, and Ma 2014) propose mechanism designs for wage determination and task allocation in crowd-sourcing platforms. The mechanisms make each worker share their cost of solving a task truthfully and then determine their wages and task assignment. However, these studies assume that all workers’ costs are asked in advance, which is not applicable to crowd-sourcing platforms where workers appear one after another. In contrast, our study assumes a realistic setting in which workers appear on the platform one by one and must immediately decide on wages and task assignments.

## 2.2 Optimization Problems with Decision-dependent Noise

Since the uncertainties (i.e., workers’ arrival probabilities) in our problem depend on the decision variables (i.e., wages), our optimization problem is classified as an optimization problem with decision-dependent/endogenous noise (Hellemo, Barton, and Tomasgard 2018). For the optimization problem, typical methods are search methods such as Bayesian optimization (Brochu, Cora, and De Freitas 2010) and random search (Bergstra and Bengio 2012). However, for our problem, they can not provide good solutions in practical time for the following reasons: (i) the dimension of the decision variables is too large to explore the feasible region adequately; (ii) it takes a lot of time to compute the exact

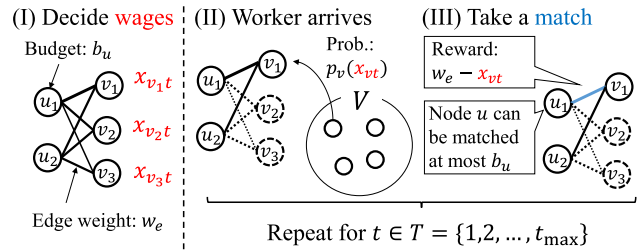


Figure 1: Illustration of system procedure

objective value or its approximation with high accuracy. In contrast, our method can output a theoretically guaranteed approximation solution quickly.

## 3. Preliminary

### 3.1 Matching Procedure

**Notation.** In this paper, we consider allocating tasks to workers who arrive one by one in each round  $t \in T := \{1, 2, \dots, t_{\max}\}$ . The task set  $U := \{u_1, \dots, u_m\}$ , the worker set  $V := \{v_1, \dots, v_n\}$ , and the bipartite graph  $G = (U, V; E)$  are given. Each edge  $(u, v) \in E$  represents that worker  $v$  is qualified to work on task  $u$ , and all  $u \in U$  and  $v \in V$  are incident to at least one  $e \in E$ . For  $e = (u, v) \in E$ , if task  $u$  is matched to worker  $v$  at time  $t$ , the platformer receives reward  $w_e \in \mathbb{R}$  from the task-holder, and the platformer pays the worker  $v$  wage  $x_{vt} \in \mathbb{R}$ . Thus, the platformer receives a total of  $w_e - x_{vt}$  rewards. Here,  $w_e$  for each  $e \in E$  is a given constant, whereas  $x_{vt}$  can be decided by the platformer. Task  $u \in U$  has budget  $b_u \in \mathbb{Z}_{>0}$ ; once  $u$  is allocated  $b_u$  times, then  $u$  cannot be allocated anymore. We assume  $b_u \leq |T|$  for all  $u \in U$  w.l.o.g. Here, we let  $k := \min_{u \in U} b_u$ .

**System Procedure.** We consider the following system procedure (see Fig. 1). (I) The platformer determines the value of wage  $x_{vt}$  for each  $v \in V$  and  $t \in T$ , and sets  $\hat{b}_u = b_u$  for all  $u \in U$  as the current budget of resource  $u$ . Then, (II) and (III) are repeated in each round  $t \in T$ . (II) Worker  $v \in V$  arrives with probability  $p_v(x_{vt})$ , or no workers arrive with probability  $1 - \sum_{v \in V} p_v(x_{vt})$ . Here,  $p_v : \mathbb{R} \rightarrow (0, r_v)$  is a given monotonically increasing function, and  $r_v \in \mathbb{R}_{>0}$  is a given constant such that  $\sum_{v \in V} r_v = 1$ . (This means that worker  $v$  appears with probability  $r_v$  and accepts the wage with probability 0 to 1.) (III) If worker  $v \in V$  arrives, the platformer chooses either (a) to assign one remaining task  $u \in \{u \in U \mid \hat{b}_u > 0\}$  to the worker, gets the reward of  $w_e - x_{vt}$ , and sets  $\hat{b}_u := \hat{b}_u - 1$ , where  $e = (u, v)$ , or (b) not assign any task.

**Differences from Matching Procedure of (Hikima et al. 2022)** Although our matching procedure is similar to that of (Hikima et al. 2022), there are three differences; (i) the existing study can treat reusable/non-reusable resources as  $u$ , while our study is limited to non-reusable ones; (ii) the existing study could set the arrival probability function  $p_v$  and the edge weight  $w_e$  to be different at each time (such as  $p_{vt}$  and  $w_{et}$ ), while our study assumes they are the same

at each time; (iii) in the existing study, budget  $b_u$  is 1 for each  $u \in U$ , whereas  $b_u \in \mathbb{Z}_{>0}$  in our study. While our matching procedure is a special case of (Hikima et al. 2022) by replicating node  $u$  at  $b_u$  times and setting  $b_u := 1$  for all  $u \in U$ , it is suitable for crowd-sourcing applications (Ho and Vaughan 2012); first, in crowd-sourcing, tasks are non-reusable because it does not need to be solved again once it has been solved; second, the reward for solving each task and each worker's arrival probability are usually considered to be independent of time. Assuming this special case, we are able to propose a fast approximation algorithm with a high approximation ratio guarantee for the profit-maximization problem described in Section 3.2.

### 3.2 Optimization Problem

We consider an optimization problem to maximize the total rewards of the platformer in the matching procedure in Section 3.1. We call our problem *Joint Optimization for Wage determination and Online task allocation* (JO-WO):

$$(JO-WO) \quad \max_{\mathbf{x} \in \mathbb{R}^{V \times T}, \pi \in \Pi} \mathbb{E}_{\xi \sim D(\mathbf{x})} [f(\pi, \mathbf{x}, \xi)].$$

Here, variable  $\pi$  represents a matching strategy in (III) of the matching procedure in Section 3.1; Symbol  $\Pi$  is the set of all strategies. A matching strategy specifies, upon arrival of worker  $v \in V$ , whether to match it, and if so, which task  $u \in U$  to match it to. Symbol  $\xi \in \{V \cup \{\emptyset\}\}^T$  is a random variable, where  $\xi_t = v$  means that worker  $v$  arrives at time  $t$ , and  $\xi_t = \emptyset$  means no workers arrive at time  $t$ .  $D(\mathbf{x})$  is a probability distribution for  $\xi \in \{V \cup \{\emptyset\}\}^T$ ; The probability mass function is  $\Pr(\xi | \mathbf{x}) = \prod_{t \in T} \Pr(\xi_t | \mathbf{x})$ , where  $\Pr(\xi_t = v | \mathbf{x}) = p_v(x_{vt})$  and  $\Pr(\xi_t = \emptyset | \mathbf{x}) = 1 - \sum_{v \in V} p_v(x_{vt})$ . Function  $f(\pi, \mathbf{x}, \xi)$  is the expected total rewards obtained from performing the matching procedure given  $(\pi, \mathbf{x}, \xi)$ .

### 3.3 Assumption on $p_v$

We assume the following throughout this paper.

**Assumption 1.** *Function  $p_v : \mathbb{R} \rightarrow (0, r_v)$  is monotonically increasing, differentiable, and bijective. Moreover,  $-\frac{p'_v(x)}{p_v(x)}$  is monotonically non-decreasing with respect to  $x$ .*

Assumption 1 is not so restrictive because many functions used in real applications satisfy Assumption 1. For example,  $p_v(x) = r_v F(x)$  satisfies Assumption 1 when  $F(x)$  is a Gauss error function or a logistic function.<sup>2</sup>

### 3.4 Minimum Cost Flow Problem

To find an approximation solution in later sections, we utilize the minimum cost (min-cost) flow problem (Ahuja,

<sup>2</sup>When  $F(x)$  is a twice-differentiable distribution function and  $\frac{F'(x)}{1-F(x)}$  is monotonically non-decreasing,  $F$  is called a log-concave function (Bagnoli and Bergstrom 2006) or Monotone Hazard Rate function (Barlow, Marshall, and Proschan 1963). This type of function is frequently used in dynamic pricing literature to model the customer's acceptance probability for the offered prices (Babaioff et al. 2015; Tong et al. 2018).

Magnanti, and Orlin 1993), which is a well-known combinatorial optimization problem. Therefore, we give a preliminary description of the min-cost flow problems here. Let  $\hat{G} = (\hat{V}, \hat{E})$  be a directed graph, where  $\hat{V}$  is a node set and  $\hat{E}$  is an edge set. There are a cost function  $c_{ij} : \mathbb{R} \rightarrow \mathbb{R}$  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,  $\ell_i \in \mathbb{R}$ , which is called the supply (demand) of the node when  $\ell_i > 0$  (when  $\ell_i < 0$ ). Given the above, the min-cost flow problem can be written as follows:

$$(MCF) \quad \min_{\mathbf{z}} \sum_{(i,j) \in \hat{E}} c_{ij}(z_{ij})$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in \hat{E}} z_{ij} - \sum_{j:(j,i) \in \hat{E}} z_{ji} = \ell_i, \quad \forall i \in \hat{V},$$

$$0 \leq z_{ij} \leq \ell_{ij}, \quad \forall (i,j) \in \hat{E}.$$

If the cost function  $c_{ij}$  is non-convex, then min-cost flow problems are generally NP-hard. However, if the cost functions are convex, several existing methods can solve min-cost flow problems efficiently. In Section 4.3, to find an approximation solution of JO-WO, we use the capacity scaling algorithm (Ahuja, Magnanti, and Orlin 1993; Végé 2016). The algorithm explained in (Ahuja, Magnanti, and Orlin 1993) solves the convex min-cost flow problems that restrict the decision variable  $\mathbf{z}$  to integer values. However, we can use this method to obtain an optimal solution in a continuous domain within a small arbitrary error by the following procedure: (i) substitute  $\epsilon \cdot y_{ij}$  into  $z_{ij}$  for each  $(i, j)$ , where  $\epsilon$  is a sufficiently small value and  $y_{ij} \in \mathbb{Z}^{|E|}$ ; (ii) find an integer optimal solution  $\mathbf{y}^*$  of the transformed problem; (iii) let  $\mathbf{z}_{ij}^* := \epsilon \cdot \mathbf{y}_{ij}^*$ . Then,  $\mathbf{z}^*$  is an optimal solution of the original problem within error  $\epsilon$ .

## 4. Proposed Method

Solving JO-WO is difficult for the following reasons. (i) We need to optimize the matching strategy  $\pi$  and variable  $\mathbf{x}$  jointly. However, even finding just the optimal matching strategy is hard (Manshadi, Gharan, and Saberi 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  $\xi$  over  $(|V| + 1)^{|T|}$  realizations.

### 4.1 Multi-period Convex Continuous Problem for Approximation Solution of JO-WO

To obtain an approximation solution of JO-WO, we introduce the following problem. It is derived by approximating the objective function  $\max_{\pi \in \Pi} \mathbb{E}_{\xi \sim D(\mathbf{x})} [f(\pi, \mathbf{x}, \xi)]$  of JO-WO by a linear optimization problem, *Expected LP* of (Alaei, Hajiaghayi, and Liaghat 2012).

$$(PA) \quad \max_{\substack{\mathbf{x} \in \mathbb{R}^{V \times T} \\ \mathbf{z} \in [0,1]^{E \times T}}} \sum_{t \in T} \sum_{e=(u,v) \in E} (w_e - x_{vt}) z_{et}$$

$$\text{s.t.} \quad \sum_{e \in \delta(v)} z_{et} \leq p_v(x_{vt}), \quad \forall v \in V, t \in T,$$

$$\sum_{t \in T} \sum_{e \in \delta(u)} z_{et} \leq b_u, \quad \forall u \in U.$$

Inspired by (Hikima et al. 2022, Lemma 1), we show the following lemma (the proof is slightly different).

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

Then, we show the following theorem from the result of (Alaei, Hajiaghayi, and Liaghat 2012). It shows that solving (PA) yields a  $\left(1 - \frac{1}{\sqrt{k+3}}\right)$ -approximation solution to JO-WO, where  $k := \min_{u \in U} b_u$ .

**Theorem 2.** Suppose that Assumption 1 holds. Let  $(\hat{x}, \hat{z})$  be an optimal solution of (PA). Consider the following matching strategy  $\pi^{\text{Alaei}}(\hat{x}, \hat{z})$ : when vertex  $v$  arrives at time  $t$ ,

1. Choose task  $u$  with probability  $\hat{z}_{et}/p_v(\hat{x}_{vt})$ , where  $e = (u, v)$ , or don't choose any task with probability  $1 - \sum_{e \in \delta(v)} \hat{z}_{et}/p_v(\hat{x}_{vt})$ .
2. If task  $u$  is chosen and  $w_e + \mathcal{E}(u, \hat{b}_u - 1, t + 1) > \mathcal{E}(u, \hat{b}_u, t + 1)$ , then take the match of  $e = (u, v)$ , where  $\mathcal{E}(u, \hat{b}_u, t)$  is the expected total rewards we get from task  $u$  with remaining capacity  $\hat{b}_u$  at time  $t$ . Otherwise, don't take any match.

Then,  $(\hat{x}, \pi^{\text{Alaei}}(\hat{x}, \hat{z}))$  is a  $\left(1 - \frac{1}{\sqrt{k+3}}\right)$ -approximation solution for JO-WO.

Here, we can compute  $\mathcal{E}(u, \hat{b}_u, t)$  for all  $u, \hat{b}_u$ , and  $t$  in advance by recursive calculations from  $t = |T|$  to  $t = 1$  (Alaei, Hajiaghayi, and Liaghat 2012).

Although (PA) is a non-convex optimization problem since function  $p_v(x_{vt})$  is non-convex, we can reduce (PA) to a convex problem under Assumption 1. First, we introduce the following minimization problem, which is obtained by eliminating  $x$  from (PA) by using  $x_{vt} := p_v^{-1}(\sum_{e \in \delta(v)} z_{et})$ :

$$\begin{aligned} (\text{PA}') \quad & \min_{z \in [0,1]^{E \times T}} \sum_{t \in T} \sum_{v \in V} p_v^{-1}\left(\sum_{e \in \delta(v)} z_{et}\right) \sum_{e \in \delta(v)} z_{et} \\ & - \sum_{t \in T} \sum_{e=(u,v) \in E} w_e z_{et} \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} z_{et} \in S_v, \quad \forall v \in V, t \in T, \\ & \sum_{t \in T} \sum_{e \in \delta(u)} z_{et} \leq b_u, \quad \forall u \in U. \end{aligned}$$

where  $S_v$  is the range of function  $p_v$ . Then, we can show the following theorem, inspired by (Hikima et al. 2022, Proposition 4).

**Theorem 3.** Suppose that Assumption 1 holds. Let the optimal solution of (PA') be  $z^*$  and  $x_{vt}^* := p_v^{-1}(\sum_{e \in \delta(v)} z_{et}^*)$  for all  $v \in V$  and  $t \in T$ . Then,  $(x^*, z^*)$  is an optimal solution for (PA).

Moreover, the following theorem holds from the result of (Hikima et al. 2021, Theorem 3).

**Theorem 4.** When Assumption 1 holds, the objective function of (PA') is convex, and then (PA') is a convex optimization problem. Specifically,  $p_v^{-1}(\sum_{e \in \delta(v)} z_{et}) \sum_{e \in \delta(v)} z_{et} - \sum_{e \in \delta(v)} w_e z_{et}$  is convex respect to  $z_{vt} := \{z_{et}\}_{e \in \delta(v)}$  for all  $(v, t) \in V \times T$ .

Therefore, by solving convex optimization problem (PA'), we can obtain a  $1 - \frac{1}{\sqrt{k+3}}$ -approximation solution

for JO-WO. Here, since  $k = \min_{u \in U} b_u \geq 1$ , the approximation ratio  $1 - \frac{1}{\sqrt{k+3}}$  is greater than or equal to  $\frac{1}{2}$ , the approximation ratio achieved in (Hikima et al. 2022). Moreover,  $1 - \frac{1}{\sqrt{k+3}}$  becomes large in practice since  $k$  is assumed to be large in crowd-sourcing applications. For example, (Ho and Vaughan 2012) assumes that  $k$  is about 100, and (Zhang, Ma, and Sugiyama 2015) assumes that  $k$  is about 120-150. In such cases, the proposed method can obtain approximation solutions with approximation ratios of 0.9 or better.

## 4.2 Reduce (PA') to Single Period Convex Optimization Problem

We can reduce (PA') to a single period convex problem. We consider the following problem:

$$\begin{aligned} (\text{SCP}) \quad & \min_{z \in [0,1]^E} \sum_{v \in V} p_v^{-1}\left(\sum_{e \in \delta(v)} z_e\right) \sum_{e \in \delta(v)} z_e - \sum_{e=(u,v) \in E} w_e z_e \\ \text{s.t.} \quad & \sum_{e \in \delta(v)} z_e \in S_v, \quad \forall v \in V, \\ & \sum_{e \in \delta(u)} z_e \leq \frac{b_u}{|T|}, \quad \forall u \in U. \end{aligned}$$

Here, we show the following theorem.

**Theorem 5.** Suppose that Assumption 1 holds. Let  $\tilde{z}$  be an optimal solution for (SCP). Then,  $z^*$  such that  $z_{e1}^* = z_{e2}^* = \dots = z_{e|T|}^* = \tilde{z}_e$  is an optimal solution for (PA').

## 4.3 Solve (SCP) by Reducing Min-cost Flow Problem

Next, we solve (SCP) by reducing it to a min-cost flow problem. First, we prepare new subscripts  $s$  and  $d$ . Let  $z_{su} := \sum_{e \in \delta(u)} z_e$  for all  $u$  and  $z_{vd} := \sum_{e \in \delta(v)} z_e$  for all  $v$ . Let  $z_{sd} \in [0, 1]$  be a slack variable. Then, (SCP) can be written as follows.

$$(\text{FP}) \quad \min_z \sum_{v \in V} p_v^{-1}(z_{vd}) z_{vd} - \sum_{e \in E} w_e z_e \quad (1)$$

$$\text{s.t.} \quad \sum_{u \in U} z_{su} + z_{sd} = 1, \quad \sum_{v \in V} z_{vd} + z_{sd} = 1, \quad (2)$$

$$z_{su} - \sum_{e \in \delta(u) \cap E} z_e = 0, \quad \forall u \in U, \quad (3)$$

$$\sum_{e \in \delta(v) \cap E} z_e - z_{vd} = 0, \quad \forall v \in V, \quad (4)$$

$$0 \leq z_{su} \leq \frac{b_u}{|T|}, \quad \forall u \in U, \quad (5)$$

$$z_{vd} \in S_v, \quad \forall v \in V, \quad (6)$$

$$0 \leq z_e \leq 1, \quad \forall e \in E, \quad (7)$$

$$0 \leq z_{sd} \leq 1. \quad (8)$$

This is a min-cost flow problem for the graph with  $U \cup V \cup \{s, d\}$  as node sets (illustrated in Fig. 2). The objective function (1) corresponds to the cost function for the flow amount of each edge;  $p_v^{-1}(z_{vd}) z_{vd}$  is the cost function for each edge of  $\{(v, d) \mid v \in V\}$ , and  $-w_e z_e$  is the cost function for each edge  $e \in E$ . The constraints (2)-(4) represent the demand/supply for each node; the supply at the node  $s$  is 1, the demand at the node  $d$  is  $-1$ , and the demands/supplies at other nodes are 0. (FP) is the problem of finding a way of sending 1 amount of flow from node  $s$  to node  $d$  in the

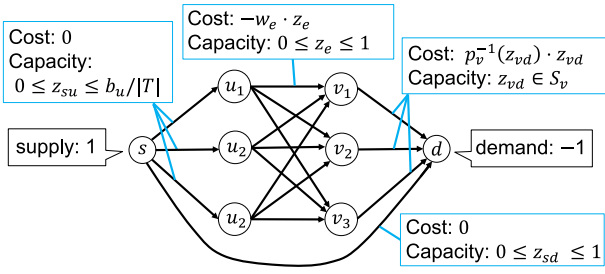


Figure 2: Illustration of (FP)

network. The constraints (5)-(8) represent the flow amount capacity for each edge.

Because (FP) is a minimum convex cost flow problem from Theorem 4, we can find a solution by the capacity scaling algorithm (Ahuja, Magnanti, and Orlin 1993; Végth 2016) within error  $\epsilon$  in  $O(|E|^2 \cdot \log(1/\epsilon) \cdot \log(|U| + |V|))$  time. For a formal convex min-cost flow problem (MCF) in Section 3.4, (Ahuja, Magnanti, and Orlin 1993, Theorem 14.1) shows that the capacity scaling algorithm finds an integer optimal solution in  $O(|\hat{E}| \cdot \log(\hat{L}) \cdot \hat{S})$  time, where  $\hat{L} := \max\{\max_{i \in \hat{V}} |\ell_i|, \max_{(i,j) \in \hat{E}} \ell_{ij}\}$ , and  $\hat{S}$  is the time complexity for solving the shortest path problem in graph  $\hat{G}$  with non-negative edge costs. Since shortest path problems can be solved by Dijkstra's algorithm with binary heap in  $O(|\hat{E}| \cdot \log|\hat{V}|)$  time, the total time complexity is  $O(|\hat{E}|^2 \cdot \log(\hat{L}) \cdot \log|\hat{V}|)$ . When we use this algorithm to find an optimal solution of a convex min-cost flow problem in a continuous domain within error  $\epsilon$ , the total time complexity is  $O(|\hat{E}|^2 \cdot \log(\hat{L}/\epsilon) \cdot \log|\hat{V}|)$  because substituting  $\epsilon \cdot y_{ij}$  into  $z_{ij}$  causes  $\hat{L}$  to be multiplied by  $1/\epsilon$ . Then, for (FP), the capacity scaling algorithm can find an optimal solution within error  $\epsilon$  in  $O(|E|^2 \cdot \log(1/\epsilon) \cdot \log(|U| + |V|))$  time.

#### 4.4 Novelty from (Hikima et al. 2022)

The novelty of our method from (Hikima et al. 2022) is the way of reduction to a continuous optimization problem in Section 4.1 and all procedures in Sections 4.2 and 4.3. In Section 4.1, we approximate the objective value of JO-WO by an optimal value of a linear problem by the results of (Alaei, Hajiaghayi, and Liaghat 2012), while the existing ones use the results of (Dickerson et al. 2018). This difference leads to an improvement in the approximation ratio from  $1/2$  to  $1 - 1/\sqrt{k+3}$  ( $\geq 1/2$ ). Furthermore, we propose a way of reducing a multi-period problem to a single-period problem in Section 4.2, and transform the reduced problem into the minimum convex cost flow problem in Section 4.3. In contrast, the existing method directly solves their reduced multi-period optimization problem by the primal-dual hybrid gradient method.

## 5. Experiments

We conducted experiments to show that our algorithm outputs solutions with higher total rewards in shorter compu-

tation times than the baselines. We performed synthetic experiments and simulation experiments using real data from a crowd-sourcing platform. The settings of both experiments are based on (Hikima et al. 2022). 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 3.6.8.

**Parameter Setting of Proposed Method** We set  $\epsilon = 0.01$  as the error tolerance when solving (FP) by the proposed method.

**Baselines** We implemented the following baselines.

**OM-CRA(Hikima et al. 2022):** This method can be used for our problem where  $b_u = 1$  for all  $u \in U$ . We apply it by replicating  $u$  at  $b_u$  times and setting  $b_u := 1$  for all  $u \in U$ . We set the allowable error for solving equation (6) in the paper of (Hikima et al. 2022) as  $10^{-8}$ , and iterations were performed until the primal and dual residuals were less than  $10^{-5}$  or until the maximum number of iterations, 20 (chosen from  $\{20, 30, 40\}^3$ ), was reached.

**BO-A:** We apply Bayesian optimization (Brochu, Cora, and De Freitas 2010) to search  $\mathbf{x}$  while adopting the  $1 - 1/\sqrt{k+3}$ -approximation matching strategy (Alaei, Hajiaghayi, and Liaghat 2012). BO-A first evaluates five random points of  $\mathbf{x}$ . Then, it runs the Bayesian optimization for 1000 seconds and outputs the solution with the highest objective value among the evaluated points. The set to search for  $\mathbf{x}$  is  $[0, 1]^{V \times T}$ .

**RS-A:** We apply the random search to variable  $\mathbf{x}$  while adopting the  $1 - 1/\sqrt{k+3}$ -approximation matching strategy (Alaei, Hajiaghayi, and Liaghat 2012). In each iteration, this method randomly generates a search point from the set  $[0, 1]^{V \times T}$  and evaluates the objective value. Then, it outputs the solution with the highest objective value among the evaluated points in 1000 seconds.

**Metric** We use *Expected Total Rewards* (ETR) for given  $\mathbf{x}$  and  $\pi$ :

$$\text{ETR} := \frac{1}{10^3} \sum_{k=1}^{10^3} f(\pi, \mathbf{x}, \boldsymbol{\xi}^k),$$

where  $\boldsymbol{\xi}^k \sim D(\mathbf{x})$ . This is the approximation of the objective function of (P) by  $10^3$  simulations.

### 5.1 Synthetic Experiments

**Synthetic Parameter Setup.** We determined the default settings of problem parameters and then performed experiments by varying each parameter. First, we explain the default settings. We set  $m := 20$ ,  $n := 40$ ,  $|E| := mn$ , and  $|T| := 200$ . Parameter  $b_u$  for each  $u$  was generated from a uniform distribution of  $[5, 15]$ , and parameter  $w_e$  for each  $e$  was generated from a uniform distribution of  $[0.4, 0.8]$ . We set  $p_v(x) := \frac{r_v}{1 + e^{-(x - q_v)/(\gamma|q_v|)}}$ , where  $\gamma := 0.1\sqrt{3}/\pi$ ,  $q_v$  is generated from a uniform distribution of  $[0.5, 0.8]$  for each

<sup>3</sup>We set 20 as the maximum number of iterations because the obtained ETR was almost the same as for 30 or 40 iterations in the simulation experiment with real data at the default settings.

$v \in V$ , and  $r_v = \frac{y_v}{\sum_{v \in V} y_v}$ . Here,  $y_v$  is generated from a uniform distribution of  $[0.01, 1]$  for each  $v$ .<sup>4</sup> Then, under the default settings, we perform experiments by changing  $m$ ,  $n$ ,  $T$ , and the closed interval for  $b_u$ .

**Experimental Results** Fig. 3 shows the results of the simulation experiments with different parameter values. First, for all parameters, the proposed method outperforms baselines in terms of ETR. This is due to the following reasons; (i) **OM-CRA** excessively conserves resources for problems with non-reusable resources, as is described in Section 2.1; (ii) **BO-A** and **RS-A** cannot adequately explore  $\mathbf{x}$  since (a) the dimension of  $\mathbf{x}$  is too large and (b) it takes a lot of time to evaluate the objective value at each search point. Moreover, in Fig. 3g, the proposed method further outperforms the baselines as  $b_u$  increases. This result is consistent with the theoretically derived approximation ratio since it increases as the minimum value ( $k$ ) of  $b_u$  increases. In terms of computation time, the proposed method outputs solutions at least 100 times faster than the baselines.

## 5.2 Simulation Experiments with Real Data

**Data Set and Parameter Setup** We used an open crowdsourcing dataset by (Buckley, Lease, and Smucker 2010). The dataset records workers' judgments on the task of checking the relevance between a given topic and a web page. We set each parameter by the data as follows:

- (i)  $U$ : From all tasks,  $m$  tasks are chosen as task set  $U$ .
- (ii)  $V$ : From all workers,  $n$  workers are chosen as the arriving worker set  $V$ .
- (iii)  $w_e$ : We set the correct judgment for each task in the data by majority vote. Let  $\phi_s^v$  be the percentage of correct answers of worker  $v$  for topic  $s$ . Then, we set  $w_e := \phi_{s(u)}^v$  for each  $e = (u, v)$ , where  $s(u)$  is the topic of task  $u$ . This setting is based on a scheme that determines the value of solving a task according to the worker's skill. When worker  $v$  has never solved tasks with topic  $s$ , we let  $\phi_s^v$  be the average percentage of correct answers of all workers in the data.

As the default settings, we set  $m := 20$ ,  $n := 40$ , and  $|T| := 200$ . Parameter  $b_u$  for each  $u$  was generated from a uniform distribution of  $[5, 15]$ . We use the same setting for  $p_v$  as synthetic experiments. Then, we performed experiments by varying  $n$ ,  $m$ ,  $T$ , and  $b_u$  as in the synthetic experiments.

**Experimental Results** Fig. 4 shows the results of the simulation experiments with different parameter values. The proposed method outperforms all baselines in terms of ETR in all parameters. In contrast to the simulation experiments, the proposed method outperforms the baselines even when  $b_u$  is small. Moreover, in terms of computation time, the proposed method outperforms the baselines in all parameters.

<sup>4</sup>This setting assumes worker  $v$  appears with probability  $r_v$  and the worker's reservation wage (the minimum wage willing to work) follows a logistic distribution with mean  $q_v$  and standard deviation of  $0.1q_v$ .

## 6. Conclusion

We tackled an optimization problem, JO-WO, to jointly optimize workers' wages and online task allocation in crowdsourcing platforms. We proposed a fast  $1 - 1/\sqrt{k+3}$ -approximation algorithm for JO-WO, where  $k$  is the minimum value among all budgets of tasks. This approximation ratio is greater than or equal to existing ones. Moreover, the proposed method empirically outputs solutions with higher objective value than existing methods. Synthetic experiments and simulation experiments with real data confirmed the effectiveness of the proposed method.

Future work includes extending our method to more diverse crowdsourcing settings; e.g. a setting that allows one worker to take multiple tasks, a setting that allows multiple workers to cooperate to solve a single task, or a setting that multiple workers appear simultaneously at each time step.

## 7. Proofs

### 7.1 Proof of Theorem 2

We introduce the following problem for a given  $\mathbf{x}$ , and denote the optimal value by  $h(\mathbf{x})$ :

$$\begin{aligned} & \text{(Expected LP)} \\ & \max_{\mathbf{z} \in [0, 1]^{E \times T}} \sum_{t \in T} \sum_{e=(u,v) \in E} (w_e - x_{vt}) z_{et} \\ & \text{s.t.} \quad \sum_{e \in \delta(v)} z_{et} \leq p_v(x_{vt}), \quad \forall v \in V, \forall t \in T, \\ & \quad \quad \sum_{t \in T} \sum_{e \in \delta(u)} z_{et} \leq b_u, \quad \forall u \in U. \end{aligned}$$

Let  $\mathbf{z}(\mathbf{x})$  be an optimal value for (Expected LP). Then, the following holds for any  $\mathbf{x} \in \mathbb{R}^{V \times T}$  and  $\pi \in \Pi$  from (Alaei, Hajiaghayi, and Liaghat 2012),

$$\begin{aligned} \left(1 - \frac{1}{\sqrt{k+3}}\right) h(\mathbf{x}) & \leq \mathbb{E}_{\xi \sim D(\mathbf{x})} [f(\pi^{\text{Alaei}}(\mathbf{x}, \mathbf{z}(\mathbf{x})), \mathbf{x}, \xi)] \\ & \leq \mathbb{E}_{\xi \sim D(\mathbf{x})} [f(\pi, \mathbf{x}, \xi)] \leq h(\mathbf{x}). \end{aligned} \quad (9)$$

Since problem (PA) is equivalent to  $\max_{\mathbf{x} \in \mathbb{R}^{V \times T}} h(\mathbf{x})$ , it yields  $h(\hat{\mathbf{x}}) \geq h(\mathbf{x})$  for any  $\mathbf{x} \in \mathbb{R}^{V \times T}$ . Then, for any  $(\mathbf{x}, \pi) \in \mathbb{R}^{V \times T} \times \Pi$ , we have

$$\begin{aligned} \left(1 - \frac{1}{\sqrt{k+3}}\right) \mathbb{E}_{\xi \sim D(\mathbf{x})} [f(\pi, \mathbf{x}, \xi)] & \leq \left(1 - \frac{1}{\sqrt{k+3}}\right) h(\mathbf{x}) \\ & \leq \left(1 - \frac{1}{\sqrt{k+3}}\right) h(\hat{\mathbf{x}}) \leq \mathbb{E}_{\xi \sim D(\hat{\mathbf{x}})} [f(\pi^{\text{Alaei}}(\hat{\mathbf{x}}, \mathbf{z}(\hat{\mathbf{x}})), \hat{\mathbf{x}}, \xi)], \end{aligned}$$

where first and third inequalities hold from (9). Since  $\mathbf{z}(\hat{\mathbf{x}}) = \hat{\mathbf{z}}$  from the definition, the above inequality means  $(\hat{\mathbf{x}}, \pi^{\text{Alaei}}(\hat{\mathbf{x}}, \hat{\mathbf{z}}))$  is a  $\left(1 - \frac{1}{\sqrt{k+3}}\right)$ -approximation solution for JO-WO.  $\square$

### 7.2 Proof of Theorem 3

Let  $(\hat{\mathbf{x}}, \hat{\mathbf{z}})$  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  $\hat{v} \in V$  and  $\hat{t} \in T$  satisfying  $\sum_{e \in \delta(\hat{v})} \hat{z}_{e\hat{t}} = 0$  to obtain a contradiction. We pick an arbitrary vertex  $\hat{e} \in \delta(\hat{v})$ . Then, there exists (sufficiently small)  $x_M$  satisfying  $w_{\hat{e}} - x_M > \sum_{(e=(u,v), t) \in E \times T \setminus \{\hat{e}, \hat{t}\}: w_e - \hat{x}_{vt} \geq 0} \{w_e - \hat{x}_{vt}\}$ .

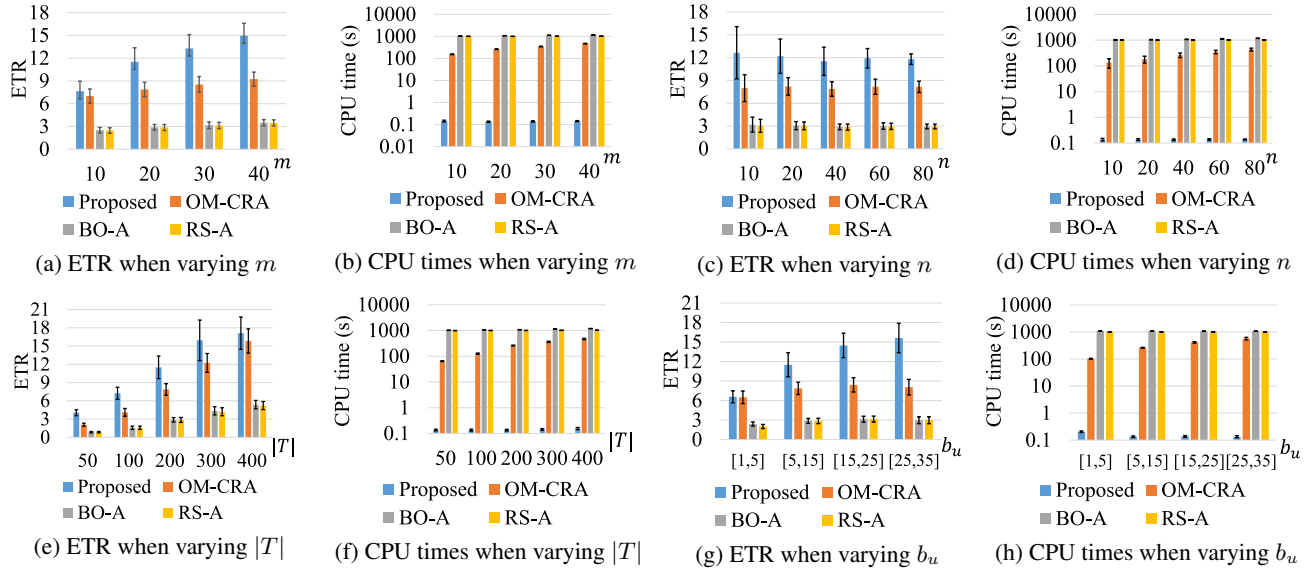


Figure 3: Results of synthetic experiments. ETR and CPU times (in seconds) with error bars of standard deviation are shown when varying parameters. The vertical axis of each graph for CPU time is a logarithmic scale. Upper (Lower) values are better in ETR (CPU times). Each result is the average of 30 simulations. The differences between the proposed method and all baselines are significant for ETR and CPU time (two-sided t-test:  $p < 0.01$ ) in all parameters other than ETR of  $b_u \in [1, 5]$  in Fig. 3g.

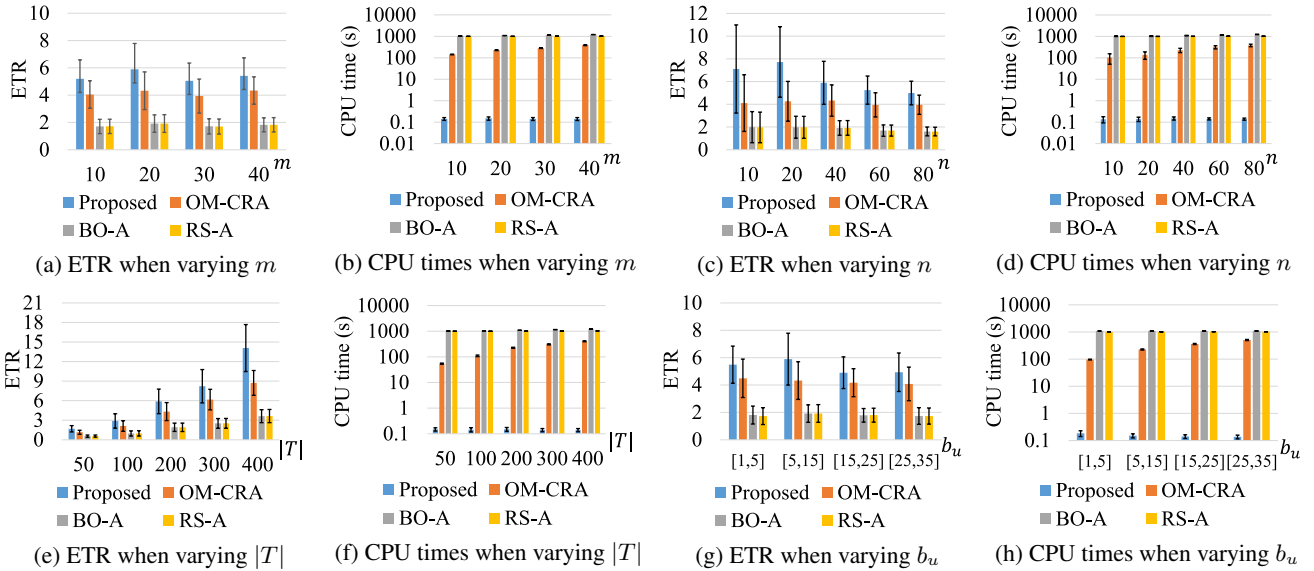


Figure 4: Results of simulation experiments with real data. ETR and CPU times (in seconds) with error bars of standard deviation are shown when varying parameters. The vertical axis of each graph for CPU time is a logarithmic scale. Upper (Lower) values are better in ETR (CPU times). Each result is the average of 30 simulations. The differences between the proposed method and all baselines are significant for ETR and CPU time (two-sided t-test:  $p < 0.01$ ) in all parameters.

Here, we let  $\epsilon := p_{\hat{v}}(x_M)$  and show that replacing  $\hat{x}_{\hat{v}\hat{t}}$  with  $x_M$ ,  $\hat{z}_{\hat{e}\hat{t}}$  ( $= 0$ ) with  $\epsilon$ , and  $\hat{z}_{et}$  with  $\max\{0, \hat{z}_{et} - \epsilon\}$  for all  $(e, t) \in E \times T \setminus (\hat{e}, \hat{t})$  increases the objective value of (PA) without impairing feasibility. The objective value of (PA) increases because  $(w_{\hat{e}} - x_M)\epsilon - \sum_{(e=(u,v),t) \in E \times T \setminus (\hat{e}, \hat{t}): w_e - \hat{x}_{vt} \geq 0} (w_e - \hat{x}_{vt}) \min\{\hat{z}_{et}, \epsilon\} \geq (w_{\hat{e}} - x_M - \sum_{(e=(u,v),t) \in E \times T \setminus (\hat{e}, \hat{t}): w_e - \hat{x}_{vt} \geq 0} (w_{et} -$

$\hat{x}_{vt}))\epsilon > 0$ . Here, for all  $e = (u, v) \in E$  and  $t \in T$ , if  $w_e - \hat{x}_{vt} < 0$ , 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  $(\hat{x}, \hat{z})$  is the optimal solution of (PA). The first constraint of (PA) is satisfied because  $\epsilon \leq p_{\hat{v}\hat{t}}(x_M)$  and  $\sum_{e \in \delta(v)} \max\{0, \hat{z}_{et} - \epsilon\} \leq \sum_{e \in \delta(v)} \hat{z}_{et} \leq$

$p_v(\hat{x}_{vt})$  for all  $(v, t) \in V \times T \setminus (\hat{v}, \hat{t})$ . The second constraint holds for all  $u \in U \setminus \hat{u}$  since the left-hand side of the inequality only decreases. For  $\hat{u}$ , the second constraint holds because  $\sum_{(e,t) \in \delta(\hat{u}) \times T \setminus (\hat{e}, \hat{t})} \max\{0, \hat{z}_{et} - \epsilon\} + \epsilon \leq \max\{\epsilon, \sum_{(e,t) \in \delta(\hat{u}) \times T} \hat{z}_{et}\} \leq b_{\hat{u}}$ . Therefore, replacing  $\hat{x}_{\hat{v}\hat{t}}$  with  $x_M$ ,  $\hat{z}_{\hat{e}\hat{t}} (= 0)$  with  $\epsilon$ , and  $\hat{z}_{et}$  with  $\max\{0, \hat{z}_{et} - \epsilon\}$  for all  $(e, t) \in E \times T \setminus (\hat{e}, \hat{t})$  increases the objective value of (PA) from  $(\hat{x}, \hat{z})$  without impairing feasibility. This contradicts the optimality of  $(\hat{x}, \hat{z})$  for (PA). Then,  $\sum_{e \in \delta(v)} \hat{z}_{et} \neq 0$  for all  $v \in V, t \in T$ .

We show that there exists an optimal solution  $(\tilde{x}, \tilde{z})$  for (PA) that satisfies  $p_v(\tilde{x}_{vt}) = \sum_{e \in \delta(v)} \tilde{z}_{et}$  for all  $v \in V, t \in T$ . Let  $(\hat{x}, \hat{z})$  be an optimal solution for (PA). Then, from constraints of (PA),  $\sum_{e \in \delta(v)} \hat{z}_{et} \leq p_v(\hat{x}_{vt})$  for all  $v \in V, t \in T$ . Suppose that there exists  $(v, t)$  satisfying  $\sum_{e \in \delta(v)} \hat{z}_{et} < p_v(\hat{x}_{vt})$  and let  $\Omega := \{(v, t) \mid \sum_{e \in \delta(v)} \hat{z}_{et} < p_v(\hat{x}_{vt})\}$ . Since  $\sum_{e \in \delta(v)} \hat{z}_{et} > 0$  and  $\lim_{x \rightarrow -\infty} p_v(x) = 0$  from Assumption 1, there exists a positive scalar  $d_{vt}$  that satisfies  $\sum_{e \in \delta(v)} \hat{z}_{et} = p_v(\hat{x}_{vt} - d_{vt})$  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  $\tilde{z}$  be  $\hat{z}$ . Then,  $(\tilde{x}, \tilde{z})$  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  $\hat{z} \geq 0$ . Therefore, in (PA), there exists an optimal solution  $(\tilde{x}, \tilde{z})$  satisfying  $\sum_{e \in \delta(v)} \tilde{z}_{et} = p_v(\tilde{x}_{vt})$  for all  $v \in V$  and  $t \in T$ . Then, (PA) with the constraints “ $x_{vt} = p_v^{-1}(\sum_{e \in \delta(v)} z_{et})$  for all  $v \in V, t \in T$ ”, that is, (PA') is equivalent to (PA). Hence,  $(x^*, z^*)$  is an optimal solution for (PA).  $\square$

### 7.3 Proof of Theorem 4

**Remark:** This proof is the same as (Hikima et al. 2021) and not our contribution. For readability, we only changed the symbols according to our paper.

Since  $-\frac{p'_v(x)}{p_v(x)}$  is monotonically non-decreasing from Assumption 1,  $\frac{p_v(x)}{p'_v(x)}$  is monotonically non-decreasing. Then,  $x + \frac{p_v(x)}{p'_v(x)}$  is monotonically increasing because  $x$  is monotonically increasing.

For all  $v \in V$  and given  $z \in S_v$ , there exists only one  $x$  that satisfies  $p_v(x) = z$  from Assumption 1. Then, the following equality holds:

$$(p_v^{-1}(z)z)' = p_v^{-1}(z) + (p_v^{-1}(z))'z = x + p_v(x)/p'_v(x). \quad (10)$$

Here, we consider  $z^1 \in S_v$  and  $z^2 \in S_v$  satisfying  $z^1 < z^2$ . Let  $x^1$  satisfy  $p_v(x^1) = z^1$  and  $x^2$  satisfy  $p_v(x^2) = z^2$ . Then,  $x^1 < x^2$  since  $p_v$  is monotonically increasing from Assumption 1. Since (10) holds and  $x + p_v(x)/p'_v(x)$  is monotonically increasing,

$$\begin{aligned} (p_v^{-1}(z^1)z^1)' &= x^1 + p_v(x^1)/p'_v(x^1) \\ &< x^2 + p_v(x^2)/p'_v(x^2) = (p_v^{-1}(z^2)z^2)'. \end{aligned}$$

Since  $(p_v^{-1}(z)z)'$  is monotonically increasing,  $p_v^{-1}(z)z$  is convex. Moreover, the objective function of (CP) is convex since  $p_v^{-1}(\sum_{e \in \delta(v)} z_{et}) \sum_{e \in \delta(v)} z_{et}$  is convex with respect

to  $z$  for all  $v \in V$  and  $t \in T$  from (Boyd and Vandenberghe 2004, Section 3.2.2).  $\square$

### 7.4 Proof of Theorem 5

From Lemma 1 and the proof of Theorem 3, there exists an optimal solution  $z'$  for (PA'). We suppose that there is  $t'$ , and  $t''$  such that  $\exists e \in E : z'_{et'} \neq z'_{et''}$ . Let  $z_t^\dagger := \{z'_{et}\}_{e \in E}$  and  $z'_t := \{z'_{et}\}_{e \in E}$ . Then, we show that  $z^\dagger$  is also an optimal solution for (PA'), where

$$z_t^\dagger := \begin{cases} \frac{z'_{t'} + z'_{t''}}{2}, & \text{if } t \in \{t', t''\}, \\ z'_t, & \text{otherwise.} \end{cases} \quad (11)$$

Here, since  $0 \leq z' \leq 1$ ,

$$0 \leq z^\dagger \leq 1. \quad (12)$$

For all  $v \in V$ , the set  $S_v$  is a connected set since the function  $p_v$  is continuous. Therefore, since  $\sum_{e \in \delta(v)} z'_{et'} \in S_v$  and  $\sum_{e \in \delta(v)} z'_{et''} \in S_v$ , it yields for all  $v \in V$  and  $t \in \{t', t''\}$  that

$$\begin{aligned} \sum_{e \in \delta(v)} z_t^\dagger &= \sum_{e \in \delta(v)} \frac{z'_{et'} + z'_{et''}}{2} \\ &= \frac{1}{2} \sum_{e \in \delta(v)} z'_{et'} + \frac{1}{2} \sum_{e \in \delta(v)} z'_{et''} \in S_v. \end{aligned} \quad (13)$$

Moreover,

$$\sum_{t \in T} \sum_{e \in \delta(u)} z_t^\dagger = \sum_{t \in T} \sum_{e \in \delta(u)} z'_{et} \leq b_u \quad (14)$$

for all  $u \in U$ . From (12), (13) and (14),  $z^\dagger$  is a feasible solution for (PA'). Furthermore, changing  $z'$  to  $z^\dagger$  increases the objective function by the following amount.

$$\begin{aligned} &g(z_t^\dagger) + g(z_{t''}^\dagger) - g(z'_{t'}) - g(z'_{t''}) \\ &= 2g\left(\frac{z'_{t'} + z'_{t''}}{2}\right) - g(z'_{t'}) - g(z'_{t''}) \\ &= 2\left(g\left(\frac{z'_{t'} + z'_{t''}}{2}\right) - \frac{g(z'_{t'}) + g(z'_{t''})}{2}\right) \leq 0, \end{aligned}$$

where  $g(z_t) := \sum_{v \in V} p_v^{-1}(\sum_{e \in \delta(v)} z_{et}) \sum_{e \in \delta(v)} z_{et} + \sum_{e=(u,v) \in E} w_e z_{et}$ . The inequality holds since the function  $g$  is convex from Theorem 4. Therefore, since changing  $z'$  to  $z^\dagger$  does not increase the objective value,  $z^\dagger$  is also an optimal solution. By repeating operation (11),  $z$  such that  $z_{e1} = z_{e2} = \dots = z_{e|T|}$  can be created from an optimal solution  $z'$  without increasing the objective value. Thus, there is an optimal solution  $z$  such that  $z_{e1} = z_{e2} = \dots = z_{e|T|}$  for (PA').

Therefore, (PA') with the constraint “ $z_{e1} = z_{e2} = \dots = z_{e|T|}$  for all  $e \in E$ ”, that is, (SCP) is equivalent to (PA'). Then,  $z^*$  is an optimal solution for (PA').  $\square$

### References

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

- Alaei, S.; Hajiaghayi, M.; and Liaghat, V. 2012. Online prophet-inequality matching with applications to ad allocation. In *EC*, 18–35.
- Anari, N.; Goel, G.; and Nikzad, A. 2014. Mechanism design for crowdsourcing: An optimal  $1-1/e$  competitive budget-feasible mechanism for large markets. In *FOCS*, 266–275.
- 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.
- Bagnoli, M.; and Bergstrom, T. 2006. Log-concave probability and its applications. In *Rationality and Equilibrium*, 217–241.
- 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.
- Bergstra, J.; and Bengio, Y. 2012. Random search for hyperparameter optimization. *Journal of machine learning research*, 13(2).
- 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*.
- Dickerson, J.; Sankararaman, K.; Srinivasan, A.; and Xu, P. 2018. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *AAAI*, 1007–1014.
- Fan, J.; Li, G.; Ooi, B. C.; Tan, K.-l.; and Feng, J. 2015. icrowd: An adaptive crowdsourcing framework. In *SIGMOD*, 1015–1030.
- Hellemo, L.; Barton, P. I.; and Tomaszgard, A. 2018. Decision-dependent probabilities in stochastic programs with recourse. *Computational Management Science*, 15(3): 369–395.
- Hikima, Y.; Akagi, Y.; Kim, H.; Kohjima, M.; Kurashima, T.; and Toda, H. 2021. Integrated Optimization of Bipartite Matching and Its Stochastic Behavior: New Formulation and Approximation Algorithm via Min-cost Flow Optimization. In *AAAI*, 3796–3805.
- Hikima, Y.; Akagi, Y.; Marumo, N.; and Kim, H. 2022. Online matching with controllable rewards and arrival probabilities. In *IJCAI*, 1825–1833.
- Ho, C.-J.; Jabbari, S.; and Vaughan, J. W. 2013. Adaptive task assignment for crowdsourced classification. In *ICML*, 534–542.
- Ho, C.-J.; and Vaughan, J. 2012. Online task assignment in crowdsourcing markets. In *AAAI*, 45–51.
- Horton, J. J.; and Zeckhauser, R. J. 2010. Algorithmic wage negotiations: Applications to paid crowdsourcing. In *CrowdConf*, 2–5.
- Manshadi, V. H.; Gharan, S. O.; and Saberi, A. 2012. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4): 559–573.
- Mao, K.; Yang, Y.; Li, M.; and Harman, M. 2013. Pricing crowdsourcing-based software development tasks. In *ICSE*, 1205–1208.
- Minder, P.; Seuken, S.; Bernstein, A.; and Zollinger, M. 2012. *Crowdmanager-combinatorial allocation and pricing of crowdsourcing tasks with time constraints*. University of Zurich.
- Singer, Y.; and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. In *WWW*, 1157–1166.
- 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.
- Tran-Thanh, L.; Stein, S.; Rogers, A.; and Jennings, N. R. 2014. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214: 89–111.
- 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.
- Wen, Y.; Shi, J.; Zhang, Q.; Tian, X.; Huang, Z.; Yu, H.; Cheng, Y.; and Shen, X. 2014. Quality-driven auction-based incentive mechanism for mobile crowd sensing. *IEEE transactions on vehicular technology*, 64(9): 4203–4214.
- Xiao, S.; Wang, Z.; Chen, M.; Tang, P.; and Yang, X. 2020. Optimal Common Contract with Heterogeneous Agents. In *AAAI*, 7309–7316.
- Zhang, H.; Ma, Y.; and Sugiyama, M. 2015. Bandit-based task assignment for heterogeneous crowdsourcing. *Neural computation*, 27(11): 2447–2475.
- Zhao, D.; Li, X.-Y.; and Ma, H. 2014. How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *INFOCOM*, 1213–1221.
- Zhao, Y.; Xia, J.; Liu, G.; Su, H.; Lian, D.; Shang, S.; and Zheng, K. 2019. Preference-aware task assignment in spatial crowdsourcing. In *AAAI*, 2629–2636.