Budgeted Online Assignment in Crowdsourcing Markets: Theory and Practice

Pan Xu, Aravind Srinivasan
Dept. of Computer Science
University of Maryland, College Park, USA
{panxu, srin}@cs.umd.edu

Kanthi K. Sarpatwar, Kun-Lung Wu
IBM Thomas J. Watson Research Center
Yorktown Heights, NY, USA
{sarpatwa, klwu}@us.ibm.com

ABSTRACT

We consider the following budgeted online assignment problem motivated by crowdsourcing. We are given a set of offline tasks that need to be assigned to workers who come from the pool of types $\{1, 2, \ldots, n\}$. For a given time horizon $\{1, 2, \ldots, T\}$, at each instant of time t, a worker j arrives from the pool in accordance with a known probability distribution $[p_{jt}]$ such that $\sum_j p_{jt} \leq 1$; j has a known subset $N(j)$ of the tasks that it can complete, and an assignment of one task i to j (if we choose to do so) should be done before task i's deadline. The assignment $e = (i, j)$ (of task $i \in N(j)$ to worker j) yields a profit w_e to the crowdsourcing provider and requires different quantities of K distinct resources, as specified by a cost vector $\mathbf{a}_e \in [0, 1]^K$; these resources could be client-centric (such as their budget) or worker-centric (e.g., a driver's limitation on the total distance traveled or number of hours worked in a period). The goal is to design an online-assignment policy such that the total expected profit is maximized subject to the budget and deadline constraints.

We propose and analyze two simple linear programming (LP)-based algorithms and achieve a competitive ratio of nearly $1/(\ell+1)$, where ℓ is an upper bound on the number of non-zero elements in any $\mathbf{a}_e$. This is nearly optimal among all LP-based approaches. We also propose several heuristics adapted from our algorithms and compare them to other non-LP-based algorithms over a large set of random instances. Experimental results show that our LP-based heuristics significantly outperform the non-LP-based ones, sometimes by nearly 90%.

CCS Concepts

•Computing methodologies → Multi-agent systems;

Keywords

approximation algorithms; online algorithms; crowdsourcing market

1. INTRODUCTION

Crowdsourcing markets (e.g., Amazon Mechanical Turk or Crowdflower) have evolved to be powerful platforms that bring together task performers (or workers) and task requesters (or consumers). In recent years, problems arising from online decision making in such settings have been attracting tremendous attention (see the survey [37]). A typical problem arising in such settings, considered by [6], is to schedule a batch of consumer tasks using a pool of workers who become available in an online fashion (i.e., in real time). More specifically, we are given a set I of *offline* tasks, where each task $i \in I$ has a deadline d_i after which it cannot be scheduled. Workers arrive in an online fashion (according to an adversarial or random permutation order) and submit bids on a subset of tasks that interest them. When a worker j arrives, a decision must be made immediately and irrevocably - either assign it an available task or reject its service. If the worker j is allocated a task i, we must pay the worker their bid amount b_{ij}. The goal is to maximize the number of tasks assigned while constrained by a given bid budget of B. Our work deals with a natural variant of this problem.

As per standard notation, we use $[n]$ to denote the set of integers $\{1, 2, \ldots, n\}$. Further let us assume a time horizon $[T]$. In this work, we model the arrival of workers as follows. At any given instant of time (referred to as *round*) $t \in [T]$, a single worker is chosen from a known pool of worker types $[n]$ in accordance with a *known* probability distribution $[p_{jt}]$ such that $\sum_j p_{jt} \leq 1$[1] (noting that such a choice is made independently for each round t). Current related works in the domain of *mechanism design* for crowdsourcing markets mainly model the arrival pattern of online workers as either random arrival order (e.g., [7]) or *known independent identical distributions (i.i.d)* (e.g., [35, 36]). Our arrival setting can be viewed a natural generalization in the way that we allow the arrival distributions change over time. Notice that we do not consider if each worker will submit her bid truthfully when designing the allocating policy, which is one of the major concerns of mechanism design.

Another key distinction from the previous models is that we consider multiple budget constraints. That is, we assume that there are K distinct resources and that each assignment $e = (i, j)$ has a bid cost vector $\mathbf{a}_e \in [0, 1]^K$, where the k^{th} component of the vector corresponds to the amount of resource type k needed by the assignment. These resources could be task-requester-centric (such as their budget) or worker-centric (e.g., a driver's limitation on the total distance traveled or number of hours worked in a period). Resource k is called integral if $a_{e,k} \in \{0, 1\}$ for all e, otherwise we refer to it as *non-integral*. We note that [6] set a constraint that each task be assigned at most once, while [25] generalize it to the setting where each task may be assigned at most $b_i \in \mathbb{Z}_+$ times. Putting this in context with our work, these settings can be viewed as special cases of our setting as follows: each task itself is an integral resource with budget either 1 or b_i respectively.

Finally, instead of maximizing the throughput (i.e., number of tasks completed), each assignment e is associated with a *known*

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

[1]Here we allow that with probability $1 - \sum_j p_{jt}$, none of the workers is chosen at t.

weight or utility w_e and we aim to maximize the expected utility collected from those successful assignments. [25] consider the case where the weights are unknown and must be learned, while workers arrive online in a random permutation order.

Our Contributions. We deal with several theoretical and practical aspects of the above budgeted online assignment problem (BOA), under the assumption that the arrival distribution is known in advance. Before discussing our contributions, we define a couple of useful parameters that can help appreciate our results better. Let ℓ_1 (resp. ℓ_2) denote the maximum number of integral (resp. non-integral) resources requested (in a non-zero amount) in any assignment cost vector $\mathbf{a}_e$.

First, we consider the simple and natural case where all the resources are integral and each assignment requests at most $\ell_1 = \ell$ resources. We present two simple LP-based algorithms, ALG_1 and ALG_2, that are non-adaptive and adaptive respectively. Here we say an online algorithm is adaptive if it somehow incorporates all information observed so far including online arrivals and outcomes of previous strategies to design the current strategy. In Section 5, we prove the following theorems.

Theorem 1.1. *There exists an online non-adaptive algorithm* ALG_1 *for the BOA problem with a competitive ratio of* $\frac{1}{\ell+1}(1-\frac{1}{\ell+1})^{\ell} \geq \frac{1}{e(\ell+1)}$*, assuming all the resources are integral.*

Theorem 1.2. *There exists an online adaptive algorithm* ALG_2 *for the BOA problem with a competitive ratio of* $(1-\varepsilon)/(\ell+1)$*, for any given* $\varepsilon > 0$*, assuming all the resources are integral.*

Our competitive ratio analysis is tight for the non-adaptive algorithm ALG_1 (as shown in Example 5.2). In other words, Theorem 1.1 states the best possible ratio that ALG_1 could get. Another notable point is that ALG_2 is nearly optimal among all LP (4.1)-based approaches, i.e., all possible algorithms using the LP (4.1) as a benchmark, since it has an integrality gap at least $\ell - 1 + 1/\ell$ ([21]). The main technical challenge facing us is to lower bound $\Pr[\wedge_k \mathcal{S}_k]$ for a certain family of *negatively correlated* events $\{\mathcal{S}_k\}$. We develop two different useful techniques (see Section 5.2 and 5.3) to tackle this challenge and use them to prove the optimality of our analyses.

Subsequently, we consider the general case of resources being both integral and non-integral (see Section 6) and show that the above theorems (i.e., Theorem 1.1 and 1.2) can be readily applied assuming that the budget of any non-integral resource is at least *moderately large* enough. More precisely, we prove these two theorems. Let B be the minimum budget for any non-integral resource.

Theorem 1.3. *For the BOA problem,* ALG_1 *yields a competitive ratio of* $\frac{1}{\ell_1+1}\Big((1-\frac{1}{\ell_1+1})^{\ell_1}-\varepsilon\Big)$*, for any* $\varepsilon > 0$*, assuming* $B \geq 2\ln(\frac{\ell_2}{\varepsilon})\Big(1+\frac{3\ell_1+2}{\ell_1^2}\Big)+2$.

Theorem 1.4. *For the BOA problem,* ALG_2 *yields a competitive ratio of* $\frac{1-2\varepsilon}{\ell_1+1}$ *for any given* $\varepsilon > 0$*, assuming* $B \geq 3\ln(\frac{\ell_2}{\varepsilon})(1+\frac{1}{\ell_1})+2$.

In the proof of Theorem 1.4, we apply the technique of *virtual algorithms* to tackle the technical challenge of upper bounding $\Pr[\sum_i X_i \geq (1+\delta)\mathbb{E}[\sum_i X_i]]$ for a family of *positively correlated* random variables $\{X_i \in [0,1]\}$. Our results show that the knowledge about arrival distributions holds a significant edge over the adversarial model or the random permutation model. Let us compare our results with those of [6]. As discussed before, their setting fits our model when $\ell_1 = \ell_2 = 1$. From Theorem 1.4, we obtain a $(\frac{1}{2}-\epsilon)$ competitive ratio assuming $B \geq 12\ln(1/\epsilon)$ while [6] obtain a ratio of $O(\frac{1}{R^{\epsilon}\ln R})$, assuming $B \geq \frac{R}{\epsilon}$ and $R \doteq \frac{\max b_{i,j}}{\min b_{i,j}}$ (i.e., the ratio of the largest bid to the smallest bid over all possible assignments). In fact, we completely remove the dependency on R and obtain a constant ratio while relaxing the lower bound assumption on B significantly. Our result may be seen as theoretical evidence to advocate the use of historical data to learn arrival distributions.

Finally, we propose various LP-based heuristics and evaluate them against certain non-LP-based approaches. Our experiments show that the LP-based approaches yield a far superior competitive ratio compared to LP-blind approaches, sometimes by nearly 90%.

2. RELATED WORK

As an offline version of our model, the classic column-sparse packing (CSP) problem has been well studied in the theoretical computer science community. The basic setting is as follows: we are given n items and K resources; each item $i \in [n]$ has a size vector $\mathbf{a}_j \in [0,1]^K$ and a profit $w_j > 0$; given a budget $\mathbf{B} \in \mathbb{R}_+^K$, the goal is to choose a subset of items such that the total profit is maximized without violating the budget constraints. More generally, our offline model is reduced to the Multidimensional Knapsack problem (MKP) when there is no restriction on the sparsity of each size vector $\mathbf{a}_j$. We see that the offline model such as MKP can fit our online model as a special case when $T = n$ and $p_{jt} = 1$ iff $j = t$ and 0 otherwise for all $j \in [n], t \in [T]$. Note that we have more restrictions here: we are not allowed to look at all the items before making decisions; instead we have to make an *instant irrevocable* decision whenever an item comes. Many common techniques such as *permutation of all items* [8] and *alteration* [9] shown useful in the offline setting, are not applicable to our online problems. Notice that any hardness result from CSP problem will also apply here though.

We now briefly describe several recent results for the CSP problem. Let k be the column sparsity, i.e., the number of non-zero entries in any column vector (this is equivalent to the parameter ℓ in our BOA problem). For the general case when each $\mathbf{a}_j \in [0,1]^K$, [9] gave a randomized algorithm with the approximation ratio of $1/(ek+o(k))$ and constructed an instance showing integrality gap of even a strengthened LP to be at least $2k-1$. As a special case of CSP when all $\mathbf{a}_j$ are binary and $\mathbf{B}$ is integral, the k-set packing problem is extensively studied before [11, 24, 26, 5]. Note that [21] showed that the natural LP relaxation for k-set packing has an integrality gap at least $k-1+1/k$. [8] considered a stochastic version where each $\mathbf{a}_j$ is a binary-vector valued random variable with each outcome having at most k non-zero elements. They obtained a $2k$-approximation algorithm. Further, they presented a $(k+1)$-approximation algorithm as well when each $\mathbf{a}_j$ has monotone outcomes. An important variant of the stochastic CSP problem, known as the stochastic matching problem, arises with $k = 2$ and has received considerable attention recently [14, 10, 1, 22].

As for our online model, there is a long line of research related to our problem: online bipartite matching and its variants, which are motivated by applications to online advertisement business. Two notable special cases, Adwords and Display Ads, have been studied extensively in recent years: [13, 15, 16, 17, 18, 33]. Both these models can be modeled as each assignment consumes only a single integral (Display Ads) or non-integral (Adwords) resource even though the potential number of distinct resources can be huge. More recently, [28] considered a natural generalization of Adwords, where there are multi-tier budgets forming a laminar structure. Regarding the online arrival assumption, there are three main categories: adversarial, random arrival order, and known distributions (see the book [32] for more details). A majority of the recent work under known distributions, focuses on the case when when the distributions in each round are independent and identical (referred to as *known i.i.d*) [19, 23, 31, 27, 12]. We refer to another variant of known distributions where the distributions can change over rounds as the *known adversarial*. Note that this is the setting we consider here and is more general compared to the *known i.i.d* as descried before. For this setting, [4] considers the online stochastic generalized assignment problem while [3] considers the online prophet-inequality matching problem. Note that most of the current online-matching models under known distributions can fit into our model as a special case except the fact that some assume the cost or bid is a random variable while we model it as deterministic here. There are several papers considering online packing LP problem as well under a random permutation order [29, 2]. To be specific, [29] presented an algorithm achieving a $(1-\epsilon)$-competitive ratio, provided $B = \Omega(\ln(\ell)/\epsilon^2)$, where B is the largest capacity ratio and ℓ is the cost-vector sparsity.

3. PROBLEM STATEMENT

In this section, we present a formal statement of our problem. Let $I = \{i \in [m]\}$ be the set of offline tasks and $J = \{j \in [n]\}$ be the set of online workers. On a finite time horizon T, each task i has a deadline $d_i \in [T]$ after which it will become unavailable. Let $G = (I, J, E)$ be the bipartite graph that models the relation between the tasks and workers: there is an edge $e = (i, j)$ iff worker j is interested in the task i. Let $N(j) = \{i : (i, j) \in E\}$ be the set of tasks that interest worker j and $N(i) = \{j : (i, j) \in E\}$ be the set of workers who are interested in task i. Each edge $e = (i, j)$ has a weight w_e denoting the profit obtained by assigning task i to worker j. Each assignment $e = (i, j)$ has a requirement for one or more of a given set of K types of resources. The requirement of an assignment e is given by a K-dimensional vector $\mathbf{a}_e \in [0, 1]^K$, where the k^{th} dimension $a_{e,k}$ represents the amount of resource k needed. Each resource type k has a budget $B_k \in \mathbb{R}_+$ that must not be violated. For each e, let $S_e = \{k \in [K] : a_{e,k} > 0\}$, i.e., the set of resources it requests.

At any instant $t \in [T]$, a worker j arrives with a probability p_{jt} such that $\sum_j p_{jt} \leq 1$ (thus, with probability $1 - \sum_j p_{j,t}$, no worker arrives at time t). Let $E_{j,t} = \{e = (i, j), i \in N(j) : d_i \geq t\}$ denote the set of *available* assignments for the worker j at time t. In this paper, we assume without loss of generality that each task can be assigned for an arbitrary number of times before its deadline. Any potential restriction on the number of assignments can easily be modeled by an additional budget constraint: the task itself is an integral resource and the corresponding budget is the upper bound on the number of assignments. For each $e \in E_{j,t}$, we say e is *safe or valid* iff for each $k \in S_e$, resource k has remaining budget larger or equal to $a_{e,k}$. When a worker j arrives at t, we have to make an immediate and irrevocable decision: either reject it or choose a safe option $e \in E_{j,t}$ [2] and get a resultant profit w_e. Once a safe assignment e is scheduled, the budget of each resource $k \in S_e$ will be reduced by $a_{e,k}$. Our goal is to design an online assignment policy such that the expected profit is maximized.

In most applications, we need to deal with two kinds of resources, namely integral and non-integral. A resource k is integral if $a_{e,k} \in \{0, 1\}$ for all $e \in E$ and $B_k \in \mathbb{Z}_+$. On the other hand a resource k is non-integral if $a_{e,k} \in [0, 1]$ and $B_k \in \mathbb{R}_+$. This captures resources such as money and time that cannot be quantified as integral. Let $\mathcal{K}_1 = \{1, 2, \cdots, K_1\}$ and $\mathcal{K}_2 = \{K_1 + 1, \cdots, K_1 + K_2\}$ denote the set of integral and non-integral resources respectively. As defined in the introduction, for each assignment e, $|S_e \cap \mathcal{K}_1| \leq \ell_1$ and $|S_e \cap \mathcal{K}_2| \leq \ell_2$.

[2]In the case when some worker j can accept multiple assignments each time, say L, we can simply add L copies of j to our graph G.

4. BENCHMARK LP

For an online algorithm ALG, the competitive ratio is defined as the ratio of the expected performance of ALG to the expected offline optimal over all possible realizations. A common technique is to use an LP (we called benchmark LP) to upper bound the latter value, thereby obtaining a lower bound on the competitive ratio.

Recall that $E_{j,t}$ is the set of available assignments for a worker j arriving at t. For any t, let $E_t = \bigcup_j E_{j,t}$ be the set of all available assignments at t. Further, for each t and $e \in E_t$, let $x_{e,t}$ be the probability that we make the assignment e at t in the offline optimal solution. Our benchmark LP can now be described as follows:

$$\text{maximize} \sum_t \sum_{e \in E_t} w_e x_{e,t} \tag{4.1}$$

$$\text{subject to} \sum_{e \in E_{j,t}} x_{e,t} \leq p_{jt} \qquad \forall j \in J, t \in [T] \tag{4.2}$$

$$\sum_t \sum_{e \in E_t} x_{e,t} a_{e,k} \leq B_k \qquad \forall k \in [K] \tag{4.3}$$

$$0 \leq x_{e,t} \leq 1 \qquad \forall e \in E, t \in [T] \tag{4.4}$$

Lemma 4.1. *The optimal value to LP (4.1) is a valid upper bound for the offline optimal.*

Our benchmark LP is essentially the same as that used in [3] and [4]. The detailed proof can be found there. We provide a rough proof here.

Proof. The simple idea is to show that all the constraints in the above LP are valid for the offline optimal. For each given t and worker j, $\sum_{e \in E_{j,t}} x_{e,t}$ can be interpreted as the sum of the expected number of assignments related to j we could make in the offline optimal, which is surely no larger than the probability that j comes at t. This justifies constraints (4.2). Any offline algorithm should satisfy the budget constraints as well and by linearity of expectation, we see constraints (4.3) are valid. □

5. THE CASE OF INTEGRAL RESOURCES

In this section, we consider the case when $K_2 = 0$, i.e., all resources are integral with $a_{e,k} \in \{0, 1\}$ and $B_k \in \mathbb{Z}_+$ for all $e \in E$ and $k \in [K]$. Let $\ell_1 = \ell$, i.e., each assignment requests at most ℓ (integral) resources.

As shown in Section 2, the k-set packing problem can be reformulated as a special case here. Thus from [21], it follows that even for the special case of unit budget, i.e., $B_k = 1$ for all $k \in [K]$, LP (4.1) has an integrality gap at least $\ell - 1 + 1/\ell$. That implies by using the LP (4.1) as the benchmark, we cannot get an online algorithm achieving a ratio beating $1/(\ell - 1 + 1/\ell)$.

5.1 A simple non-adaptive algorithm

In this section, we present a simple LP-based non-adaptive algorithm. Suppose $\{x^*_{e,t} | t \in [T], e \in E_t\}$ is an optimal solution for the LP (4.1). The main idea behind our algorithm (described in Algorithm 1) is as follows: at each time t when some worker j arrives, if safe make the assignment $e \in E_{j,t}$ with probability $\alpha x^*_{e,t}/p_{j,t}$, where $\alpha \in (0, 1]$ is a parameter that will be optimized later.

We note that the last step of Algorithm 1 is well defined because $\sum_{e \in \hat{E}_{j,t}} \alpha x^*_{e,t}/p_{jt} \leq \sum_{e \in E_{j,t}} x^*_{e,t}/p_{jt}$, which is at most 1.

Theorem 5.1. *By choosing* $\alpha = \frac{1}{2\ell}$, ALG_1 *achieves an online competitive ratio of at least* $\frac{1}{4\ell}$.

Proof. WLOG assume that $t = T$ and fix an assignment $e \in E_T$. Recall that S_e is the set of resources requested by e. For each $k \in S_e$,

Algorithm 1: A simple non-adaptive algorithm (ALG_1)

1 For each time t, assume some worker j arrives.
2 Let $\hat{E}_{j,t} \subseteq E_{j,t}$ be the set of *safe* available assignments we can make for j.
3 If $\hat{E}_{j,t} = \emptyset$, then reject j; otherwise sample at most one assignment $e \in \hat{E}_{j,t}$ with probability $\alpha x^*_{e,t}/p_{jt}$.

let $\mathcal{S}_k$ be the event that e is *safe* at T with respect to a resource k. We now lower bound the value $\Pr[\wedge_{k \in S_e} \mathcal{S}_k]$. Fix one such $k \in S_e$. Let U_k be the usage of resource k at the beginning of $t = T$ and $X_{e',t'}$ be the indicator random variable for assignment $e' \in E_{t'}$ chosen at $t' \in [T-1]$. We have $U_k = \sum_{t'<T} \sum_{e' \in E_{t'}} X_{e',t'} a_{e',k}$. By definition, e is safe with respect to resource k iff $U_k \leq B_k - 1$. Observe that $\mathbb{E}[X_{e',t'}] \leq \alpha x^*_{e',t'}$. By Markov inequality we see

$$\Pr[U_k \leq B_k - 1] = 1 - \Pr[U_k \geq B_k] \geq 1 - \alpha \tag{5.1}$$

Thus we get

$$\Pr[\wedge_{k \in S_e} \mathcal{S}_k] = \Pr\left[\bigwedge_{k \in S_e} \Big(U_k \leq B_k - 1\Big)\right] \geq 1 - \ell\alpha \tag{5.2}$$

So we get that for the given (e, t), e will be made with probability at least $\alpha x^*_{e,t}(1 - \ell\alpha)$. By setting $\alpha = \frac{1}{2\ell}$, we get that each assignment e is made with probability at least $x^*_{e,t}/(4\ell)$. ◻

5.2 A tight analysis for ALG_1 with unit budget

In this section, we consider a special case when $B_k = 1$ for all $k \in K$ and show a *tight* analysis for ALG_1. Consider the following example.

Example 5.1. *Consider an unweighted star graph $G = (I, J, E)$ where $|I| = 1, |J| = 3, E = (e_1, e_2, e_3)$ with $T = 2$ and $d_1 = T$ (no deadline constraints). Suppose at $t = 1$, $j = 1, 2$ arrives with equal probability $1/2$ and at $t = 2$, $j = 3$ will arrive with probability 1. Let e_1, e_2, e_3 denote respectively the assignment we consider when $j = 1$ comes at $t = 1$, $j = 2$ comes at $t = 1$ and $j = 3$ comes at $t = 2$. Let $K = 2$ with $\mathbf{B} = (1, 1)$ and $\mathbf{a}_{e_1} = (1, 0)$, $\mathbf{a}_{e_1} = (0, 1)$ and $\mathbf{a}_{e_3} = (1, 1)$. Suppose LP (4.1) offers us such an optimal solution: $x^*_{e_1} = x^*_{e_2} = 1/2$ and $x^*_{e_3} = 1/2$ (notice that unweighted). Let us analysis the assignment e_3 when $j = 3$ comes at $t = 2$ by running* ALG_1.

According to ALG_1, *at $t = 1$ we will choose e_j with probability α whenever $j = 1$ or $j = 2$ comes. Notice that at $t = 2$, the first and the second resource are each safe with respective probability $1 - \alpha/2$ and both of the two are safe with probability $1 - \alpha$.* ◻

The above example suggests us two things: (1) the events that two different resources are safe can be *negatively* correlated. This means we can not apply the FKG inequality which is widely used in the offline version [9, 8, 10] to replace the union bound in inequality (5.2); (2) we could potentially strengthen the lower bound that each resource is safe, which is currently obtained by Markov inequality (5.1). Now we follow these ideas to present a tight analysis for ALG_1 for the case of unit budget.

Theorem 5.2. *By choosing $\alpha = \frac{1}{\ell+1}$,* ALG_1 *has an online competitive ratio of $\frac{1}{\ell+1}(1 - \frac{1}{\ell+1})^\ell$ with unit budget.*

Proof. As before, we consider the case that $t = T$ and an assignment $e \in E_T$. For each $t' < T$ and $k \in S_e$, let $E_{k,t'} = \{e' | e' \in E_{t'}, S_{e'} \ni k\}$ be the set of assignments which are available at t' and participate in the budget constraint of k. Let $B_{k,t'}$ be the (random) budget of k at the beginning of t'. Define $A_{k,t'} = (B_{k,t'+1} = 1 | B_{k,t'} = 1)$ and

$$A_{t'} = \wedge_{k \in S_e} A_{k,t'} = \Big(\wedge_{k \in S_e} B_{k,t'+1} = 1 | \wedge_{k \in S_e} B_{k,t'} = 1\Big)$$

We see that

$$\Pr[A_{k,t'}] = 1 - \sum_{e' \in E_{k,t'}} \alpha x^*_{e',t'}, \Pr[A_{t'}] \geq 1 - \sum_{e' \in \cup_{k \in S_e} E_{k,t'}} \alpha x^*_{e',t'}$$

It follows that

$$\Pr[\wedge_{k \in S_e} \mathcal{S}_k] = \prod_{t'<t} \Pr[A_{t'}] \geq \prod_{t'<t} \left(1 - \sum_{e' \in \cup_{k \in S_e} E_{k,t'}} \alpha x^*_{e',t'}\right) \tag{5.3}$$

The above inequality can be made tight when $\{E_{k,t'} | k \in S_e\}$ is disjoint for each t'. Here are two useful observations. The first one is $\sum_{e' \in \cup_{k \in S_e} E_{k,t'}} \alpha x^*_{e',t'} \leq \sum_{e' \in E_{t'}} \alpha x^*_{e',t'} \leq \alpha$. The second one is

$$\sum_{t'<T} \sum_{e' \in \cup_{k \in S_e} E_{k,t'}} \alpha x^*_{e',t'} \leq \sum_{k \in S_e} \sum_{t'<T} \sum_{e' \in E_{k,t'}} \alpha x^*_{e',t'} \leq \alpha\ell$$

These two observations lead to the fact that the rightmost expression of inequality (5.3) has a minimum value of $(1-\alpha)^\ell$. Therefore e will be made at t with overall probability $x^*_{e,t}\alpha(1-\alpha)^\ell$. By choosing $\alpha = 1/(\ell+1)$, we prove our claim. ◻

The example below shows the above analysis is tight.

Example 5.2. *Consider a star graph $G = (I, J, E)$ where $|I| = 1, |J| = \ell + 1, E = \{e_j | j \in [J]\}$ with $T = J$. Let $d_1 = T$, i.e., no deadline constraints. For each $t \in [T]$, $p_j = 1$ if $j = t$ and 0 otherwise. In other words, at each time $t \in [T]$, only worker $j = t$ will come surely and no one else. Suppose we use $\mathbf{a}_j$ and x^*_j to denote the terms $\mathbf{a}_{e_j}$ and $x^*_{e_j,t=j}$ before. Let $K = \ell$ with $\mathbf{B} = \mathbf{1}$ (dimension of K) and $\mathbf{a}_j = \mathbf{e}_j$ for each $j \leq \ell$, where $\mathbf{e}_j$ is the jth standard-basis unit vector, and $\mathbf{a}_j = \mathbf{1}$ for $j = \ell + 1$. Suppose LP (4.1) offers us such an optimal solution: $x^*_j = 1 - \epsilon$ for each $j \leq \ell$ and $x^*_{\ell+1} = \epsilon$.*

Now focus on the assignment $e = e_J$ when $j = J$ comes at $t = T$. Let us analyze the probability that e is safe at T, denoted by $\Pr[\mathcal{S}_{e,T}]$, in ALG_1 *with some parameter $\alpha \in (0, 1)$. Notice that e will be safe at $t = T$ iff none of $e_j, j \leq \ell$ is made before. According to* ALG_1, *each time t, $e_{j=t}$ will be made with probability equal to $\frac{\alpha x^*_j}{p_j} = \alpha(1-\epsilon)$. That implies $\Pr[\mathcal{S}_{e,T}] = (1 - \alpha(1-\epsilon))^\ell$, which matches our lower bound as shown in the proof of Theorem 5.2.* ◻

5.3 A tight analysis for ALG_1 with general integral budget

In Section 5.2, we give a tight analysis for ALG_1 for the case of unit budget. Intuitively, we should be in a better situation when each B_k is larger than 1. For example, by the Chernoff bound, we see that the probability that the usage of resource k at T overflows B_k should decrease exponentially as B_k gets larger. In this section, we give a tight analysis for ALG_1 by extending the result in Theorem 5.2 to the case of general integral budget.

Let us present an equivalent but simpler model of our problem. Suppose we have K types of balls and for each type $k \in [K]$, the number of balls is $B_k \in \mathbb{Z}_+$. We have a set of choices $E = \{e | e \in E\}$ and each choice is associated with a binary vector $\mathbf{a}_e \in \{0, 1\}^K$, which has at most ℓ non-zero elements. Once we make the choice e, we will take one ball of type k whenever $a_{e,k} = 1$. For each time $t \in [T]$, one choice e will arrive with probability $x^*_{e,t}$ such that $\sum_{e \in E} x^*_{e,t} \leq 1$ for each t. Each time t, for whatever choice comes, we will accept it non-adaptively with some probability $\alpha \in (0, 1)$. Consider a fixed choice e and $t = T$ and let $S_e \subseteq K$ be the set of types of balls choice e will take. For each $k \in S_e$, let $\mathcal{S}_k$ be the event that at $t = T$, we still have at least one ball of type k left. Our question is that how the adversary minimize $\Pr[\wedge_{k \in S_e} \mathcal{S}_k]$ subject to the constraints (1) $\sum_{t \in [T-1]} x^*_{e,t} a_{e,k} \leq B_k$ for each $k \in S_e$ and (2) $\sum_{e \in E} x^*_{e,t} \leq 1$ for each t. The equivalence between this new model and our original problem can be seen as follows: (1) each assignment corresponds a choice here; (2) for some assignment e with deadline t, we set $x^*_{e,t'} = 0$ for all $t' > t$. Thus we can safely ignore the deadline issue as far as ALG_1 is considered.

Consider a given $k \in S_e$. Let $E_k = \{e \in E | a_{e,k} = 1\}$ be the set of choices e that participate in the resource constraint of k. Let $x^*_{k,t} = \sum_{e \in E_k} x^*_{e,t}$. Notice that $x^*_{k,t} \leq 1$ and at time t, one of the choices in E_k arrives with probability $x^*_{k,t}$. Let $A_{k,t}$ be the indicator random variable that one of the choices in E_k arrives at t and $A_k = \sum_{t \leq T-1} A_{k,t}$, which denotes the random number of arrivals of choices in E_k over $T - 1$ rounds. For an integral A and B, let $p(A, \alpha, B) \doteq \Pr[Z \leq B - 1]$ where $Z \sim \mathrm{Bi}(A, \alpha)$ (binomial distribution) and we assume $p(A, \alpha, B) = 1$ for any $0 \leq A \leq B - 1$. Now consider a given set $A = \{A_k | k \in S_e\}$.

Lemma 5.1.

$$\Pr[\mathcal{S}_k | A_k] \geq p(A_k, \alpha, B_k), \quad \Pr[\wedge_{k \in S_e} \mathcal{S}_k | A] \geq \prod_{k \in S_e} p(A_k, \alpha, B_k)$$

Proof. Consider a given k and A_k. Given A_k trials and each time we take one ball independently with probability at most α. Thus we end at at least $B_k - 1$ balls with probability at least $p(A_k, \alpha, B_k)$. Notice that the events $\{(\mathcal{S}_k | A_k) | k \in S_e\}$ are positively correlated by the FKG inequality [20], which yields the second inequality. □

Lemma 5.2.

$$\Pr[\wedge_{k \in S_e} \mathcal{S}_k] \geq \prod_{k \in S_e} \exp\left(\mathbb{E}\left[\ln(p(A_k, \alpha, B_k))\right]\right)$$

Proof. First notice that $\Pr[\wedge_{k \in S_e} \mathcal{S}_k] = \mathbb{E}_A\left[\Pr[\wedge_{k \in S_e} \mathcal{S}_k | A]\right]$ by conditioning on the event A. From Lemma 5.1, we see the latter should be at least $\mathbb{E}_A\left[\prod_{k \in S_e} p(A_k, \alpha, B_k)\right]$. Thus

$$\Pr[\wedge_{k \in S_e} \mathcal{S}_k] = \mathbb{E}_A\left[\Pr[\wedge_{k \in S_e} \mathcal{S}_k | A]\right] \geq \mathbb{E}_A\left[\prod_{k \in S_e} p(A_k, \alpha, B_k)\right]$$

$$= \mathbb{E}_A\left[\exp\left(\sum_k \ln(p(A_k, \alpha, B_k))\right)\right] \geq \exp\left(\sum_k \mathbb{E}\left[\ln(p(A_k, \alpha, B_k))\right]\right)$$

$$= \prod_{k \in S_e} \exp\left(\mathbb{E}\left[\ln(p(A_k, \alpha, B_k))\right]\right)$$

□

The inequality in the second line to the third line is due to Jensen's inequality. Recall that $A_k = \sum_{t \leq T-1} A_{k,t}$ where $A_{k,t}$ is a Bernoulli random variable indicating if a choice $e \in E_k$ arrives at t. Notice that $\mathbb{E}[A_k] = \sum_{t \leq T-1} x^*_{k,t} \leq B_k$.

Lemma 5.3. *For any $\alpha \in [0, \frac{1}{2}]$ and integer $B_k \geq 1$,*

$$\mathbb{E}_{A_k}[\ln(p(A_k, \alpha, B_k))] \geq \ln(1 - \alpha)$$

We can show that in the worst scenario, the adversary will designate each A_k as a Poisson random variable with mean B_k such that $\mathbb{E}_{A_k}[\ln(p(A_k, \alpha, B_k))]$ gets minimized. The full proof of Lemma 5.3 can be seen in the full version. Now we have all ingredients to prove Theorem 1.1.

Proof. The proof is very similar to that of Theorem 5.2. Consider a given assignment e and $t = T - 1$ w.l.o.g. Notice that $\alpha = \frac{1}{\ell+1} \leq \frac{1}{2}$. From Lemma 5.2 and 5.3, we see that $\Pr[\wedge_{k \in S_e} \mathcal{S}_k] \geq (1 - \alpha)^{\ell}$. Thus by plugging in $\alpha = \frac{1}{\ell+1}$, we prove our claim. □

5.4 Simulation-based adaptive algorithm

In this section, we present a simulation-based algorithm. The main idea is as follows. Suppose we aim to develop an online algorithm achieving a ratio of $\gamma \in [0, 1]$. Consider an assignment $e = (i, j) \in E_t$ when worker j arrived at some time t. Let $\mathcal{S}_{e,t}$ be the event that e is safe conditioning on the arrival of e at t. By simulating the current strategy up to t, we can get an estimation of $\Pr[\mathcal{S}_{e,t}]$, say $\beta_{e,t}$, within an arbitrary small error. Therefore in the case e is safe at t, we can sample it with probability $\frac{x_{e,t}}{p_{j,t}} \frac{\gamma}{\beta_{e,t}}$, which leads to the fact that e is sampled with probability $\gamma x_{e,t}$ unconditionally.

The simulation-based attenuation technique has been used to attack other stochastic optimization problems as well such as stochastic knapsack [30] and stochastic matching [1]. Assume for now we can always get an accurate estimation $\beta_{e,t}$ of $\Pr[\mathcal{S}_{e,t}]$ for all t and e (It is easy to see that the sampling error can be folded into a multiplicative factor of $(1 - \epsilon)$ in the competitive ratio by standard Chernoff bounds). The formal statement of our algorithm, denoted by ALG_2, is as follows.

Algorithm 2: Simulation-based adaptive algorithm (ALG_2)

For each time t, assume some worker j arrives.

Let $\hat{E}_{j,t} \subseteq E_{j,t}$ be the set of *safe* available assignments we can make for j.

If $\hat{E}_{j,t} = \emptyset$, then reject j; otherwise sample an assignment $e \in \hat{E}_{j,t}$ with probability $\frac{x^*_{e,t}}{p_{j,t}} \frac{\gamma}{\beta_{e,t}}$.

Note that $\beta_{e,t}$ is the value of $\Pr[\mathcal{S}_{e,t}]$, which assumes to be known exactly through simulation. To ensure the above algorithm works with parameter γ, it suffices to show that $\beta_{e,t} \geq \gamma$ for all possible t and e.

Lemma 5.4. *By choosing $\gamma = 1/(\ell + 1)$, we have $\beta_{e,t} \geq \gamma$ for all $t \in [T]$ and $e \in E_t$.*

Proof. The proof is similar to that of Theorem 5.1. Consider a given t and $e \in E_t$. Focus on a given $k \in S_e$ and let $U_{k,t}$ be the usage of resource k at the beginning of t. For each $t' < t$ and $e' \in E_{t'}$, let $X_{e',t'}$ be the indicator random variable that e' is chosen at t'. Notice that $U_{k,t} = \sum_{t' < t} X_{e',t'} a_{e',k}$.

Now we prove by induction on t. For the base case $t = 1$, we see $\beta_{e,t} = 1$ for all $e \in E_t$. Thus we claim is valid. Assume our claim works for all $t' < t$, which leads to the fact that for all $e' \in E_{t'}$ with $t' < t$, e' will be made at t' with probability *exactly* equal to $x^*_{e',t'}\gamma$. In other words, $\mathbb{E}[X_{e',t'}] = x^*_{e',t'}\gamma$. Consider the event that e is safe at t with respect to resource k. By Markov's inequality, we have

$$\Pr[U_{k,t} \leq B_k - 1] = 1 - \Pr[U_{k,t} \geq B_k] \geq 1 - \gamma$$

Thus we have

$$\Pr[\mathcal{S}_{e,t}] = \Pr\left[\bigwedge_{k \in S_e} \left(U_{k,t} \leq B_k - 1\right)\right] \geq 1 - \ell\gamma \geq \gamma$$

The last inequality is valid since $\gamma \le 1/(\ell+1)$. □

The above Lemma validates ALG_2. By manipulating the simulation error in a proper way as shown in [1, 30], we can make sure that final ratio will have a relative error at most ε for any given $\varepsilon > 0$. Thus we prove our claim for Theorem 1.2. Note that the running time will depend on $1/\varepsilon$ polynomially.

6. EXTENSION TO COMBINED INTEGRAL AND NON-INTEGRAL RESOURCES

Recall that $\mathcal{K}_2$ is the set of non-integral resources and for each $k \in \mathcal{K}_2$, all $a_{e,k} \in [0,1]$. Let $B = \min_{k \in [K_2]} B_k$ and we assume B is large. In this section, we discuss how to extend the results in Section 5 here when non-integral resources are added with the large B assumption. In particular, we are interested in how large B should be such that we lose at most ϵ in the competitive ratio. By default we assume $\mathcal{K}_1 \neq \emptyset$ and $\ell_1 \ge 1$.

6.1 Extension of ALG_1

In this section, we analyze the performance of ALG_1 with parameter $\alpha = 1/(\ell_1+1) \le 1/2$ when non-integral resources are added. Recall that in ALG_1, each assignment e is made at t non-adaptively with probability at most $\alpha x^*_{e,t}$. Let $X_{e,t}, Y_{e,t}$ indicate if e is made at t and if e is safe at t respectively. Let $Z_{e,t}$ indicate if e comes *and* gets sampled at t when e is safe at t. Here we treat $Z_{e,t}$ is Bernoulli random variable with mean $\alpha x^*_{e,t}$ and *independent* from $Y_{e,t}$ in the following way: when e comes at t while e is not safe, we continue to set $Z_{e,t} = 1$ with probability $\alpha x^*_{e,t}/p_{j,t}$ and 0 otherwise, i.e., pretending e is safe. Observe that (1) $X_{e,t} = Y_{e,t} Z_{e,t} \le Z_{e,t}$; (2) For any two random variables $Z_{e,t}$ and $Z_{e',t'}$, the two will be independent if $t \neq t'$ and negatively correlated if $t = t'$. Now we start to prove Theorem 1.3.

Proof. Focus on a given t and an assignment $e \in E_t$. Let $S_1 = S_e \cap \mathcal{K}_1$ and $S_2 = S_e \cap \mathcal{K}_2$. Let $\mathcal{S}_{k,t}$ be the event that e is safe with respect to resource k at t. From the analysis of Theorem 1.1, we see that $\Pr[\wedge_{k \in S_1} \mathcal{S}_{k,t}] \ge (1-\alpha)^{\ell_1}$. Now we focus on analyzing the value $\Pr[\wedge_{k \in S_2} \mathcal{S}_{k,t}]$. Let $U_{k,t}$ be the usage of resource k at the beginning of t, i.e., $U_{k,t} = \sum_{t'<t} \sum_{e' \in E_{t'}} X_{e',t'} a_{e',k}$.

Notice that for each $k \in S_2$,

$$\begin{aligned}\Pr[\mathcal{S}_{k,t}] &\ge 1 - \Pr[U_k \ge B_k - 1] \\ &= 1 - \Pr[\sum_{t'<t} \sum_{e' \in E_{t'}} X_{e',t'} a_{e',t'} \ge B_k - 1] \\ &\ge 1 - \Pr[\sum_{t'<t} \sum_{e' \in E_{t'}} Z_{e',t'} a_{e',t'} \ge B_k - 1]\end{aligned}$$

Let $H_{k,t} = \sum_{t'<t} \sum_{e' \in E_{t'}} Z_{e',t'} a_{e',t'}$. Notice that (1) $\mathbb{E}[H_{k,t}] \le \alpha B_k$ and (2) former discussion shows that $\{Z_{e',t'} | e' \in E_{t'}, t' < t\}$ are 1-correlated as defined in [34]. Thus from there, we can effectively view them as "independent" and apply the Chernoff bound to upper bound the value $\Pr[Z_{k,t} \ge B_k - 1]$. WLOG assume $B_k = B$ and we have

$$\begin{aligned}\Pr[H_{k,t} \ge B-1] &\le \exp\left(\frac{-\alpha B(\frac{B-1}{\alpha B} - 1)^2}{\frac{B-1}{\alpha B} + 1}\right) \\ &= \exp\left(-\frac{1-\alpha-1/B}{1+\alpha-1/B}(B(1-\alpha)-1)\right) \\ &\le \exp\left(-\frac{1}{2}\frac{1-\alpha}{1+\alpha}(B(1-\alpha)-1)\right)\end{aligned}$$

To get the last inequality we assume $B \ge 4$. Thus

$$\Pr[\wedge_{k \in S_2} \mathcal{S}_{k,t}] \ge 1 - \ell_2 \exp\left(-\frac{1}{2}\frac{1-\alpha}{1+\alpha}(B(1-\alpha)-1)\right) \doteq 1 - \epsilon$$

We solve that it will suffice $B \ge 2\ln(\frac{\ell_2}{\epsilon})\left(1 + \frac{3\ell_1+2}{\ell_1^2}\right) + 2$. In this case, we get a competitive ratio of $\frac{1}{\ell_1+1}\left((1-\frac{1}{\ell_1+1})^{\ell_1} - \epsilon\right)$. □

6.2 Extension of ALG_2

Suppose we aim for a competitive ratio of $\gamma = \frac{1-\epsilon}{\ell_1+1}$ for ALG_2 where the multiplicative loss ϵ is due to the adding of non-integral resources (we ignore all simulation errors first and handle them later). This implies, for each time t and assignment e, we try to maintain that e is made at t with probability equal to $\frac{1-\epsilon}{\ell_1+1}$. From the analysis in Section 5.4, it would suffice to show at each time t, e is safe with probability $\beta_{e,t} \ge \gamma$. Focus on a given assignment e and let $\mathcal{S}_{k,t}$ be the event that e is safe at t with respect to the resource k. Let $S_1 = S_e \cap \mathcal{K}_1$ and $S_2 = S_e \cap \mathcal{K}_2$. From the proof of Lemma 5.4, we see that all integral resources are safe at t with probability $\Pr[\wedge_{k \in S_1} \mathcal{S}_{k,t}] \ge 1 - \frac{(1-\epsilon)\ell_1}{\ell_1+1}$. Thus the remaining issue is to show that $\Pr[\wedge_{k \in S_2} \mathcal{S}_{k,t}] \ge 1-\epsilon$, which by union bound leads to the fact that $\beta_{e,t} = \Pr[\wedge_{k \in S_e} \mathcal{S}_{k,t}] \ge \gamma = \frac{1-\epsilon}{\ell_1+1}$.

Section 6.1 shows that when B is large, all non-integral resources are almost safe throughout T in ALG_1 by applying Chernoff bound and union bound. As for ALG_2, the same analysis failed due to the following challenges: (1) we cannot upper bound $X_{e,t}$ by some *independent or negatively correlated* $Z_{e,t}$ as before; (2) $\{X_{e,t}\}$ itself can be positively correlated as shown in the following example.

Example 6.1. *Consider an unweighted star graph $G = (I, J, E)$ where $|I| = 1, |J| = 3, E = (e_1, e_2, e_3)$ with $T = 2$. Suppose at $t = 1$, $j = 1, 2$ arrives with equal probability $p_j = 1/2$ and at $t = 2$, $j = 3$ will arrive with probability $p_j = 1$. Let e_1, e_2, e_3 denote respectively the assignment we consider when $j = 1, 2$ comes at $t = 1$ and $j = 3$ comes at $t = 2$. Let $K = 2$ with $\mathbf{B} = (1, 1)$ and $\mathbf{a}_{e_1} = (1, 0)$, $\mathbf{a}_{e_1} = (0, 1)$ and $\mathbf{a}_{e_3} = (0, 1)$. Suppose LP (4.1) offers us the following optimal solution: $x^*_{e_1} = x^*_{e_2} = 1/2$ and $x^*_{e_3} = 1/2$. In our context, $\ell_1 = 1, \gamma = 1/2$ and ALG_2 goes as follows: at $t = 1$, e_1 and e_2 will be made with probability $1/2$ when each comes; at the beginning of $t = 2$, e_3 is safe with probability $\beta = 3/4$ and accordingly, it will be made with probability $\frac{x^*_3}{p_3}\frac{\gamma}{\beta} = \frac{1}{3}$ when it comes.*

Recall that $X_{e,t}$ indicates if the assignment e is made at t. We can verify that $\Pr[X_{e_1,t=1} = 1] = x^*_1/2 = 1/4$ and $\Pr[X_{e_3,t=2} = 1] = x^*_3/2 = 1/4$. $\Pr[X_{e_3,t=2} = 1 | X_{e_1,t=1} = 1] = 1/3$, that is because e_3 is safe with probability 1 at $t = 2$ conditioning on $X_{e_3,t=2} = 1$. □

We use the technique of *virtual algorithms* to attack the potential positive correlation among $\{X_{e,t}\}$. Suppose we run ALG_2 with some parameter γ up to the time t such that for each e' and $t' < t$, $\Pr[X_{e',t'} = 1] = \gamma x^*_{e',t'}$. Now we try to lower bound the value $\beta_{k,t} \doteq \Pr[\mathcal{S}_{k,t}]$ for a given e and $k \in S_2$ with $S_2 = S_e \cap \mathcal{K}_2$.

Consider the simple setting where only one non-integral resource k is involved. Suppose we run ALG_1 with parameter $\alpha = \frac{\gamma}{1-\delta}$ as a virtual algorithm up to time t and let $\beta'_{k,t}(\delta) = \Pr[\mathcal{S}'_{k,t}]$ be the probability that e is safe at time t with respect to resource k in the virtual algorithm. Here $\delta = o(1)$ when $B \to \infty$.

Lemma 6.1. *For any δ with $\beta'_{k,t}(\delta) \ge 1-\delta$, we have $\beta_{k,t} \ge 1-\delta$.*

Proof. Consider a feasible δ with $\beta'_{k,t}(\delta) \ge 1-\delta$. For each e' and $t' < t$, let $X'_{e',t'}$ indicate that e' is made at t' in the virtual algorithm.

We see $\Pr[X'_{e',t'}] = \frac{\gamma x^*_{e',t'}}{1-\delta} \Pr[e' \text{ is safe at } t'] \geq \gamma x^*_{e',t'}$. Notice that in our algorithm ALG_2 with parameter γ, each assignment e' will be made with probability equal to $\gamma x^*_{e',t'}$. Therefore we claim that in ALG_2, $\beta_{k,t} = \Pr[\mathcal{S}_{k,t}] \geq \Pr[\mathcal{S}'_{k,t}] = \beta'_{k,t} \geq 1-\delta$. □

Now we have all ingredients to prove Theorem 1.4.

Proof. Focus on an assignment e and t. Ignore the simulation error first and we try to show that when $B \geq 3\ln(\frac{\ell_2}{\epsilon})(1+\frac{1}{\ell_1})+2$, $\Pr[\mathcal{S}_{k,t}] \geq 1-\frac{\epsilon}{\ell_2}$ for each $k \in S_2$.

Lemma 6.1 tells us that we just need to find a feasible δ such that $\beta'_{k,t} \geq 1-\delta$. In this case, we have $\Pr[\mathcal{S}_{k,t}] \geq 1-\delta$ and setting $\epsilon = \ell_2\delta$ will complete the proof. Consider the virtual algorithm ALG_1 with parameter $\alpha = \frac{(1-\epsilon)/(\ell_1+1)}{1-\delta}$ and let $H_{k,t} = \sum_{t'<t}\sum_{e' \in E_{t'}} Z_{e',t'} a_{e',t'}$ where $\Pr[Z_{e',t'} = 1] = \alpha$ for each e' and $t' < t$. Notice that $\Pr[\mathcal{S}'_{k,t}] \geq 1 - \Pr[H_{k,t} \geq B-1]$ and $\mathbb{E}[H_{k,t}] \leq \alpha B$. WLOG assume $\mathbb{E}[H_{k,t}] = \alpha B$ and $\ell_1 \geq 2$. Let $\Delta = \frac{B-1}{\alpha B} - 1$. We have

$$\begin{aligned}\Delta &= (1-\frac{1}{B})\frac{(1-\delta)(1+\ell_1)}{1-\epsilon} - 1 \\ &= \frac{\ell_1+\epsilon-\delta-\delta\ell_1}{1-\epsilon} - \frac{1}{B}\frac{(1-\delta)(1+\ell_1)}{1-\epsilon} \geq \frac{1-\delta}{1-\epsilon}\Big(\ell_1 - \frac{\ell_1+1}{B}\Big) \geq 1\end{aligned}$$

The last inequality assumes that $B \geq 3 \geq 1 + \frac{2}{\ell_1 - 1}$. Therefore by the Chernoff Bound, we have

$$\begin{aligned}\Pr[H_{k,t} \geq B-1] &= \Pr\left[H_{k,t} \geq \mathbb{E}[H_{k,t}](1+\Delta)\right] \\ &\leq \exp\left(-\frac{1}{3}\frac{B(1-\epsilon)}{(\ell_1+1)(1-\delta)}\frac{1-\delta}{1-\epsilon}\Big(\ell_1 - \frac{\ell_1+1}{B}\Big)\right) \\ &= \exp\left(-\frac{1}{3}B\Big(1-\frac{1}{B}-\frac{1}{\ell_1+1}\Big)\right)\end{aligned}$$

which implies that

$$\Pr[\mathcal{S}_{k,t}] \geq \Pr[\mathcal{S}'_{k,t}] \geq 1 - \exp\left(-\frac{1}{3}B\Big(1-\frac{1}{B}-\frac{1}{\ell_1+1}\Big)\right)$$

When $B \geq 3\ln(\frac{\ell_2}{\epsilon})(1+\frac{1}{\ell_1})+2$, we can verify that the right-hand side value at least $1-\delta = 1-\frac{\epsilon}{\ell_2}$. Thus we prove our claim that for each $k \in S_2$, $\Pr[\mathcal{S}_{k,t}] \geq 1-\frac{\epsilon}{\ell_2}$, which yields that ALG_2 achieves a ratio of $(1-\epsilon)/(\ell_1+1)$. After incorporating the simulation error, we will have an additional multiplicative factor $(1-\epsilon)$ in the competitive ratio. Thus we prove Theorem 1.4.

□

7. EXPERIMENTAL EVALUATION

In this section, we propose and evaluate a number of heuristic algorithms for the BOA problem. We start with the case when only integral resources are involved. Section 5 shows that non-adaptive ALG_1 and adaptive ALG_2 can achieve a ratio of at least $\frac{1}{\ell+1}\frac{1}{e}$ and $\frac{1}{\ell+1}$ respectively, where ℓ is the upper bound of integral resources requested by each assignment. In our experiments, we show that the performance is far better than these theoretical worst case bounds (such bounds hold only for some extremely specialized cases such as the one shown in Example 5.2).

Our experimental setup is as follows.

1. For each j, recall that $N(j)$ is the set of tasks that interest j. We generate $N(j)$ by sampling each $i \in [m]$ independently with some probability, say 0.3. We propose to study the sensitivity to this parameter further in the future.
2. Let P_1 be the arrival probability matrix of size $n \times T$ such that $P_1(i,j) = p_{i,j}$. We first generate a random "seed" matrix P_0 of size $n \times T_1$ such that for each $t \in [T_1]$, the values in the t^{th} column of P_0 are uniformly distributed over $[0,1]$ *conditioning on the column sum is* 1, i.e., $\sum_t P_0(i,t) = 1$. We achieve this by running the file "randfixedsum.m" due to Roger Stafford [3]. Once we have a fixed P_0, we generate P_1 by sampling one column from P_0 uniformly for T times. Notice that if we generate P_1 in the direct way as P_0, then each j will have almost the same arrivals over T rounds since T assumes to be very large. In our case we set $T_1 = m \ll T$ and we hope we can create some potential bias of the arrivals over all $j \in [n]$ and that can pass to P_1.
3. Let E be the set of assignments generated as shown in the first point. For each assignment $e \in E$, we independently choose a uniform value $w_e \in [0,1]$.
4. Recall that $\mathcal{K}_1$ and $\mathcal{K}_2$ are the set of integral and non-integral resources respectively. We generate a budget B_k by uniformly sampling an integer from $[UB] = \{1,2,3,\ldots,UB\}$ for each $k \in \mathcal{K}_1$ and from $[LB, 5*LB]$ for each $k \in \mathcal{K}_2$ respectively. Here UB and LB are parameters specified in advance.
5. Recall S_e is the set of resources requested by e. For each e, we first generate a random permutation π_1 over $\mathcal{K}_1$ and then set $S_e \cap \mathcal{K}_1$ as the first $\lceil \rho_0 * K_1 \rceil$ elements of π_1. Set $a_{e,k} = 1$ for each $k \in S_e \cap \mathcal{K}_1$. We then generate another random permutation π_2 over $\mathcal{K}_2$ and set $S_e \cap \mathcal{K}_2$ as the first $\lceil \rho_0 * K_2 \rceil$ elements of π_2. Sample a uniform value from $[0,1]$ for $a_{e,k}$ for each $k \in S_e \cap \mathcal{K}_2$. Here $\rho_0 \in [0,1]$ is a parameter given in advance.
6. For each e, let d_e be the deadline of e. We sample a random integer from $[T/2, T]$ uniformly as d_e for each $e \in E$. In this experiment we consider a relative more flexible setting: allow assignments with respect to a single task to have potentially distinctive deadlines.

Let $\mathsf{ALG}_1(\alpha)$ denote the algorithm shown in Section 5.1 with parameter α. Theorem 1.1 shows that $\mathsf{ALG}_1(\frac{1}{\ell+1})$ can achieve a ratio at least $\frac{1}{\ell+1}\frac{1}{e}$. Our experimental results suggest that it will be too conservative for the choice of $\alpha = \frac{1}{\ell+1}$. This inspires us to propose the following four heuristics. All these four algorithms are non-adaptive essentially except the last one. Consider some time t when j comes and let $E_{j,t} = \{e = (i,j) | i \in N(j), d_e \geq t\}$ be the set of available (not necessarily safe) assignments related to j.

1. NAdap: sample an assignment $e \in E_{j,t}$ with probability $\frac{x^*_{e,t}}{\sum_{e \in E_{j,t}} x^*_{e,t}}$. Make it iff e is safe.
2. $\mathsf{ALG}_1(1)$: sample an assignment $e \in E_{j,t}$ with probability $\frac{x^*_{e,t}}{p_{j,t}}$. Make it iff e is safe.
3. USamp: sample an assignment $e \in E_{j,t}$ uniformly from $E_{j,t}$. Make it iff e is safe.
4. Greedy: choose the assignment $e \in E_{j,t}$, which has the largest weight w_e among all safe options in $E_{j,t}$.

Remark: (1) the first two are both LP-based non-adaptive algorithms; the third is non-adaptive but blind to the LP solution; the last one is adaptive and blind to the LP solution as well, the strategy

[3] https://www.mathworks.com/matlabcentral/fileexchange/9700-random-vectors-with-fixed-sum/content/randfixedsum.m

gets updated as the set of safe options shrinks in later rounds; (2) the second can be viewed as the first one plugged with an attenuation factor $\frac{\sum_{e \in E_{j,t}} x^*_{e,t}}{p_{j,t}} \leq 1$. (3) we did not test ALG_2 since the implementation is really time-consuming even on moderate problem size.

For each set of parameters $\mathcal{P} = (m, n, K_1, K_2, T, UB, LB, \rho_0)$, we generate a set $\mathcal{I}(\mathcal{P})$ of 5 random instances as described before. For each instance $I \in \mathcal{I}(\mathcal{P})$, we run the above five algorithms each on I for 100 times and take the mean as the final performance. For each given instance I, let $\mathsf{OPT}(I)$ be the LP optimal value and $\mathsf{ALG}(I)$ be the final performance on I. We define $\rho(\mathsf{ALG}, I) = \mathsf{ALG}(I)/\mathsf{OPT}(I)$, which is the ratio of performance of ALG to the LP value on I. For each set of parameters $\mathcal{P} = (m, n, K_1, K_2, T, UB, LB, \rho_0)$, we generate 5 random instances as described before and set the mean ratio as $\rho(\mathsf{ALG}, \mathcal{P})$ for each ALG. The results can be seen in Figures 1, 2 and 3. The detailed discussion can be found in the full version.

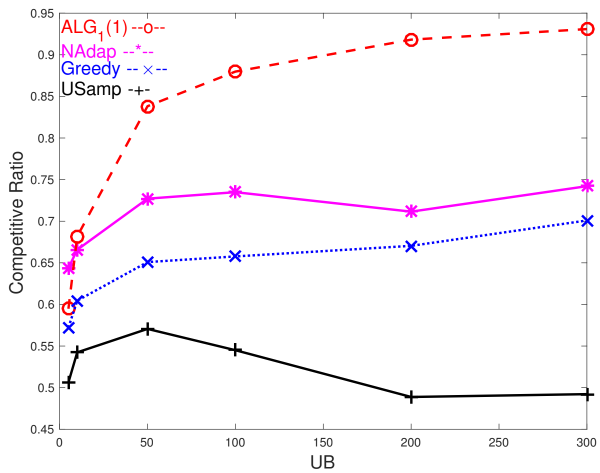

Figure 1: Performance of the four algorithms as UB increases where: $m = 10, n = 50, K_1 = 90, K_2 = 0, T = 3000, \rho_0 = 0.1$. The best LP-based heuristic $\mathsf{ALG}_1(1)$ (red-colored) strictly beats the best LP-blind strategy Greedy (blue-colored).

Acknowledgments

Part of this work is done when Pan Xu interned at the IBM T. J. Watson Research Center during the summer of 2016. Aravind Srinivasan's research was supported in part by NSF Awards CNS-1010789 and CCF-1422569, and by a research award from Adobe, Inc. Pan Xu's research was supported in part by NSF Awards CNS-1010789 and CCF-1422569.

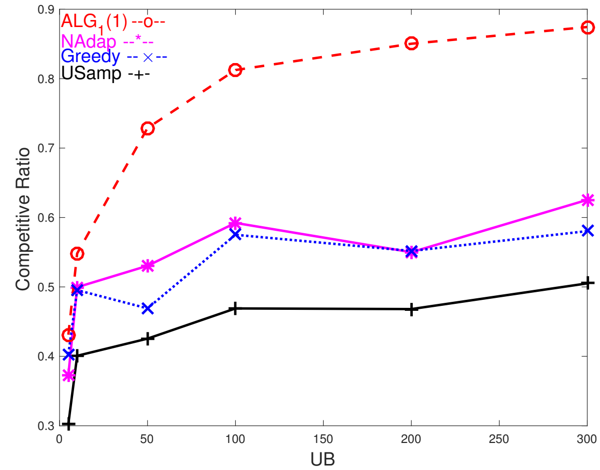

Figure 2: Performance of the four algorithms as UB increases where $m = 10, n = 50, K_1 = 90, K_2 = 0, T = 3000, \rho_0 = 0.5$. The best LP-based heuristic $\mathsf{ALG}_1(1)$ (red-colored) strictly beats the best LP-blind strategy Greedy (blue-colored).

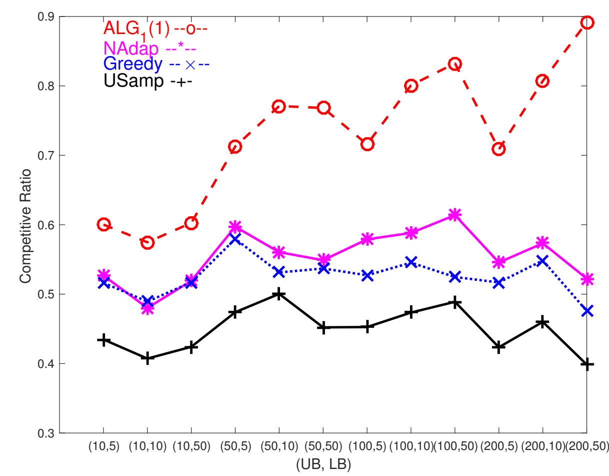

Figure 3: Performance of the four algorithms as (UB, LB) increases where: $m = 10, n = 50, K_1 = 50, K_2 = 40, T = 2000, \rho_0 = 0.5$. The best LP-based heuristic $\mathsf{ALG}_1(1)$ (red-colored) strictly beats the best LP-blind strategy Greedy (blue-colored).

REFERENCES

[1] M. Adamczyk, F. Grandoni, and J. Mukherjee. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*, pages 1–12. Springer, 2015.

[2] S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

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

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

[5] E. M. Arkin and R. Hassin. On local search for weighted k-set packing. *Mathematics of Operations Research*, 23(3):640–648, 1998.

[6] S. Assadi, J. Hsu, and S. Jabbari. Online assignment of heterogeneous tasks in crowdsourcing markets. In *Third AAAI Conference on Human Computation and Crowdsourcing*, 2015.

[7] A. Badanidiyuru, R. Kleinberg, and Y. Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 128–145. ACM, 2012.

[8] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.

[9] N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. Solving packing integer programs via randomized rounding with alterations. *Theory of Computing*, 8(1):533–565, 2012.

[10] A. Baveja, A. Chavan, A. Nikiforov, A. Srinivasan, and P. Xu. Improved bounds in stochastic matching and optimization. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 40. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[11] P. Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. In *Scandinavian Workshop on Algorithm Theory*, pages 214–219. Springer, 2000.

[12] B. Brubach, K. A. Sankararaman, A. Srinivasan, and P. Xu. New algorithms, better bounds, and a novel model for online stochastic matching. *arXiv preprint arXiv:1606.06395*, 2016.

[13] N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.

[14] N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. *Automata, Languages and Programming*, pages 266–278, 2009.

[15] N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78. ACM, 2009.

[16] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 29–38. ACM, 2011.

[17] N. R. Devanur, B. Sivan, and Y. Azar. Asymptotically optimal algorithm for stochastic adwords. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 388–404. ACM, 2012.

[18] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *International Workshop on Internet and Network Economics*, pages 374–385. Springer, 2009.

[19] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 117–126. IEEE, 2009.

[20] C. M. Fortuin, P. W. Kasteleyn, and J. Ginibre. Correlation inequalities on some partially ordered sets. *Comm. Math. Phys.*, 22(2):89–103, 1971.

[21] Z. Füredi, J. Kahn, and P. D. Seymour. On the fractional matching polytope of a hypergraph. *Combinatorica*, 13(2):167–180, 1993.

[22] A. Gupta and V. Nagarajan. A stochastic probing problem with applications. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 205–216. Springer, 2013.

[23] B. Haeupler, V. S. Mirrokni, and M. Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *International Workshop on Internet and Network Economics*, pages 170–181. Springer, 2011.

[24] E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating k-set packing. *computational complexity*, 15(1):20–39, 2006.

[25] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In *AAAI*, volume 12, pages 45–51, 2012.

[26] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1):68–72, 1989.

[27] P. Jaillet and X. Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2013.

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

[29] T. Kesselheim, A. Tönnis, K. Radke, and B. Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 303–312. ACM, 2014.

[30] W. Ma. Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1154–1163. Society for Industrial and Applied Mathematics, 2014.

[31] V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.

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

[33] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM (JACM)*, 54(5):22, 2007.

[34] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997.

[35] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1157–1166. ACM, 2013.

[36] A. Singla and A. Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1167–1178. ACM, 2013.

[37] A. Slivkins and J. W. Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges. *ACM SIGecom Exchanges*, 12(2):4–23, 2014.

[38] A. Subramanian, G. S. Kanth, S. Moharir, and R. Vaze. Online incentive mechanism design for smartphone crowd-sourcing. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, pages 403–410. IEEE, 2015.