Online Stochastic Generalized Assignment Problem with Demand Learning

Yanwen Li Zihao Li Limeng Liu Hao Wang Zhenzhen Yan
Nanyang Technological University
{yanwen.li, hao_wang, yanzz}@ntu.edu.sg
{zihao004, limeng001}@e.ntu.edu.sg

Abstract

We study an online generalized assignment problem under a stochastic arrival model. Items arrive in an online manner, following a known independent and identical distribution. The demand associated with each arrival is stochastic and is drawn from an unknown type-specific distribution. The demand is only revealed after the resource allocation decision has been made. Upon the arrival of each online item, two decisions need to be made: either pack it into an offline bin and deduce a certain capacity, or reject it. Successful matches between offline bins and online items result in variable rewards, which may differ for different matching pairs. The objective is to maximize the total reward of the packing scheme while adhering to capacity constraints. We utilize the idea of exploration-exploitation to derive an algorithm that simultaneously learns about the demand and allocates online items to offline bins. We present a non-asymptotic parametric guarantee when the demand distributions are Bernoulli distributions. Finally, we conduct numerical experiments to evaluate the effectiveness of our algorithm among different demand distributions.

1 Introduction

Online generalized assignment problem (GAP) is a classical optimization problem in online resource allocation. In this problem, we are presented with a collection of offline bins, each with its own capacity, and a sequence of online items with varying demands. Upon the arrival of each online item, we can allocate it to an available offline bin, which results in a reduction from the bin's capacity corresponding to the item's demand. Alternatively, we can choose to reject the item irrevocably. Our objective is to maximize the total reward without violating any bin capacity constraints.

In this paper, we study the online GAP under a stochastic arrival model. The arrival process is assumed to follow a known independent and identical distribution (i.i.d.). The demand associated with each arrival is drawn from an unknown type-specific distribution and is only revealed after a resource allocation decision has been made. This problem setting is applicable in numerous real-world scenarios. For example, in cloud computing, tasks are allocated to computers and require a certain amount of processing time. However, the precise processing time is frequently undisclosed until the task is executed. In ride-sharing, the online requests arrive randomly on the platform and each request demands a random number of seats. However, the exact demand size remains unknown until a vehicle is assigned to the request. In food delivery, delivery requests emerge randomly in the system and each request may occupy a different capacity of the deliveryman's vehicle. However, the specific capacity consumed by each request is not revealed until the resource allocation decision is made. In all of these scenarios, the demand for packing an item into a bin can only be observed after the packing decision has been made, which is the setting to be explored in this paper.

There has been an increasing number of literature analyzing the stochastic demand model in the online GAP [Alaei et al., 2013, Alaei, 2014, Jiang et al., 2022, Stein et al., 2020, Wang et al., 2022]. But

Preprint. Under review.

most of them allow demand to be observed before making the packing decision. Alaei et al. [2013] is the most related one. Specifically, they assume the demand distribution for packing each item type into each bin is known before the online process, but the specific realization of the demand is only learned after the item has been packed. They proposed a threshold algorithm with a competitive ratio of $1 - \frac{1}{\sqrt{k}}$, under the assumption that no item can consume more than $\frac{1}{k}$ of a bin's capacity. Our paper differs from Alaei et al. [2013] in the following perspectives. First, we assume the demand distribution is *unknown* prior to the online process, which is in contrast to the assumption of known demand distribution in Alaei et al. [2013]. Second, our arrival process is assumed to be i.i.d., whereas their analysis only requires independence. To the best of our knowledge, this paper is the first to solve the online GAP with unknown demand distribution.

It is worth highlighting that being able to accommodate unknown demand distribution makes a significant contribution in practice. In many applications, the demand distribution is not known in advance. For instance, consider a market with newly launched products, where limited data is available to determine the demand distribution beforehand. Consequently, the distribution needs to be learned during the online resource allocation process. Another example can be seen in online advertising, where advertisers bid for ad placements on websites or search engines. The demand for ad placements, as indicated by user clicks or impressions, is often unknown in advance. Advertisers must make allocation decisions without precise knowledge of the demand distribution due to the unpredictable nature of user behavior and engagement.

On the other hand, it is challenging to deal with the unknown demand distribution. Many existing algorithms are designed under the assumption of either observed demand or pre-known demand distribution. For instance, a commonly used large-small technique (see Feldman et al. [2021], Naori and Raz [2019], Stein et al. [2020]) is to categorize items into large and small groups based on their demand size and subsequently handle them differently. However, this approach is not applicable in our model since the precise demand is not available until the packing decision is made.. Another commonly employed idea is to reduce the original problem to variants of the magician problem. This cannot be easily adapted to our model either due to its heavy dependence on known demand distributions in the algorithm design and subsequent analysis (e.g., Alaei et al. [2013], Jiang et al. [2022]). In general, the lack of knowledge regarding demand distribution makes it difficult to adapt allocation strategies to unforeseen demand scenarios. In our paper, we tackle the challenge by adopting the idea of exploration-exploitation to dynamically learn the demand distribution and guide the packing decision.

Contributions. We summarize our main contributions as follows. In this paper, we study the online GAP under a known independent and identically distributed arrival process, which is commonly assumed in the literature of online resource allocation [Feldman et al., 2009, Huang et al., 2022, Jaillet and Lu, 2014]. Each online arrival requests a random demand which is drawn from an unknown type-specific distribution and generates a type-specific reward if being matched to a bin. Upon each arrival, we need to make an immediate and irrevocable decision: whether to pack this item into a bin or reject it outright. The specific demand of the present item can only be revealed after the decision is made. Our goal is to maximize the total reward of the packing without violating the capacity constraints.

In this setting, we adopt the idea of exploration-exploitation to first learn the demand distribution and then leverage it to guide our packing decisions. If all demand distributions are Bernoulli distributions, we present a parametric performance guarantee, as shown in Theorem 4.6. We further test our algorithm on synthetic datasets. We introduce three heuristics derived from our main algorithm and demonstrate their effectiveness by comparing them to a greedy benchmark. Importantly, we find that these heuristics consistently perform well across a variety of demand distributions, indicating their robustness and versatility.

Our work is the first to study the online GAP with unknown demand distribution. We derive a multi-phase algorithm to learn demand (explore) and optimize the allocation decision (exploit). We further provide a non-trivial analysis of the derived algorithm.

Related work. Online GAP has been widely studied in the recent years [Aggarwal et al., 2011, Albers et al., 2020, Feldman et al., 2009, Jiang et al., 2022, Kesselheim et al., 2018]. Feldman et al. [2009] is the first work to consider the online GAP problem and showed that there exists an algorithm that can achieve a competitive ratio of $1 - \frac{1}{e}$ under the free disposal assumption. Here, the free disposal assumption means that we can allocate more items to a bin but only a subset of these items without violating the capacity constraint can gain a reward. Without free disposal, Aggarwal et al. [2011] demonstrated that there is no online algorithm with a positive competitive ratio if the arrival order of the online items is determined by an adversary. Some other studies consider a random order arrival, where the arriving order of online items is assumed to be uniformly selected from all possible permutations. In this context, Albers et al. [2020] introduced a randomized algorithm that achieves a competitive ratio of $\frac{1}{6.99}$, which improves the ratio $\frac{1}{8.06}$ proposed by [Kesselheim et al., 2018]. To the best of our knowledge, it is the best ratio under the random arrival model.

Note that the aforementioned works assume a deterministic demand for each online item. Recently, some studies have started to consider stochastic demands under a stochastic arrival model. In this setting, the reward and demand associated with packing each arriving item follow a known joint distribution, and the specific reward and demand pair for each item is revealed upon its arrival, before a resource allocation decision has been made. Jiang et al. [2022] introduced an algorithm that achieves a tight performance guarantee of $\frac{1}{3+e^{-2}}$ under this setting.

Some other works study the online GAP under the setting that we can only learn the demand realization after packing the item or rejecting it [Alaei et al., 2013, Bhalgat, 2011, Xu, 2022], which is the problem setting considered in this paper. Among them, Alaei et al. [2013] is the most related one. They also study a stochastic demand model but they assume demand distribution is known in advance, whereas our paper specifically addresses the challenge of dealing with an unknown demand distribution. Under the stochastic arrival model, Alaei et al. [2013] reduced the problem to a generalized magician problem (GMP) and proposed a threshold algorithm that achieves a competitive ratio of $1 - \frac{1}{\sqrt{k}}$, under the assumption that each online item can take at most $\frac{1}{k}$ fraction of any bin's capacity. If every bin has k-unit capacity and the demand follows a Bernoulli distribution, Alaei [2014] reduced the problem to a standard magician problem (MP) and established a competitive ratio of $1 - \frac{1}{\sqrt{k+3}}$. We adopt their reduction idea in our analysis. Specifically, we also consider reducing our original problem to GMP. However, the unknown demand distribution adds significant complexity to the analysis. The algorithms for the classical GMP or MP used in Alaei et al. [2013], Alaei [2014] heavily depend on having full knowledge of the demand distributions, which is not available to us. To tackle this challenge, we employ the idea of exploration-exploitation to dynamically learn the demand distribution and guide our packing decisions. The integration of the exploration-exploitation framework into GMP/MP introduces substantial intricacies to the analysis.

2 Preliminaries

We first introduce some notations and abbreviations in this paper. Let $[m]$ denote the set of $\{1, 2, \cdots, m\}$ where m is a positive integer. We use "CDF" to represent the cumulative distribution function and "LP" to denote the linear program.

We consider an online stochastic generalized assignment problem (GAP) with unknown demand, where we need to allocate online items to offline bins while the demand of each item is revealed after a packing decision is made. Denote U as the set of offline bins, and each bin $u \in U$ has an initial capacity of c_u units. We assume all capacities c_u are integers at least 2 since Proposition 1 in Alaei et al. [2013] has demonstrated the impossibility of achieving a positive ratio when the capacity is equal to 1 even with known demand distribution. Let V denote the set of online item types. We assume that each arriving item is sampled from V according to a known independent and identically distributed (i.i.d.) arrival process.

Suppose there are T periods in total. At each period $t \in [T]$, we sample an item of type v with probability p_v where $\sum_{v \in V} p_v = 1$ and the probabilities $\{p_v\}$ are known to us. After the arrival of the tth item, which is of type v, we need to decide whether to allocate it to an offline bin or reject the item immediately and irrevocably. The demand of the item d'_t will be revealed according to an unknown distribution D_v only after the decision has been made. If we reject the item, we will not get any reward. If the item of type v is allocated to bin u and d'_t is not larger than the remaining capacity of u, the allocation is successful. Then the capacity of u is deducted by d'_t, and we will earn a non-negative reward r_{uv}, which is known to us. But if the remaining capacity of the allocated bin is smaller than d'_t, the allocation is failed. As a result, we get zero reward and the capacity of u remains unchanged. The objective is to maximize the total reward in the total time horizon of T. For ease of notation, we denote the number of offline bins and the number of online item types by n and m, respectively.

The demand distribution D_v of each online type v is unknown to us. But without loss of generality, we assume the support of D_v is a subset of $[0,1]$ and expected value is d_v which is also unknown. For the sake of analysis, we further assume there exists a known constant $\underline{d} > 0$ such that $d_v \geq \underline{d}$ for each $v \in V$.

Competitive Ratio. We use the competitive ratio to measure the performance of an algorithm. Let $\text{ALG}(I)$ denote the expected total reward given by an online algorithm ALG for a problem instance I of our model. The expectation is taken over the random arrivals of the items, the revealed random demand of each item, and the randomized (if needed) algorithm. The benchmark is the offline optimal algorithm OPT which knows the online arrivals and demands at the beginning and maximizes the total reward. Similarly, we use $\text{OPT}(I)$ to denote the total expected reward achieved by OPT for the instance I. The competitive ratio of ALG is defined as the minimum ratio of $\text{ALG}(I)$ over $\text{OPT}(I)$ among all instances of our model. For simplicity of the notation, we will drop I in the following analysis when there is no ambiguity, e.g., let OPT denote the total expected reward by the offline optimal algorithm.

Linear Program Benchmark. We use the following linear program (LP) (1) to assist the competitive ratio analysis.

$$\begin{aligned} \textbf{max} & \sum_{u \in U, v \in V} r_{uv} x_{uv} && && (1) \\ \textbf{s.t.} & \sum_{u \in U} x_{uv} \leq p_v T, && \forall v \in V, && (1a) \\ & \sum_{v \in V} d_v x_{uv} \leq c_u, && \forall u \in U, && (1b) \\ & x_{uv} \geq 0, && \forall u \in U, v \in V, && (1c) \end{aligned}$$

where x_{uv} are the decision variables and denote the expected number of matches between u and v. Constraints (1a) impose an upper bound on the number of matches for type v, which is determined by the expected number of arrivals of type v during the total time horizon of T. Constraints (1b) are the capacity constraints for offline bins. The objective is to maximize the expected total reward. To facilitate the subsequent algorithm design, we use $LP(\mathbf{d}, T)$ to represent the above LP which allows us to modify the expected values of demand and the total time horizon of T. In the following lemma, we show that the optimal value of LP (1) is an upper bound of OPT. This lemma can be proved following a similar idea as in the proof of Theorem 1 in Alaei et al. [2013]. But due to the page limits, we defer all the proofs in this paper to the appendix.

Lemma 2.1. *The optimal value of LP (1) is an upper bound of the offline optimal* OPT.

Note that the expected demand d_vs are unknown to us, hence LP (1) can not be computed accurately. We need to invoke Lemma 2.2 to estimate the mean of an unknown distribution, based on which we will show how to estimate this LP later in Section 4.

Lemma 2.2 (Estimation of Mean, [Hoeffding, 1994])**.** *Suppose $X_1, \ldots, X_t$ are independent and identically distributed random variables with the expectation μ taking values in $[0,1]$. Denote $\bar{X}$ as the average of $X_1, \ldots, X_t$. For any $\delta > 0$, $Pr[\bar{X} \geq (1+\delta)\mu] < e^{-2\delta^2\mu^2 t}$.*

This lemma can be shown using Hoeffding's inequality by setting the range of the support to $[0,1]$.

3 Modified GMP

In this section, similar to Alaei et al. [2013], we first consider a subroutine of our algorithm, which is a modified version of the generalized magician problem.

Definition 3.1 (modified Generalized Magician Problem (modified GMP))**.** The modified GMP is defined as follows. Given a planning horizon of T and a bin of capacity k, the goal is to establish an allocation policy for the online arriving items. For each time $t \in [T]$, an item arrives whose demand Y_t is unknown but follows a fixed distribution D taking values in $[0,1]$. The distribution D is unknown to us. But we can estimate a distribution $\tilde{D}$ supported in $[0,1]$ satisfying the following conditions: (1) $F_{\tilde{D}}(x) \leq F_D(x)$ for all x; (2) $T \cdot \mathbb{E}[X] \leq k$, where X follows $\tilde{D}$. After each arrival, we need to decide whether to pack the item into a bin or reject it. The item can only be accepted to a bin if the bin has at least one unit of capacity remaining. The realization y_t of Y_t is only revealed after the decision is made, and if the item is accepted, the bin's capacity will be reduced by y_t.

In contrast to the GMP used in Alaei et al. [2013], our modified GMP does not assume any knowledge about the actual demand distribution D. As a result, the previous algorithm and its performance guarantee cannot be directly applied to our model. Therefore, we need to develop a new procedure to effectively address this problem. We first introduce an additional notation here. If a specific procedure ensures that every item can be allocated with a probability of at least γ, we call this a γ-conservative magician or the conservative ratio of this procedure is γ. We next introduce how to find a γ-conservative magician for our modified GMP.

Inspired by Alaei et al. [2013], we design the following procedure called *modified γ-magician*, where γ is a fixed parameter in $[0, 1]$. For each time $t \in [T]$, denote W_t and $\tilde{W}_t$ as the amount of used capacity before the arrival of item t if the demand of all arrivals follows the actual distribution D and the imaginary distribution $\tilde{D}$, respectively. We then choose the smallest threshold θ_t such that $\Pr[\tilde{W}_t \leq \theta_t] \geq \gamma$. Here, the probability is considered in the ex-ante sense, meaning that it does not depend on the realization of previous arrivals. Let I_t be the indicator random variable which takes value 1 if we pack item t into the bin when the demand follows D. We have:

$$\Pr[I_t = 1 | W_t] = \begin{cases} 1 & \text{if } W_t < \theta_t, \\ r_t & \text{if } W_t = \theta_t, \\ 0 & \text{if } W_t > \theta_t. \end{cases}$$

Here, $r_t := \frac{\gamma - \Pr[\tilde{W}_t < \theta_t]}{\Pr[\tilde{W}_t = \theta_t]}$. We can define $\tilde{I}_t$ conditioned on $\tilde{W}_t$ for the case that the demand follows $\tilde{D}$ in a similar manner. Based on the definition of W_t and $\tilde{W}_t$, we can compute W_{t+1} and $\tilde{W}_{t+1}$ by applying the update rule $W_{t+1} = W_t + I_t D_t$ and $\tilde{W}_{t+1} = \tilde{W}_t + \tilde{I}_t \tilde{D}_t$, respectively.

Analysis. We next analyze the performance guarantee of this procedure. We first consider the case when the demand follows the imaginary distribution $\tilde{D}$. Note that the imaginary distribution $\tilde{D}$ is known to the decision maker. Furthermore, both the threshold θ_t and r_t are determined by $\tilde{D}$. Hence we can analyze the GMP under the imaginary distribution $\tilde{D}$ using the same tools established in Alaei et al. [2013]. From Theorem 2 and Theorem 6 in Alaei et al. [2013] (for arbitrary distribution), Theorem 4 and Theorem 17 in Alaei [2014] (for Bernoulli distribution), we have the following result.

Claim 3.2. *For any $\gamma \leq 1 - \frac{1}{\sqrt{k}}$ ($\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$ when the demand distribution is Bernoulli distribution), we have Pr$[\tilde{I}_t] = \gamma$ and $\theta_t \leq k - 1$ for each $t \in [T]$.*

We then get the conservative ratio of this procedure in the following claim. Note that $\theta_t \leq k - 1$ for all $t \in [T]$, hence we can always pack item t into the bin successfully. As a result, $\Pr[I_t]$ is exactly the allocated probability of the t-th item. Thus, we have the following claim.

Claim 3.3. *The conservative ratio of modified γ-magician is $\min_{t \in [T]}$ Pr$[W_t < \theta_t] + r_t \cdot$Pr$[W_t = \theta_t]$.*

Next, we show in Theorem 3.4 that if both distributions D and $\tilde{D}$ are Bernoulli distributions whose value is either 0 or 1, the conservative ratio of the modified γ-magician is at least γ.

Theorem 3.4. *If both the actual demand distribution D and the imaginary demand distribution $\tilde{D}$ are Bernoulli distributions, our procedure* modified γ-magician *is a γ-conservative magician of the modified GMP for $\gamma \leq 1 - \frac{1}{\sqrt{k+3}}$.*

We sketch the proof as follows. First, we need to show that the CDF $F_{\tilde{W}_t}$ of $\tilde{W}_t$ and the CDF F_{W_t} of W_t satisfy $F_{\tilde{W}_t}(x) \leq F_{W_t}(x)$ for all x and $t \in [T]$. For Bernoulli distributions, this can be separated into two claims: (1) $F_{\tilde{W}_{t+1}}(x) \leq F_{W_{t+1}}(x)$ for all $x < \theta_t$; (2) $\Pr_{\tilde{W}_{t+1}}[x \leq \theta_t] \leq \Pr_{W_{t+1}}[x \leq \theta_t]$ for each $t \in [T-1]$. Both can be shown by induction. Second, we need to compare the term $\gamma = \Pr[\tilde{I}_t] = \Pr[\tilde{W}_t < \theta_t] + r_t \cdot \Pr[\tilde{W}_t = \theta_t]$ and the term $\Pr[W_t < \theta_t] + r_t \cdot \Pr[W_t = \theta_t]$ for each $t \in [T]$. The details are referred to the appendix as mentioned earlier.

4 Main algorithm

In this section, we formally solve the original problem by using the modified γ-magician procedure. Algorithm 1 presents our algorithm.

```
Algorithm 1: GAP Algorithm
Input: Online arrivals of items
Output: A packing of online items in offline bins
Parameter: Size of sampling phase $T_0 \leq T$, ratio parameter $\gamma$, and error parameter $\epsilon > 0$

 1: for each time $t \in [T]$ do
 2:     item $t$ of type $v_t$ arrives
 3:     if $t \leq T_0$ then
 4:         {Sampling phase}
 5:         reject this item, observe the demand $d'_t$ of item $t$
 6:     end if
 7:     if $t = T_0$ then
 8:         {Estimation}
 9:         for each $v \in V$ do
10:             calculate $\bar{d}_v$ as the average of all previous observed $d'_t$ with $v_t = v$
11:             use $\tilde{d}_v = \bar{d}_v + \epsilon$ as the estimate of $d_v$
12:         end for
13:         solve $LP(\tilde{\boldsymbol{d}}, T')$ where $T' = T - T_0$, and get the solution $\tilde{\boldsymbol{x}}$
14:         for each $v \in V$ do
15:             calculate the empirical distribution $F_v$ according to all previous arrivals whose type is $v$
16:             define a CDF function $\tilde{F}_v$ s.t. $\tilde{F}_v(x) = \max\{0, F_v(x) - \epsilon\}$ for $x < 1$ and $\tilde{F}_v(x) = 1$ for
                $x \geq 1$, and use it as the estimate of the CDF of $D_v$
17:         end for
18:         for each $u \in U$ do
19:             define a CDF function $\tilde{F}_u$ s.t. $\tilde{F}_u(x) = (1 - \sum_{v \in V} \frac{\tilde{x}_{uv}}{T'}) + \sum_{v \in V} \frac{\tilde{x}_{uv}}{T'} \tilde{F}_v(x)$ for
                $x \in [0, 1]$, and use it as the CDF $F_{\tilde{D}}$ in the modified GMP for bin $u$
20:         end for
21:     end if
22:     if $t > T_0$ then
23:         {Magician phase}
24:         choose a bin $u \in U$ with probability $\frac{\tilde{x}_{uv_t}}{p_{v_t} T'}$
25:         decide whether packing item $t$ into bin $u$ according to the modified $\gamma$-magician procedure
            for bin $u$, observe the demand $d'_t$ of item $t$
26:         update $\tilde{W}_t$ in the modified $\gamma$-magician procedure for all bins
27:     end if
28: end for
```

In our algorithm, we employ the idea of exploration-exploitation. Initially, we refrain from allocating the first few items and instead utilize them solely to obtain estimates of the expected value and CDF of the demand distributions. Next, we approach each bin as a modified GMP problem, utilizing the estimates and the modified γ-magician procedure to decide the options for subsequent arrivals. Specifically, Steps 3-6 perform the sampling phase, where we directly reject all arrivals and only record the demand realization at Step 5. We use T_0 to denote the size of the sampling phase. After the sampling phase, we calculate our estimates in Steps 9-20. Steps 9-12 estimate the expected value of the demand distribution. In particular, Step 11 is derived from the modification of the CDF function at Step 16, where the updated $\tilde{d}_v$ remains an upper bound of the expected value of the distribution $\tilde{F}_v$, ensuring the assumption $T \cdot \mathbb{E}[X] \leq k$ made in the modified GMP holds for all bins, as stated in Constraints (1b) of our LP. Steps 14- 17 estimate the CDF function of the demand distribution. The modification of the CDF function at Step 16 is to ensure the assumption $F_{\tilde{D}}(x) \leq F_D(x)$ made in the modified GMP holds. Finally, Steps 18-20 initialize the CDF $F_{\tilde{D}}$ of the modified GMP for each bin and Steps 22-27 make the decision according to the corresponding modified GMP.

Analysis. We now analyze the performance guarantee of our algorithm. We first derive the following three lemmas to get the error bound of the estimates. We fix an error parameter $\epsilon > 0$ and a phase parameter $\alpha = \frac{T_0}{T}$, where T_0 denotes the size of the sampling phase. Denote N as $\min_{v \in V} p_v T$.

Lemma 4.1. *With a probability of at least* $1 - me^{-\frac{\alpha N}{8}}$*, the number of arrivals of items of each type is at least* $\frac{\alpha N}{2}$*.*

We then define E as the event that the number of arrivals for items of each type is at least $\frac{\alpha N}{2}$.

Lemma 4.2. *Conditioning on E, for any* $\delta > 0$*, with a probability of at least* $1 - me^{-\delta^2 \underline{d}^2(\alpha N)}$*, we have* $\tilde{d}_v \leq (1 + \delta + \epsilon/\underline{d}) d_v$ *for all* $v \in V$*.*

Lemma 4.3. *Conditioning on E, with a probability of at least* $1 - 2me^{-\epsilon^2(\alpha N)}$*, we have* $\tilde{F}_v(x) \leq F_v(x)$ *for all* x *and* $v \in V$*. Here,* $F_v(x)$ *is the exact CDF of the demand distribution* D_v*.*

We can apply the union bound and Lemma A.2 to prove Lemma 4.1. Lemmas 4.2 and 4.3 can be proved by making slight adjustments to Lemmas 2.2 and A.1. Here, the descriptions of Lemmas A.1 and A.2 are deferred to the appendix, which are some direct results from Dvoretzky et al. [1956], Mitzenmacher and Upfal [2017].

Next we compare the optimal values of the original LP and $LP(\tilde{\boldsymbol{d}}, T')$ by adjusting the optimal solution to the original LP. Here, we first disregard the probability used in Lemmas 4.2 and 4.3 and assume the conclusions to be valid.

Lemma 4.4. *The optimal value* OPT' *of* $LP(\tilde{\boldsymbol{d}}, T')$ *is at least* $\frac{1-\alpha}{1+\delta+\epsilon/\underline{d}}\mathrm{OPT}$.

Note that the optimal value of $LP(\tilde{\boldsymbol{d}}, T')$ is $\sum_{u \in U} \sum_{v \in V} r_{uv} \tilde{x}_{uv}$. If we have a γ'-conservative magician for each bin, then we get the expected total reward of our algorithm to be $\gamma' \sum_{v \in V} T' p_v \cdot \frac{\tilde{x}_{uv_t}}{p_{v_t} T'} r_{uv} = \gamma' \mathrm{OPT}'$, where the term $T' p_v$ represents the expected arrivals of items of type v and the term $\frac{\tilde{x}_{uv_t}}{p_{v_t} T'}$ is the choosing probability. Hence, it suffices to show that the allocation problem for each bin u can be reduced to a modified GMP.

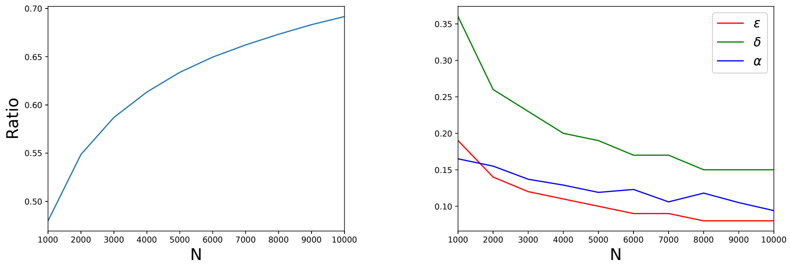
Lemma 4.5. *For each bin u, defining* D *and* $\tilde{D}$ *as described in Algorithm 1, it reduces to a modified GMP.*

By combining these lemmas, considering their probabilities, and further applying Theorem 3.4, we can conclude the following theorem.

Theorem 4.6. *Fix phase parameter* $\alpha = \frac{T_0}{T} \in (0, 1)$*, error parameters* $\epsilon > 0$ *and* $\delta > 0$*. Denote* N *as* $\min_{v \in V} p_v T$ *and* k *as* $\min_{u \in U} c_u$*. If all demand distributions are Bernoulli distributions, with a probability of at least* $1 - me^{-\frac{\alpha N}{8}} - me^{-\delta^2 \underline{d}^2(\alpha N)} - 2me^{-\epsilon^2(\alpha N)}$*, Algorithm 1 can achieve a competitive ratio of* $\frac{1-\alpha}{1+\delta+\epsilon/\underline{d}}(1 - \frac{1}{\sqrt{k+3}})$*.*

We first observe that the competitive ratio exhibits an increasing trend with the minimum capacity k, which aligns with the pattern observed for the ratio derived by Alaei [2014] under known Bernoulli demand distributions. We next analyze the change of the optimal competitive ratio and the related parameters when N changes. Here, we fix $\underline{d} = 0.5$ and assume α can choose an arbitrary value in $(0, 1)$. This assumption on α is considered mild, especially when T is not small (e.g., at least 100), since the difference between two consecutive choices of α is sufficiently small. Hence it makes a negligible impact even if we assume α can take any value between 0 and 1.

To compare the competitive ratio with that derived by Alaei [2014] under known Bernoulli demand distributions, we first calculate the optimal parameters that maximize the ratio between our competitive ratio and theirs. The ratio is given by $\frac{1-\alpha}{1+\delta+\epsilon/\underline{d}}$ according to Theorem 4.6. It is worth noting that under the calculated optimal parameters, the probability characterized in Theorem 4.6 is always at least 0.9. In other words, the ratio holds with a high probability. We plot the results in Figure 1, where Figure 1a provides the optimal ratio between two competitive ratios and Figure 1b presents the corresponding choices of parameters ϵ, δ and α when N varies from 1000 to 10000. According to Figure 1a, our algorithm can achieve approximately 47% of the competitive ratio obtained under known demand distribution even if N is only 1000 and this ratio increases with N. When N increases to 10000, we can achieve a competitive ratio that is almost 70% of the competitive ratio derived under known demand distribution. Figure 1b provides a guideline in choosing parameters. Specifically, it suggests that when N is small, we should increase our error tolerance ϵ and allow more explorations (a larger α) to remain good performance.



(a) Optimal ratio under $\underline{d} = 0.5$

(b) Optimal parameters under $\underline{d} = 0.5$

Figure 1: Illustration of Theorem 4.6.

5 Experiments

In this section, we perform numerical experiments[1] to compare our GAP algorithm with a baseline algorithm, which is a greedy algorithm, using synthetic datasets with various distributions. Specifically, we test Bernoulli, uniform, and truncated normal distributions for the demand to showcase the effectiveness and efficiency of our algorithm.

Data description. We first introduce how to build an instance I of our problem. We assume the total number of bins is $n = 10$, the total number of online types is $m = 10$, and the capacity of each bin is set to k. The rewards, denoted by r_{uv}, between bin u and item type v are generated from a uniform distribution $\mathcal{U}[10, 20]$. To determine the arrival probabilities, we start by generating a number η_v for each type v from a uniform distribution $\mathcal{U}[0, 1]$. We then normalize these values to obtain the arrival probabilities p_v for each type. Specifically, p_v is calculated as $p_v = \frac{\eta_v}{\sum_{v \in V} \eta_v}$. For the demand distributions, we consider three types: Bernoulli $\mathcal{B}(q)$, uniform $\mathcal{U}[a, b]$ and truncated normal $\mathcal{T}(a, b, \mu, \sigma^2)$. The specific definitions and parameters for these distributions can be found in the appendix. When testing a particular type of distribution, all online types share the same demand distribution, but the parameters can vary. The parameter(s) of a given online type's demand distribution are randomly chosen (refer to the appendix for more details). Then given an instance I and the number of online arrivals T, we generate a realization s that includes a sequence of online items and their realized demand according to the instance I.

Algorithms. We test our GAP algorithm with $\gamma = 1 - 1/\sqrt{k}$, 0.5 or 0.8, respectively. Note that $1 - 1/\sqrt{k}$ is the upper bound established in Claim 3.2 for general distributions. The other two tested values can be larger than this bound. We choose these parameters to demonstrate that our algorithm maintains good performance even when γ is beyond the bound. The baseline algorithm, which is a greedy algorithm denoted by GRD is designed as follows. For every arrival i of type v, allocate it to the safe offline bin u with the highest reward r_{uv}, where "safe" refers to an offline bin that has at least one unit of capacity available. In this section, we fix the size of the sampling phase to $T_0 = 100$. We only present the results for the case where $\epsilon = 0$, while the results for other values of ϵ are provided in the appendix. Detailed descriptions of the runtime can be also found in the appendix.

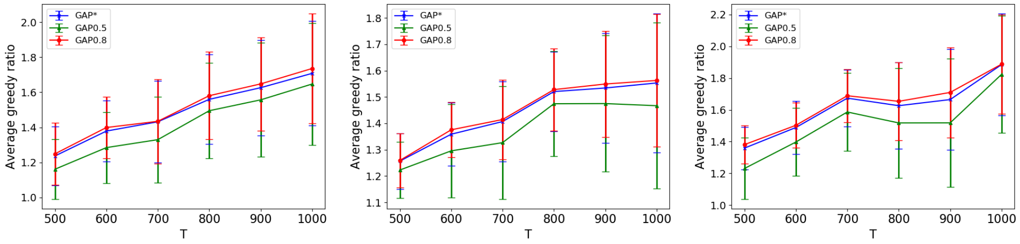
Metric. Because of the NP-hardness of the offline GAP (see Section 2.3 of Mehta et al. [2013]), we use GRD as our benchmark rather than the offline GAP solution. For a realization s, we compute the ratio between the reward achieved by each tested algorithm and the reward obtained by the GRD algorithm. We refer to this ratio as the "greedy ratio". Given k and T, we conduct experiments using $N_I = 20$ different instances, and for each instance, we test $N_s = 5$ different realizations. We present the average greedy ratio along with the standard error across all realizations. We focus on reporting the average greedy ratio rather than the average rewards of the tested algorithms. This is because our main interest lies in the relative gap between our algorithms and the GRD benchmark, and the ratio allows for a meaningful comparison that is independent of the scale of the rewards. We refer the interested reader to the appendix for the comparison results in terms of average rewards.

[1] The experiment is performed by Think Book 14 G2 ITL, with processer: 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz 2.80 GHz.

Results and discussions. The results of our numerical study are shown in Figures 2 and 3. In general, our algorithms outperform GRD as the greedy ratio is larger than 1, except when $k \leq 4, T = 500$ for the uniform distribution and $k \leq 3, T = 500$ for the truncated normal distribution. The standard error is tolerable. We note that GAP0.8 has the best performance among all cases, despite the fact that the value of γ may violate the condition in Theorem 3.4. This implies that in practical simulations, a more aggressive packing strategy from our heuristics can yield better performance.

In Figure 2, we fix the initial capacity of each offline bin $k = 10$, and vary the number of online arrivals T from 500 to 1000. We can see that our heuristics always outperform GRD and the greedy ratios are increasing as T goes larger. To see it, note that T_0 is fixed at 100 in our heuristics. Consequently, when T is large, the sampling part becomes relatively small (i.e., phase parameter $\alpha = \frac{T_0}{T}$ is small but αN remains fixed). As a result, the competitive ratio of our algorithm improves, as indicated by the competitive ratio derived in Theorem 4.6. In contrast, GRD is not affected by the value of T, resulting in increasing greedy ratios over time.

In Figure 3, we keep the number of online arrivals fixed at $T = 500$ and test for different values of $k = 2, 3, \cdots, 10$. In the case of the Bernoulli distribution, the greedy ratios of our heuristics decrease as k increases. This can be attributed to the fact that the reward of GRD increases with k, while the rewards of our heuristics remain relatively stable. Notably, the greedy ratio of GAP0.8 is consistently greater than 1 and decreases as k increases.

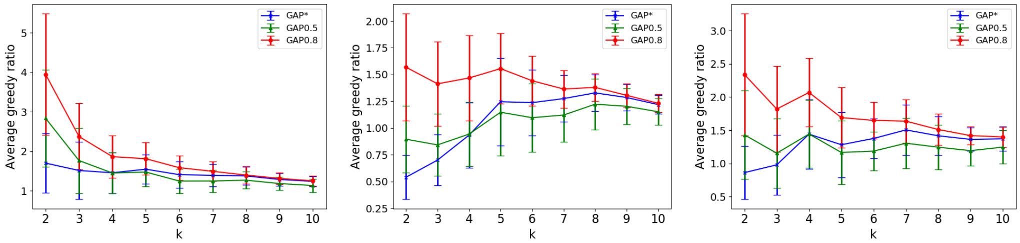


(a) Bernoulli distribution

(b) Uniform distribution

(c) Truncated normal distribution

Figure 2: $k = 10, T = 500, 600, \cdots, 1000$.



(a) Bernoulli distribution

(b) Uniform distribution

(c) Truncated normal distribution

Figure 3: $T = 500$, $k = 2, 3, \cdots, 10$.

To summarize, we test three heuristics based on Algorithm 1 and compare them with the greedy benchmark over three different distributions. Our heuristics can outperform the greedy algorithm in most cases and GAP0.8 provides the best performance among all tested cases.

6 Conclusions

In this paper, we investigate the online GAP with known i.i.d. arrivals and unknown type-specific demand distributions. We consider a scenario where the realization of demand is revealed only after decisions are made. We propose an algorithm that utilizes exploration-exploitation techniques to learn the demand distributions and guide the allocation decision. Theoretically, we present a non-asymptotic parametric guarantee when the demand distributions are Bernoulli distributions. Numerically, we test three heuristics based on our algorithm and show the effectiveness of our heuristics among different demand distributions.

References

Saeed Alaei, MohammadTaghi Hajiaghayi, and Vahid Liaghat. The online stochastic generalized assignment problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques: 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings*, pages 11–25. Springer, 2013.

Saeed Alaei. Bayesian combinatorial auctions: Expanding single buyer mechanisms to many buyers. *SIAM Journal on Computing*, 43(2):930–972, 2014.

Jiashuo Jiang, Will Ma, and Jiawei Zhang. Tight Guarantees for Multi-unit Prophet Inequalities and Online Stochastic Knapsack, June 2022. URL `http://arxiv.org/abs/2107.02058`. arXiv:2107.02058 [cs].

Clifford Stein, Van-Anh Truong, and Xinshang Wang. Advance service reservations with heterogeneous customers. *Management Science*, 66(7):2929–2950, 2020.

Hao Wang, Zhenzhen Yan, and Xiaohui Bei. A nonasymptotic analysis for re-solving heuristic in online matching. *Production and Operations Management*, 31(8):3096–3124, 2022.

Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes with applications to Bayesian selection problems. *SIAM Journal on Computing*, 50(2):255–300, 2021.

David Naori and Danny Raz. Online multidimensional packing problems in the random-order model. *arXiv preprint arXiv:1907.00605*, 2019.

Jon Feldman, Nitish Korula, Vahab Mirrokni, Shanmugavelayutham Muthukrishnan, and Martin Pál. Online ad assignment with free disposal. In *Internet and Network Economics: 5th International Workshop, WINE 2009, Rome, Italy, December 14-18, 2009. Proceedings 5*, pages 374–385. Springer, 2009.

Zhiyi Huang, Xinkai Shu, and Shuyi Yan. The power of multiple choices in online stochastic matching. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 91–103, 2022.

Patrick Jaillet and Xin Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2014.

Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1253–1264. SIAM, 2011.

Susanne Albers, Arindam Khan, and Leon Ladewig. Improved online algorithms for knapsack and gap in the random order model. *arXiv preprint arXiv:2012.00497*, 2020.

Thomas Kesselheim, Klaus Radke, Andreas Tonnis, and Berthold Vocking. Primal beats dual on online packing lps in the random-order model. *SIAM Journal on Computing*, 47(5):1939–1964, 2018.

Anand Bhalgat. A (2+ e)-approximation algorithm for the stochastic knapsack problem. *Unpublished manuscript*, 2011.

Pan Xu. Exploring the tradeoff between competitive ratio and variance in online-matching markets. *arXiv preprint arXiv:2209.07580*, 2022.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *The collected works of Wassily Hoeffding*, pages 409–426, 1994.

Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.

Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

Aranyak Mehta et al. Online matching and ad allocation. *Foundations and Trends® in Theoretical Computer Science*, 8(4):265–368, 2013.