Online Bin Covering with Frequency Predictions

Magnus Berg
University of Southern Denmark, Odense, Denmark

Shahin Kamali
York University, Toronto, Canada

Abstract

We study the bin covering problem where a multiset of items from a fixed set $S \subseteq (0,1]$ must be split into disjoint subsets while maximizing the number of subsets whose contents sum to at least 1. We focus on the online discrete variant, where S is finite, and items arrive sequentially. In the purely online setting, we show that the competitive ratios of best deterministic (and randomized) algorithms converge to $\frac{1}{2}$ for large S, similar to the continuous setting. Therefore, we consider the problem under the prediction setting, where algorithms may access a vector of frequencies predicting the frequency of items of each size in the instance. In this setting, we introduce a family of online algorithms that perform near-optimally when the predictions are correct. Further, we introduce a second family of more robust algorithms that presents a tradeoff between the performance guarantees when the predictions are perfect and when predictions are adversarial. Finally, we consider a stochastic setting where items are drawn independently from any fixed but unknown distribution of S. Using results from the PAC-learnability of probabilities in discrete distributions, we introduce a purely online algorithm whose average-case performance is near-optimal with high probability for all finite sets S and all distributions of S.

2012 ACM Subject Classification Theory of computation → Packing and covering problems; Theory of computation → Online learning algorithms; Theory of computation → Online algorithms

Keywords and phrases Bin Covering, Online Algorithms with Predictions, PAC Learning, Learning-Augmented Algorithms

Digital Object Identifier 10.4230/LIPIcs.SWAT.2024.10

Related Version *Full Version*: https://doi.org/10.48550/arXiv.2401.14881

Funding *Magnus Berg*: Supported in part by the Independent Research Fund Denmark, Natural Sciences, grant DFF-0135-00018B and in part by the Innovation Fund Denmark, grant 9142-00001B, Digital Research Centre Denmark, project P40: Online Algorithms with Predictions.
Shahin Kamali: Supported in part by Natural Sciences and Engineering Research Council of Canada (NSERC) [funding reference number DGECR-2018-00059].

1 Introduction

Bin Covering is a classical NP-complete [5] optimization problem where the input is a multiset of items, each with a size between 0 and 1. The objective is to split the items into disjoint subsets, called *bins*, while maximizing the number of bins whose contents sum to at least 1 [22]. The problem is often considered a dual to the bin packing problem, which asks for minimizing the number of bins, subject to each bin having a sum of at most 1.

In the online setting [18, 14, 5], items arrive one by one, and whenever an item arrives, an algorithm has to irrevocably place the item in an existing bin or open a new bin to place the item in. The existing results mostly consider a continuous setting in which items take any real value from $(0,1]$, and it is well known that a simple greedy strategy, *Dual-Next-Fit* (DNF), achieves an optimal competitive ratio of $\frac{1}{2}$ [5].

In this paper, we consider a discrete variant of Online Bin Covering, where item sizes belong to a finite, known set $S \subseteq (0,1]$. We abbreviate this problem by DBC_S. The special case when $S = \{\frac{i}{k} \mid i = 1, \ldots, k\}$ has been studied in the previous work. For example, Csirik, Johnson, and Kenyon [15] developed online algorithms with good average-case performance based on the *Sum of Squares* algorithm for Online Discrete Bin Packing [17, 16]. In this paper, we study a more general setting where S may be *any* finite subset of $(0, 1]$.

For measuring and comparing the quality of online algorithms for the DBC_S problem, we rely on the classical *competitive analysis* framework [9, 23], where one measures the quality of an online algorithm by comparing the performance of the algorithm to the performance of an optimal offline algorithm optimizing for the best worst-case guarantee.

1.1 Previous Work

The possibilities for creating algorithms for Online Bin Covering are well-studied. In the continuous setting, where items can take any size in $(0, 1]$, Assmann et al. [5] proved that DNF is $\frac{1}{2}$-competitive, and Csirik and Totik [18] presented an impossibility result showing that this is best possible. Later, Epstein [20] proved that the same impossibility result holds for randomized algorithms as well. Online Bin Covering has been studied under the advice setting [10, 12], where algorithms can access an advice tape that has encoded information about the input sequence. The aim is to determine how much additional information, measured by the number of bits needed to encode the information, is necessary and sufficient to achieve a certain competitive ratio and how well algorithms can perform when they are given a certain amount of information. For example, it is known that $\Theta(\log \log n)$ bits of advice are necessary and sufficient to achieve algorithms with a competitive ratio strictly better than $\frac{1}{2}$ [10], and that $O(b + \log(n))$ bits is sufficient to create an asymptotically $\frac{2}{3}$-competitive algorithm [12], where b is the number of bits needed to encode a rational value.

In recent years, developments in machine learning have inspired questions about how online algorithms may benefit from machine-learned advice [24, 25], commonly referred to as *predictions*. Unlike the advice model, the predictions may be erroneous or even adversarial. Online algorithms with predictions is a rapidly growing field (see, e.g., [1]) that aims at deriving online algorithms that provide a tradeoff between *consistency* and *robustness*. The consistency of an online algorithm refers to its competitive ratio when predictions are error-free; ideally, the consistency of an algorithm is 1 or close to 1. On the other hand, robustness refers to the competitive ratio assuming adversarial predictions; ideally, the robustness of an algorithm is close to the competitive ratio of the best purely online algorithm (with no prediction). These ideal cases, however, are not always realizable simultaneously, and one often settle for a consistency/robustness trade-off [25, 2, 27, 11, 3], giving explicit bounds on an algorithm's consistency as a function of its robustness, and vice versa.

To the authors' knowledge, no previous work on Bin Covering with predictions exists. The related Bin Packing problem, however, is previously studied under the prediction setting [4, 2].

1.2 Contribution

Our contributions for DBC_S can be summarized as follows. Throughout, we let $k = |S|$. In the continuous setting, where items take *any* real value in $(0, 1]$, no improvements in the competitive ratio can be achieved via predictions that are of size independent of input length, even if the predictions are error-free. This follows from a result of [10] that states any algorithm with an advice of size $o(\log \log n)$ is no better than $\frac{1}{2}$-competitive. Due to this negative result, we relax the problem and assume items come from a fixed, finite set. This relaxed setting is also studied for the related bin packing problem [4].

Purely online setting. We establish the following result on purely online algorithms for DBC_{F_k}, where $F_k = \{\frac{i}{k} \mid i = 1, 2, \ldots, k\}$, based on ideas from [18] and [20] (all missing proofs can be found in the full paper [8]).

▶ **Theorem 1.** *Let* ALG *be any deterministic or randomized online algorithm for* DBC_{F_k}*, with* $k \geqslant 5$*. Then,* ALG*'s competitive ratio is at most* $\frac{1}{2} + \frac{1}{H_{k-1}}$*, where* $H_{k-1} = \sum_{i=1}^{k-1} \frac{1}{i}$*.*

A consequence of Theorem 1 is the well-known fact [18, 20] that the competitive ratio of any deterministic or randomized algorithm for Online Bin Covering is at most $\frac{1}{2}$. This shows that Online Bin Covering is still a hard problem, even after discretization.

Prediction setting. We study DBC_S, where predictions concerning the frequency of item sizes are available. We start with an impossibility result that establishes a consistency/robustness tradeoff for this prediction scheme (Theorem 2). We then present an online algorithm, named *Group Covering*, which is near-optimal when the predictions are error-free, for all finite sets $S \subseteq (0, 1]$ (Theorem 5). Further, we create a family of hybrid algorithms that accepts a parameter λ, quantifying one's trust in the predictions. We establish a consistency/robustness tradeoff that bounds the consistency and robustness of these hybrid algorithms as a function of λ (Theorems 9 and 10).

Stochastic setting. Motivated by the work of Csirik, Johnson, and Kenyon [15], we study the purely online problem under a stochastic setting, where item sizes follow an unknown distribution. Unlike [15], which assumes items are of sizes $\frac{i}{k}$, for $i = 1, 2, \ldots, k$, we do not make any assumption about input set S. We use a PAC-learning bound [13, 26] to create a family of online algorithms without predictions, whose expected performance ratio [15] is near-optimal with high probability, for any finite set S, and any unknown distribution D of S (Theorem 12).

2 Preliminaries

2.1 Online Discrete Bin Covering

Fix a finite set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$. An instance for *S-Discrete Bin Covering* is a sequence $\sigma = \langle a_1, a_2, \ldots, a_n \rangle$ of items, where $a_i \in S$, for $i \in [n]$. The task of an algorithm ALG is to place the items in σ into bins $B_1, B_2, \ldots, B_t$, maximizing the number of bins, B, for which $\sum_{a \in B} a \geqslant 1$. For any bin, B, we call $\text{lev}(B) = \sum_{a' \in B} a'$ the *level* of B. We assume that algorithms are aware of S. In the online setting, the items are presented one-by-one to ALG, and upon receiving an item a, ALG has to place a in a bin. This decision is irrevocable. We abbreviate *Online S-Discrete Bin Covering* by DBC_S. Throughout, we assume that $k \geqslant 2$, and we set $F_k = \left\{\frac{i}{k} \mid \text{for } i = 1, 2, \ldots, k\right\}$, and abbreviate DBC_{F_k} by DBC_k.

2.2 Performance Measures

Given an online maximization problem, Π, an online algorithm, ALG, for Π, and an instance, σ, of Π, we let $\text{ALG}[\sigma]$ be ALG's solution on instance σ and $\text{ALG}(\sigma)$ be the profit of $\text{ALG}[\sigma]$. If ALG is deterministic, then the *competitive ratio* of ALG is

$$\text{CR}_{\text{ALG}} = \sup\{c \in (0, 1] \mid \exists b > 0 \colon \forall \sigma \colon \text{ALG}(\sigma) \geqslant c \cdot \text{OPT}(\sigma) - b\},$$

where OPT is an offline optimal algorithm for Π. Further, ALG is *c-competitive* if $c \leqslant \text{CR}_{\text{ALG}}$.

For a fixed finite set $S = \{s_1, s_2, \dots, s_k\} \subseteq (0, 1]$, and a fixed (unknown) distribution D of S, the *asymptotic expected ratio* [19, 15] of an online algorithm, ALG, is

$$\text{ER}^{\infty}_{\text{ALG}}(D) = \liminf_{n\to\infty} \mathbb{E}_D \left[\frac{\text{ALG}(\sigma_n(D))}{\text{OPT}(\sigma_n(D))}\right], \tag{1}$$

where $\sigma_n(D)$ is a sequence of n independent identically distributed random variables, $\sigma_n(D) = \langle X_1, X_2, \dots, X_n\rangle$[1], where $X_i \sim D$, for all $i = 1, 2, \dots, n$.

When an algorithm, ALG, has access to predictions, the *consistency* of ALG, and the *robustness* of ALG, is ALG's competitive ratio when the predictions are error-free and adversarial, respectively. Throughout, we let $[n] = \{1, 2, \dots, n\}$.

3 Predictions Setting

In this section, we assume that algorithms are given a *frequency prediction*, which, for a fixed instance σ, and each item $s_i \in S$, predicts what fraction of items in σ are of size s_i.

Formally, given a finite set $S = \{s_1, s_2, \dots, s_k\} \subseteq (0, 1]$, and an instance, σ, of DBC_S, we let n_i^σ be the number of items of size s_i in σ, n^σ be the number of items in σ, and $f_i^\sigma = \frac{n_i^\sigma}{n^\sigma}$. We call f_i^σ the *frequency* of items of size s_i in σ, and set $\boldsymbol{f^\sigma} = (f_1^\sigma, f_2^\sigma, \dots, f_k^\sigma)$. When there can be no confusion, we abbreviate n_i^σ, n^σ, f_i^σ, and $\boldsymbol{f^\sigma}$, by n_i, n, f_i, and $\boldsymbol{f}$, respectively.

Throughout, we abbreviate *Online S-Discrete Bin Covering with Frequency Predictions* by $\text{DBC}_S^{\mathcal{F}}$. An instance for $\text{DBC}_S^{\mathcal{F}}$ is a tuple $(\sigma, \hat{\boldsymbol{f}})$ consisting of a sequence of items, σ, and a vector of predicted frequencies $\hat{\boldsymbol{f}} = \left(\hat{f}_1, \hat{f}_2, \dots, \hat{f}_k\right)$.

It is well-known that probabilities in discrete distributions are PAC-learnable, as shown in [13]. That is, there exists a polynomial-time algorithm that learns the probabilities in discrete distributions to arbitrary precision with a confidence that is arbitrarily close to 1, given sufficiently many random samples (see [26] for a formal definition of PAC-learnability). This makes frequency predictions easily attainable when historical data is available.

3.1 A Consistency-Robustness Trade-Off for $\text{DBC}_k^{\mathcal{F}}$

In the following, by a *wasteful* algorithm, we mean an algorithm that sometimes places an item, a, in a bin, B, for which $\text{lev}(B) \geqslant 1$ before a was placed in B. Any wasteful algorithm can be trivially converted to an equally good (possibly better) algorithm that avoids placing items into already-covered bins. Therefore, in what follows, we assume that all algorithms, including OPT, are non-wasteful.

▶ **Theorem 2.** *Any $(1-\alpha)$-consistent deterministic algorithm for $\text{DBC}_k^{\mathcal{F}}$ is at most 2α-robust.*

Proof. Let ALG be any deterministic online algorithm for $\text{DBC}_k^{\mathcal{F}}$. Consider the instance $(\sigma_1^n, \hat{\boldsymbol{f}})$, with $\hat{\boldsymbol{f}} = (\hat{f}_1, \hat{f}_2, \dots, \hat{f}_k)$, where

$$\sigma_1^n = \left\langle \left\langle \frac{k-1}{k} \right\rangle^n, \left\langle \frac{1}{k} \right\rangle^n \right\rangle \quad \text{and} \quad \hat{f}_i = \begin{cases} \frac{1}{2}, & \text{if } s_i \in \left\{\frac{1}{k}, \frac{k-1}{k}\right\} \\ 0, & \text{otherwise.} \end{cases}$$

Clearly, $\hat{\boldsymbol{f}}$ is a perfect prediction for σ_1^n, and $\text{OPT}(\sigma_1^n) = n$. Hence, by the consistency of ALG, there exists a constant b, such that

$$\text{ALG}(\sigma_1^n, \hat{\boldsymbol{f}}) \geqslant (1-\alpha) \cdot \text{OPT}(\sigma_1^n) - b = (1-\alpha) \cdot n - b. \tag{2}$$

[1] The particular choice of notation for X_i's is due to the items being random variables.

Let $\mathcal{B}_i$, for $i = 1, 2$, be the collection of bins that ALG places i items of size $\frac{k-1}{k}$ in. Then, $\text{ALG}(\sigma_1^n, \hat{\boldsymbol{f}}) \leqslant |\mathcal{B}_1| + |\mathcal{B}_2| + \frac{n-|\mathcal{B}_1|}{k}$. Since ALG is non-wasteful, $n = |\mathcal{B}_1| + 2 \cdot |\mathcal{B}_2|$, and so, by Equation (2), we have that $(1-\alpha) \cdot (|\mathcal{B}_1| + 2 \cdot |\mathcal{B}_2|) - b \leqslant |\mathcal{B}_1| + \frac{(k+2)\cdot|\mathcal{B}_2|}{k}$, which implies

$$\frac{n \cdot \left(1 - 2 \cdot \alpha - \frac{2}{k}\right) - 2 \cdot b}{1 - \frac{2}{k}} \leqslant |\mathcal{B}_1| \,. \tag{3}$$

Hence, since ALG is $(1-\alpha)$-consistent, it has created at least $\frac{n\cdot\left(1-2\cdot\alpha-\frac{2}{k}\right)-2\cdot b}{1-\frac{2}{k}}$ bins that contain exactly one item of size $\frac{k-1}{k}$ after processing the first n items.

Next, consider the instance $(\sigma_2^n, \hat{\boldsymbol{f}})$, with imperfect predictions, where $\sigma_2^n = \left\langle \frac{k-1}{k} \right\rangle^n$. Since the first n requests of σ_1^n and σ_2^n are identical, ALG cannot distinguish the instances $(\sigma_1^n, \hat{\boldsymbol{f}})$ and $(\sigma_2^n, \hat{\boldsymbol{f}})$ until it has seen the first n items. Hence, since ALG is deterministic, it distributes the first n items identically on the two instances. Given that $n = |\mathcal{B}_1| + 2 \cdot |\mathcal{B}_2|$, Equation (3) implies that

$$\text{ALG}(\sigma_2^n, \hat{\boldsymbol{f}}) \leqslant |\mathcal{B}_2| = \frac{n - |\mathcal{B}_1|}{2} \leqslant \frac{1}{2} \cdot \left(n - \frac{n\left(1 - 2 \cdot \alpha - \frac{2}{k}\right) - 2 \cdot b}{1 - \frac{2}{k}} \right) = \frac{2 \cdot n \cdot \alpha + 2 \cdot b}{2 - \frac{4}{k}}.$$

Since $\text{OPT}(\sigma_2^n) = \frac{n}{2}$, then, for all $n \in \mathbb{Z}^+$, $\frac{\text{ALG}(\sigma_2^n, \hat{\boldsymbol{f}})}{\text{OPT}(\sigma_2^n)} \leqslant \frac{\frac{2\cdot n\cdot\alpha+2\cdot b}{2-\frac{4}{k}}}{\frac{n}{2}} = \frac{4\cdot n\cdot\alpha+4\cdot b}{n\cdot\left(2-\frac{4}{k}\right)} \leqslant 2 \cdot \alpha - \frac{2\cdot b}{n}$, and thus ALG is at most $2 \cdot \alpha$-robust. ◀

Note that the impossibility result of Theorem 2 holds even for the special case of $S = F_k$. In fact, since we only use items from $\{\frac{1}{k}, \frac{k-1}{k}\}$ in input sequences of the proof, Theorem 2 can be stated for all finite sets $S \subseteq (0, 1]$, for which $\{\frac{1}{k}, \frac{k-1}{k}\} \subseteq S$.

3.2 A Near-Optimally Consistent Algorithm for $\text{DBC}_S^{\mathcal{F}}$

In this section, inspired by the *Profile Packing* algorithm from [4], we present a family of algorithms named *Group Covering*, parameterized by a parameter, ε, that receives frequency predictions, and outputs a $(1-\varepsilon)$-approximation of the optimal solution, assuming predictions are error-free. In other words, the algorithm achieves a consistency that is arbitrarily close to optimal. For a fixed $\varepsilon > 0$, we let GC_ε be the Group Covering algorithm with parameter ε.

The Strategy of Group Covering

Fix a finite set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$. A *non-wasteful bin type* is an ordered l-tuple $(a_1, a_2, \ldots, a_l)$ of items, with $l \geqslant 1$ and $a_i \in S$, for all $i \in [l]$, such that a_1 was placed in the bin first, then a_2, and so on, and such that $\sum_{i=1}^{l-1} a_i < 1$. Observe that this definition implies an ordering of the items in bin types, which is essential for our purpose. For example, the bin type $(1/2, 1/2, \varepsilon)$ is wasteful, as the bin is already covered after placing the second item of size $1/2$, but the bin type $(1/2, \varepsilon, 1/2)$ is non-wasteful, as removing the top item will make the bin no longer covered. Note that non-covered bins are also constitute a non-wasteful bin type. We let $\mathcal{T}_S$ denote the collection of all possible non-wasteful bin types given S, and set $\tau_S = |\mathcal{T}_S|$ and $t_{\max} = \max_{t\in\mathcal{T}_S}\{|t|\}$. For example, if $S = \left\{\frac{1}{k}, \frac{k-1}{k}\right\}$ then,

$$\mathcal{T}_S = \left\{ \left(\underbrace{\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}}_{i \text{ times}} \right) \mid i \in [k] \right\} \cup \left\{ \left(\underbrace{\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}}_{i \text{ times}}, \frac{k-1}{k} \right) \mid i \in [k-1] \right\} \cup \left\{ \left(\frac{k-1}{k}\right), \left(\frac{k-1}{k}, \frac{1}{k}\right), \left(\frac{k-1}{k}, \frac{k-1}{k}\right) \right\},$$

$\tau_S = 2k + 2$, and $t_{\max} = k$.

Given an instance of $\text{DBC}_S^{\mathcal{F}}$, $(\sigma, \hat{\boldsymbol{f}})$, GC_ε works as follows. In its initialization phase (before any item is placed), it creates an optimal solution to the following multiset, σ_{sub}, created based on $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$ (which it knows) and the frequency prediction:

$$\sigma_{\text{sub}} = \langle \lfloor \hat{f}_1 \cdot m_{k,\varepsilon} \rfloor, \lfloor \hat{f}_2 \cdot m_{k,\varepsilon} \rfloor, \ldots, \lfloor \hat{f}_k \cdot m_{k,\varepsilon} \rfloor \rangle,$$

where $m_{k,\varepsilon} = m_\varepsilon + k$, and $m_\varepsilon = \lceil 3 \cdot \tau_S \cdot t_{\max} \cdot \varepsilon^{-1} \rceil$. In this optimal solution, we maintain a *placeholder* of size a for any item $a \in \sigma_{\text{sub}}$. A placeholder of size a is a virtual item of size a, which reserves space for an item of size a. We let $P_{\hat{\boldsymbol{f}},\varepsilon}$ be the copy of $\text{OPT}[\sigma_{\text{sub}}]$ containing placeholders. To finish the initialization, GC_ε opens the first *group*, $G^1_{\hat{\boldsymbol{f}},\varepsilon}$; a copy of $P_{\hat{\boldsymbol{f}},\varepsilon}$.

When an item, a, arrives, GC_ε searches for a placeholder of size a in the open groups, searching in $G^1_{\hat{\boldsymbol{f}},\varepsilon}$ first, then $G^2_{\hat{\boldsymbol{f}},\varepsilon}$ second, and so on. If such a placeholder exists, GC_ε replaces the placeholder with a. If no such placeholder exists, GC_ε checks whether $P_{\hat{\boldsymbol{f}},\varepsilon}$ contains such a placeholder, by checking whether $a \in \sigma_{\text{sub}}$. If so, then GC_ε opens a new group, $G^i_{\hat{\boldsymbol{f}},\varepsilon}$, i.e. a new copy of $P_{\hat{\boldsymbol{f}},\varepsilon}$, and it replaces a newly created placeholder with a. Otherwise, GC_ε places a in an *extra*-bin using DNF. Extra bins are reserved for items that GC_ε did not expect to receive any of (items whose predicted frequency is 0 and thus are not in σ_{sub}). Pseudocode for GC_ε are given in Algorithm 1.

Analysis of GC_ε

We say that a group, $G^i_{\hat{\boldsymbol{f}},\varepsilon}$, is *completed* if all its placeholders have been replaced by items, and let g_ε be the number of groups that GC_ε completes. Recall that, by construction, GC_ε first completes $G^1_{\hat{\boldsymbol{f}},\varepsilon}$, then $G^2_{\hat{\boldsymbol{f}},\varepsilon}$, and so on.

▶ **Lemma 3.** *Fix any finite set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$, any $\varepsilon \in (0, 1)$, and any instance $(\sigma, \hat{\boldsymbol{f}})$ for $DBC_S^{\mathcal{F}}$, with $\hat{\boldsymbol{f}} = \boldsymbol{f}$. Then, $\left\lfloor \frac{n}{m_{k,\varepsilon}} \right\rfloor \leqslant g_\varepsilon \leqslant \left\lfloor \frac{n}{m_\varepsilon} \right\rfloor$.*

Throughout, we let $\mathbf{p}(N)$ be the profit of a solution N for an input σ. Observe that $\mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right) = \mathbf{p}\left(G^i_{\hat{\boldsymbol{f}},\varepsilon}\right)$, for all $i \in [g_\varepsilon]$, i.e. all completed groups have the same profit.

▶ **Lemma 4.** *Fix any set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$, any $\varepsilon \in (0, 1)$, and any instance, $(\sigma, \hat{\boldsymbol{f}})$, for $DBC_S^{\mathcal{F}}$, with $\hat{\boldsymbol{f}} = \boldsymbol{f}$ and $n^\sigma > m^2_{k,\varepsilon} + m_{k,\varepsilon}$. Then, $g_\varepsilon \cdot \mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant (1 - \varepsilon) \cdot \text{OPT}(\sigma)$.*

Proof. We show this by creating a solution, N, based on $\text{OPT}[\sigma]$, such that

(i) $\mathbf{p}(N) \geqslant \left(1 - \frac{\varepsilon}{3}\right) \cdot \text{OPT}(\sigma)$, and

(ii) $g_\varepsilon \cdot \mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant \left(1 - \frac{2\cdot\varepsilon}{3}\right) \cdot \mathbf{p}(N)$.

Since $\varepsilon \in (0, 1)$, it suffices to prove (i) and (ii), because (i) and (ii) imply that

$$g_\varepsilon \cdot \mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant \left(1 - \frac{2 \cdot \varepsilon}{3}\right) \cdot \left(1 - \frac{\varepsilon}{3}\right) \cdot \text{OPT}(\sigma) \geqslant (1 - \varepsilon) \cdot \text{OPT}(\sigma).$$

Construction of N. Initially, let N be a copy of $\text{OPT}[\sigma]$. Since OPT is non-wasteful, all bins in $\text{OPT}[\sigma]$ are filled according to non-wasteful bin types. For each non-wasteful bin type $t \in \mathcal{T}_S$, remove between 0 and $g_\varepsilon - 1$ bins of type t from N, such that the number of bins of type t becomes divisible by g_ε.

Proof of (i). Since $\text{OPT}(\sigma) \geqslant \frac{n^\sigma}{t_{\max}}$, Lemma 3 implies that

$$\mathbf{p}(N) \geqslant \text{OPT}(\sigma) - (g_\varepsilon - 1) \cdot \tau_S \geqslant \text{OPT}(\sigma) - \frac{n^\sigma}{m_\varepsilon} \cdot \tau_S$$
$$\geqslant \text{OPT}(\sigma) - \text{OPT}(\sigma) \cdot \frac{\tau_S \cdot t_{\max}}{m_\varepsilon} \geqslant \left(1 - \frac{\varepsilon}{3}\right) \cdot \text{OPT}(\sigma).$$

Algorithm 1 GC_ε.

Input: a $\mathrm{DBC}_S^{\mathcal{F}}$-instance. $(\sigma, \hat{\boldsymbol{f}})$
$j, l \leftarrow 1$
Compute τ_S, $t_{\max}$, and $k = |S|$
$m_\varepsilon \leftarrow \lceil 3 \cdot \tau_S \cdot t_{\max} \cdot \varepsilon^{-1} \rceil$
$m_{k,\varepsilon} \leftarrow m_\varepsilon + k$
$\sigma_{\text{sub}} \leftarrow \langle \lfloor \hat{f}_1 \cdot m_{k,\varepsilon} \rfloor, \lfloor \hat{f}_2 \cdot m_{k,\varepsilon} \rfloor, \dots, \lfloor \hat{f}_k \cdot m_{k,\varepsilon} \rfloor \rangle$
$P_{\hat{\boldsymbol{f}},\varepsilon} \leftarrow \emptyset$
for all $B \in \mathrm{OPT}[\sigma_{\text{sub}}]$ **do**
  $B' \leftarrow \emptyset$ ▷ Create a new empty bin
  for all $a \in B$ **do**
    $B' \leftarrow B' \cup \{p_a\}$ ▷ Add a placeholder of size a to B'
  $P_{\hat{\boldsymbol{f}},\varepsilon} \leftarrow P_{\hat{\boldsymbol{f}},\varepsilon} \cup B'$ ▷ Add a copy of B containing placeholders to $P_{\hat{\boldsymbol{f}},\varepsilon}$
$G^1_{\hat{\boldsymbol{f}},\varepsilon} \leftarrow P_{\hat{\boldsymbol{f}},\varepsilon}$ ▷ Open the first group
while receiving items, a, **do**
  not_placed $\leftarrow$ `true` ▷ Marks whether a still has to be placed
  for $i = 1, 2, \dots, l$ **do** ▷ Go through open groups chronologically
    if not_placed **then** ▷ To avoid trying to place a multiple times
      if $\exists B \in G^i_{\hat{\boldsymbol{f}},\varepsilon} : p_a \in B$ **then** ▷ Search for p_a in $G^i_{\hat{\boldsymbol{f}},\varepsilon}$
        $B \leftarrow B \setminus \{p_a\} \cup \{a\}$ ▷ Swap out placeholder, p_a, for a
        not_placed $\leftarrow$ `false` ▷ a has been placed in a bin
  if not_placed **then** ▷ Checking whether a has been placed
    if $\lfloor \hat{f}_a \cdot m_{k,\varepsilon} \rfloor \neq 0$ **then** ▷ Checking whether $a \in \sigma_{\text{sub}}$
      $l \leftarrow l + 1$
      $G^l_{\hat{\boldsymbol{f}},\varepsilon} \leftarrow \mathrm{OPT}[\sigma_{\text{sub}}]$ ▷ Open a new group
      Determine $B \in G^l_{\hat{\boldsymbol{f}},\varepsilon}$ such that $p_a \in B$, and $B \leftarrow B \setminus \{p_a\} \cup \{a\}$
    else ▷ $a \notin \sigma_{\text{sub}}$
      $B^E_j \leftarrow B^E_j \cup \{a\}$ ▷ Place a in a *extra* bin using DNF
      if $\mathrm{lev}(B^E_j) \geqslant 1$ **then**
        $j \leftarrow j + 1$
        $B^E_j \leftarrow \emptyset$

Proof of (ii). Since the number of occurrences of each bin type in N is divisible by g_ε, we may consider N as g_ε identical copies of a smaller covering $\overline{N}$. Since we do not add any items when creating N, and thus $\overline{N}$, we have $n_i^{\overline{N}} \leqslant \left\lfloor \frac{n_i^\sigma}{g_\varepsilon} \right\rfloor$, for all $i \in [k]$, where $n_i^{\overline{N}}$ denotes the number of items of size i in $\overline{N}$. Then, for all $i \in [k]$, we can write

$$n_i^{\overline{N}} \leqslant \left\lfloor \frac{n_i^\sigma}{g_\varepsilon} \right\rfloor \leqslant \left\lfloor \frac{n_i^\sigma}{\left\lfloor \frac{n^\sigma}{m_{k,\varepsilon}} \right\rfloor} \right\rfloor \leqslant \left\lfloor \frac{n_i^\sigma}{\frac{n^\sigma}{m_{k,\varepsilon}} - 1} \right\rfloor = \left\lfloor \frac{n_i^\sigma}{\frac{n^\sigma - m_{k,\varepsilon}}{m_{k,\varepsilon}}} \right\rfloor = \left\lfloor n_i^\sigma \cdot \frac{m_{k,\varepsilon}}{n^\sigma - m_{k,\varepsilon}} \right\rfloor .$$

Given that $\frac{m_{k,\varepsilon}}{n^\sigma - m_{k,\varepsilon}} = \frac{m_{k,\varepsilon}}{n^\sigma} + \frac{m_{k,\varepsilon}^2}{n^\sigma \cdot (n^\sigma - m_{k,\varepsilon})}$, and that $n^\sigma > m_{k,\varepsilon}^2 + m_{k,\varepsilon}$, we may conclude

$$n_i^{\overline{N}} \leqslant \left\lfloor \frac{n_i^\sigma \cdot m_{k,\varepsilon}}{n^\sigma} + \frac{m_{k,\varepsilon}^2}{n^\sigma - m_{k,\varepsilon}} \right\rfloor \leqslant \left\lfloor \frac{n_i^\sigma \cdot m_{k,\varepsilon}}{n^\sigma} \right\rfloor + 1 = \lfloor f_i \cdot m_{k,\varepsilon} \rfloor + 1.$$

Hence, $\overline{N}$ contains at most one more item of size s_i than $G^j_{\hat{\boldsymbol{f}},\varepsilon}$, for all $i \in [k]$, and all $j \in [g_\varepsilon]$. Then, for all $j \in [g_\varepsilon]$, the following holds:

$$\mathbf{p}\left(G^j_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant \mathbf{p}\left(\overline{N}\right) - k. \tag{4}$$

Next, we devise a lower bound for $\mathbf{p}\left(\overline{N}\right)$. Since $\textsc{Opt}(\sigma) \geqslant \frac{n^\sigma}{t_{\max}}$,

$$\mathbf{p}\left(\overline{N}\right) = \frac{\mathbf{p}(N)}{g_\varepsilon} \geqslant \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot \textsc{Opt}(\sigma)}{g_\varepsilon} \geqslant \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot n^\sigma}{t_{\max}\cdot g_\varepsilon} \geqslant \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot n^\sigma}{t_{\max}\cdot \frac{n^\sigma}{m_\varepsilon}}$$
$$= \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot m_\varepsilon}{t_{\max}} \geqslant \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot \frac{3\cdot\tau_S\cdot t_{\max}}{\varepsilon}}{t_{\max}} \geqslant \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot 3\cdot\tau_S}{\varepsilon} \geqslant \frac{\left(1-\frac{\varepsilon}{3}\right)\cdot k}{\frac{\varepsilon}{3}}.$$

Hence, $k \leqslant \frac{\frac{\varepsilon}{3}\cdot\mathbf{p}\left(\overline{N}\right)}{1-\frac{\varepsilon}{3}}$, and so, by Equation (4), $\mathbf{p}\left(G^j_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant \mathbf{p}\left(\overline{N}\right) - \frac{\frac{\varepsilon}{3}\cdot\mathbf{p}\left(\overline{N}\right)}{1-\frac{\varepsilon}{3}} \geqslant \left(1-\frac{2\cdot\varepsilon}{3}\right)\cdot\mathbf{p}\left(\overline{N}\right)$. Since $\mathbf{p}(N) = g_\varepsilon \cdot \mathbf{p}\left(\overline{N}\right)$ and $\mathbf{p}\left(G^j_{\hat{\boldsymbol{f}},\varepsilon}\right) = \mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right)$, for all $j \in [g_\varepsilon]$, we conclude $g_\varepsilon \cdot \mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant g_\varepsilon \cdot \left(1-\frac{2\cdot\varepsilon}{3}\right)\cdot\mathbf{p}\left(\overline{N}\right) = \left(1-\frac{2\cdot\varepsilon}{3}\right)\cdot\mathbf{p}(N)$, which establishes (ii). ◀

Given Lemma 4, it is straightforward to deduce the following theorem, which is the main result of this section.

► **Theorem 5.** *For any set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0,1]$, and any $\varepsilon \in (0,1)$, there exists a constant, b, such that for all instances $(\sigma, \hat{\boldsymbol{f}})$, with $\boldsymbol{f} = \hat{\boldsymbol{f}}$, it holds that $GC_\varepsilon(\sigma, \hat{\boldsymbol{f}}) \geqslant (1-\varepsilon)\cdot \textsc{Opt}(\sigma) - b$. That is, GC_ε is a $(1-\varepsilon)$-consistent algorithm for $DBC_S^{\mathcal{F}}$.*

While the above theorem shows that GC_ε is almost optimally consistent, the same cannot be said about its robustness. Consider the instance $(\sigma^n, \hat{\boldsymbol{f}})$ where $\sigma^n = \left\langle \frac{1}{k}\right\rangle^n$ and $\hat{\boldsymbol{f}}$ predicts that half of the items are of size $\frac{1}{k}$, and half of the items are of size $\frac{k-1}{k}$, a wrong prediction for σ^n. Based on the predictions $\hat{\boldsymbol{f}}$, GC_ε creates $\left\lfloor \frac{m_{k,\varepsilon}}{2}\right\rfloor$ bins that contain placeholders for one item of size $\frac{1}{k}$, and one item of size $\frac{k-1}{k}$. Since no item of size $\frac{k-1}{k}$ appears in the input, GC_ε never covers a bin, and since $\textsc{Opt}(\sigma^n) = \left\lfloor \frac{n}{k}\right\rfloor$, GC_ε is not robust. In the next section, we introduce a strategy for improving the robustness of GC_ε.

3.3 Robustifying GC_ε

For each purely online algorithm, $\textsc{Alg}$ (e.g. DNF), we create a family of *hybrid algorithms* that combines GC_ε with $\textsc{Alg}$ to improve the robustness of GC_ε. Formally, for any algorithm, $\textsc{Alg}$, we create the family $\{\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}\}_{\lambda,\varepsilon}$, of hybrid algorithms, parametrized by $\varepsilon \in (0,1)$ and a *trust level*, $\lambda \in \mathbb{Q}^+$. Throughout, we assume that λ is given as a fraction, $\lambda = \frac{\kappa}{\ell}$, for some $\kappa \in \mathbb{N}$ and $\ell \in \mathbb{Z}^+$. For any item $a \in S$, $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ maintains a counter for the number of items of size a in the input observed so far. Upon receiving an item a, $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ counts the number of occurrences of a, denoted c_a, and if $c_a \pmod{\ell} \leqslant \ell - \kappa - 1$, it uses $\textsc{Alg}$ to place a in a bin that only $\textsc{Alg}$ places items into, and otherwise, it uses GC_ε to place a in a bin that only GC_ε places items into. The pseudo-code for $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ is given in Algorithm 2.

For the analysis of $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$, we associate, to any instance σ of DBC_S, a $(\ell+1)$-tuple, $(\sigma_1, \sigma_2, \ldots, \sigma_\ell, \sigma_e)$ called the *ℓ-splitting of σ*, which is created as follows. Process the items one-by-one, in the order they appear in σ; when processing an item a, place it in σ_{i+1} if $c_a \pmod{\ell} \equiv i$, where c_a is the number of items of size a previously recorded. After processing all items in σ, we compute the number of items of size s_i, for any $s_i \in S$, in each σ_j, for all $i \in [k]$ and all $j \in [\ell]$. If there are equally many items of size s_i in all σ_j, we are done. If, on the other hand, there exists some $i \in [k]$ and some $j \in [\ell]$ such that $\sigma_1, \sigma_2, \ldots, \sigma_j$ contains one more item of size s_i than $\sigma_{j+1}, \sigma_{j+2}, \ldots, \sigma_\ell$, then we remove one item of size s_i from all of $\sigma_1, \sigma_2, \ldots, \sigma_j$, and place it in σ_e instead. The pseudo-code for this process is given in the full paper [8].

Algorithm 2 $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$.

Input: An instance for $\text{DBC}^{\mathcal{F}}_S$, $(\sigma, \hat{\boldsymbol{f}})$
Determine $\kappa, \ell \in \mathbb{Z}^+$ such that $\lambda = \frac{\kappa}{\ell}$
Run Lines 2-13 of GC_ε (see Algorithm 1), given the prediction $\hat{\boldsymbol{f}}$
Run initialization part of $\textsc{Alg}$, if such exists
for all $i \in [k]$ **do**
  $c_{s_i} \leftarrow 0$
while receiving items, a, **do**
  $j \leftarrow c_a \pmod{\ell}$ ▷ $a \in \sigma_{j+1}$
  if $j \leqslant \ell - \kappa - 1$ **then**
    Ask $\textsc{Alg}$ to place a
  else ▷ $\ell - \kappa \leqslant j \leqslant \ell - 1$
    Ask GC_ε to place a ▷ See Lines 14-30 in Algorithm 1
  $c_a \leftarrow c_a + 1$

By construction, the ℓ-splitting of σ decomposes σ into ℓ smaller instances, σ_i for $i \in [\ell]$, that all contain the same multiset of items, but possibly in different orders, and an *excess* instance σ_e, which contain the remaining items from σ. By construction, $n^{\sigma_e} \leqslant (\ell - 1) \cdot k$.

Bounding the Performance of the Optimal Packing

In what follows, we present an upper bound for the number of bins covered by $\textsc{Opt}$. Throughout, given ℓ instances, $\sigma_1, \sigma_2, \ldots, \sigma_\ell$, we set $\bigcup_{i=1}^{\ell} \sigma_i = \langle \sigma_1, \sigma_2, \ldots, \sigma_\ell \rangle$.

▶ **Observation 6.** *Let $\sigma_1, \sigma_2, \ldots, \sigma_\ell$ be any instances for DBC_S, then $\sum_{i=1}^{\ell} \textsc{Opt}(\sigma_i) \leqslant \textsc{Opt}\left(\bigcup_{i=1}^{\ell} \sigma_i\right)$.*

▶ **Lemma 7.** *Let $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$ be any finite set, let σ by any instance of DBC_S, and let $(\sigma_1, \sigma_2, \ldots, \sigma_\ell, \sigma_e)$ be the ℓ-splitting of σ. Then, $\textsc{Opt}(\sigma) \leqslant \sum_{i=1}^{\ell} \textsc{Opt}(\sigma_i) + (\ell - 1) \cdot (k + \tau_S)$.*

Proof. We split this proof into two parts, by showing that

(i) $\textsc{Opt}\left(\bigcup_{i=1}^{\ell} \sigma_i\right) \leqslant \sum_{i=1}^{\ell} \textsc{Opt}(\sigma_i) + (\ell - 1) \cdot \tau_S$, and

(ii) $\textsc{Opt}(\sigma) \leqslant \textsc{Opt}\left(\bigcup_{i=1}^{\ell} \sigma_i\right) + (\ell - 1) \cdot k$.

Proof of (i). We use a similar strategy as in the proof of Theorem 5. To this end, let N be the solution obtained by removing at most $\ell - 1$ bins of each non-wasteful bin type from a copy of $\textsc{Opt}\left[\bigcup_{i=1}^{\ell} \sigma_i\right]$ (recall that $\textsc{Opt}$ is non-wasteful) such that the number of each bin type in N is divisible by ℓ. Then, $\mathbf{p}(N) \geqslant \textsc{Opt}\left(\bigcup_{i=1}^{\ell} \sigma_i\right) - (\ell - 1) \cdot \tau_S$. Therefore, it suffices to compare the profit of $\bigcup_{i=1}^{\ell} \textsc{Opt}[\sigma_i]$ to $\mathbf{p}(N)$. Since $\sigma_1, \sigma_2, \ldots, \sigma_\ell$ all contain the same multiset of items (but possibly in a different order), it holds that $\textsc{Opt}(\sigma_i) = \textsc{Opt}(\sigma_j)$, for all $i, j \in [\ell]$. Further, by construction, N is the union of ℓ identical smaller coverings, $\overline{N}$, for which $n_i^{\overline{N}} \leqslant n_i^{\sigma_i}$, for all $i \in [k]$. Therefore, $\textsc{Opt}(\sigma_i) \geqslant \mathbf{p}\left(\overline{N}\right)$, for all $i \in [k]$, and we can write $\sum_{i=1}^{\ell} \textsc{Opt}(\sigma_i) = \ell \cdot \textsc{Opt}(\sigma_1) \geqslant \ell \cdot \mathbf{p}\left(\overline{N}\right) = \mathbf{p}(N)$, which completes the proof of (i).

Proof of (ii). Since $n^{\sigma_e} \leqslant (\ell - 1) \cdot k$, we can write $\textsc{Opt}\left(\bigcup_{i=1}^{\ell} \sigma_i\right) \geqslant \textsc{Opt}(\sigma) - (\ell - 1) \cdot k$. Adding $(\ell - 1) \cdot k$ to both sides establishes (ii) and thus completes the proof. ◀

A Bound on the Performance of GC_ε

We compare the number of bins covered by GC_ε on a subset of the instances in the ℓ-splitting of an instance, σ, to that of $\textsc{Opt}$ on σ. To this end, observe that if σ is a DBC_S-instance, where $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0,1]$, and $(\sigma_1, \sigma_2, \ldots, \sigma_\ell, \sigma_e)$ is the ℓ-splitting of σ, then $n_j^{\sigma_i} = \left\lfloor \frac{n_j^\sigma}{\ell} \right\rfloor$, for all $j \in [k]$ and all $i \in [\ell]$.

▶ **Lemma 8.** *Fix any set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0,1]$, any $\varepsilon \in (0,1)$, and any instance $(\sigma, \hat{\boldsymbol{f}})$ of DBC_S, for which $\boldsymbol{f} = \hat{\boldsymbol{f}}$, and let $(\sigma_1, \sigma_2, \ldots, \sigma_\ell, \sigma_e)$ be the ℓ-splitting of σ, for some $\ell \in \mathbb{Z}^+$. Then, for any $j \in \mathbb{Z}^+$, with $j \leqslant \ell$, there exists a constant b such that $GC_\varepsilon\left(\left(\bigcup_{i=\ell-j+1}^{\ell} \sigma_i\right), \hat{\boldsymbol{f}}\right) \geqslant \frac{j \cdot (1-\varepsilon) \cdot \textsc{Opt}(\sigma)}{\ell} - b$.*

Proof. Let $\tilde{\sigma_j} = \bigcup_{i=\ell-j+1}^{\ell} \sigma_i$, and set $b = m_{k,\varepsilon}^2 + m_{k,\varepsilon} + k \cdot \ell$. If $n^\sigma \leqslant b$, the right-hand side is non-positive, and the left-hand side is non-negative, and the lemma's statement follows.

Hence, assume that $n^\sigma > b$. Let $C = \mathrm{GC}_\varepsilon[\sigma, \hat{\boldsymbol{f}}]$, and let g_ε be the number of groups, $G^i_{\hat{\boldsymbol{f}},\varepsilon}$, that GC_ε completes on instance $(\sigma, \hat{\boldsymbol{f}})$. By Lemma 4, we have $g_\varepsilon \cdot \mathbf{p}\left(G^1_{\hat{\boldsymbol{f}},\varepsilon}\right) \geqslant (1-\varepsilon) \cdot \textsc{Opt}(\sigma)$. Since $G^i_{\hat{\boldsymbol{f}},\varepsilon}$ is only dependent on ε, S, and $\hat{\boldsymbol{f}}$, GC_ε creates the same groups, $G^i_{\hat{\boldsymbol{f}},\varepsilon}$, on instance $(\sigma, \hat{\boldsymbol{f}})$ as on instance $(\tilde{\sigma_j}, \hat{\boldsymbol{f}})$. In the following, we prove a lower bound for the number of groups that GC_ε completes on instance $(\tilde{\sigma_j}, \hat{\boldsymbol{f}})$, as a function of g_ε.

Since C completely covers g_ε copies of $G^i_{\hat{\boldsymbol{f}},\varepsilon}$, then $n_i^\sigma \geqslant g_\varepsilon \cdot \lfloor f_i^\sigma \cdot m_{k,\varepsilon} \rfloor$ for all $i \in [k]$. Moreover, given that each σ_i contains exactly $\left\lfloor \frac{n_i^\sigma}{\ell} \right\rfloor$ items of size s_i, we have

$$n_i^{\tilde{\sigma}_j} \geqslant j \cdot \left\lfloor \frac{n_i^\sigma}{\ell} \right\rfloor \geqslant \frac{j \cdot n_i^\sigma}{\ell} - j \geqslant \frac{j \cdot g_\varepsilon}{\ell} \cdot \lfloor f_i^\sigma \cdot m_{k,\varepsilon} \rfloor - j \geqslant \left\lfloor \frac{j \cdot g_\varepsilon}{\ell} \right\rfloor \cdot \lfloor f_i^\sigma \cdot m_{k,\varepsilon} \rfloor - j.$$

This implies that, GC_ε fills in all placeholders for items of size s_i in $\left\lfloor \frac{j \cdot g_\varepsilon}{\ell} \right\rfloor$ groups, except at most j, on instance $(\tilde{\sigma}_j, \hat{\boldsymbol{f}})$, for all $i \in [k]$. Hence,

$$\mathrm{GC}_\varepsilon(\tilde{\sigma}_j, \hat{\boldsymbol{f}}) \geqslant \left\lfloor \frac{j \cdot g_\varepsilon}{\ell} \right\rfloor \cdot \mathbf{p}\left(G^i_{\hat{\boldsymbol{f}},\varepsilon}\right) - k \cdot j \geqslant \left(\frac{j \cdot g_\varepsilon}{\ell} - 1\right) \cdot \mathbf{p}\left(G^i_{\hat{\boldsymbol{f}},\varepsilon}\right) - k \cdot j.$$

Since $\mathbf{p}\left(G^i_{\hat{\boldsymbol{f}},\varepsilon}\right) \leqslant m_{k,\varepsilon}$, we conclude the following, which completes the proof:

$$\mathrm{GC}_\varepsilon(\tilde{\sigma}_j, \hat{\boldsymbol{f}}) \geqslant \frac{j \cdot g_\varepsilon}{\ell} \cdot \mathbf{p}\left(G^i_{\hat{\boldsymbol{f}},\varepsilon}\right) - k \cdot j - m_{k,\varepsilon} \geqslant \frac{j \cdot (1-\varepsilon) \cdot \textsc{Opt}(\sigma)}{\ell} - b.$$ ◀

A Trust-Parametrized Family of Hybrid Algorithms

In what follows, we wrap up the analysis of $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ by stating and proving the main results of this section. By construction, $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ (see Algorithm 2) distributes the items that arrive between GC_ε and $\textsc{Alg}$ in a way determined by λ. Whenever λ becomes close to 1, $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ assigns a larger fraction of items to GC_ε, and when λ gets close to 0, $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ assigns more items to $\textsc{Alg}$. In particular, $\textsc{Hyb}^{1,\varepsilon}_{\textsc{Alg}} = \mathrm{GC}_\varepsilon$, and $\textsc{Hyb}^{0,\varepsilon}_{\textsc{Alg}} = \textsc{Alg}$. Clearly, $\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}$ cannot create a perfect ℓ-splitting online, since it cannot correctly identify the items that are placed in σ_e. It can, however, get sufficiently close.

▶ **Theorem 9.** *For any finite set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0,1]$, any purely online $DBC_S^{\mathcal{F}}$-algorithm, $\textsc{Alg}$, any $c \leqslant \textsc{cr}_{\textsc{Alg}}$, any $\varepsilon \in (0,1)$, and any $\lambda \in \mathbb{Q}^+$, there exists a constant $b \in \mathbb{Z}^+$, such that for all instances $(\sigma, \hat{\boldsymbol{f}})$, the following holds, assuming $\boldsymbol{f} = \hat{\boldsymbol{f}}$:*

$$\textsc{Hyb}^{\lambda,\varepsilon}_{\textsc{Alg}}(\sigma, \hat{\boldsymbol{f}}) \geqslant (\lambda \cdot (1-\varepsilon) + (1-\lambda) \cdot c) \cdot \textsc{Opt}(\sigma) - b.$$

Proof. Let $b_{\textsc{Alg}}$ be the additive constant of $\textsc{Alg}$, $b_{\mathrm{GC}_\varepsilon} = m_{k,\varepsilon}^2 + m_{k,\varepsilon} + k \cdot \ell$. Then, we set $b = b_{\textsc{Alg}} + b_{\mathrm{GC}_\varepsilon} + (\ell - 1) \cdot (k + \tau_S)$. If $n^\sigma \leqslant b$, the result follows trivially. Hence, assume that $n^\sigma > b$.

Let $(\sigma_1, \sigma_2, \ldots, \sigma_\ell, \sigma_e)$ be the ℓ-splitting of σ, and let $\sigma_e^{\textsc{Alg}}$ and $\sigma_e^{\mathrm{GC}_\varepsilon}$ be the collection of instances from σ_e that $\textsc{Alg}$ and GC_ε receive, respectively. Then, by definition of $\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}$,

$$\begin{aligned}\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}[\sigma, \hat{\boldsymbol{f}}] &= \textsc{Alg}\left[\left(\bigcup_{i=1}^{\ell-\kappa} \sigma_i\right) \cup \sigma_e^{\textsc{Alg}}\right] \cup \mathrm{GC}_\varepsilon\left[\left(\bigcup_{i=\ell-\kappa+1}^{\ell} \sigma_i\right) \cup \sigma_e^{\mathrm{GC}_\varepsilon}, \hat{\boldsymbol{f}}\right] \\ &\geqslant \textsc{Alg}\left(\bigcup_{i=1}^{\ell-\kappa} \sigma_i\right) + \mathrm{GC}_\varepsilon\left(\left(\bigcup_{i=\ell-\kappa+1}^{\ell} \sigma_i\right), \hat{\boldsymbol{f}}\right).\end{aligned}$$

Set $b' = b_{\textsc{Alg}} + b_{\mathrm{GC}_\varepsilon}$. Then, by c-competitiveness of $\textsc{Alg}$ and Lemma 8, we can write

$$\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}(\sigma, \hat{\boldsymbol{f}}) \geqslant c \cdot \textsc{Opt}\left(\bigcup_{i=1}^{\ell-\kappa} \sigma_i\right) + \lambda \cdot (1 - \varepsilon) \cdot \textsc{Opt}(\sigma) - b'.$$

Since $\textsc{Opt}(\sigma_i) = \textsc{Opt}(\sigma_j)$ for all $i, j \in [\ell]$ then, by Observation 6, we have $\sum_{i=1}^{\ell-\kappa} \textsc{Opt}(\sigma_i) \leqslant \textsc{Opt}\left(\bigcup_{i=1}^{\ell-\kappa} \sigma_i\right)$. Therefore, from the above inequality, we can conclude

$$\begin{aligned}\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}(\sigma, \hat{\boldsymbol{f}}) &\geqslant c \cdot \left(\sum_{i=1}^{\ell-\kappa} \textsc{Opt}(\sigma_i)\right) + \lambda \cdot (1 - \varepsilon) \cdot \textsc{Opt}(\sigma) - b' \\ &= (1 - \lambda) \cdot c \cdot \left(\sum_{i=1}^{\ell} \textsc{Opt}(\sigma_i)\right) + \lambda \cdot (1 - \varepsilon) \cdot \textsc{Opt}(\sigma) - b'.\end{aligned}$$

Combining Lemma 7 and the above bound for $\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}(\sigma, \hat{\boldsymbol{f}})$, we can conclude the following, which completes the proof:

$$\begin{aligned}\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}(\sigma, \hat{\boldsymbol{f}}) &\geqslant (1 - \lambda) \cdot c \cdot (\textsc{Opt}(\sigma) - (\ell - 1) \cdot (k + \tau_S)) + \lambda \cdot (1 - \varepsilon) \cdot \textsc{Opt}(\sigma) - b' \\ &\geqslant ((1 - \lambda) \cdot c + \lambda \cdot (1 - \varepsilon)) \cdot \textsc{Opt}(\sigma) - b.\end{aligned}$$ ◀

The above theorem gives an explicit formula for the consistency of $\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}$ as a function of the trust-level, λ, $\varepsilon \in (0, 1)$, and the performance guarantee of $\textsc{Alg}$. A similar proof can be used to establish a guarantee on the robustness of $\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}$.

▶ **Theorem 10.** *For any finite set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0, 1]$, any purely online algorithm, $\textsc{Alg}$, for DBC_S, any $c \leqslant \mathrm{CR}_{\textsc{Alg}}$, and any ε, there exists a constant $b \in \mathbb{Z}^+$, such that for all instances $(\sigma, \hat{\boldsymbol{f}})$, $\textsc{Hyb}_{\textsc{Alg}}^{\lambda,\varepsilon}(\sigma, \hat{\boldsymbol{f}}) \geqslant (1 - \lambda) \cdot c \cdot \textsc{Opt}(\sigma) - b$.*

4 Stochastic Setting

In this section, we consider a setting for DBC_S where item sizes are generated independently at random from an unknown distribution. This setting has already been studied for the more restricted DBC_k problem, where Csirik, Johnson and Kenyon used variants of the Bin Packing algorithm "Sum-of-Squares", first introduced in [17, 16], to develop algorithms for DBC_k. Rather than designing algorithms that perform well in the worst case, they aimed to design algorithms that perform well on average. Specifically, they develop an algorithm, called SS^*, with $\mathrm{ER}_{SS^*}^{\infty}(D) = 1$ (see Equation (1) for the definition of $\mathrm{ER}_{SS^*}^{\infty}(D)$), for all discrete distributions D of F_k, with rational probabilities.

In this section, we use a PAC-learning bound for learning frequencies in discrete distributions to derive a family of algorithms called *purely online group covering* ($\{\text{POGC}_\varepsilon^\delta\}_{\varepsilon,\delta}$). These algorithms are parametrized by two real numbers $\varepsilon, \delta \in (0,1)$, satisfying that, for *all* finite sets $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0,1]$, there exists a constant $b \in \mathbb{R}^+$, such that for all (unknown) distributions $D = \{p_1, p_2, \ldots, p_k\}$ of S, allowing irrational probabilities, the following holds:

$$P\left(\text{POGC}_\varepsilon^\delta(\sigma_n(D)) \geqslant (1-\varepsilon) \cdot \text{OPT}(\sigma_n(D)) - b\right) \geqslant 1-\delta, \tag{5}$$

where $\sigma_n(D)$ is defined in the preliminaries. Observe that this guarantee is true, even for adversarial S and D. Clearly, Equation (5) implies that

$$P(\text{ER}^\infty_{\text{POGC}_\varepsilon^\delta}(D) \geqslant 1-\varepsilon) \geqslant 1-\delta. \tag{6}$$

The guarantee from Equation (5) is, however, stronger than Equation (6), in that the additive term in Equation (5) is constant, whereas the additive term for $\text{POGC}_\varepsilon^\delta$ in Equation (6) may be a function of n. As pointed out in [6], having only constant loss before giving a multiplicative performance guarantee is a desirable property.

We formalize the strategy of $\text{POGC}_\varepsilon^\delta$ in Algorithm 3. In words; the algorithm works by defining a "sample size", Φ, as a function of k, ε and δ. Intuitively, observing Φ items from the input prefix is sufficient to make predictions about the frequency of items with respect to D that are ε-accurate with confidence $1-\delta$. We formalize this in Proposition 11. In the process of learning D, $\text{POGC}_\varepsilon^\delta$ places the first Φ items using DNF while observing the item frequencies. After placing the first Φ item, $\text{POGC}_\varepsilon^\delta$ uses the observed frequencies to make an estimate - prediction - about the item frequencies and applies $\text{GC}_{\frac{\varepsilon}{2}}$ to place the remaining items.

Algorithm 3 $\text{POGC}_\varepsilon^\delta$.

Input: A DBC_S-instance, σ
$ss \leftarrow 0$ ▷ Sample size
Compute τ_S, $t_{\max}$, and $k = |S|$
$m_{\frac{\varepsilon}{2}} \leftarrow \lceil 6 \cdot \tau_S \cdot t_{\max} \cdot \varepsilon^{-1} \rceil$
$m_{k,\frac{\varepsilon}{2}} \leftarrow m_{\frac{\varepsilon}{2}} + k$
$\Phi \leftarrow \max\left\{16 \cdot k \cdot (m_{k,\frac{\varepsilon}{2}}+1)^2, 32 \cdot (m_{k,\frac{\varepsilon}{2}}+1)^2 \cdot \ln\left(\frac{2}{1-\sqrt{1-\delta}}\right)\right\}$
for all $i \in [k]$ **do**
    $c_{s_i} \leftarrow 0$ ▷ Number of items of size s_i
while receiving items, a, **and** $ss < \Phi$ **do**
    $c_a \leftarrow c_a + 1$
    Place a in a DNF-marked bin using DNF
    $ss \leftarrow ss + 1$
for $i = 1, 2, \ldots, k$ **do**
    $\hat{f}_i^\Phi = \frac{c_{s_i}}{\Phi}$
$\hat{\boldsymbol{f}}^{\boldsymbol{\Phi}} = \left(\hat{f}_1^\Phi, \hat{f}_2^\Phi, \ldots, \hat{f}_k^\Phi\right)$
Run Lines 2-13 of $\text{GC}_{\frac{\varepsilon}{2}}$ (see Algorithm 1), given the prediction $\hat{\boldsymbol{f}}^{\boldsymbol{\Phi}}$
while receiving items, a, **do**
    Place a using $\text{GC}_{\frac{\varepsilon}{2}}$ ▷ See Lines 14-30 in Algorithm 1

Before formalizing and proving the claim from Equation (5), we review a PAC-learning bound for learning frequencies in discrete distributions [13].

Sampling Complexity of Learning Frequencies

We refer to [13] for a proof of the following well-known fact that establishes an upper bound for the sampling complexity of PAC-learning frequencies:

▶ **Proposition 11** ([13]). *For any finite set $S = \{s_1, s_2, \ldots, s_k\} \subseteq (0,1]$, there exists an algorithm, $\mathcal{A}$, and a map $\Phi_{\mathcal{A}}\colon \mathbb{R}^+ \times (0,1) \to \mathbb{Z}^+$, such that for any $\gamma \in \mathbb{R}^+$, any $\delta \in (0,1)$, any (unknown) discrete distribution $D = \{p_1, p_2, \ldots, p_k\}$ of S, and any $n \geqslant \Phi_{\mathcal{A}}(\gamma,\delta)$, letting $\{X_i\}_{i=1}^n$ be a sequence of independent identically distributed random variables, with $X_i \sim D$,*

$$P\left(L^1(\mathcal{A}(X_1, X_2, \ldots, X_n), D) \leqslant \gamma\right) \geqslant 1-\delta,$$

where L^1 is the usual L^1-distance. For learning frequencies in discrete distributions, $\mathcal{A}$ is the algorithm which outputs the predicted distribution:

$$\mathcal{A}(X_1, X_2, \ldots, X_n) = \left\{ \hat{p}_i \;\middle|\; i \in [k] \text{ and } \hat{p}_i = \frac{1}{n} \cdot \sum_{j=1}^{n} \mathbb{1}_{\{s_i\}}(X_j) \right\},$$

and, for any $\gamma \in \mathbb{R}^+$ and $\delta \in (0,1)$, the map $\Phi_{\mathcal{A}}$ is given by

$$\Phi_{\mathcal{A}}(\gamma,\delta) = \max\left\{ \frac{4 \cdot k}{\gamma^2}, \frac{8}{\gamma^2} \cdot \ln\left(\frac{2}{\delta}\right) \right\}.$$

4.1 Analysis of POGC$_\varepsilon^\delta$

We formalize and prove the claim from Equation (5):

▶ **Theorem 12.** *For all finite sets $S = \{s_1, s_2, \ldots, s_k\} \subset (0,1]$, and all $\varepsilon, \delta \in (0,1)$, there exists a constant $b \in \mathbb{Z}^+$, such that for all discrete distributions $D = \{p_1, p_2, \ldots, p_k\}$ of S, and all $n \in \mathbb{Z}^+$, the following holds:*

$$P\left(POGC_\varepsilon^\delta(\sigma_n(D)) \geqslant (1-\varepsilon) \cdot \textsc{Opt}(\sigma_n(D)) - b \right) \geqslant 1-\delta,$$

where $\sigma_n(D) = \langle X_1, X_2, \ldots, X_n \rangle$, and $\{X_i\}_{i=1}^n$ is a sequence of independent identically distributed random variables with $X_i \sim D$, for all $i \in [n]$.

Proof. Set $\Phi = \max\left\{16 \cdot k \cdot (m_{k,\frac{\varepsilon}{2}}+1)^2, 32 \cdot (m_{k,\frac{\varepsilon}{2}}+1)^2 \cdot \ln\left(\frac{2}{1-\sqrt{1-\delta}}\right)\right\}$, and $b = \max\{2 \cdot \Phi, m_{k,\frac{\varepsilon}{2}}^2 + m_{k,\frac{\varepsilon}{2}} + \Phi\}$, and observe that b is independent of the input length n. By similar arguments as in the proof of Lemma 8, we assume that $n \geqslant b$. For ease of notation, we set $\tilde{\varepsilon} = \frac{\varepsilon}{2}$.

Throughout this proof, we split $\sigma_n(D)$ into two subsequences, σ_a and σ_s. Formally, we set $\sigma_a = \langle X_1, X_2, \ldots, X_\Phi \rangle$, and $\sigma_s = \langle X_{\Phi+1}, X_{\Phi+2}, \ldots, X_n \rangle$. By construction, POGC$_\varepsilon^\delta$ uses DNF on the first Φ items while counting the number of items of each size. After observing the first Φ items, it creates the predicted distribution $\hat{\boldsymbol{f}}^{\boldsymbol{\Phi}} = \mathcal{A}(X_1, X_2, \ldots, X_\Phi)$, by Lines 13-15 in Algorithm 3. By construction of Φ and Proposition 11, we can write

$$P\left(L^1(\hat{\boldsymbol{f}}^{\boldsymbol{\Phi}}, D) \leqslant \frac{1}{2 \cdot (m_{k,\tilde{\varepsilon}}+1)} \right) \geqslant \sqrt{1-\delta}.$$

Therefore, by construction of $\hat{\boldsymbol{f}}^{\boldsymbol{\Phi}}$ and the definition of L^1, the following holds:

$$P\left(\sum_{i=1}^{k} \left| \hat{f}_i^{\Phi} - p_i \right| \leqslant \frac{1}{2 \cdot (m_{k,\tilde{\varepsilon}}+1)} \right) \geqslant \sqrt{1-\delta}.$$

Denote by $\boldsymbol{f^{\sigma_s}}$ the true frequencies of $\sigma_s = \langle X_{\Phi+1}, X_{\Phi+2}, \ldots, X_n \rangle$. Since $n \geqslant 2 \cdot \Phi$, we know that $|\sigma_s| \geqslant \Phi$, and so, by similar arguments as above,

$$P\left(\sum_{i=1}^{k} |f_i^{\sigma_s} - p_i| \leqslant \frac{1}{2 \cdot (m_{k,\tilde{\varepsilon}} + 1)}\right) \geqslant \sqrt{1-\delta}.$$

Let $E_{\boldsymbol{\hat{f}^\Phi}}$ be the event $\sum_{i=1}^{k} \left|\hat{f}_i^\Phi - p_i\right| \leqslant \frac{1}{2 \cdot (m_{k,\tilde{\varepsilon}}+1)}$, and $E_{\boldsymbol{f^{\sigma_s}}}$ be the event $\sum_{i=1}^{k} |f_i^{\sigma_s} - p_i| \leqslant \frac{1}{2 \cdot (m_{k,\tilde{\varepsilon}}+1)}$. Since $E_{\boldsymbol{\hat{f}^\Phi}}$ and $E_{\boldsymbol{f^{\sigma_s}}}$ are independent, we have $P\left(E_{\boldsymbol{\hat{f}^\Phi}} \textbf{ and } E_{\boldsymbol{f^{\sigma_s}}}\right) \geqslant 1 - \delta$ Therefore, with probability at least $1 - \delta$, we have

$$L^1(\boldsymbol{\hat{f}^\Phi}, \boldsymbol{f^{\sigma_s}}) = \sum_{i=1}^{k} \left|\hat{f}_i^\Phi - f_i^{\sigma_s}\right| \leqslant \sum_{i=1}^{k} \left|\hat{f}_i^\Phi - p_i\right| + \sum_{i=1}^{k} |f_i^{\sigma_s} - p_i| < \frac{1}{m_{k,\tilde{\varepsilon}}}. \tag{7}$$

This means that the predictions $\text{POGC}_\varepsilon^\delta$ creates are very close to the true frequencies of the remainder of the instance, σ_s, with high probability.

Next, by construction of $\text{POGC}_\varepsilon^\delta$, we deduce that $\text{POGC}_\varepsilon^\delta(\sigma_n(D)) \geqslant \text{GC}_{\tilde{\varepsilon}}(\sigma_s, \boldsymbol{\hat{f}^\Phi})$. Then, as long as we can verify that the inequality

$$\text{GC}_{\tilde{\varepsilon}}(\sigma_s, \boldsymbol{\hat{f}^\Phi}) \geqslant (1 - \varepsilon) \cdot \text{OPT}(\sigma_s), \tag{8}$$

holds whenever $L^1(\boldsymbol{\hat{f}^\Phi}, \boldsymbol{f^{\sigma_s}}) < \frac{1}{m_{k,\tilde{\varepsilon}}}$, we deduce that

$$\begin{aligned} \text{POGC}_\varepsilon^\delta(\sigma_n(D)) &\geqslant \text{GC}_{\tilde{\varepsilon}}(\sigma_s, \boldsymbol{\hat{f}^\Phi}) \\ &\geqslant (1 - \varepsilon) \cdot \text{OPT}(\sigma_s) \\ &\geqslant (1 - \varepsilon) \cdot \text{OPT}(\sigma_n(D)) - 2 \cdot \Phi. \end{aligned}$$

Since $P(L^1(\boldsymbol{\hat{f}^\Phi}, \boldsymbol{f^{\sigma_s}}) < \frac{1}{m_{k,\tilde{\varepsilon}}}) \geqslant 1 - \delta$, by Equality 7, we can write

$$P\left(\text{POGC}_\varepsilon^\delta(\sigma_n(D)) \geqslant (1 - \varepsilon) \cdot \text{OPT}(\sigma_n(D)) - 2 \cdot \Phi\right) \geqslant 1 - \delta,$$

which completes the proof.

It remains to prove that Equation (8) holds whenever $L^1(\boldsymbol{\hat{f}^\Phi}, \boldsymbol{f^{\sigma_s}}) < \frac{1}{m_{k,\tilde{\varepsilon}}}$. To this end, assume that $L^1(\boldsymbol{\hat{f}^\Phi}, \boldsymbol{f^{\sigma_s}}) < \frac{1}{m_{k,\tilde{\varepsilon}}}$. Let $g_{\tilde{\varepsilon}}$ be the number of groups that $\text{GC}_{\tilde{\varepsilon}}$ would complete on instance $(\sigma_s, \boldsymbol{f^{\sigma_s}})$, that is, with perfect predictions. Moreover, let $P_{\sigma_s,\tilde{\varepsilon}} = \text{OPT}[\langle \lfloor f_1^{\sigma_s} \cdot m_{k,\tilde{\varepsilon}} \rfloor, \ldots, \lfloor f_k^{\sigma_s} \cdot m_{k,\tilde{\varepsilon}} \rfloor \rangle]$, and $P_{\Phi,\tilde{\varepsilon}} = \text{OPT}[\langle \lfloor \hat{f}_1^\Phi \cdot m_{k,\tilde{\varepsilon}} \rfloor, \ldots, \lfloor \hat{f}_k^\Phi \cdot m_{k,\tilde{\varepsilon}} \rfloor \rangle]$, where items have been replaced with placeholders.

First, we compare the number of items of size s_i in $P_{\sigma_s,\tilde{\varepsilon}}$ compared to $P_{\Phi,\tilde{\varepsilon}}$. To this end, for all $i \in [k]$, set $\mu_i = \left|\lfloor \hat{f}_i^\Phi \cdot m_{k,\tilde{\varepsilon}} \rfloor - \lfloor f_i^{\sigma_s} \cdot m_{k,\tilde{\varepsilon}} \rfloor\right|$. Then,

$$\mu_i \leqslant \left|\hat{f}_i^\Phi \cdot m_{k,\tilde{\varepsilon}} - f_i^{\sigma_s} \cdot m_{k,\tilde{\varepsilon}}\right| + 1 = \left|\hat{f}_i^\Phi - f_i^{\sigma_s}\right| \cdot m_{k,\tilde{\varepsilon}} + 1.$$

Since $L^1(\boldsymbol{\hat{f}^\Phi}, \boldsymbol{f^{\sigma_s}}) < \frac{1}{m_{k,\tilde{\varepsilon}}}$, we get that $\sum_{i=1}^{k} \left|\hat{f}_i^\Phi - f_i^{\sigma_s}\right| < \frac{1}{m_{k,\tilde{\varepsilon}}}$, which implies that $\left|\hat{f}_i^\Phi - f_i^{\sigma_s}\right| < \frac{1}{m_{k,\tilde{\varepsilon}}}$, for all $i \in [k]$. Therefore, we have $\mu_i < 2$ for all $i \in [k]$, and since $\mu_i \in \mathbb{N}$, we get that $\mu_i \in \{0, 1\}$, for all $i \in [k]$.

Next, we lower bound $\text{GC}_{\tilde{\varepsilon}}(\sigma_s, \boldsymbol{\hat{f}^\Phi})$, as a function of $\mathbf{p}(P_{\Phi,\tilde{\varepsilon}})$ and $g_{\tilde{\varepsilon}}$. Since $\text{GC}_{\tilde{\varepsilon}}$ would complete $g_{\tilde{\varepsilon}}$ groups on instance $(\sigma_s, \boldsymbol{f^{\sigma_s}})$, then, for all $i \in [k]$, σ_s contains at least $g_{\tilde{\varepsilon}} \cdot \lfloor f_i^{\sigma_s} \cdot m_{k,\tilde{\varepsilon}} \rfloor$ items of size s_i. Since $\mu_i \in \{0, 1\}$ for all $i \in [k]$, then, on instance $(\sigma_s, \boldsymbol{\hat{f}^\Phi})$, $\text{GC}_{\tilde{\varepsilon}}$ fills all placeholders of size s_i in $g_{\tilde{\varepsilon}}$ groups, except at most $g_{\tilde{\varepsilon}}$. Hence,

$$\text{GC}_{\tilde{\varepsilon}}(\sigma_s, \boldsymbol{\hat{f}^\Phi}) \geqslant g_{\tilde{\varepsilon}} \cdot \mathbf{p}(P_{\Phi,\tilde{\varepsilon}}) - g_{\tilde{\varepsilon}} \cdot k.$$

For the rest of this proof, we use an argument as in the proof of Theorem 5. To this end, let N be the covering obtained by creating a copy of $\text{OPT}[\sigma_s]$, from which we have removed a number of bins of type $t \in \mathcal{T}_S$, such that the number of bins of type t is divisible by $g_{\tilde{\varepsilon}}$, for all $t \in \mathcal{T}_S$. By similar arguments as in Lemma 4, we get that $\mathbf{p}(N) \geqslant (1-\frac{\tilde{\varepsilon}}{3}) \cdot \text{OPT}(\sigma_s)$.

Next, observe that N is comprised of $g_{\tilde{\varepsilon}}$ identical coverings $\overline{N}$. Since $n \geqslant b$, we can write $|\sigma_s| \geqslant m^2_{k,\tilde{\varepsilon}} + m_{k,\tilde{\varepsilon}}$. Hence, by a similar argument as in the proof of Lemma 4, we have $n_i^{\overline{N}} \leqslant n_i^{P_{\sigma_s,\tilde{\varepsilon}}} + 1 \leqslant n_i^{P_{\Phi,\tilde{\varepsilon}}} + 2$, for all $i \in [k]$, and thus $\mathbf{p}(P_{\Phi,\tilde{\varepsilon}}) \geqslant \mathbf{p}(\overline{N}) - 2 \cdot k$. Moreover, as in Lemma 4, it holds that $k \leqslant \frac{\frac{\tilde{\varepsilon}}{3} \cdot \mathbf{p}(\overline{N})}{1-\frac{\tilde{\varepsilon}}{3}}$, and we can write

$$\mathbf{p}(P_{\Phi,\tilde{\varepsilon}}) \geqslant \mathbf{p}\big(\overline{N}\big) - 2 \cdot \frac{\frac{\tilde{\varepsilon}}{3} \cdot \mathbf{p}\big(\overline{N}\big)}{1-\frac{\tilde{\varepsilon}}{3}} \geqslant (1-\tilde{\varepsilon}) \cdot \mathbf{p}\big(\overline{N}\big) .$$

Conclusively, from the above-established inequalities, we can conclude the following, which completes the proof:

$$\begin{aligned}\mathrm{GC}_{\tilde{\varepsilon}}(\sigma_s, \hat{\boldsymbol{f}}^{\boldsymbol{\Phi}}) &\geqslant g_{\tilde{\varepsilon}} \cdot (\mathbf{p}(P_{\Phi,\tilde{\varepsilon}}) - k) \geqslant g_{\tilde{\varepsilon}} \cdot \left((1-\tilde{\varepsilon}) \cdot \mathbf{p}\big(\overline{N}\big) - \frac{\frac{\tilde{\varepsilon}}{3} \cdot \mathbf{p}\big(\overline{N}\big)}{1-\frac{\tilde{\varepsilon}}{3}} \right) \\ &\geqslant g_{\tilde{\varepsilon}} \cdot \left(1 - \frac{5}{3} \cdot \tilde{\varepsilon}\right) \cdot \mathbf{p}\big(\overline{N}\big) \geqslant \left(1 - \frac{5}{3} \cdot \tilde{\varepsilon}\right) \cdot \left(1 - \frac{\tilde{\varepsilon}}{3}\right) \cdot \text{OPT}(\sigma_s) \\ &= (1 - 2 \cdot \tilde{\varepsilon}) \cdot \text{OPT}(\sigma_s) = (1-\varepsilon) \cdot \text{OPT}(\sigma_s).\end{aligned}$$

◀

5 Concluding Remarks

We studied the power of frequency predictions in improving the performance of online algorithms for the discrete bin cover problem. In particular, we showed that when input is adversarially generated, frequency predictions (from historical data) can help design algorithms with adjustable trade-offs between consistency and robustness. Specifically, one can achieve near-optimal solutions, assuming predictions are error-free. On the other hand, when input is generated stochastically, we showed that frequencies could be learned from an input prefix of constant length to achieve solutions that are arbitrarily close to optimal with arbitrarily high confidence. An interesting variant of the problem concerns inputs generated adversarially but permuted randomly. This setting is in line with recent work on the analysis of algorithms with random order input (see, e.g., [21, 7]). We expect that our algorithm for the stochastic setting can still be applied to this setting to achieve close to optimal solutions with high confidence, although a different analysis is needed.

References

1 Algorithms with predictions. URL: `https://algorithms-with-predictions.github.io/`. Accessed: 2024-01-26.

2 Spyros Angelopoulos, Christoph Dürr, Shendan Jin, Shahin Kamali, and Marc P. Renault. Online computation with untrusted advice. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 52:1–52:15. Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.ITCS.2020.52`.

3 Spyros Angelopoulos and Shahin Kamali. Contract scheduling with predictions. *J. Artif. Intell. Res.*, 77:395–426, 2023.

4 Spyros Angelopoulos, Shahin Kamali, and Kimia Shadkami. Online bin packing with predictions. *Journal of Artificial Intelligence Research*, 78:1111–1141, 2023. `doi:10.1613/jair.1.14820`.

5 Susan F. Assmann, David S. Johnson, Daniel J. Kleitman, and Josepth Y.-T. Leung. On a dual version of the one-dimensional packing problem. *Journal of Algorithms*, 5:502–525, 1984. doi:10.1016/0196-6774(84)90004-X.

6 Siddhartha Banerjee and Daniel Freund. Uniform loss algorithms for online stochastic decision-making with applications to bin packing. In *Abstracts of the 2020 SIGMETRICS*, pages 1–2. ACM, 2020. doi:10.1145/3393691.3394224.

7 Magnus Berg, Joan Boyar, Lene M. Favrholdt, and Kim S. Larsen. Online minimum spanning trees with weight predictions. In *Algorithms and Data Structures (WADS)*, pages 136–148. Springer, Cham, 2023. doi:10.1007/978-3-031-38906-1_10.

8 Magnus Berg and Shahin Kamali. Online bin covering with frequency predictions, 2024. arXiv:2401.14881.

9 Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, New York, NY, USA, 1998.

10 Joan Boyar, Lene M. Favrholdt, Shahin Kamali, and Kim S. Larsen. Online bin covering with advice. *Algorithmica*, 83:795–821, 2021. doi:10.1007/s00453-020-00728-0.

11 Joan Boyar, Lene M. Favrholdt, Shahin Kamali, and Kim S. Larsen. Online interval scheduling with predictions. In *18th International Symposium on Algorithms and Data Structures (WADS)*, volume 14079, pages 193–207, 2023.

12 Andrej Brodnik, Bengt J. Nilsson, and Gordana Vujovic. Online bin covering with exact parameter advice. *arXiv:2309.13647*, 2023. doi:10.48550/arXiv.2309.13647.

13 Clément L. Canonne. A short note on learning discrete distributions. *arXiv:2002.11457*, 2020. doi:10.48550/arXiv.2002.11457.

14 János Csirik, J. B. G. Frenk, Martine Labbé, and Shuzhong Zhang. Two simple algorithms for bin covering. *Acta Cybernetica*, 14:13–25, 1999.

15 János Csirik, David S. Johnson, and Claire Kenyon. Better approximation algorithms for bin covering. In *12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 557–566. SIAM, 2001.

16 János Csirik, David S. Johnson, Claire Kenyon, James B. Orlin, Peter W. Shor, and Richard R. Weber. On the sum-of-squares algorithm for bin packing. *Journal of the ACM*, 53:1–65, 2006. doi:10.1145/1120582.1120583.

17 János Csirik, David S. Johnson, Claire Kenyon, Peter W. Shor, and Richard R. Weber. A self organizing bin packing heuristic. In *Algorithms Engeneering and Experimentation (ALENEX)*, pages 250–269. Springer, 1999. doi:10.1007/3-540-48518-X_15.

18 János Csirik and Vilmos Totik. Online algorithms for a dual version of bin packing. *Discrete Applied Mathematics*, 21:163–167, 1988. doi:10.1016/0166-218X(88)90052-2.

19 János Csirik and Gerhard H. Woeginger. On-line packing and covering problems. In *Online Algorithms*, pages 147–177. Springer, Berlin, Heidelberg, 2005. doi:10.1007/BFb0029568.

20 Leah Epstein. Online variable sized covering. *Information and Computation*, 171:294–305, 2001. doi:10.1006/inco.2001.3087.

21 Anupam Gupta, Gregory Kehne, and Roie Levin. Random order online set cover is as easy as offline. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 1253–1264. IEEE, 2021. doi:10.1109/FOCS52979.2021.00122.

22 Klaus Jensen and Roberto Solis-Oba. An asymptotic fully polynomial time approximation scheme for bin covering. *Theoretical Computer Science*, 306:543–551, 2003. doi:10.1016/S0304-3975(03)00363-3.

23 Dennis Komm. *An Introduction to Online Computation: Determinism, Randomization, Advice*. Springer Cham, Switzerland, 2016. doi:10.1007/978-3-319-42749-2.

24 Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. *Journal of the ACM*, 68:1–25, 2021. doi:10.1145/3447579.

25 Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ml predictions. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*, pages 9684–9693. Curran Associates, Inc., 2018.

26 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning Theory: From Theory to Algorithms*. Cambridge University Press, Cambridge, England, 2014. `doi:10.1017/CBO9781107298019`.

27 Alexander Wei and Fred Zhang. Optimal robustness-consistency trade-offs for learning-augmented online algorithms. In *34th Conference on Neural Information Processing Systems (NeurIPS)*, pages 8042–8053. Curran Associates, Inc., 2020.