On-line sequential bin packing

András György[1] and **Gábor Lugosi**[2] and **György Ottucsák**[3]
[1]Machine Learning Research Group, Computer and Automation Research Institute, Budapest, Hungary [*]
gya@szit.bme.hu
[2] ICREA and Department of Economics, Universitat Pompeu Fabra, Barcelona, Spain [†]
gabor.lugosi@gmail.com
[3] GusGus AB, Budapest, Hungary
oti@szit.bme.hu

Abstract

We consider a sequential version of the classical bin packing problem in which items are received one by one. Before the size of the next item is revealed, the decision maker needs to decide whether the next item is packed in the currently open bin or the bin is closed and a new bin is opened. If the new item doesn't fit, it is lost. If a bin is closed, the remaining free space in the bin accounts for a loss. The goal of the decision maker is to minimize the loss accumulated over n periods. The main result of the paper is an algorithm that has a cumulative loss not much larger than any strategy that uses a fixed threshold at each step to decide whether a new bin is opened.

1 Introduction

In the classical *off-line* bin packing problem, an algorithm receives *items* (also called *pieces*) $x_1, x_2, \ldots, x_n \in (0, 1]$. We have an infinite number of bins, each with capacity 1, and every item is to be assigned to a bin. Further, the sum of the sizes of the items assigned to any bin cannot exceed its capacity. A bin is empty if no item is assigned to it, otherwise, it is used. The goal of the algorithm is to minimize the number of used bins. This is one of the classical NP-hard problems and heuristic and approximation algorithms have been investigated thoroughly, see, e.g., Coffman, Garey, and Johnson [3].

Another well-studied version of the problem is the so-called *on-line* bin packing problem. Here items arrive one by one and each item x_t must be assigned to a bin (with free space at least x_t) immediately, without any knowledge of the next pieces. In this setting the goal is the same as in the off-line problem, that is, the number of used bins is to be minimized, see, e.g., Seiden [8], He and Dósa [6].

In both the off-line and on-line problems the algorithm has access to the bins in arbitrary order. In this paper we abandon this assumption and introduce a more restricted version that we call *sequential bin packing*. In this setting items arrive one by one (just like in the on-line problem) but in each round the algorithm has only two possible choices: assign the given item to the (only) open bin or to the "next" empty bin (in this case this will be the new open bin), and items cannot be assigned anymore to closed bins. An algorithm thus determines a sequence of binary decisions $i_1, \ldots, i_n$ where $i_t = 0$ means that the next item is assigned to the open bin and $i_t = 1$ means that a new bin is opened and the next item is assigned to that bin. Of course, if $i_t = 0$, then it may happen that the item x_t doesn't fit in the open bin. In that case the item is "lost." If the decision is $i_t = 1$ then the remaining empty space in the last closed bin is counted as a loss. The measure of performance we use is the total sum of all lost items and wasted empty space.

Just as in the original bin packing problem, we may distinguish off-line and on-line versions of the sequential bin packing problem. In the *off-line sequential* bin packing problem the entire sequence $x_1, \ldots, x_n$ is known to the algorithm at the outset. Note that unlike in the classical bin packing problem, the order of the items is relevant. This problem turns out to be computationally significantly easier than its non-sequential counterpart. In Section 3 we present a simple algorithm with running time of $O(n^2)$ that minimizes the total loss in the off-line sequential bin packing problem.

Much more interesting is the on-line variant of the sequential bin packing problem. Here the items x_t are revealed one by one, *after* the corresponding decision i_t has been made. In other words, each decision has to be made without any knowledge on the size of the item. Formulated this way, the problem is reminiscent to an on-line *prediction problem*, see [2]. However, unlike in standard formulations of on-line prediction, here the loss the predictor suffers depends not only on the outcome x_t and decision i_t but also on the "state" defined by the fullness of the open bin.

Our goal is to extend the usual bin packing problems to situations in which one can handle only one bin at a time, and items must be processed immediately so they cannot wait for bin changes. To motivate the on-line sequential model, one may imagine a simple revenue management problem in which a decision maker has a unit storage capacity at his disposal. A certain product arrives in packages of different size and after each arrival, it has to be decided whether the stored packages are shipped or not. (Storing of the product is costly.) If the stored goods are shipped, the entire storage

[*]The first author acknowledges support by the Hungarian Scientific Research Fund (OTKA F60787) and the Mobile Innovation Center of Hungary.

[†]The second author acknowledges support by the Spanish Ministry of Science and Technology grant MTM2006-05650 and by the PASCAL Network of Excellence under EC grant no. 506778.

capacity becomes available again. If they are not shipped one waits for the arrival of the next package. However, if the next package is too large to fit in the remaining open space, it is lost.

In another example of application, a sensor collects measurements that can be compressed to variable size (these are the items). The sensor communicates its measurements by sending frames of some fixed size (bins). Since it has limited memory, it cannot store more data than one frame. To save energy, the sensor must maximize its throughput (the proportion of useful data in each frame) and at the same time minimize data loss (this trade-off is reflected in the definition of the loss function).

Just like in on-line prediction, we compare the performance of an algorithm with the best in a pool of reference algorithms (experts). Arguably the most natural comparison class contains all algorithms that use a fixed threshold to decide whether a new bin is opened. In other words, reference predictors are parameterized by a real number $p \in (0, 1]$. An expert with parameter p simply decides to open a new bin whenever the remaining free space in the open bin is less than p. We call such an expert a *constant-threshold* strategy. The main result of this paper is the construction of a randomized algorithm for the sequential on-line bin packing problem that achieves a cumulative loss (measured as the sum of the total wasted capacity and lost items) that is less than the total loss of the best constant-threshold strategy (determined in hindsight) plus a quantity of the order of $n^{2/3} \log^{1/3} n$.

The principal difficulty of the problem lies in the fact that each action of the decision maker takes the problem in a new "state" (determined by the remaining empty space in the open bin) which has an effect on future losses. Moreover, the state of the algorithm is typically different from the state of the experts which makes comparison difficult. In related work, Merhav, Ordentlich, Seroussi, and Weinberger [7] considered a similar setup in which the loss function has a "memory," that is, the loss of a predictor depends on the loss of past actions. Furthermore, Even-Dar, Kakade and Mansour [4] considered the MDP case where the adversarial reward function changes according to some fixed stochastic dynamics. However, there are several main additional difficulties in the present case. First, unlike in [7], but similarly to [4], the loss function has an unbounded memory as the state may depend on an arbitrarily long sequence of past predictions. Second, the state space is infinite (the $[0, 1)$ interval) and the class of experts we compare to is also infinite, in contrast to both of the above papers. However, the special properties of the bin packing problem make it possible to design a prediction strategy with small regret.

Note that the MDP setting of [4] would be a too pessimistic approach to our problem, as in our case there is a strong connection between the rewards in different states, thus the absolute adversarial reward function results in an overestimated worst case. Also in the present case, state transitions are deterministically given by the outcome, the previous state, and the action of the decision maker, while in the setup of [4] transitions are stochastic and depend only on the state and the decision of the algorithm, but not on the reward (or on the underlying individual sequence generating the reward).

We also mention here the similar *on-line bin packing with rejection* problem where the algorithm has an opportunity to reject some items and the loss function is the sum of the number of the used bins and the "costs" of the rejected items[1] (see He and Dósa [6]). However, instead of the number of used bins we use the sum of idle capacities (missed or free spaces) in the used bins to measure the loss instead of the number of the used bins.

The following example may help explain the difference between various versions of the problem.

Example 1 *Let the sequence of the items be* $\langle 0.4, 0.5, 0.2, 0.5, 0.5, 0.3, 0.5, 0.1\rangle$. *Then the cumulative loss of the optimal off-line bin packing is* 0 *and it is* 0.4 *in the case of sequential off-line bin packing (see Figure 1). In the sequential case the third item (0.2) has been rejected.*

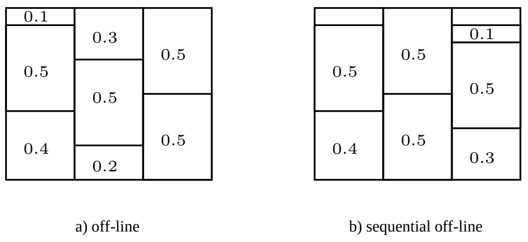

Figure 1: The difference between the optimal solutions for the off-line and sequential off-line problems.

The rest of the paper is organized as follows. In Section 2 the problem is defined formally. In Section 3 the complexity of the off-line sequential bin packing problem is analyzed. The main results of the paper are presented in Section 4.

2 Setup

We use a terminology borrowed from the theory of on-line prediction with expert advice. Thus, we call the sequential decisions of the on-line algorithm *predictions* and we use *forecaster* as a synonym for algorithm.

We denote by $I_t \in \{0, 1\}$ the action of the forecaster at time t (that is, when $t - 1$ items have been received). Action 0 means that the next item will be assigned to the open bin and action 1 represents the fact that a new bin is opened and the next item is assigned to the next empty bin. Note that we assume that we start with an open empty bin, thus for any reasonable algorithm, $I_1 = 0$, and we will restrict our attention to such algorithms. The sequence of decisions up to time t is denoted by $\mathbf{I}_t \in \{0, 1\}^t$.

Denote by $\widehat{s}_t \in [0, 1)$ the free space in the open (last) bin at time $t \geq 1$, that is, after having placed the items $x_1, x_2, \ldots, x_t$ according to the sequence $\mathbf{I}_t$ of actions. This is the *state* of the forecaster. More precisely, the state of the forecaster is defined, recursively, as follows: As at the beginning we have an empty bin, $\widehat{s}_0 = 1$. For $t = 1, 2, \ldots, n$,

[1]In sequential bin packing we assume that the cost of the items coincides with their size. In this case the optimal solution of bin-packing with rejection is to reject all items.

- $\widehat{s}_t = 1 - x_t$, when the algorithm assigns the item to the next empty bin (i.e., $I_t = 1$);
- $\widehat{s}_t = \widehat{s}_{t-1}$, when the assigned item does not fit in the open bin (i.e., $I_t = 0$ and $\widehat{s}_{t-1} < x_t$);
- $\widehat{s}_t = \widehat{s}_{t-1} - x_t$, when the assigned item fits in the open bin (i.e., $I_t = 0$ and $\widehat{s}_{t-1} \geq x_t$).

This may be written in a more compact form:

$$\begin{aligned} \widehat{s}_t &= \widehat{s}_t(I_t, x_t, \widehat{s}_{t-1}) \\ &= I_t(1 - x_t) + (1 - I_t)(\widehat{s}_{t-1} - \mathbb{I}_{\{\widehat{s}_{t-1} \geq x_t\}}) \end{aligned} \quad (1)$$

where $\mathbb{I}_{\{\cdot\}}$ denotes the indicator function of the event in brackets, that is, it equals 1 if the event is true and 0 otherwise. The loss suffered by the forecaster at round t is

$$\ell(I_t, x_t \mid \widehat{s}_{t-1}),$$

where the loss function ℓ is defined by

$$\ell(0, x \mid s) = \begin{cases} 0, & \text{if } s \geq x; \\ x, & \text{otherwise} \end{cases} \quad (2)$$

and

$$\ell(1, x \mid s) = s\ . \quad (3)$$

The goal of the forecaster is to minimize its cumulative loss defined by

$$\widehat{L}_t = L_{\mathbf{I}_t, t} = \sum_{s=1}^{t} \ell(I_s, x_s \mid \widehat{s}_{s-1})\ .$$

In the off-line version of the problem, the entire sequence $x_1, \ldots, x_n$ is given and the solution is the optimal sequence $\mathbf{I}_n^*$ of actions

$$\mathbf{I}_n^* = \arg \min_{\mathbf{I}_n \in \{0,1\}^n} L_{\mathbf{I}_n, n}\ .$$

In the on-line version of the problem the forecaster does not know the size of the next items, and the sequence of items can be completely arbitrary. We allow the forecaster to randomize its decisions, that is, at each time instance t, I_t is allowed to depend on a random variable U_t where $U_1, \ldots, U_n$ are i.i.d. uniformly distributed random variables in $[0, 1]$.

Since we allow the forecaster to randomize, it is important to clarify that the entire sequence of items are determined *before* the forecaster starts making decisions, that is, $x_1, \ldots, x_n \in (0, 1]$ are fixed and cannot depend on the randomizing variables. (This is the so-called *oblivious adversary* model known in the theory of sequential prediction, see, e.g., [2].)

The performance of a sequential on-line algorithm is measured by its cumulative loss. It is natural to compare it to the cumulative loss of the off-line solution $\mathbf{I}_n^*$. However, it is easy to see that in general it is impossible to achieve an on-line performance that is comparable to the optimal solution. (This is in contrast with the non-sequential counterpart of the bin packing problem in which there exist on-line algorithms for which the number of used bins is within a constant factor of that of the optimal solution.)

So in order to measure the performance of a sequential on-line algorithm in a meaningful way, we adopt an approach extensively used in on-line prediction (the so-called "experts" framework). We define a set of reference forecasters, the so-called *experts*. The performance of the algorithm is evaluated relative to this set of experts, and the goal is to perform asymptotically as well as the best expert from the reference class.

Formally, let $f_{E,t} \in \{0, 1\}$ be the decision of an expert E at round t, where $E \in \mathcal{E}$ and $\mathcal{E}$ is the set of the experts. This set may be finite or infinite, we consider both cases below. Similarly we denote the state of expert E with $s_{E,t}$ after the t-th item has been revealed. Then the loss of expert E at round t is

$$\ell(f_{E,t}, x_t \mid s_{E,t-1})$$

and the cumulative loss of expert E is

$$L_{E,n} = \sum_{t=1}^{n} \ell(f_{E,t}, x_t \mid s_{E,t-1}).$$

The goal of the algorithm is to perform almost as well as the best expert from the reference class $\mathcal{E}$. Ideally, the normalized difference of the cumulative losses (the so-called *regret*) should vanish as n grows, that is, one wishes to achieve

$$\limsup_{n \to \infty} \frac{1}{n} (\widehat{L}_n - \inf_{E \in \mathcal{E}} L_{E,n}) \leq 0$$

with probability one, regardless of the sequence of items. This property is called *Hannan consistency*, see [5]. The model of sequential on-line bin packing with expert advice is given in Figure 2.

SEQUENTIAL ON-LINE BIN PACKING PROBLEM WITH EXPERT ADVICE

Parameters: set $\mathcal{E}$ of experts, state space $\mathcal{S} = [0, 1)$, action space $\mathcal{A} = \{0, 1\}$, nonnegative loss function $\ell : (\mathcal{A} \times (0, 1] | \mathcal{S}) \to [0, 1)$, number n of items.
Initialization: $\widehat{s}_0 = 1$ and $s_{E,0} = 1$ for all $E \in \mathcal{E}$.
For each round $t = 1, \ldots, n$,

(a) each expert forms its action $f_{E,t} \in \mathcal{A}$;
(b) the forecaster observes the actions of the experts and forms its own decision $I_t \in \mathcal{A}$;
(c) the next item $x_t \in (0, 1]$ is revealed;
(d) the algorithm incurs loss $\ell(I_t, x_t \mid \widehat{s}_{t-1})$ and each expert incurs loss $\ell(f_{E,t}, x_t \mid s_{E,t-1})$. The states of the experts and the algorithm are updated.

Figure 2: Sequential on-line bin packing problem with expert advice.

In Section 4 we design sequential on-line bin packing algorithms for two cases. In the first (and simpler) case we assume that the class $\mathcal{E}$ of experts is finite. In the second case we consider the (infinite) class of experts defined by constant-threshold strategies. But before turning to the on-line problem, we show how the off-line problem can be solved by a simple quadratic-time algorithm.

3 Sequential off-line bin packing

As it is well known, most variants of the bin packing problem are NP-hard, including bin packing with rejection [6], and maximum resource bin packing [1]. In this section we show that the sequential bin packing problem is significantly easier. Indeed, we offer an algorithm to find the optimal sequential strategy with time complexity $O(n^2)$ where n is the number of the items.

The key property is that after the t-th item has been received, the 2^t possible sequences of decisions cannot lead to more than t different states.

Lemma 1 *For any fixed sequence of items $x_1, x_2, \ldots, x_n$ and for every $1 \le t \le n$,*

$$|\mathcal{S}_t| \le t,$$

where

$$\mathcal{S}_t = \{s : s = s_{\mathbf{I}_t,t}, \mathbf{I}_t \in \{0,1\}^t\}$$

and $s_{\mathbf{I}_t,t}$ is the state reached after the sequence $\mathbf{I}_t$ of decisions.

Proof: The proof goes by induction. Note that since $I_1 = 0$, we always have $s_{\mathbf{I}_1,1} = 1 - x_1$, and therefore $|\mathcal{S}_1| = 1$. Now assume that $|\mathcal{S}_{t-1}| \le t-1$. At time t, the state of every sequence of decisions with $I_t = 0$ belongs to the set $\mathcal{S}'_t = \{s' : s' = s - \mathbb{I}_{\{s \ge x_t\}} x_t, s \in \mathcal{S}_{t-1}\}$ and the state of those with $I_t = 1$ becomes $1 - x_t$. Therefore,

$$|\mathcal{S}_t| \le |\mathcal{S}'_t| + 1 \le |\mathcal{S}_{t-1}| + 1 \le t$$

as desired. ■

To describe a computationally efficient algorithm to compute $\mathbf{I}^*_n$, we set up a graph with the set of possible states as a vertex set (there are $O(n^2)$ of them by Lemma 1) and we show that the shortest path on this graph yields the optimal solution of the sequential off-line bin packing problem.

To formalize the problem, consider a finite directed acyclic graph with a set of vertices $V = \{v_1, \ldots, v_{|V|}\}$ and a set of edges $E = \{e_1, \ldots, e_{|E|}\}$. Each vertex $v_k = v(s_k, t_k)$ of the graph is defined by a time index t_k and a state $s_k \in \mathcal{S}_{t_k}$ and corresponds to state s_k reachable after t_k steps. To show the latter dependence, we will write $v_k \in \mathcal{S}_{t_k}$. Two vertices (v_i, v_j) are connected by an edge if and only if $v_i \in \mathcal{S}_{t-1}$, $v_j \in \mathcal{S}_t$ and state v_j is reachable from state v_i. That is, by choosing either action 0 or action 1 in state v_i, the new state becomes v_j after item x_t has been placed. Each edge has a label and a weight: the label corresponds to the action (zero or one) and the weight equals the loss, dependig on the initial state, the action, and the size of item. Figure 3 shows the proposed graph. Moreover a sink vertex $v_{|V|}$ is introduced that is connected with all vertices in $\mathcal{S}_n$. These edges have weight equal to the loss of the final states. The losses of these edges only depend on the initial state of the edges. More precisely, for $(v_i, v_{|V|})$ the loss is $1 - v_i$, where $v_i \in \mathcal{S}_n$.

Notice that there is a one to one coresspondence between paths from v_1 to $v_{|V|}$ and possible sequences of actions of length n. Furthermore, the total weight of each path (calculated as the sum of the weights on the edges of the path) is

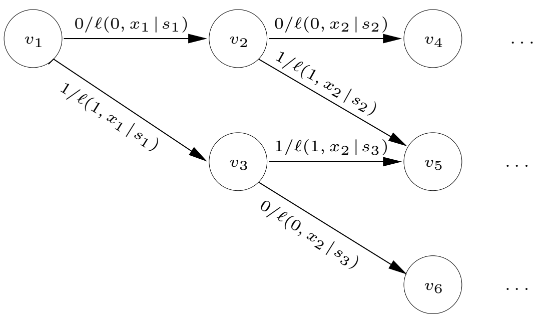

Figure 3: The graph corresponding to the off-line sequential bin packing problem.

equal to the loss of the corresponding sequence of actions. Thus, if we find a path with minimal total weight from v_1 to $v_{|V|}$, we also find the optimal sequence of actions for the off-line bin packing problem. It is well known that this can be done in $O(|V| + |E|)$ time. Now by Lemma 1, $|V| \le n(n+1)/2 + 1$, where the additional vertex accounts for the sink. Moreover it is easy to see that $|E| \le n(n-1) + n = n^2$. Hence the total time complexity of finding the off-line solution is $O(n^2)$

4 Sequential on-line bin packing

In this section we study the sequential on-line bin packing problem with expert advice, as described in Section 2. We deal with two special cases. First we consider finite classes of experts (i.e., reference algorithms) without any assumption on the form or structure of the experts. We construct a randomized algorithm that, with large probability, achieves a cumulative loss not larger than that of the best expert plus $O(n^{2/3} \ln^{1/3} N)$ where $N = |\mathcal{E}|$ is the number of experts. Then we consider the class of all constant-threshold experts and show that a regret of the order $O(n^{2/3} \ln^{1/3} n)$ may be achieved with high probability.

The following simple lemma is a key ingredient of the results of this section. It shows that in sequential on-line bin packing the cumulative loss is not sensitive to the initial states in the sense that the cumulative loss depends on the initial state in a minor way.

Lemma 2 *Let $i_1, \ldots, i_m \in \{0,1\}$ be a fixed sequence of decisions and let $x_1, \ldots, x_m \in (0,1]$ be a sequence of items. Let $s_0, s'_0 \in [0,1)$ be two different initial states. Finally, let $s_0, \ldots, s_m$ and $s'_0, \ldots, s'_m$ denote the sequences of states generated by $i_1, \ldots, i_m$ and $x_1, \ldots, x_m$ starting from initial states s_0 and s'_0, respectively. Then*

$$\left| \sum_{t=1}^{m} \ell(i_t, x_t \mid s'_{t-1}) - \sum_{t=1}^{m} \ell(i_t, x_t \mid s_{t-1}) \right| \le s'_0 + s_0 \le 2\,.$$

Proof: Let m' denote the smallest index for which $i_{m'} = 1$.

Note that $s_{t-1} = s'_{t-1}$ for all $t > m'$. Therefore, we have

$$\begin{aligned}
&\sum_{t=1}^{m} \ell(i_t, x_t \mid s'_{t-1}) - \sum_{t=1}^{m} \ell(i_t, x_t \mid s_{t-1}) \\
&\quad = \sum_{t=1}^{m'} \ell(i_t, x_t \mid s'_{t-1}) - \sum_{t=1}^{m'} \ell(i_t, x_t \mid s_{t-1}) \\
&\quad = \sum_{t=1}^{m'-1} \ell(0, x_t \mid s'_{t-1}) - \sum_{t=1}^{m'-1} \ell(0, x_t \mid s_{t-1}) \\
&\qquad + \ell(1, x_{m'} \mid s'_{m'-1}) - \ell(1, x_{m'} \mid s_{m'-1}) .
\end{aligned}$$

Now using the definition of the loss (see (2) and (3)), we write

$$\begin{aligned}
&\sum_{t=1}^{m} \ell(i_t, x_t \mid s'_{t-1}) - \sum_{t=1}^{m} \ell(i_t, x_t \mid s_{t-1}) \\
&\quad = \sum_{t=1}^{m'-1} x_t (\mathbb{I}_{\{s'_{t-1} < x_t\}} - \mathbb{I}_{\{s_{t-1} < x_t\}}) \\
&\qquad + s'_{m'-1} - s_{m'-1} \\
&\quad \le \sum_{t=1}^{m'-1} x_t (1 - \mathbb{I}_{\{s_{t-1} < x_t\}}) + s'_{m'-1} - s_{m'-1} \\
&\quad \le \sum_{t=1}^{m'-1} x_t (1 - \mathbb{I}_{\{s_{t-1} < x_t\}}) + s'_0 \\
&\quad \le s_0 + s'_0
\end{aligned}$$

where the next-to-last inequality holds because $s'_{m'-1} \le s'_0$ and $s_{m'-1} \ge 0$, and the last inequality follows from the fact that

$$\begin{aligned}
0 \le s_{m'-1} &= s_{m'-2} - \mathbb{I}_{\{s_{m'-2} \ge x_{m'-1}\}} x_{m'-1} \\
&= s_{m'-3} - \mathbb{I}_{\{s_{m'-3} \ge x_{m'-2}\}} x_{m'-2} \\
&\quad - \mathbb{I}_{\{s_{m'-2} \ge x_{m'-1}\}} x_{m'-1} \\
&= s_0 - \sum_{t=1}^{m'-1} \mathbb{I}_{\{s_{t-1} \ge x_t\}} x_t .
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\sum_{t=1}^{m} \ell(i_t, x_t \mid s_{t-1}) - \sum_{t=1}^{m} \ell(i_t, x_t \mid s'_{t-1}) \\
&\qquad\qquad \le s'_0 + s_0
\end{aligned}$$

and the statement follows. ■

The following example shows that upper bound of the lemma is tight.

Example 2 *Let $x_1 = s_0$, $s'_0 < s_0$, and $m' = 2$. Then*

$$\begin{aligned}
&\sum_{t=1}^{m} \ell(i_t, x_t \mid s'_{t-1}) - \sum_{t=1}^{m} \ell(i_t, x_t \mid s_{t-1}) \\
&\quad = \ell(0, x_1 \mid s'_0) + \ell(1, x_2 \mid s'_1) \\
&\qquad - \big(\ell(0, x_1 \mid s_0) + \ell(1, x_2 \mid s_1)\big) \\
&\quad = \ell(0, s_0 \mid s'_0) + \ell(1, x_2 \mid s'_0) \\
&\qquad - \big(\ell(0, s_0 \mid s_0) + \ell(1, x_2 \mid 0)\big) \\
&\quad = s_0 + s'_0 - (0 + 0) .
\end{aligned}$$

4.1 Finite sets of experts

First we consider the on-line sequential bin packing problem when the goal of the algorithm is to keep its cumulative loss close to the best in a finite set of experts. In other words, we assume that the class of experts is finite, say $|\mathcal{E}| = N$, but we do not assume any additional structure of the experts. The ideas presented here will be used below when we consider the infinite class of constant-threshold experts.

The proposed algorithm partitions the time period $t = 1, \dots, n$ into segments of length m where $m < n$ is a positive integer whose value will be specified later. This way we obtain $n' = \lfloor n/m \rfloor$ segments of length m, and, if m does not divide n, an extra segment of length less than m. At the beginning of each segment, the algorithm selects an expert randomly, according to an exponentially weighted average distribution. During the entire segment, the algorithm follows the advice of the selected expert. By changing actions so rarely, the algorithm achieves a certain synchronization with the chosen expert, since the effect of the difference in the initial states is minor, according to Lemma 2. (A similar idea was used in [7] in a different context.) The algorithm is described in Figure 4. Recall that each expert $E \in \mathcal{E}$ recommends an action $f_{E,t} \in \{0,1\}$ at every time instance $t = 1, \dots, n$. Since we have N experts, we may identify $\mathcal{E}$ with the set $\{1, \dots, N\}$. Thus, experts will be indexed by the positive integers $i \in \{1, \dots, N\}$. At the beginning of each segment, the algorithm chooses expert i randomly, with probability $p_{i,t}$, where the distribution $\mathbf{p}_t = (p_{1,t}, \dots, p_{N,t})$ is specified below. The random selection is made independently for each segment.

The following theorem establishes a performance bound of the algorithm. Recall that $\widehat{L}_n$ denotes the cumulative loss of the algorithm while $L_{i,n}$ is that of expert i.

Theorem 3 *Let n, $N \ge 1$, $\eta > 0$, $1 \le m \le n$, and $\delta \in (0,1)$. For any sequence $x_1, \dots, x_n \in (0,1]$ of items, the cumulative loss $\widehat{L}_n$ of the randomized strategy defined above satisfies, with probability at least $1 - \delta$,*

$$\begin{aligned}
&\widehat{L}_n - \min_{i=1,\dots,N} L_{i,n} \\
&\quad \le \frac{m \ln N}{\eta} + \frac{n\eta}{8} + \sqrt{\frac{nm}{2} \ln \frac{1}{\delta}} + \frac{2n}{m} + 2m
\end{aligned}$$

In particular, choosing $m = (16n/\ln(N/\delta))^{1/3}$ and $\eta = \sqrt{8m \ln N/n}$, one has

$$\begin{aligned}
&\widehat{L}_n - \min_{i=1,\dots,N} L_{i,n} \\
&\quad \le \frac{3}{\sqrt[3]{2}} n^{2/3} \ln^{1/3} \frac{N}{\delta} + 4 \left(\frac{2n}{\ln(N/\delta)} \right)^{1/3} .
\end{aligned}$$

Proof: We introduce an auxiliary quantity, the so-called *hypothetical loss*, defined as the loss the algorithm would suffer if it had been in the same state as the selected expert. This hypothetical loss does not depend on previous decisions of the algorithm. More precisely, the true loss of the algorithm at time instance t is $\ell(I_t, x_t \mid \widehat{s}_t)$ and its hypothetic loss is $\ell(I_t, x_t \mid s_{J_t,t})$. Introducing the notation

$$\ell_{i,t} = \ell(f_{i,t}, x_t \mid s_{i,t}) ,$$

```
SEQUENTIAL ON-LINE BIN PACKING ALGORITHM

Parameters: Real number $\eta > 0$ and $m \in \mathbb{N}^+$.
Initialization: $w_{i,0} = 1$ and $s_{i,0} = 1$ for $i = 1, \ldots, N$, and $\widehat{s}_0 = 1$.
For each round $t = 1, \ldots, n$,
  (a) If $((t-1) \bmod m) = 0$ then
        – calculate the updated probability distribution
              $p_{i,t} = \frac{w_{i,t-1}}{\sum_{j=1}^{N} w_{j,t-1}}$
          for $i = 1, \ldots, N$;
        – randomly select an expert $J_t \in \{1, \ldots, N\}$ according to the probability distribution $\mathbf{p}_t = (p_{1,t}, \ldots, p_{N,t})$;
      otherwise, let $J_t = J_{t-1}$.
  (b) Follow the chosen expert: $I_t = f_{J_t,t}$.
  (c) The size of next item $x_t \in (0, 1]$ is revealed.
  (d) The algorithm incurs loss
              $\ell(I_t, x_t \mid \widehat{s}_{t-1})$
      and each expert $i$ incurs loss $\ell(f_{i,t}, x_t \mid s_{i,t-1})$. The states of the experts and the algorithm are changed.
  (e) Update the weights
              $w_{i,t} = w_{i,t-1} e^{-\eta \ell(f_{i,t}, x_t \mid s_{i,t-1})}$
      for all $i \in \{1, \ldots, N\}$.
```

Figure 4: Sequential on-line bin packing algorithm.

the hypothetical loss of the algorithm is just

$$\ell(I_t, x_t \mid s_{J_t,t}) = \ell(f_{J_t,t}, x_t \mid s_{J_t,t}) = \ell_{J_t,t} \ .$$

Now it follows by a well-known result of randomized on-line prediction (see, e.g., [2, Corollary 4.2]) that the hypothetical loss of the sequential on-line bin packing algorithm satisfies, with probability at least $1 - \delta$,

$$\begin{aligned} &\sum_{t=1}^{n} \ell_{J_t,t} - \min_{i=1,\ldots,N} \sum_{t=1}^{n} \ell_{i,t} \qquad (4) \\ &\leq \quad m \left(\frac{\ln N}{\eta} + \frac{n'\eta}{8} + \sqrt{\frac{n'}{2} \ln \frac{1}{\delta}} \right) + m \ , \end{aligned}$$

where $n' = \lfloor \frac{n}{m} \rfloor$ and the last m term comes from bounding the difference on the last, not necessarily complete segment.

Now we may decompose the regret as follows:

$$\begin{aligned} &\widehat{L}_n - \min_{i=1,\ldots,n} L_{i,n} \\ &= \left(\widehat{L}_n - \sum_{t=1}^{n} \ell_{J_t,t} \right) \\ &\quad + \left(\sum_{t=1}^{n} \ell_{J_t,t} - \min_{i=1,\ldots,n} L_{i,n} \right) . \end{aligned}$$

The second term on the right-hand side is bounded using (4). To bound the first term, observe that by Lemma 2,

$$\begin{aligned} &\widehat{L}_n - \min_{i=1,\ldots,n} L_{i,n} \\ &= \sum_{t=1}^{n} \ell(I_t, x_t \mid \widehat{s}_{t-1}) - \sum_{t=1}^{n} \ell(I_t, x_t \mid s_{J_t-1}) \\ &\leq m + \sum_{s=0}^{n'-1} \sum_{t=1}^{m} \big(\ell(I_{sm+t}, x_{sm+t} \mid \widehat{s}_{sm+t-1}) \\ &\quad - \ell(I_{sm+t}, x_{sm+t} \mid s_{J_{sm+t-1},sm+t-1}) \big) \\ &\leq m + 2n' \end{aligned}$$

where in the second inequality we bounded the difference on the last segment separately. ∎

4.2 Constant-threshold experts

Now we are prepared to address the sequential on-line bin packing problem when the goal is to perform almost as well as the best in the class of all constant-threshold strategies. Recall that a constant-threshold strategy is parameterized by a number $p \in (0, 1]$ and it opens a new bin if and only if the remaining empty space in the bin is less than p. More precisely, if the state of the algorithm defined by expert with parameter p is $s_{p,t-1}$, then at time t the expert's advice is $\mathbb{I}_{\{s_{p,t-1} < p\}}$. To simplify notation, we will refer to each expert with its parameter, and, similarly to the previous section, $f_{p,t}$ and $s_{p,t}$ will denote the decision of expert p at time t, and its state after the decision, respectively.

The difficulty in this setup is that there are uncountably many constant-threshold experts. In this section we provide a solution to this problem by reducing it to the case of finite expert classes. The main observation that enables this reduction is that on any sequence of n items, experts can exhibit only a finite number of different behaviors. In a sense, the "effective" number of experts is not too large and this fact may be exploited by the algorithm.

For $t = 1, \ldots, n$ we call two experts *t-indistinguishable* (with respect to the sequence of items $x_1, \ldots, x_t$) if their decision sequences are identical up to time t. This property defines a natural partitioning of the class of experts into maximal t-indistinguishable sets, where any two experts that belong to the same set are t-indistinguishable, and experts from different sets are not t-indistinguishable. Obviously, there are no more than 2^t maximal t-indistinguishable sets. This bound, although finite, is still too large to be useful. However, it turns out that the number of maximal t-indistinguishable sets only grows quadratically with t.

The first step in proving this fact is the next lemma that shows that the maximal t-indistinguishable expert sets are intervals.

Lemma 4 *Let $1 \geq p > r > 0$ be such that expert p and expert r are t-indistinguishable. Then for any $p > q > r$ expert q is t-indistinguishable from both experts p and r. Thus, the maximal t-indistinguishable expert sets form subintervals of* $(0, 1]$.

Proof: By the assumption of the lemma the decision sequences of experts p and r coincide, that is,

$$f_{p,u} = f_{r,u} \qquad \text{and} \qquad s_{p,u} = s_{r,u}$$

for all $u = 1, 2, \ldots, t$. Let $t_1, t_2, \ldots$ denote the time instances when expert p (or expert r) assigns the next item to the next empty bin (i.e., $f_{p,u} = 1$ for $u = t_1, t_2, \ldots$). If expert q also decides 1 at time t_k for some k, then it will decide 0 for $t = t_k + 1, \ldots, t_{k+1} - 1$ since so does expert p and $p > q$, and will decide 1 at time t_{k+1} as $q > r$. Thus the decision sequence of expert q coincides with that of expert p and r for time instances $t_k + 1, \ldots, t_{k+1}$ in this case. Since all experts start with the empty bin at time 0, the statement of the lemma follows by induction. ∎

Based on the lemma we can identify the t-indistinguishable sets by their end points. Let $\mathcal{Q}_t = \{q_{1,t}, \ldots, q_{N_t,t}\}$ denote the set of the end points after receiving t items, where $N_t = |\mathcal{Q}_t|$ is the number of maximal t-indistinguishable sets, and $q_{0,t} = 0 < q_{1,t} < q_{2,t} < \cdots < q_{N_t,t} = 1$. Then the t-indistinguishable sets are $(q_{k-1,t}, q_{k,t}]$ for $k = 1, \ldots, N_t$. The next result shows that the number of maximal t-indistinguishable sets cannot grow too fast.

Lemma 5 *The number of the maximal t-indistinguishable sets is at most quadratic in the number of the items t. More precisely, $N_t \leq 1 + (t-1)t/2$ for any $1 \leq t \leq n$.*

Proof: The proof is by induction. First, $N_1 = 1$ (and $\mathcal{Q}_1 = \{1\}$) since the first decision of each expert is 1. Now assume that $N_{t-1} \leq 1 + (t-2)(t-1)/2$ for some $1 \leq t \leq n-1$. When the next item x_t arrives, an expert p with state s decides 1 in the next step if and only if $0 \leq s - x_t < p$. Therefore, as each expert belonging to the same indistinguishable set has the same state, the k-th maximal $(t-1)$-indistinguishable interval with state s is split into two subintervals if and only if $q_{k-1,t-1} < s - x_t \leq q_{k,t-1}$ (experts in this interval with parameters larger than $s - x_t$ will form one subset, and the ones with parameter at most $s - x_t$ will form the other one). As the number of possible states at time $t-1$ is at most $t-1$ by Lemma 1, it follows that at most $t-1$ intervals can be split, and so $N_t \leq N_{t-1} + t - 1 \leq 1 + (t-1)t/2$, where the second inequality holds by the induction hypothesis. ∎

This lemma makes it possible to apply our earlier algorithm for the case of finite expert classes. However, note that the number of "distinguishable" experts, that is, the number of the maximal indistinguishable sets, constantly grows with time, and each indistinguishable set contains a continuum number of experts. Therefore we need to redefine the algorithm carefully. This may be done by a two-level random

```
SEQUENTIAL ON-LINE BIN PACKING ALGORITHM
WITH CONSTANT-THRESHOLD EXPERTS

Parameters: $\eta > 0$ and $m \in \mathbb{N}^+$.
Initialization: $w_{0,1} = 1$, $N_1 = 1$, $\mathcal{Q}_1 = \{1\}$, $s_{1,0} = 1$ and $\widehat{s}_0 = 1$.
For each round $t = 1, \ldots, n$,

  (a) If $((t-1) \bmod m) = 0$ then
        – for $i = 1, \ldots, N_t$, compute the probabilities
              $p_{i,t} = \frac{w_{i,t-1}}{\sum_{j=1}^{N_t} w_{j,t-1}}$;
        – randomly select an interval $J_t \in \{1, \ldots, N_t\}$ according to the probability
          distribution $\mathbf{p}_t = (p_{1,t}, \ldots, p_{N_t,t})$;
        – choose an expert $p_t$ uniformly from the interval $(q_{J_t-1,t}, q_{J_t,t}]$;
      otherwise, let $p_t = p_{t-1}$.
  (b) Follow the decision of expert $p_t$: $I_t = f_{p_t,t}$.
  (c) $x_t \in (0, 1]$, the size of the next item is revealed.
  (d) The algorithm incurs loss
              $\ell(I_t, x_t \mid \widehat{s}_{t-1})$
      and each expert $p \in (0, 1]$ incurs loss $\ell(f_{p,t}, x_t \mid s_{p,t-1})$, where $p \in [0, 1)$.
  (e) Compute the state $\widehat{s}_t$ of the algorithm by (2), and calculate the auxiliary
      weights and states of the expert sets for all $i = 1, \ldots, N_t$ by
              $\tilde{w}_{i,t} = w_{i,t-1} e^{-\eta \ell(f_{i,t}, x_t | s_{i,t-1})}$
              $\tilde{s}_{i,t} = f_{i,t}(1 - x_t)$
                        $+ (1 - f_{i,t})(s_{i,t} - \mathbb{I}_{\{s_{i,t} \geq x_t\}})$.
  (f) Update the end points of the intervals:
              $\mathcal{Q}_{t+1} = \mathcal{Q}_t \cup \bigcup_{i=1}^{N_t} \{\tilde{s}_{i,t} : q_{i-1,t} < \tilde{s}_{i,t} \leq q_{i,t}\}$
      and $N_{t+1} = |\mathcal{Q}_{t+1}|$.
  (g) Assign the new states and weights to the $(t+1)$-indistinguishable sets
              $s_{i,t+1} = \tilde{s}_{j,t}$ and $w_{i,t+1} = \tilde{w}_{j,t}$
      for all $i = 1, \ldots, N_{t+1}$ and $j = 1, \ldots, N_t$ such that $q_{j-1,t} < q_{i,t+1} \leq q_{j,t}$.
```

Figure 5: Sequential on-line bin packing algorithm with constant-threshold experts.

choice of the experts: first we choose an indistinguishable expert set, then we pick one expert from this set randomly. The resulting algorithm is given in Figure 5.

Up to step (e) the algorithm is essentially the same as in the case of finitely many experts. The two-level random choice of the expert is performed in step (a). In step (f) we update the t-indistinguishable sets, and usually introduce new indistinguishable expert sets. Because of these new expert sets, the update of the weights $w_{i,t}$ and the states $s_{i,t}$ are performed in two steps, (e) and (g), where the actual update is made in step (e), and reordering of these quantities according to the new indistinguishable sets is performed in step (g) together with the introduction of the weights and states for the newly formed expert sets.

The performance and complexity of the algorithm is given in the next theorem.

Theorem 6 *Let $N = 1+n(n-1)/2, m = (16n/\ln(n^2/\delta))^{1/3}$ and $\eta = 4\sqrt{m \ln n/n}$ and $\delta \in (0,1)$. Then the regret of the algorithm defined above is bounded, with probability at least $1-\delta$, by*

$$\widehat{L}_n - \inf_{p\in(0,1]} L_{p,n} \le \frac{3}{\sqrt[3]{2}} n^{2/3} \ln^{1/3} \frac{n^2}{\delta} + 4\left(\frac{2n}{\ln(n^2/\delta)}\right)^{1/3} .$$

Moreover, the algorithm can be implemented with time complexity $O(n^3)$ and space complexity $O(n^2)$.

Proof: It is easy to see that the two-level choice of the expert p_t ensures that the algorithm is the same as for the finite expert class with the experts defined by $\mathcal{Q}_n$. Thus, Theorem 3 can be used to bound the regret, where the number of experts is N_t. By Lemma 5, the latter is bounded by $N < n^2$, which finishes the proof of the first statement.

For the second part note that the algorithm has to store the states, the intervals, the weights and the probabilities, each on the order of $O(n^2)$ based on Lemma 5. Concerning time complexity, the algorithm has to update the weights and states in each round (requiring $O(n^2)$ computations per round), and has to compute the probabilities in every m step, which requires $O(n^3/m)$ computations. Thus the time complexity of the algorithm is $O(n^3)$. ∎

The next example reveals that the loss of the best expert can be arbitrarily far from that of the optimal sequential off-line packing.

Example 3 *Let the sequence of items be*

$$\langle\, \underbrace{\varepsilon, 1{-}\varepsilon, \varepsilon, 1{-}\varepsilon, \ldots, \varepsilon, 1{-}\varepsilon}_{2k}, \varepsilon, \underbrace{1, 1, \ldots, 1}_{k} \,\rangle,$$

where the number of items is $n = 3k+1$ and $0 < \varepsilon < 1$. The optimal sequential off-line packing is achieved if we drop the first or the $(2k+1)$-th ε term; then the total loss is ε. In contrast to this, the loss of the constant-threshold experts is $1-\varepsilon+k$ independently of the choice of the parameter p. Namely, if $p \le 1-\varepsilon$ then the loss is 0 for the first $2k$ items, but after the algorithm is stuck and suffers $k+1-\varepsilon$ loss. If $p > 1-\varepsilon$, then the loss is k for the first $2k$ items and after that $1-\varepsilon$ for the rest of the sequence.

5 Conclusions

In this paper we provide an extension of the classical bin packing problems to an on-line sequential scenario. In this setting items are received one by one, and before the size of the next item is revealed, the decision maker needs to decide whether the next item is packed in the currently open bin or the bin is closed and a new bin is opened. If the new item doesn't fit, it is lost. If a bin is closed, the remaining free space in the bin accounts for a loss. The goal of the decision maker is to minimize the loss accumulated over n periods.

As the main result of the paper, we give an algorithm that has a cumulative loss not much larger than any finite set of reference algorithms, and, more importantly, another algorithm that has a cumulative loss not much larger than any strategy that uses a fixed threshold at each step to decide whether a new bin is opened. An interesting aspect of the problem is that the loss function has an (unbounded) memory. The presented solutions rely on the fact that one can "synchronize" the loss function in the sense that no matter in what state an algorithm is started, its loss may change only by a small additive constant. The second result is obtained by a covering of the uncountable set of constant-threshold experts such that the cardinality of the chosen finite set of experts grows only quadratically with the sequence length. The approach in the paper can easily be extended to any control problem where the loss function has such a synchronizable property.

References

[1] J. Boyar, L. Epstein, L.M. Favrholdt, J.S. Kohrt, K.S. Larsen, M.M. Pedersen, and S. Wøhlk. The maximum resource bin packing problem. *Theoretical Computer Science*, 362:127–139, 2006.

[2] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[3] E.G. Coffman, M.R. Garey, and D.S. Johnson. *Approximation algorithms for bin packing: a survey*. In *Approximation algorithms for NP-hard problems*, pp. 46–93, PWS Publishing Co., Boston, MA, 1997.

[4] E. Even-Dar, S.M. Kakade, and Y. Mansour. Experts in a Markov Decision Process. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pp. 401–408. MIT Press,Cambridge, MA, 2005.

[5] J. Hannan. Approximation to Bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pp. 97–139. Princeton University Press, 1957.

[6] Y. He and Gy. Dósa. Bin packing and covering problems with rejection. *Lecture Notes in Computer Science 3595*, pp. 885–894, 2005.

[7] N. Merhav, E. Ordentlich, G. Seroussi, and M. J. Weinberger. On sequential strategies for loss functions with memory. *IEEE Transactions on Information Theory*, 48:1947-1958, 2002.

[8] S.S. Seiden. On the online bin packing problem. in *Proceedings of the 28th International Colloquium on Automata, Languages and Programming*, pp. 237 - 248, 2001.