CHOICE-MEMORY TRADEOFF IN ALLOCATIONS

NOGA ALON, ORI GUREL-GUREVICH, AND EYAL LUBETZKY

Abstract. In the classical balls-and-bins paradigm, where n balls are placed independently and uniformly in n bins, typically the number of bins with at least two balls in them is $\Theta(n)$ and the maximum number of balls in a bin is $\Theta(\frac{\log n}{\log\log n})$. It is well known that when each round offers k independent uniform options for bins, it is possible to typically achieve a constant maximal load if and only if $k = \Omega(\log n)$. Moreover, it is possible **whp** to avoid any collisions between $n/2$ balls if $k > \log_2 n$.

In this work, we extend this into the setting where only m bits of memory are available. We establish a tradeoff between the number of choices k and the memory m, dictated by the quantity km/n. Roughly put, we show that for $km \gg n$ one can achieve a constant maximal load, while for $km \ll n$ no substantial improvement can be gained over the case $k = 1$ (i.e., a random allocation).

For any $k = \Omega(\log n)$ and $m = \Omega(\log^2 n)$, one can achieve a constant load **whp** if $km = \Omega(n)$, and the load is unbounded if $km = o(n)$. Similarly, if $km > Cn$ then $n/2$ balls can be allocated without any collisions **whp**, whereas for $km < \varepsilon n$ there are typically $\Omega(n)$ collisions. Furthermore, we show that the load is **whp** at least $\frac{\log(n/m)}{\log k + \log\log(n/m)}$. In particular, for $k \asymp \operatorname{polylog} n$, if $m = n^{1-\delta}$ the optimal maximal load is $\Theta(\frac{\log n}{\log\log n})$ (the same as in the case $k = 1$), while $m = 2n$ suffices to ensure a constant load. Finally, we analyze non-adaptive allocation algorithms and give tight upper and lower bounds for their performance.

1. Introduction

The balls-and-bins paradigm (see, e.g., [6],[8]) describes the process where b balls are placed independently and uniformly at random in n bins. Many variants of this classical occupancy problem were intensively studied, having a wide range of applications in Computer Science.

It is well-known that when $b = \lambda n$ for λ fixed and $n \to \infty$, the load of each bin tends to Poisson with mean λ and the bins are asymptotically independent. In particular, for $b = n$, the typical number of empty bins at the end of the process is $(1/\mathrm{e} + o(1))n$. The typical maximal load in that case is $(1 + o(1))\frac{\log n}{\log\log n}$ (cf. [7]). In what follows, we say that an event holds with high probability (**whp**) if its probability tends to 1 as $n \to \infty$.

The extensive study of this model in the context of load balancing was pioneered by the celebrated paper of Azar et. al. [3] (see the survey [10]) that analyzed the effect of a choice between k independent uniform bins on

the maximal load, in an online allocation of n balls to n bins. It was shown in [3] that the GREEDY algorithm (choose the least loaded bin of the k) is optimal and achieves a maximal-load of $\log_k \log n$ **whp**, compared to a load of $\frac{\log n}{\log \log n}$ for the original case $k = 1$. Thus, $k = 2$ random choices already significantly reduce the maximal load, and as k further increases, the maximal load drops until it becomes constant at $k = \Omega(\log n)$.

In the context of online bipartite matchings, the process of dynamically matching each client in a group A of size $n/2$ with one of k independent uniform resources in a group B of size n precisely corresponds to the above generalization of the balls-and-bins paradigm: Each ball has k options for a bin, and is assigned to one of them by an online algorithm that should avoid collisions (no two balls can share a bin). It is well known that the threshold for achieving a perfect matching in this case is $k = \log_2 n$: For $k \geq (1+\varepsilon)\log_2 n$, **whp** every client can be exclusively matched to a target resource, and if $k \leq (1-\varepsilon)\log_2 n$ then $\Omega(n)$ requests cannot be satisfied.

In this work, we study the above models in the presence of a constraint on the memory that the online algorithm has at its disposal. We find that a tradeoff between the choice and the memory governs the ability to achieve a perfect allocation as well as a constant maximal load. Surprisingly, the threshold separating the subcritical regime from the supercritical regime takes a simple form, in terms of the product of the number of choices k, and the size of the memory in bits m:

- If $km \gg n$ then one can allocate $(1-\varepsilon)n$ balls in n bins without any collisions **whp**, and consequently achieve a load of 2 for n balls.
- If $km \ll n$ then *any* algorithm for allocating εn balls **whp** creates $\Omega(n)$ collisions and an unbounded maximal load.

Roughly put, when $km \gg n$ the amount of choice and memory at hand suffices to guarantee an essentially best-possible performance. On the other hand, when $km \ll n$, the memory is too limited to enable the algorithm to make use of the extra choice it has, and no substantial improvement can be gained over the case $k = 1$, where no choice is offered whatsoever.

Our first main result establishes the exact threshold of the choice-memory tradeoff for achieving a constant maximal-load. As mentioned above, one can verify that when there is unlimited memory, the maximal load is **whp** uniformly bounded iff $k = \Omega(\log n)$. Thus, assuming that $k = \Omega(\log n)$ is a prerequisite for discussing the effect of limited memory on this threshold.

Theorem 1. *Consider n balls and n bins, where each ball has $k = \Omega(\log n)$ uniform choices for bins, and $m = \Omega(\log^2 n)$ bits of memory are available. If $km = \Omega(n)$, one can achieve a maximal-load of $O(1)$* **whp**. *Conversely, if $km = o(n)$, any algorithm* **whp** *creates a load that exceeds any constant.*

Consider the case $k = \Theta(\log n)$. The naïve algorithm for achieving a constant maximal-load in this setting requires n bits of memory ($2n$ bits of memory always suffice; see Subsection 1.3). Surprisingly, the above theorem implies that $O(n/\log n)$ bits of memory already suffice, and this is tight.

As we later show, one can extend the upper bound on the load, given in Theorem 1, to $O(\frac{n}{km})$ (useful when $\frac{n}{km} \leq \frac{\log n}{\log\log n}$), whereas the lower bound tends to ∞ with $\frac{n}{km}$. This further demonstrates how the quantity $\frac{n}{km}$ governs the value of the optimal maximal load. Indeed, Theorem 1 will follow from Theorems 3 and 4 below, which determine that the threshold for a perfect matching is $km = \Theta(n)$.

Again consider the case of $k = \Theta(\log n)$, where an online algorithm with unlimited memory can achieve an $O(1)$ load **whp**. While the above theorem settles the memory threshold for achieving a constant load in this case, one can ask what the optimal maximal load would be below the threshold. This is answered by the next theorem, which shows that in this case, e.g., $m = n^{1-\delta}$ bits of memory yield no significant improvement over an algorithm which makes random allocations.

Theorem 2. *Consider n/k balls and n bins, where each ball has k uniform choices for bins, and m bits of memory are available. For any algorithm, the maximal load is at least $(1+o(1))\frac{\log(n/m)}{\log\log(n/m)+\log k}$* **whp**. *Specifically, if $m = n^{1-\delta}$ for some $\delta > 0$ fixed and $k = O(\mathrm{polylog}(n))$, then the maximal load is $\Theta\big(\frac{\log n}{\log\log n}\big)$* **whp**.

Recall that a load of order $\frac{\log n}{\log\log n}$ is what one would obtain using a random allocation of n balls in n bins. The above theorem states that, when $m = n^{1-\delta}$ and $k \leq \mathrm{polylog}(n)$, any algorithm would create such a load already after n/k rounds.

Before describing our remaining results, we note that the lower bounds in our theorems in fact apply to a more general setting. In the original model, in each round the online algorithm chooses one of k uniformly chosen bins, thus inducing a distribution on the location of the next ball. Clearly, this distribution has the property that no bin has a probability larger than k/n.

Our theorems applies to a relaxation of the model, where the algorithm is allowed to dynamically choose a distribution Q_t for each round t, which is required to satisfy the above property (i.e., $\|Q_t\|_\infty \leq k/n$). We refer to these distributions as *strategies*.

Observe that indeed this model gives more power to the online algorithm: For instance, if $k = 2$ (and the memory is unlimited), an algorithm in the relaxed model can allocate $n/2$ balls perfectly (by assigning 0 probability to the occupied bins), whereas in the original model collisions occur already with $n^{2/3}\log n$ balls **whp**.

Furthermore, we also relax the memory constraint on the model. Instead of treating the algorithm as an automaton with 2^m states, we only impose the restriction that there are at most 2^m different strategies to choose from. In other words, at time t, the algorithm knows the entire history (the exact location of each ball so far), and needs to choose one of its 2^m strategies for the next round. In this sense, our lower bounds are for the case of limited communication complexity rather than limited space complexity.

We note that all our bounds remain valid when each round offers k choices with repetitions.

1.1. **Tradeoff for perfect matching.** The next two theorems address the threshold for achieving a perfect matching when allocating $(1-\delta)n$ balls in n bins for some fixed $0 < \delta < 1$ (note that for $\delta = 0$, even with unlimited memory, one needs $k = \Omega(n)$ choices to avoid collisions **whp**). The upper and lower bounds obtained for this threshold are tight up to a multiplicative constant, and again pinpoint its location at $km = \Theta(n)$. The constants below were chosen to simplify the proofs and could be optimized.

Theorem 3. *For $\delta > 0$ fixed, consider $(1-\delta)n$ balls and n bins, where each ball has $2 \le k \le O(n/\log n)$ uniform choices for bins, and there are m bits of memory available. Let $L > 0$ be an arbitrarily large constant, and suppose*

$$km \le \varepsilon n \quad \text{for a suitably small fixed } \varepsilon = \varepsilon(L) > 0 \ .$$

Then any *algorithm* **whp** *either creates a load of n^{ε} or has $\Omega(n)$ collisions and a load of L. Furthermore, the maximal load is* **whp** $\Omega(\log\log(n/(km))$.

Theorem 4. *For $\delta > 0$ fixed, consider $(1-\delta)n$ balls and n bins, where each ball has k uniform choices for bins, and m bits of memory are available. The following holds for any $k \ge (2/\delta)\log n$ and $m \ge \log n \cdot \log_2 \log n$: If*

$$km \ge Cn \text{ for some } C = C(\delta) > 0 \ ,$$

then a perfect allocation (no collisions) can be achieved **whp**.

In light of the above, for any value of k, the online allocation algorithm given by Theorem 4 is optimal with respect to its memory requirements.

1.2. **Non-adaptive algorithms.** In the non-adaptive case the algorithm is again allowed to choose a fixed (possibly randomized) strategy for selecting the placement of ball number t in one of the k possible randomly chosen bins given in step t. Therefore, each such algorithm consists of a sequence $Q_1, Q_2, \ldots, Q_n$ of n pre-determined strategies, where Q_t is the strategy for selecting the bin in step number t. Here we show that even if $k = \frac{n \log\log n}{\log n}$, the maximum load is **whp** at least $(1-o(1))\frac{\log n}{\log\log n}$, that is, it is essentially

as large as in the case $k = 1$. It is also possible to obtain tight bounds for larger values of k. We illustrate this by considering the case $k = n/2$.

Theorem 5. *Consider the problem of allocating n balls into n bins, where each ball has k uniform choices for bins, using a non-adaptive algorithm.*

(i) The maximum load in any non-adaptive algorithm with $k \leq \frac{n \log\log n}{\log n}$ is **whp** *at least $(1 - o(1))\frac{\log n}{\log\log n}$.*

(ii) The maximum load in any non-adaptive algorithm with $k = n/2$ is **whp** *$\Omega(\sqrt{\log n})$. This is tight, that is, there is a non-adaptive algorithm with $k = n/2$ so that the maximum load in it is $O(\sqrt{\log n})$* **whp**.

1.3. **Range of parameters.** In the above theorems and throughout the paper, the parameter k may assume values up to n. As for the memory, one may naïvly use $n \log_2 L$ bits to store the status of n bins, each containing at most L balls. The next observation shows that the $\log_2 L$ factor is redundant:

Observation. *At most $n + b - 1$ bits of memory suffice to keep track of the number of balls in each bin when allocating b balls in n bins.*

Indeed, one can maintain the number of balls in each bin using a vector in $\{0, 1\}^{n+b-1}$, where 1-bits stand for separators between the bins. In light of this, the original case of unlimited memory corresponds to the case $m = 2n$.

1.4. **Main techniques.** The key argument in the lower bound on the performance of the algorithm with limited memory is analyzing the expected number of new collisions that a given step introduces. We wish to estimate this value with an error probability smaller than 2^{-m}, so it would hold **whp** for all of the 2^m possible strategies for this step.

To this end, we generalize the standard Azuma-Hoeffding martingale concentration inequality, and in turn use it to obtain the above mentioned bounds on the error probabilities. Theorem 2.1 bounds the probability of deviation of a martingale, in terms of the bound on its increments and the cumulative variance. The novelty here is that this theorem does not require a uniform bound on individual variances (as it appears in standard versions), and rather treats them as random variables. The proof of this theorem uses probabilistic tools rather than analytical ones.

For the upper bounds, the algorithm essentially partitions the bins into blocks, where for different blocks it maintains an accounting of the occupied bins with varying resolution. Once a block exceeds a certain threshold of occupied bins, it is discarded and a new block takes its place.

1.5. **Organization.** The rest of this paper is organized as follows. In Section 2 we prove the generalized Azuma-Heoffding concentration inequality (Theorem 2.1). Section 3 contains the lower bounds on the collisions and

load, thus proving Theorem 3. Section 4 provides algorithms for achieving a perfect-matching and for achieving a constant load, respectively proving Theorem 4 and completing the proof of Theorem 1. In Section 5 we extend the analysis of the lower bound to prove Theorem 2. Section 6 discusses non-adaptive allocations, and contains the proof of Theorem 5.

Remark. The problem of balanced allocations with limited memory was proposed to us by Itai Benjamini. In a recent independent work, Benjamini and Makarychev [4] settled the special case of the problem for $k = 2$ (i.e., when there are two choices for bins at each round). While our focus was mainly the regime $k = \Omega(\log n)$ (where one can readily achieve a constant maximal load when there is unlimited memory), our results also apply for smaller values of k, and extend the lower bounds of [4] to any $k \leq \text{polylog}(n)$.

2. A generalized Azuma-Hoeffding type inequality

In this section, we prove the following Martingale concentration inequality, which will later be one of the key ingredients in proving the lower bound in the main theorem. This result extends the Azuma martingale inequality, which involves an a-priori bound on the variance of each of the individual increments, into one that incorporates an estimate on the sum of these. For related results, see [5] and the references therein.

Theorem 2.1. *Let $(X_i)_{i=0}^n$ be a martingale with respect to the filter $(\mathcal{F}_i)$. Suppose that $|X_{i+1} - X_i| \leq M$ for all i, and write $V_i = \sum_{j=1}^{i} \mathrm{Var}(X_i \mid \mathcal{F}_{i-1})$. Then for an absolute constant $c > 0$ and any $\lambda, \ell > 0$ and integer n we have*

$$\mathbb{P}\left(X_n \geq X_0 + \lambda \ , \ V_n \leq \ell\right) \leq \exp\left[-c\lambda^2/(\ell + M\lambda)\right] \ .$$

As a special case of Theorem 2.1, note that whenever each of the terms $\mathrm{Var}(X_i \mid \mathcal{F}_{i-1})$ is bounded by some *constant* σ_i^2, then $V_n \leq \sum_i \sigma_i^2$ with probability 1, and we obtain the following well-known result (cf., e.g., [9]):

Corollary 2.2. *Let $(X_i)_{i=0}^n$ be a martingale with respect to the filter $(\mathcal{F}_i)$. Suppose that $|X_{i+1} - X_i| \leq M$ and $\mathrm{Var}(X_i \mid \mathcal{F}_{i-1}) \leq \sigma_i^2$ hold for every i. Then for some absolute constant $c > 0$ and any $\lambda > 0$ and integer n,*

$$\mathbb{P}(X_n \geq X_0 + \lambda) \leq \exp\Big[- c\lambda^2 / \Big(\sum_{i=1}^{n} \sigma_i^2 + M\lambda \Big) \Big] \ .$$

In addition, an immediate corollary of Theorem 2.1 provides a useful bound for the case where the deviation λ exceeds V_n with probability 1:

Corollary 2.3. *Let $(X_i)_{i=0}^n$ be a martingale with respect to the filter $(\mathcal{F}_i)$. Suppose that $|X_{i+1} - X_i| \leq M$ for all i, and that $\sum_{j=1}^{n} \mathrm{Var}(X_i \mid \mathcal{F}_{i-1}) \leq \sigma^2$.*

Then for an absolute constant $c > 0$, any integer n and any $\lambda \geq \sigma^2/M$,

$$\mathbb{P}\left(X_n \geq X_0 + \lambda\right) \leq \exp\left(-c\lambda/M\right) \ .$$

Proof of Theorem 2.1. Define the following stopping times for (X_i):

$$\tau_j \triangleq \min\left\{t > \tau_{j-1} : \left|X_t - X_{\tau_{j-1}}\right| \geq 2M\right\} \quad \text{for } j = 1, 2, \ldots; \ \tau_0 \triangleq 0 \ .$$

By the Optional Stopping Theorem, we have that $(X_{t\wedge(\tau_1\wedge n)})$ is a martingale, and since in addition it is bounded at all times (by $|X_0| + Mn$) and so is our stopping-time, we further have that $\mathbb{E}X_{\tau_1\wedge n} = X_0$. The exact same reasoning gives that for all j

$$\mathbb{E}[X_{\tau_j\wedge n} \mid \mathcal{F}_{\tau_{j-1}}] = X_{\tau_{j-1}} \ ,$$

giving rise to the following definition of a martingale:

$$Z_j \triangleq X_{\tau_j\wedge n} \ .$$

Further define $J \triangleq \min\{j : \tau_j \geq n\}$, and notice that $X_n = Z_J$. We aim to show that, as long as the variance of X is suitably large, one can derive a lower bound on the number of steps made by X along a single move of Z, providing an upper bound on the value of J. To show this, first consider Z_1 (the same analysis will then be applied to all j). It is standard to define

$$Y_i \triangleq (X_i - X_0)^2 - V_i \ ,$$

and obtain that (Y_i) is a martingale, since by definition

$$\begin{aligned}\mathbb{E}[Y_{i+1} - Y_i \mid \mathcal{F}_i] &= \mathbb{E}[(X_{i+1} - X_0)^2 \mid \mathcal{F}_i] - (X_i - X_0)^2 - \mathrm{Var}(X_{i+1} \mid \mathcal{F}_i) \\ &= \mathbb{E}[X_{i+1} \mid \mathcal{F}_i]^2 - X_i^2 = 0 \ ,\end{aligned}$$

where in the last equality we used the fact that (X_i) is itself a martingale. Consider the following stopping times:

$$\begin{aligned}\tau_1^\sigma &\triangleq \min\left\{t : V_t \geq M^2\right\} \ , \\ \tau_1^* &\triangleq \tau_1^\sigma \wedge \tau_1 \wedge n = \min\left\{t : (V_t \geq M^2) \text{ or } (|X_t - X_0| \geq 2M)\right\} \wedge n \ ,\end{aligned}$$

and note that $|Y_{\tau_1^*\wedge t}| < 11M^2$, as $|X_{i+1} - X_i| \leq M$ and $\mathrm{Var}(X_{i+1} \mid \mathcal{F}_i) \leq M^2$ for all i. Optional Stopping with respect to the (bounded) stopping-time τ_1^* again gives that $(Y_{\tau_1^*\wedge t})$ is also a martingale and that furthermore

$$\mathbb{E}Y_{\tau_1^*} = Y_0 = 0 \ .$$

Similarly, Optional Stopping gives that $\mathbb{E}X_{\tau_1^*} = X_0$, and combining this with the fact that $V_{\tau_1^*} < 2M^2$ we deduce that

$$\mathrm{Var}(X_{\tau_1^*}) = \mathbb{E}(X_{\tau_1^*} - X_0)^2 = \mathbb{E}V_{\tau_1^*} < 2M^2 \ .$$

Thus, Chebyshev's inequality gives

$$\mathbb{P}(|X_{\tau_1^*} - X_0| \geq 2M) \leq \frac{\mathrm{Var}(X_{\tau_1^*})}{(2M)^2} \leq \frac{1}{2} \ .$$

Altogether, we have that

$$\mathbb{P}\left((\tau_1^\sigma \wedge n) \leq (\tau_1 \wedge n)\right) \geq \frac{1}{2} \ .$$

Similarly, defining

$$\tau_j^\sigma \triangleq \min\left\{t > \tau_{j-1} : V_t - V_{\tau_{j-1}} \geq M^2\right\} \ , \ \tau_j^* \triangleq \tau_j^\sigma \wedge \tau_j \wedge n \ ,$$

the same analysis implies that for all j we have

$$\mathbb{P}\left((\tau_j^\sigma \wedge n) \leq (\tau_j \wedge n) \mid \mathcal{F}_{\tau_{j-1}}\right) \geq \frac{1}{2} \ .$$

Recalling that $J = \min\{j : \tau_j \geq n\}$, let $K \triangleq \#\{j < J : \tau_j^\sigma \leq \tau_j\}$. The above inequality implies that the number of events $(\tau_j^\sigma \leq \tau_j < n)$ that occur, given that $J > r$, stochastically dominates a binomial variable $\mathrm{Bin}(r, \frac{1}{2})$:

$$(K \mid J > r) \succeq \mathrm{Bin}(r, \tfrac{1}{2}) \ .$$

Each such event increases the value of V by at least M^2. Thus, setting

$$r \triangleq \frac{5\ell}{M^2} \vee \frac{\lambda}{M} \ , \tag{2.1}$$

we get

$$\begin{aligned}
\mathbb{P}(V_{\tau_r} \leq \ell \mid J > r) &\leq \mathbb{P}\Big(K \leq \frac{\ell}{M^2} \mid J > r\Big) \leq \frac{\ell}{M^2}\binom{r}{\ell/M^2} 2^{-r} \\
&\leq \frac{\ell}{M^2}\Big(\mathrm{e}\left(rM^2/\ell\right) 2^{-\frac{9}{10}(rM^2/\ell)}\Big)^{\ell/M^2} 2^{-r/10} \\
&< \frac{\ell}{M^2}\Big(\frac{2}{3}\Big)^{\ell/M^2} 2^{-r/10} < 2^{-r/10} < \exp(-r/5) \ , \qquad (2.2)
\end{aligned}$$

where all inequalities hold for any $\ell, \lambda > 0$ by the requirement $r \geq 5\ell/M^2$. By (2.1) and (2.2), for any $\lambda \leq 5\ell/M$ (in which case $\frac{\lambda^2}{\ell} \leq 25\frac{\ell}{M^2}$) we have

$$\mathbb{P}(V_{\tau_r} \leq \ell \mid J > r) < \mathrm{e}^{-\ell/M^2} \leq \mathrm{e}^{-\frac{\lambda^2}{25\ell}} \ ,$$

whereas for any $\lambda \geq 5\ell/M$

$$\mathbb{P}(V_{\tau_r} \leq \ell \mid J > r) < \mathrm{e}^{-\frac{\lambda}{5M}} \ .$$

Combining the two bounds, we conclude that in both cases

$$\mathbb{P}(V_{\tau_r} \leq \ell \mid J > r) < \exp\Big[-\frac{\lambda^2}{25\ell \ \vee \ 5\lambda M}\Big] \ . \tag{2.3}$$

On the other hand, we claim that, roughly put, when $J \le r$ the probability of $|X_t - X_0|$ exceeding λ becomes suitably small. More precisely, put

$$\tau' \triangleq \min\{t : |Z_j - Z_0| \ge \lambda\} \ , \ Z'_j = Z_{j\wedge\tau'} \ .$$

A final application of Optional Stopping gives that (Z'_j) is a martingale, hence $\mathbb{E}Z'_j = Z'_0 = X_0$. Furthermore, as the increments of (Z'_j) are bounded by $3M$, the standard Azuma inequality guarantees that for any t,

$$\mathbb{P}(|Z'_t - X_0| \ge \lambda) \le \exp\Big[-\frac{\lambda^2}{2(3M)^2 t}\Big] = \exp\left[-\lambda^2/\left(18M^2 t\right)\right] \ . \qquad (2.4)$$

The following inequality will now allow us to put all the pieces together into a bound on $|X_n - X_0|$:

$$\begin{aligned}\mathbb{P}(|X_n - X_0| \ge \lambda \ , \ V_n \le \ell) \\ &\le \mathbb{P}(|X_n - X_0| \ge \lambda \ , \ J \le r) + \mathbb{P}(V_n \le \ell \ , \ J > r) \\ &= \mathbb{P}(|Z_J - X_0| \ge \lambda \ , \ J \le r) + \mathbb{P}(V_n \le \ell \ , \ J > r) \\ &\le \mathbb{P}(\tau' \le J \le r) + \mathbb{P}(V_n \le \ell \ , \ J > r) \ .\end{aligned}$$

By (2.3) and the fact that (V_j) is increasing in j,

$$\mathbb{P}(V_n \le \ell \ , \ J > r) < \exp\Big[-\frac{\lambda^2}{25(\ell \ \vee \ \lambda M)}\Big] \ ,$$

whereas applying (2.4) with $j = r$ gives

$$\begin{aligned}\mathbb{P}(\tau' \le J \le r) \le \mathbb{P}(\tau' \le r) = \mathbb{P}(|Z'_r - X_0| \ge \lambda) &\le \exp\left[-\lambda^2/\left(18rM^2\right)\right] \\ &\le \exp\Big[-\frac{\lambda^2}{18(5\ell \ \vee \ \lambda M)}\Big] \ .\end{aligned}$$

Altogether, we obtain that

$$\mathbb{P}(|X_n - X_0| \ge \lambda \ , \ V_n \le \ell) \le 2\exp\Big[-\frac{\lambda^2}{100(\ell \ \vee \ \lambda M)}\Big] \ ,$$

as required. ■

3. Lower bounds on the collisions and load

Theorem 1, establishing that the quantity km (the choice times the memory) determines either if a perfect allocation is possible, or if any allocation would necessarily produce nearly linearly many bins with arbitrarily large load.

The main ingredient in proving the lower bound on the number of bins with arbitrarily large load is an analogous bound for the number of collisions, i.e., pairs of balls that share a bin, defined as follows: Let $N_t(i)$ denote the

number of balls in bin i after performing t rounds; the number of collisions at time t is then

$$\mathrm{Col}_2(t) \triangleq \sum_{i=1}^{n} \binom{N_t(i)}{2} .$$

The following theorem provides a lower bound on $\mathrm{Col}_2(t)$ in the current regime of km:

Theorem 3.1. *In the setting of Theorem 1, there exists an absolute constant $c > 0$ so that the following holds with probability $1 - O(n^{-2})$:*

(i) For all $t \geq c \cdot k(m + \log n)$ we have

$$\mathbb{E}\,\mathrm{Col}_2(t) \geq \frac{1}{8} t^2/n \ .$$

(ii) For all $t \geq \big[c \cdot k(m + \log n) \ \vee \ Ln^{2/3} \log n\big]$, where $L(n)$ is a function such that $2 \leq L \leq n^{\varepsilon}$, either the maximal load is at least L or

$$\mathrm{Col}_2(t) \geq \frac{1}{10} t^2/n \ .$$

Notice that Theorem 3 immediately follows from the above theorem, by choosing $t = n/2$ and $L = n^{\varepsilon}$: Indeed, except with probability $O(n^{-2})$, either the maximal-load is n^{ε}, or we have $\mathrm{Col}_2(n) \geq n/40$. It thus remains to prove Theorem 3.1.

Proof of Theorem 3.1. The outline of the proof is as follows: We first relax the problem into one where the algorithm comprises a (randomly and adaptively chosen) sequence of distributions for the actual allocation of the balls. The martingale concentration inequality of Section 2 would then be used to show that the expected number of collisions between these distributions approximates the actual number of collisions between the balls. A lower bound on $\mathbb{E}\,\mathrm{Col}_2(t)$ is then derived by analyzing the best possible structure of these distributions, and is then translated to a bound on $\mathrm{Col}_2(t)$ using another application of the martingale inequality.

As noted in the Introduction, we relax the problem by allowing the algorithm to choose any distribution $\mu = (\mu(1), \ldots, \mu(n))$ for the location of the next ball, as long as it satisfies $\|\mu\|_\infty \leq k/n$.

We also relax the memory constraint as follows. The algorithm has a pool of at most 2^m different strategies, and may choose any of them at a given step without any restriction (basing its dynamic decision on the entire history).

To summarize, the algorithm has a pool of at most 2^m strategies. In each given round, it chooses a strategy μ from this pool based on the entire history, and a ball then falls to a bin distributed according to μ.

Let $\nu = (\nu(1), \ldots, \nu(n))$ be an arbitrary probability distribution on $[n]$ satisfying $\|\nu\|_\infty \leq k/n$, and denote by $Q_s = (Q_s(1), \ldots, Q_s(n))$ the strategy of the algorithm at time s. It will be convenient from time to time to treat these distributions as vectors in $\mathbb{R}^n$.

By the above discussion, Q_s is a random variable whose values belong to some a-priori set $\{\mu_1, \ldots, \mu_{2^m}\}$. We further let J_s denote the actual position of the ball at time s (drawn according to the distribution Q_s).

Given the strategy at time s, let x_s denote the probability of a collision between ν and Q_s, i.e., that the ball that is distributed according to Q_s will collide with the one distributed according to ν. We let v_s be the inner product of Q_s and ν, which measures the expectation of such collisions.

$$x_s^\nu \triangleq \nu(J_s) \ ,$$
$$v_s^\nu \triangleq \langle Q_s, \nu \rangle = \sum_{i=1}^{n} Q_s(i)\nu(i) = \mathbb{E}[x_s^\nu \mid \mathcal{F}_{s-1}] \ .$$

Further define the cumulative sums of v_s^ν and x_s^ν as follows:

$$X_t^\nu \triangleq \sum_{s=1}^{t} x_s^\nu \ ,$$
$$V_t^\nu \triangleq \sum_{s=1}^{t} v_s^\nu \ .$$

To justify these definitions, notice that for each $Q_1, \ldots, Q_t$,

$$X_{s-1}^{Q_s} = \sum_{i=1}^{t-1} Q_s(J_i) = \sum_{i=1}^{n} Q_s(i)|\{r < s : J_r = i\}| = \sum_{i=1}^{n} Q_s(i)N_{s-1}(i) \ ,$$

and so $X_{s-1}^{Q_s}$ (up to the factor k/n) is the expected number of collisions that will be contributed by the ball $J_s \sim Q_s$ given the entire history $\mathcal{F}_{s-1}$. Summing over s, we have that

$$\mathbb{E}\,\mathrm{Col}_2(t) = \sum_{s=1}^{t} \mathbb{E}X_{s-1}^{Q_s} \ ,$$

thus estimating the quantities $X_{s-1}^{Q_s}$ will provide a bound on the expected number of collisions. The next lemma shows that X_t is well approximated by V_t, thereby reducing the problem to analyzing the properties of the Q_i-s.

Lemma 3.2. *Let X_s^ν and V_s^ν be defined as above, and let $c > 0$ be some absolute constant. Then with probability at least $1 - n^{-5}\mathrm{e}^{-m}$, for any t, every $\nu \in \{\mu_1 \ldots, \mu_{2^m}\}$ and all $h \geq c\|\nu\|_\infty(m + \log n)$, we have that $V_s^\nu \geq 2h$ implies $X_s^\nu \geq h$.*

Proof. Throughout the proof of the lemma, we omit the superscripts ν in the quantities X_s^ν and V_s^ν.

Fix ν, suppose $b = \|\nu\|_\infty$ ($\leq k/n$) and define

$$Z_t \triangleq (V_t - X_t)/b \ .$$

Clearly, (Z_t) is a martingale, since the definition that $J_s \sim Q_s$ gives

$$\mathbb{E}[Z_t - Z_{t-1} \mid \mathcal{F}_{t-1}] = \mathbb{E}[(v_t - x_t)/b \mid \mathcal{F}_{t-1}] = 0 \ .$$

Moreover, the increments of this martingale are bounded, and so are their variations: Indeed, as $\|\nu\|_\infty \leq b$, we have that $0 \leq x_s/b \leq 1$, and so

$$\mathrm{Var}\left((v_s - x_s)/b \mid \mathcal{F}_{s-1}\right) = \mathrm{Var}\left(x_s/b \mid \mathcal{F}_{s-1}\right) \leq \mathbb{E}\left[x_s/b \mid \mathcal{F}_{s-1}\right] = v_s/b \ ,$$

giving that

$$|Z_t - Z_{t-1}| \leq 1 \ ,$$

$$\mathrm{Var}(Z_t \mid \mathcal{F}_{t-1}) \leq v_t/b \ , \text{ and thus } \sum_{s=1}^{t} \mathrm{Var}(Z_t \mid \mathcal{F}_{t-1}) \leq V_t/b \ .$$

Also note that for any h, we have $Z_s \geq h/b$ iff $X_s \leq V_s - h$. Thus, applying Theorem 2.1 to (Z_s), we obtain that for some fixed $c > 0$ and any $h > 0$,

$$\mathbb{P}(X_s \leq h \ , \ 2h \leq V_s \leq 4h) \leq \mathbb{P}\left(Z_s \geq h/b \ , \ V_s \leq 4h\right) \leq \exp\left(-ch/b\right) \ .$$

Summing the above over $h, 2h, 3h, \ldots$ we obtain that for any $h > 0$ such that $\exp(-ch/b) \leq \frac{1}{2}$,

$$\mathbb{P}(X_s \leq h \ , \ V_s \geq 2h) \leq 2\exp(-ch/b) \ .$$

In particular, choosing

$$h \triangleq c^{-1} b(2m + 3\log n) = c'\|\nu\|_\infty(2m + 6\log n)$$

implies that whenever $V_s \geq 2h$, we have $X_s \geq h$ except with probability $2n^{-6}\mathrm{e}^{-2m}$. Summing over the pool of at most 2^m predetermined strategies ν and at most n different time-points completes the proof of the lemma. ■

Having shown that X_t^ν is well approximated by $V_t\nu$, and recalling that we are interested in estimating $X_{s-1}^{Q_s}$, we now turn our attention to the possible values of $V_{s-1}^{Q_s}$.

Claim 3.3. *For any strategies $Q_1, \ldots, Q_t$ we have that*

$$\sum_{s=1}^{t-1} V_{s-1}^{Q_s} \geq \frac{t(t-k)}{2n} \ .$$

Proof. By our definitions, for the strategies $Q_1, \ldots, Q_t$ we have

$$\sum_{s=1}^{t} V_{s-1}^{Q_s} = \sum_{s=1}^{t}\sum_{r=1}^{s-1} \langle Q_r, Q_s \rangle = \sum_{i=1}^{n} \sum_{r<s\le t} Q_r(i) Q_s(i)$$
$$= \frac{1}{2}\sum_{i=1}^{n} \left[\Big(\sum_{s=1}^{t} Q_s(i) \Big)^2 - \sum_{s=1}^{t} Q_s(i)^2 \right] . \tag{3.1}$$

Recalling the definition of the strategies Q_i, we have that

$$\begin{cases} 0 \le Q_s(i) \le k/n & \text{for all } i \text{ and } s, \\ \sum_{i=1}^{n} Q_s(i) = 1 & \text{for all } s. \end{cases}$$

Therefore,

$$\sum_{i=1}^{n}\sum_{s=1}^{t} Q_s(i)^2 \le \frac{k}{n} \sum_{i=1}^{n}\sum_{s=1}^{t} Q_s(i) = \frac{kt}{n} .$$

On the other hand, by Cauchy-Schwartz,

$$\sum_{i=1}^{n} \Big(\sum_{s=1}^{t} Q_s(i) \Big)^2 \ge \frac{1}{n} \Big(\sum_{i=1}^{n}\sum_{s=1}^{t} Q_s(i) \Big)^2 = \frac{t^2}{n} .$$

Plugging these two estimates in (3.1) we deduce that

$$\sum_{s=1}^{t} V_{s-1}^{Q_s} \ge \frac{t(t-k)}{2n} ,$$

as required. ∎

While the above claim tells us that the average size of $V_{s-1}^{Q_s}$ is fairly large (has order at least $(t-k)/n$), we wish to obtain bounds corresponding to individual distributions Q_s. As we next show, this sum indeed enjoys a significant contribution from indices s where $V_{s-1}^{Q_s} = \Omega(k(m+\log n)/n)$. More precisely, setting $h = ck(m+\log n)/n$ as in Lemma 3.2, we claim that

$$\sum_{s=1}^{t} V_{s-1}^{Q_s} \mathbf{1}_{\{V_{s-1}^{Q_s} > 2h\}} \ge \frac{t^2}{4n} . \tag{3.2}$$

To see this, observe that if

$$t \ge t_0 \triangleq 4ck(m+\log n) , \tag{3.3}$$

where c is the absolute constant Lemma 3.2, then

$$\sum_{s=1}^{t} V_{s-1}^{Q_s} \mathbf{1}_{\{V_{s-1}^{Q_s} \le 2h\}} \le t \cdot 2ck(m+\log n)/n \le \frac{t^2}{2n} .$$

Combining this with Claim 3.3 yields (3.2).

We may now apply Lemma 3.2, and obtain that, except with probability $n^{-2}\mathrm{e}^{-m}$, whenever $V_{s-1}^{Q_s} \geq 2h$ we have $X_{s-1}^{Q_s} \geq \frac{1}{2}V_{s-1}^{Q_s}$, and so

$$\sum_{s=1}^{t} X_{s-1}^{Q_s} \geq \frac{1}{2}\sum_{s=1}^{t} V_{s-1}^{Q_s}\mathbf{1}_{\{V_{s-1}^{Q_s}>2h\}} \geq \frac{t^2}{8n} \ . \tag{3.4}$$

Altogether, we established that

$$\mathbb{E}\,\mathrm{Col}_2(t) = \sum_{s=1}^{t} X_{s-1}^{Q_s} \geq \frac{t^2}{8n} \ \text{ for all } t \geq t_0. \tag{3.5}$$

This proves Part (i) of Theorem 3.1. It remains to establish concentration for $\mathrm{Col}_2(t)$ under the added assumption that $t \geq Ln^{2/3}\log n$ for some $2 \leq L \leq n^{\varepsilon}$. Recalling that

$$\mathrm{Col}_2(t+1) = \mathrm{Col}_2(t) + N_t(J_{t+1}) \ ,$$

we let

$$Y_t \triangleq \mathrm{Col}_2(t) - \sum_{s=1}^{t-1} X_s^{Q_t} \ ,$$

and obtain that (Y_t) is a martingale, as

$$\begin{aligned}\mathbb{E}[Y_{t+1} - Y_t \mid \mathcal{F}_t] &= \mathbb{E}[N_t(J_{t+1}) \mid \mathcal{F}_t] - X_t^{Q_{t+1}} \\ &= \sum_i Q_{t+1}(i)N_t(i) - X_t^{Q_{t+1}} = 0 \ .\end{aligned}$$

Set the following stopping-time for reaching a maximal-load of L:

$$\tau_L \triangleq \min\Big\{t : \max_j N_t(j) \geq L\Big\} \ .$$

By Optional Stopping, we have that $Y_{t\wedge\tau_L}$ is a martingale, and therefore $\mathbb{E}Y_{t\wedge\tau_L} = Y_0 = 0$. Moreover,

$$|Y_{t+1\wedge\tau_L} - Y_{t\wedge\tau_L}| \leq 2L \ ,$$

and so an application of Azuma's inequality implies that for any t,

$$\begin{aligned}\mathbb{P}\left(|Y_{t\wedge\tau_L}| \geq \tfrac{1}{100}t^2/n\right) &\leq \exp\Big[-\Omega\Big(\frac{(t^2/n)^2}{tL^2}\Big)\Big] \\ &\leq \exp\left[-\Omega\big(t^3/(nL)^2\big)\right] \leq \exp\left[-\Omega(L\log^3 n)\right] \ .\end{aligned}$$

It follows that the following holds for all $t \geq (\frac{1}{8}t^2/n \vee Ln^{2/3}\log n)$, except with probability $\exp(-\Omega(\log^4 n))$: Either the maximal-load is at least L, or

$$\mathrm{Col}_2(t) \geq \mathbb{E}\,\mathrm{Col}_2(t) - \frac{t^2}{100n} \ ,$$

and (3.5) now concludes the proof of Theorem 3.1. ■

3.1. **Boosting the subcritical regime to unbounded maximal load.** While Theorem 3.1 given above provides a careful analysis for the number of 2-collisions, i.e., pairs of balls sharing a bin, one can iteratively apply this theorem, with very few modifications, in order to obtain that the number of C-collisions (a set of C balls sharing a bin) has order $\Omega(n^{1-o(1)})$, or else the maximal load is at least n^{ε}. The proof of this result hinges on the following generalization of Theorem 3.1.

Theorem 3.4. *Consider the following balls and bins setting:*

1. *We have $km \leq \varepsilon n$ for some suitably small $\varepsilon > 0$.*
2. *The online adaptive algorithm has a pool of 2^m possible strategies, where each strategy μ satisfies $\|\mu\|_\infty \leq k/n$. The algorithm selects a (random) sequence of strategies $Q_1, \ldots, Q_n$ adapted to the filter $(\mathcal{F}_i)$.*
3. *Let $A_1 \subset \ldots \subset A_n \subset [n]$ denote a random increasing sequence of subsets adapted to the filter $(\mathcal{F}_i)$, i.e. $A_i \in \mathcal{F}_i$.*
4. *There are n rounds, where in round t a new potential location for a ball is chosen according to Q_t. If this location belongs to A_t, a ball is positioned there (otherwise, nothing happens).*

Define $T = \sum_s Q_s(A_s)$. Then for some absolute constant $c > 0$ and any function $\log n \leq L \leq n^{\varepsilon}$ we have

$$\mathbb{P}\Big(T \geq \big(ck(m + \log n) \vee Ln^{2/3} \log n\big) \ , \ \mathrm{Col}_2(n) < \tfrac{1}{10} T^2/n\Big) \leq n^{-2} \ .$$

Proof. As the proof follows the same arguments of Theorem 3.1, we restrict our attention to describing the modifications that are required for the new statement to hold.

Define the following sub-distribution of Q_s with respect to A_s:

$$Q'_s \triangleq Q_s \mathbf{1}_{A_s} \ .$$

As before, given Q_s, the strategy at time s, define the following parameters:

$$x^\nu_s \triangleq \nu(J_s) \ , \ v^\nu_s \triangleq \sum_{i=1}^n Q'_s(i)\nu(i) \ ,$$

and let the cumulative sums of v^ν_s and x^ν_s be denoted by:

$$X^\nu_t \triangleq \sum_{s=1}^t x^\nu_s \ , \ V^\nu_t \triangleq \sum_{s=1}^t v^\nu_s \ .$$

We claim that the statement of Lemma 3.2 (showing that a lower bound on V_t is with high probability a lower bound on X_t) holds as is with respect to the above definitions. Indeed, the martingale concentration argument is valid without any changes, and the only delicate point is the identity of the target strategy ν, which we next address.

Let ν be a candidate for a given time $r > t$. Before, ν was taken from an pre-given pool of 2^m strategies, whereas now our designated ν would be the Q'_r at the future point of time r. As such, Q'_r may be dynamically influenced by the random variable A_r, destroying the union bound over all the possible strategies! The crucial observation that resolves this issue is the following:

Observation 3.5. *Let $r > t$ and let ν be a candidate strategy for time r. Then $V_t^\nu = V_t^{\nu'}$ and $X_t^\nu = X_t^{\nu'}$ for any increasing sequence $A_1, \ldots, A_r$.*

To see this, first consider X_t^ν and $X_t^{\nu'}$. If J_s for some $1 \le s \le t$ had a non-zero contribution to X_t^ν, then by definition $i \in A_s \subset A_t$ and

Indeed, this follows from monotonicity, since any $i \in A_s$ that qualifies this index to contribute to $\langle Q'_s, \nu \rangle$ must also belong to A_r.

The equivalent of Claim 3.3 follows by definition of T as $\sum_s \sum_i Q'_s(i)$, and the rest of the arguments hold unmodified. ■

We next show how to infer the results regarding an unbounded maximal load and C-collisions for any fixed C from Theorem 3.4. To do so, perform iterations of this theorem as follows. In step $\ell = 0, 1, 2, \ldots$, we define the increasing sequence (A_t) by:

$$A_t \triangleq \{i \in [n] : N_t(i) \ge \ell\} \ .$$

For some $L = L(n)$ to be specified later, and stop the process once we obtain that

$$T < \big(ck(m + \log n) \vee Ln^{2/3} \log n\big) \ .$$

Consider the process at its end. By summing all the error probabilities in Theorem 3.1, we may assume that as long as the process was alive, we had $\mathrm{Col}_2(n) \ge \frac{1}{10} T^2/n$. Now, suppose that the maximal load at the end of the process is less than L. It then follows that at the end of step ℓ, we had at least $\mathrm{Col}_2(n)/L$ balls which incurred collisions in this step. This gives that

$$T_{\ell+1} \ge \frac{T_\ell^2}{10nL} \ , \ T_0 = n$$

and in other words

$$T_\ell \ge \frac{n}{(10L)^{2^\ell - 1}} \ .$$

The stopping rule of $T < \varepsilon km$ means that we stop at the first ℓ such that

$$(10L)^{2^\ell - 1} \ge n/(\varepsilon km) \ ,$$

that is, just as

$$\ell \ge \log_2 \Big(1 + \frac{\log(n/(km)) + \log(1/\varepsilon)}{\log(10L)}\Big) \ .$$

As the maximal load is at least $L \wedge \ell$, selecting $L = \Omega(\log\log(n/(km)))$, gives $\ell = \Theta(L)$, and implies a final lower bound of $\Omega(\log\log(n/(km)))$. This concludes the proof of Theorem 3. ∎

4. Algorithms for perfect matching and constant load

In this section, we prove Theorem 4 by providing an algorithm that avoids collisions **whp** using only $O(n/k)$ bits of memory, which is the minimum possible by Theorem 3. The case $km = \Omega(n)$ of Theorem 1 will then follow from repeated applications of this algorithm.

Perfect allocation algorithm for $(1-\delta)n$ balls

1. For $\ell = \lfloor \frac{n}{\lfloor m/2 \rfloor} \rfloor$, partition the bins into contiguous blocks $B_1, \ldots, B_\ell$ each comprising $\lfloor m/2 \rfloor$ bins. Ignore any remaining unused bins.
2. Set $d = \left\lceil \log_2 \left(\frac{1+\varepsilon}{C\delta} \log n \right) \right\rceil$, and define the arrays $A_0, \ldots, A_{d-1}$:
   - A_j comprises 2^j contiguous blocks (a total of $\sim 2^{j-1}m$ bins).
   - For each contiguous (non-overlapping) 4^j-tuple of bins in A_j, we keep a single bit that holds whether any of its bins is occupied.
   - Pointers to the first block of A_j are chosen so that the union $\cup_j A_j$ forms a contiguous collection of bins.
3. Repeat the following procedure until exhausting all rounds:
   - Let j be the minimal integer so that a bin of A_j, marked as empty, appears in the current selection of k bins. If no such j exists, the algorithm announces failure.
   - Allocate the ball into this bin, and mark its 4^j-tuple as occupied.
   - If the fraction of empty 4^j-tuples remaining in A_j just dropped below $\delta/2$, relocate the array A_j to a fresh contiguous set of empty 2^j blocks (immediately beyond the last allocated block). If there are less than 2^j available new blocks, the algorithm fails.
4. Once $(1-\delta)n$ rounds are performed, the algorithm stops.

Throughout the proof of the algorithm, assume that in each round we are presented with k independent uniform indices of bins, possibly with repetitions. Clearly, an upper bound for the maximal load in this relaxation of the model translates into one for the original model (k choices without repetitions).

4.1. First version of the algorithm.

We begin with a description and a proof of a simpler version of the algorithm, suited for the case where

$$km \geq (3/\delta) n \log n \ . \tag{4.1}$$

This version will serve as the base for the analysis. For simplicity, assume first that $m \mid n$.

FIRST VERSION OF ALLOCATION ALGORITHM FOR $(1-\delta)n$ BALLS

1. Let $B_1, \ldots, B_\ell$ be an arbitrary partition of the n bins into $\ell \triangleq n/m$ blocks, each containing m bins. Put $r \triangleq \lfloor (1-\delta)m \rfloor$.
2. Throughout stage $j \in [\ell]$, only the m bins belonging to B_j are tracked. At the beginning of the stage, all bins in the block are marked empty.
3. Stage j comprises r rounds, in each of which:
   - The algorithm attempts to place a ball in an arbitrary empty bin of B_j if possible.
   - If no empty bin of B_j is offered, the algorithm declares failure.
4. Once $(1-\delta)n$ rounds are performed, the algorithm stops.

To verify that this algorithm indeed produces a perfect allocation **whp**, examine a specific round of stage j, and condition on the event that so far the algorithm did not fail. In particular, its accounting of which bins are occupied in B_j is accurate, and at least $m - r = (\delta - o(1))m$ bins in B_j are still empty (notice that by our assumption $m = \Omega(\log n)$, and so $m \to \infty$ with n).

Let Miss_j denote the event that the next ball precludes all of the empty bins of B_j in its k choices, we have

$$\mathbb{P}(\mathsf{Miss}_j) \le \Big(1 - \frac{m-r}{n}\Big)^k \le \mathrm{e}^{-(\delta - o(1))\frac{km}{n}} \le n^{-3+o(1)} \,, \tag{4.2}$$

by assumption (4.1). A union bound over the n rounds now yields (with room to spare) that the algorithm succeeds **whp**.

The case where m does not divide n is treated similarly: Set $\ell = \lfloor \frac{n}{\lfloor m/2 \rfloor} \rfloor$, and partition the bins into blocks that now hold $\lfloor m/2 \rfloor$ bins each, except for the final block B_ℓ which would have between $\lfloor m/2 \rfloor$ and $m-1$ bins. As before, in stage j we attempt to allocate $\lfloor (1-\delta)|B_j| \rfloor$ balls into B_j, while relying on the property that B_j has at least $(\delta - o(1))|B_j| \ge (\delta - o(1))m/2$ empty bins. This gives

$$\mathbb{P}(\mathsf{Miss}_j) \le \mathrm{e}^{-(\delta - o(1))\frac{km/2}{n}} \le n^{-3/2+o(1)} \,,$$

as required.

4.2. **Second version of the algorithm.** We now wish to adapt the above algorithm to the following case:

$$km \log_2 m \ge (20/\delta) \log(5/\delta) n \log n \,, \quad \log^3 n \le m \le \frac{n}{\log n} \,. \tag{4.3}$$

Notice that if $m \geq n^{\varepsilon}$, the above requirement is essentially that $km = \Omega_{\varepsilon}(n)$. The full version of the algorithm eliminates this dependency on ε.

SECOND VERSION OF THE ALGORITHM FOR $(1-\delta)n$ BALLS

1. For $\ell = \lfloor \frac{n}{\lfloor m/2 \rfloor} \rfloor$, partition the bins into contiguous blocks $B_1, \ldots, B_\ell$ each comprising $\lfloor m/2 \rfloor$ bins. Ignore any remaining unused bins.
2. Set $d = \lfloor \frac{1}{4} \log_2 m \rfloor$, and define the arrays $A_0, \ldots, A_{d-1}$:
   - A_j is one of the blocks $B_1, \ldots, B_\ell$.
   - For each contiguous (non-overlapping) 2^j-tuple of bins in A_j, we keep a single bit that holds whether any of its bins is occupied.
3. Repeat the following procedure until exhausting all rounds:
   - Let j be the minimal integer so that a bin of A_j, marked as empty, appears in the current selection of k bins. If no such j exists, the algorithm announces failure.
   - Allocate the ball into this bin, and mark its 2^j-tuple as occupied.
   - If the fraction of empty 2^j-tuples remaining in A_j just dropped below $\delta/2$, relocate the array A_j to a fresh block (immediately beyond the last allocated block). If no such block is found, the algorithm fails.
4. Once $(1-\delta)n$ rounds are performed, the algorithm stops.

Since the array A_j contains $2^{-j}(m/2)$ different 2^j-tuples, the amount of memory required to maintain the status of all tuples is

$$\frac{m}{2} \sum_{j=0}^{d-1} 2^{-j} = (1 - 2^{-d})m \leq m - m^{3/4} \ .$$

In addition, we keep an index for each A_j, holding its position among the ℓ blocks. By definition of d and ℓ, this amounts to at most

$$d \log_2 \ell \leq (\log_2 n)^2 < m^{3/4}$$

bits of memory, where the last inequality holds for any large n by (4.3).

We first show that the algorithm does not fail to find a bin of A_j marked as empty. At any given point, each A_j has a fraction of at least $\delta/2$ bins marked as empty. Hence, recalling (4.2), the probability of missing all the bins marked as empty in $A_0, \ldots, A_{d-1}$ is at most

$$\begin{aligned} \exp\Big(-\big(\frac{\delta}{2} - o(1)\big)\frac{km}{2n} d \Big) &\leq \exp\Big(-\big(\frac{\delta}{2} - o(1)\big)\frac{10 \log n}{\delta \log_2 m} \log\big(\frac{20}{\delta}\big)\frac{1}{4}\log_2 m \Big) \\ &\leq n^{-\log(5/\delta)5/4 - o(1)} < n^{-5/4} \ , \end{aligned}$$

where the last inequality holds for large n. Therefore, **whp** the algorithm never fails to find an array A_j with an empty bin among the k choices.

It remains to show that, whenever the algorithm relocates an array A_j, there is always a fresh block available.

By the above analysis, the probability that a ball is allocated in A_j for $j \geq 1$ at a given round is at most

$$\exp\Big(-\big(\frac{\delta}{2}-o(1)\big)\frac{km/2}{n}j\Big) \leq \exp\Big(-\big(\frac{\delta}{2}-o(1)\big)\frac{10\log n}{\delta\log_2 m}\log\big(\frac{20}{\delta}\big)j\Big)$$
$$\leq \exp(-3\log(5/\delta)j) \triangleq p_j \ ,$$

where the last inequality holds for any sufficiently large n.

Let N_j denote the number of balls that were allocated in blocks of type j throughout the run of the algorithm. Clearly, N_j is stochastically dominated by a binomial random variable $\mathrm{Bin}(n, p_j)$. Hence, known estimates for the binomial distribution (see, e.g., [2]) imply that for all j,

$$\mathbb{P}(N_j > np_j + C\sqrt{n}\log n) \leq n^{-C} \ .$$

The total number of blocks needed for A_j is at most

$$\left\lceil \frac{2^j N_j}{(1-\frac{\delta}{2})\frac{m}{2}} \right\rceil \ ,$$

and hence the total number of blocks needed is **whp** at most

$$\left\lceil \sum_{j=0}^{d-1} \frac{2^j(1-\delta)np_j + C2^j\sqrt{n}\log n}{(1-\frac{\delta}{2})\frac{m}{2}} \right\rceil \leq \sum_{j=0}^{d-1} \frac{2^j(1-\delta)np_j}{(1-\frac{\delta}{2})\frac{m}{2}} + O\Big(\frac{n^{3/4}\log n}{m}\Big) \ .$$

Since

$$\sum_{j=1}^{d-1} 2^j p_j = \sum_{j=1}^{d-1} \exp\Big(j(\log 2 - 3\log(5/\delta))\Big) < 2\cdot 2(\delta/5)^3 < \delta/5$$

(with room to spare), the total number of blocks needed is **whp** at most

$$\frac{(1+\delta/5)(1-\delta)n}{(1-\frac{\delta}{2})\frac{m}{2}} + O\Big(\frac{n^{3/4}\log n}{m}\Big) < \Big\lfloor \frac{n}{\lfloor m/2\rfloor} \Big\rfloor$$

for any sufficiently large n.

4.3. **Final version of the algorithm.** The main disadvantage in the second version of the algorithm is that the size of each A_j was fixed at $m/2$ bins. Since the resolution of each A_j is in 2^j-tuples, we are limited to at most $\log_2 m$ arrays. However, the probability of missing all the arrays $A_0, \ldots, A_{d-1}$ has to compete with n, hence the requirement that m would be polynomial in n.

To remedy this, the algorithm uses arrays with varying sizes, namely 2^j blocks for A_j. The resolution of each array is now in 4^j-tuples, i.e., A_j

contains at most $2^j\lfloor m/2\rfloor/4^j$ tuples. Thus, the number of memory bits required for all arrays is at most

$$\frac{m}{2}\sum_{j=0}^{d-1}2^{-j}=(1-2^{-d})m\le m-O(m/\log n)\ .$$

The following calculation shows that indeed there are sufficiently many blocks to initially accommodate all the arrays:

$$(2^d-1)\lfloor m/2\rfloor\le\frac{1+\varepsilon}{2C\delta}m\log n\le\frac{km}{2(1+\varepsilon)C}=\frac{n}{2(1+\varepsilon)}\ ,$$

where we used the assumption that $k\ge\frac{1+\varepsilon}{\delta}\log n$, and took $km=Cn$.

(1) Pointers for the location of the arrays:

$$d\log_2 n=(\log_2\log n+O(1))\log_2 n=(1+o(1))\log_2 n\cdot\log_2\log n\ .$$

(2) Arrays have enough tuples: A_{d-1} has about

$$\frac{1}{2}m/2^{d-1}=m/2^d=\frac{C\delta}{1+\varepsilon}m/\log n$$

4^{d-1}-tuples, and the assumption $m=\Omega(\log n\log\log n)$ guarantees this is large.

(3) Bins wasted on 4^j-tuples throughout the $(1-\delta)n$ rounds: stochastically bounded by a binomial variable $\mathrm{Bin}\,((1-\delta)n,p_j)$, where

$$p_j=\mathrm{e}^{-C\delta(2^j-1)}4^j\ .$$

Dominating value: $j=1$, and so if

$$C>(1/\delta)\log(10/\delta)\ ,$$

we have that

$$p_j\le\delta/10\ ,$$

thus the wasted rounds vanish against the extra allocation of $(\delta/2)n$ rounds.

This completes the proof of Theorem 4. ■

By joining bins together (and using the fact that our upper bounds apply to k choices with repetitions), we may extend the analysis to $km=\Omega(n)$ and obtain a constant load. This implies the upper bound in Theorem 1.

5. Improved lower bounds for poly-logarithmic choices

5.1. **Proof of Theorem 2.** Our proof of this case is an extension of the proof of Theorem 3 to estimate the number of q-collisions for general q:

$$\mathrm{Col}_q(t)\triangleq\sum_{i=1}^{n}\binom{N_t(i)}{q}\ .$$

The analysis hinges on a recursion on q, for which we need to achieve bounds on a generalized quantity, a linear function of the q-collisions vector:

$$X_t^{f;q} \triangleq \sum_{s_1<\ldots<s_q\le t} \sum_i f(i)\mathbf{1}_{\{J_{s_1}=i\}}\cdots\mathbf{1}_{\{J_{s_q}=i\}} = \sum_i f(i)\binom{N_t(i)}{q}\,, \quad (5.1)$$

$$V_t^{f;q} \triangleq \sum_{s_1<\ldots<s_q\le t} \sum_i f(i)Q_{s_1}(i)\cdots Q_{s_q}(i)\,. \quad (5.2)$$

Our objective is to obtain lower bounds for $X_t^{f;q}$ with $f\equiv 1$, as clearly $\mathrm{Col}_q(t) = X_t^{\mathbf{1};q}$. In general, our f will be the product of different strategies Q_i, a fact which would allow us to formulate a recursion relation between the $V_t^{f;q}$-s and an approximate recursion for the $X_t^{f;q}$. This is achieved by the next lemma, where here in and throughout the proof we set

$$L \triangleq \log(n/m) \quad (5.3)$$

to denote a maximal load we do not expect to reach (except if the algorithm is far from optimal).

Lemma 5.1. *There exists an absolute constant $c>0$ so that either the maximal load exceeds L, or the following holds for all $q<L$, every $t\le n/k$ and every $f\in\{\mathbf{1},\mu_1,\ldots,\mu_{2^m}\}^L$, except with probability $n^{-5}\mathrm{e}^{-m}$.*

$$\textit{If}\quad V_t^{f;q} \ge c\frac{(2L)^{q+1}}{q!}m\|f\|_\infty \quad \textit{then}\quad X_t^{f;q} \ge 2^{-q}\,V_t^{f;q}\,. \quad (5.4)$$

Proof. The key property of the quantities $V_t^{f;q}$, which justified the inclusion of the inner products with f, is the following recursion relation:

$$V_t^{f;q+1} = \sum_{s<t} V_s^{(Q_{s+1}\cdot f);q} \qquad \text{for any } q\ge 1 \text{ and any } t. \quad (5.5)$$

We now wish to write a similar recursion for the variables $X_t^{f;q}$. As opposed to the variables $V_t^{f;q}$, which satisfied the above recursion combinatorially, here the recursion will only be an approximation. Notice that

$$\begin{aligned} X_{t+1}^{f;q+1} - X_t^{f;q+1} &= f(J_{t+1})\left(\binom{N_t(J_{t+1})+1}{q+1} - \binom{N_t(J_{t+1})}{q+1}\right) \\ &= f(J_{t+1})\binom{N_t(J_{t+1})}{q}\,, \end{aligned}$$

and hence

$$\mathbb{E}\left[X_{t+1}^{f;q+1} - X_t^{f;q+1} \mid \mathcal{F}_t\right] = \sum_i Q_{t+1}(i)f(i)\binom{N_t(i)}{q} = X_t^{Q_{t+1}\cdot f;q}\,.$$

We may thus define $Z_t = Z_t^{f;q}$ by

$$Z_t^{f;q} \triangleq X_t^{f;q+1} - \sum_{s<t} X_s^{(Q_{s+1}\cdot f);q}\,,$$

and obtain that Z_t is a martingale, as $\mathbb{E}\left[Z_{t+1} - Z_t \mid \mathcal{F}_t\right] = 0$. Now apply the martingale concentration inequality as follows: our bounded increment is $M \leq \|f\|_\infty \binom{L}{q}$, where L is the maximal load of a bin, giving that

$$\mathbb{P}(X_t^{f;q+1} < h \ , \ \sum_{s<t} X_s^{(Q_t \cdot f);q} > 2h) \leq \exp\left(-ch/\left(\|f\|_\infty \binom{L}{q}\right)\right) \ .$$

Therefore, if

$$h > c' m L \|f\|_\infty \binom{L}{q}$$

then the above event does not occur for any f that is the product of up to L different strategies and all t except with probability $n^{-10}\mathrm{e}^{-m}$. Therefore, setting

$$h_{f;q} \triangleq c\frac{(2L)^{q+1}}{q!} m \|f\|_\infty$$

for an appropriate aboslute constant $c > 0$, we have

$$X_t^{f;q+1} \geq \frac{1}{2} \sum_{s<t} X_s^{(Q_t \cdot f);q}$$

provided that $\sum_{s<t} X_s^{(Q_t \cdot f);q} > 2^{-q} h_{f;q}$ except with probability $n^{-10}\mathrm{e}^{-m}$.

We now proceed to prove (5.4) by induction on q. For $q = 1$, notice that

$$\begin{aligned} X_t^{f;1} &= \sum_{s \leq t} \sum_i f(i) \mathbf{1}_{\{J_s = i\}} = \sum_i f(i) N_t(i) \ , \\ V_t^{f;1} &= \sum_{s \leq t} \sum_i f(i) Q_s(i) \ . \end{aligned}$$

Furthermore, as the definitions of $X_t^{f;q}$ and $Z_t^{f;q}$ also apply to the case $q = 0$, we then have

$$\begin{aligned} X_t^{f;0} &= \sum_i f(i) \ , \text{ and} \\ Z_t^{f;0} &= X_t^{f;1} - \sum_{s<t} \sum_i Q_{s+1}(i) f(i) = X_t^{f;1} - V_t^{f;1} \ . \end{aligned}$$

The requirement $V_t^{f;1} \geq c\frac{(2L)^{q+1}}{q!} m \|f\|_\infty$ precisely ensures that $V_t^{f;1} \geq h_{f;q}$. Therefore, the above discussion guarantees that $X_t^{f;1} \geq \frac{1}{2} V_t^{f;1}$ except with probability $n^{-10}\mathrm{e}^{-m}$.

It remains to establish the induction step. The induction hypothesis for q states that whenever $V_t^{f;q} \geq h_{f;q}$ we also have $X_t^{f;q} \geq 2^{-q} V_t^{f;q}$ except with

probability $n^{-10}\mathrm{e}^{-m}$. Therefore,

$$\begin{aligned}\sum_{s<t} X_s^{(Q_{s+1}\cdot f);q} &\geq 2^{-q}\sum_{s<t} V_s^{(Q_{s+1}\cdot f);q}\cdot \mathbf{1}_{\{V_s^{(Q_{s+1}\cdot f);q} > h_{Q_{s+1}\cdot f;q}\}} \\ &\geq 2^{-q}\Big(\sum_{s<t} V_s^{(Q_{s+1}\cdot f);q} - t\cdot h_{Q_{s+1}\cdot f;q}\Big) \\ &\geq 2^{-q}\Big(V_t^{f;q+1} - t\cdot c\frac{(3L)^{q+1}}{q!}m\|Q_{s+1}\cdot f\|_\infty\Big)\ , \qquad (5.6)\end{aligned}$$

where in the last inequality we applied the recursion relation (5.5). Recalling that Q_{s+1} is a strategy, the following holds for all $t \leq n/k$:

$$t\|Q_{s+1}\cdot f\|_\infty \leq t\|Q_{s+1}\|_\infty\|f\|_\infty \leq t\frac{k}{n}\|f\|_\infty \leq \|f\|_\infty\ .$$

Plugging this into (5.6) we obtain that for all $t \leq n/k$,

$$\sum_{s<t} X_s^{(Q_{s+1}\cdot f);q} \geq 2^{-q}\Big(V_t^{f;q+1} - c\frac{L^{q+1}}{q!}m\|f\|_\infty\Big)\ . \qquad (5.7)$$

It now follows that if $V_t^{f;q+1} \geq h_{f;q+1} = c\frac{(2L)^{q+2}}{(q+1)!}m\|f\|_\infty$, then in particular

$$V_t^{f;q+1} \geq 2c\frac{(2L)^{q+1}}{q!}m\|f\|_\infty = 2h_{f;q} \quad \text{for all } q \leq L-1.$$

Thus, under this assumption, (5.7) takes the following form:

$$\sum_{s<t} X_s^{(Q_{s+1}\cdot f);q} \geq 2^{-q}V_t^{f;q+1} \geq 2^{-q}h_{f;q}\ .$$

This entitles us to apply the martingale concentration result on Z_t, and obtain that except with probability $n^{-10}\mathrm{e}^{-m}$,

$$X_t^{f;q+1} \geq \frac{1}{2}\sum_{s<t} X_s^{(Q_{s+1}\cdot f);q} \geq 2^{-(q+1)}V_t^{f;q+1}\ ,$$

completing the induction step.

Summing the error probabilities over the induction steps for every $q < L$ concludes the proof of the lemma. ■

It remains to apply the above lemma to deduce the maximal load of $\Omega(\frac{\log n}{\log\log n})$ for $k = \mathrm{polylog}(n)$. Recalling that $m \leq n^{1-\delta}$ for some fixed $\delta > 0$, let $0 < \varepsilon < \delta/2$ and choose the following parameters:

$$q = (1-\varepsilon)\frac{\log(n/m)}{\log k + \log\log(n/m)}\ ,\ f = \mathbf{1}\ ,\ t = n/k\ .$$

Lemma 5.1 now gives that, either the maximal load exceeds $L = \log(n/m)$, or **whp** the following statements holds:

$$\text{If} \quad V_{n/k}^{\mathbf{1};q} \geq c\frac{(2L)^{q+1}}{q!}m \quad \text{then} \quad X_{n/k}^{\mathbf{1};q} \geq 2^{-q}\,V_{n/k}^{\mathbf{1};q}\ .$$

Notice that for the above value of q, we have $2^{-q} = (n/m)^{o(1)}$, thus showing that $V_{n/k}^{\mathbf{1};q} \geq (n/m)^{\varepsilon}$ would imply that the maximal load exceeds q **whp**. The following lemma, which provides a lower bound on $V_t^{\mathbf{1};q}$, is thus the final ingredient required for the proof of the theorem:

Lemma 5.2. *For all t, k and q, all $Q_1, \ldots, Q_t$ and any fixed $\alpha > 0$ we have*

$$V_t^{\mathbf{1};q} \geq \frac{\alpha(t-(1+\alpha)kq)^q}{(1+\alpha)n^{q-1}q!} \ .$$

Proof. Recall that

$$V_t^{\mathbf{1};q} = \sum_{s_1<\ldots<s_q\leq t} \sum_{i=1}^{n} (Q_{s_1}\cdots Q_{s_q})(i) \ ,$$

and observe that for all $i \in [n]$,

$$\sum_{\substack{s_1\leq\ldots\leq s_q\leq t \\ |\{s_1,\ldots,s_q\}|<q}} (Q_{s_1}\cdots Q_{s_q})(i) \leq (q-1)\frac{k}{n} \sum_{s_1\leq\ldots\leq s_{q-1}\leq t} (Q_{s_1}\cdots Q_{s_{q-1}})(i) \ ,$$

where we obtained an upper bound on the number of choices for q indices in $[t]$ with repetitions by selecting $q-1$ such indices and duplicating one of them. The factor of k/n results from the fact that $\|Q_s\|_\infty \leq k/n$ for all s by the definition of our strategies. Defining

$$r_i \stackrel{\triangle}{=} \sum_{s\leq t} Q_s(i) \ ,$$

it then follows that

$$\begin{aligned} V_t^{\mathbf{1};q} &\geq \frac{1}{q!}\sum_i \left(r_i^q - r_i^{q-1}\frac{kq}{n}\right) \geq \frac{1}{q!}\sum_i r_i^{q-1}\left(r_i - \frac{kq}{n}\right)\mathbf{1}_{\{r_i>(1+\alpha)kq/n\}} \\ &\geq \frac{\alpha}{(1+\alpha)q!}\sum_i r_i^q \mathbf{1}_{\{r_i>(1+\alpha)kq/n\}} \ . \end{aligned}$$

Applying Cauchy-Schwartz, we infer that

$$V_t^{\mathbf{1};q} \geq \frac{\alpha n}{(1+\alpha)q!}\left(\frac{\sum_i r_i\mathbf{1}_{\{r_i>(1+\alpha)kq/n\}}}{n}\right)^q = \frac{\alpha\Big(\sum_i r_i\mathbf{1}_{\{r_i>(1+\alpha)kq/n\}}\Big)^q}{(1+\alpha)n^{q-1}q!} \ .$$

The proof of the lemma now follows from noticing that

$$\sum_i r_i\mathbf{1}_{\{r_i\leq(1+\alpha)kq/n\}} \leq (1+\alpha)kq \ ,$$

whereas $\sum_i r_i = \sum_{s\leq t}\sum_i Q_s(i) = t$. ∎

To complete the proof using Lemma 5.2, apply this lemma for $\alpha = 1$, $t = n/k$ and $kq = n^{o(1)}$, giving that

$$V_{n/k}^{1;q} \geq \Big(\frac{1}{2} + o(1)\Big) nk^{-q}/q! \ .$$

Therefore,

$$\frac{c(2L)^{q+1}m/q!}{V_{n/k}^{\mathbf{1};q}} \leq (2+o(1))c(2L)^{q+1}mk^q/n \leq 3c(2kL)^{q+1}(m/n) \ ,$$

where the last inequality holds for any sufficiently large n. Since our choice of q is such that

$$(2kL)^{q+1} = \mathrm{e}^{(1+o(1))q(\log L + \log k)} = (n/m)^{1-\varepsilon-o(1)} \ ,$$

we get $V_{n/k}^{\mathbf{1};q} \geq (n/m)^{\varepsilon/2}$ for any large n, and so the maximal load is **whp** at least q. This concludes the proof of Theorem 2. ■

5.2. **A corollary for non-adaptive algorithms.** We end this section with a corollary of Theorem 2 for the case of non-adaptive algorithms, i.e. the strategies $Q_1, \ldots, Q_n$ are fixed ahead of time. Namely, we show that for $k = O(\frac{n \log\log n}{\log n})$ the optimal maximal load is **whp** $\Theta(\frac{\log n}{\log\log n})$, of the same order as the one for $k = 1$. Theorem 5, proved in Section 6, provides a stronger version of this result (asymptotically tight).

Corollary 5.3. *Consider the allocation problem of n balls into n bins, where each ball has k independent uniform choices. If $k = O(\frac{n \log\log n}{\log n})$, then any non-adaptive algorithm* **whp** *creates a maximal-load of at least $\Omega(\frac{\log n}{\log\log n})$. In particular, if $k \leq \frac{n \log\log n}{\log n}$ then the load is at least $(\frac{1}{2} - o(1))\frac{\log n}{\log\log n}$* **whp**.

Proof. Let $Q_1, \ldots, Q_n$ be the optimal sequence of strategies for the problem. Using definitions (5.1) and (5.2) with $f \equiv 1$, we have the following for all q:

$$X_t^{\mathbf{1};q} = \sum_i \binom{N_t(i)}{q} = \mathrm{Col}_q(t) \ , \text{ and}$$
$$V_t^{\mathbf{1};q} = \mathbb{E} X_t^{\mathbf{1};q} \ .$$

Fix $\varepsilon > 0$. Applying Lemma 5.2 with $\alpha = \varepsilon/(1-\varepsilon)$, we get

$$V_n^{\mathbf{1};q} \geq \frac{\alpha(n-(1+\alpha)kq)^q}{(1+\alpha)n^{q-1}q!} = \varepsilon\Big(1 - \frac{kq/(1-\varepsilon)}{n}\Big)^q \cdot \frac{n}{q!} \ . \tag{5.8}$$

Recalling our assumption that $k = O(\frac{n \log\log n}{\log n})$, let $C > 0$ be such that

$$k \leq C\frac{\log\log n}{\log n} n \ ,$$

and set

$$q = \frac{1-\varepsilon}{2(C \vee 1)} \cdot \frac{\log n}{\log\log n} \ .$$

This choice has $k \leq \frac{1-\varepsilon}{2}(n/q)$ and $q! \leq n^{\frac{1-\varepsilon}{2}+o(1)}$. Combining it with (5.8),

$$\begin{aligned}\mathbb{E}X_q^{\mathbf{1};q} = V_n^{\mathbf{1};q} &\geq \varepsilon \exp\left(- \frac{kq^2/(1-\varepsilon)}{n} / \Big(1 - \frac{kq/(1-\varepsilon)}{n}\Big)\right)\frac{n}{q!} \\ &\geq \varepsilon \exp(-q)\frac{n}{n^{(1-\varepsilon)/2+o(1)}} = n^{(1+\varepsilon)/2-o(1)} > n^{1/2+\varepsilon/4} ,\end{aligned}$$

where the last inequality holds for any sufficiently large n. As long as the maximal load at time t does not exceed L, we have that

$$0 \leq X_{t+1}^{\mathbf{1};q} - X_t^{\mathbf{1};q} \leq \binom{L}{q} .$$

Hence, using the standard Azuma inequality on Doob's martingale for $X_t^{\mathbf{1};q}$, combined with an application of Optional Stopping for the first time the maximal load exceeds l, the following holds for $t = n$ and $\lambda = n^{1/2+\varepsilon/4}$:

$$\begin{aligned}\mathbb{P}\left(\mathrm{Col}_q(n) < \lambda\right) &\leq \mathbb{P}\left(|X_n^{\mathbf{1};q} - \mathbb{E}X_n^{\mathbf{1};q}| > \lambda\right) \\ &\leq \exp\left(- \frac{\lambda^2}{2n\binom{L}{q}^2}\right) = \exp(-\Omega(n^{\varepsilon/4-o(1)})) .\end{aligned}$$

We deduce that **whp** the maximal load is at least $q = \Omega\left(\frac{\log n}{\log\log n}\right)$. ■

6. Tight bounds for non-adaptive allocations

In this section we present the proof of Theorem 5. Throughout the proof we assume, whenever this is needed, that n is sufficiently large. To simplify the presentation we omit all floor and ceiling signs whenever these are not crucial. We need the following lemma.

Lemma 6.1. *Let $p_1, p_2, \ldots, p_n$ be reals satisfying $0 \leq p_i \leq \frac{\log n}{\log\log n}$ for all i, such that $\sum_{i=1}^n p_i \geq 1-\varepsilon$, where $0 \leq \varepsilon \leq 1$ (and ε may be a function of n). Let $X_1, X_2, \ldots, X_n$ be independent indicator random variables, where $\mathbb{P}(X_i = 1) = p_i$ for all i, and put $X = \sum_{i=1}^n X_i$. Then*

$$\mathbb{P}\Big(X \geq (1-\varepsilon)\frac{\log n}{\log\log n}\Big) \geq \frac{1}{n^{1-\varepsilon}}.$$

Proof. Without loss of generality assume that $p_1 \geq p_2 \geq \cdots \geq p_n$. Define a family of k pairwise disjoint blocks $B_1, B_2, \ldots, B_k \subset \{1, 2, \ldots, n\}$, where $k \geq (1-\varepsilon)\frac{\log n}{\log\log n}$ so that for each i, $1 \leq i \leq k$,

$$\frac{2}{\log n} \leq \sum_{j \in B_i} p_j \leq \frac{\log\log n}{\log n}.$$

This can be easily done greedily; the first block consists of the indices $1, 2, \ldots, r$ where r is the smallest integer so that $\sum_{j=1}^r p_j \geq \frac{2}{\log n}$. Note that it is possible that $r = 1$, and that since the sequence p_j is monotone

decreasing, $\sum_{j=1}^{r} p_j \leq \frac{\log\log n}{\log n}$. Assuming we have already partitioned the indices $\{1,\ldots,r\}$ into blocks, and assuming we still do not have $(1-\varepsilon)\frac{\log n}{\log\log n}$ blocks, let the next block be $\{r+1,\ldots,s\}$ with s being the smallest integer exceeding r so that $\sum_{j=r+1}^{s} p_j \geq \frac{2}{\log n}$. Note that if $p_{r+1} \geq \frac{2}{\log n}$ then $s = r+1$, that is, the block consists of a single element, and otherwise $\sum_{j=r+1}^{s} p_j < \frac{4}{\log n} < \frac{\log\log n}{\log n}$. Thus, in any case the sum above is at least $\frac{2}{\log n}$ and at most $\frac{\log\log n}{\log n}$. Since the total sum of the reals p_j is at least $1-\varepsilon$ this process does not terminate before generating $k \geq (1-\varepsilon)\frac{\log n}{\log\log n}$ blocks, as needed.

Fix a family of $k = (1-\varepsilon)\frac{\log n}{\log\log n}$ blocks as above. Note that for each fixed block B_i in the family, the probability that $\sum_{j\in B_i} X_j \geq 1$ is at least

$$\sum_{j\in B_i} p_j - \sum_{j,q\in B_i, j<q} p_j p_q \geq \sum_{j\in B_i} p_j - \frac{1}{2}(\sum_{j\in B_i} p_j)^2 \geq \frac{2}{\log n} - \frac{2}{\log^2 n} > \frac{1}{\log n}.$$

It thus follows that the probability that for each of the k blocks B_i in the family $\sum_{j\in B_i} X_j \geq 1$ is at least $(\frac{1}{\log n})^k = \frac{1}{n^{1-\varepsilon}}$, completing the proof of the lemma. ■

Proof of Theorem 5. We begin with the proof of Part (i).

Let $Q_1, Q_2, \ldots, Q_n$ be the strategies defining a non-adaptive algorithm, where Q_t is the strategy for placing ball number t. As mentioned in the previous sections, each such strategy Q_t gives rise to a probability distribution $(p_{it} : 1 \leq i \leq n)$ on the bins, where p_{it} is the probability that the ball in round t will be placed in bin number i. Clearly

$$p_{it} \leq k/n = \frac{\log\log n}{\log n} \quad \text{for all } i \text{ and } t\ ,$$

and

$$\sum_{1\leq i\leq n} p_{it} = 1 \quad \text{for all } t\ .$$

The sum of entries of each column of the n by n matrix p_{it} is 1, and hence the total sum of its entries is n. If it contains a row i so that the sum of entries in this row is at least, say, $\log n$, then the expected number of balls in bin number i by the end of the process is $\sum_{t=1}^{n} p_{it} \geq \log n$. As the variance is

$$\sum_{t+1}^{n} p_{it}(1-p_{it}) \leq \sum_{t=1}^{n} p_{it}\ ,$$

it follows by Chebyshev's Inequality, (or by Hoeffding's Inequality) that with high probability the actual number of balls placed in bin number i exceeds $\frac{\log n}{2} > \frac{\log n}{\log\log n}$, showing that in this case the desired result holds.

We thus assume that the sum of entries in each row is at most $\log n$. As the average sum in a row is 1, there is a row whose total sum is at least 1. Omit this row, and note that since its total sum is at most $\log n$, the sum of all remaining entries of the matrix is still at least $n - \log n$, and hence the average sum of a row in it is at least $\frac{n-\log n}{n-1} > 1 - \frac{\log n}{n}$. Therefore there is another row of total sum at least this quantity. Omitting this row and proceeding in this manner we can define a set of rows so that the sum in each of them is large. Note that as long as we defined at most $\frac{n}{\log^2 n}$ rows, the total sum of the remaining elements of the matrix is still at least $n - \frac{n}{\log n}$, and hence there is another row of total sum at least $1 - \frac{1}{\log n}$. We have thus shown that there is a set I of $\frac{n}{\log^2 n}$ rows, so that

$$\sum_{t=1}^{n} p_{it} \geq 1 - \frac{1}{\log n} \quad \text{for each } i \in I \ .$$

By Lemma 6.1 (with, say, $\varepsilon = \frac{4 \log\log n}{\log n}$), for each $i \in I$, the probability that in bin number i there are at least $(1 - \frac{4 \log\log n}{\log n}) \frac{\log n}{\log\log n}$ balls is at least $\frac{\log^4 n}{n}$. As these events for distinct values of i are negatively correlated, a simple argument similar to the one in [1, Lemma 2.4] shows that the probability that none of these events holds is at most the product of the probabilities that all these events fail, which is at most

$$\left(1 - \frac{\log^4 n}{n}\right)^{n/\log^2 n} \leq \mathrm{e}^{-\log^2 n}.$$

This completes the proof of Part (i).

The proof of Part (ii), when $k = n/2$, is more complicated. Note that here it is not enough to assume that in the strategy Q_t each probability p_{it} is at most $k/n = \frac{1}{2}$, since if Q_t assigns probability $\frac{1}{2}$ to $i = t$ and $i = (t+1)$ (with the indices reduced modulo n), the maximum load will be at most 2. However, it is easy to see that in fact each strategy Q_t is more restricted. Indeed, the total probability it can assign to r bins does not exceed

$$1 - \frac{\binom{n-r}{k}}{\binom{n}{k}},$$

which is roughly $1 - 2^{-r}$. This has to be used in the proof. The upper bound is obtained by the natural algorithm which places the ball in round t in the first possible bin (among the k given choices) that follows bin number t in the cyclic order of the bins. The details will be given in the full version of this manuscript. ■

Acknowledgments

We thank Yossi Azar and Allan Borodin for useful discussions. We also thank Itai Benjamini for proposing the problem of balanced allocations with limited memory.

References

[1] N. Alon, B. Bollobás, J. H. Kim, and V. H. Vu, *Economical covers with geometric applications*, Proc. London Math. Soc. **86** (2003), no. 2, 273–301.

[2] N. Alon and J. H. Spencer, *The probabilistic method*, 3rd ed., John Wiley & Sons Inc., Hoboken, NJ, 2008.

[3] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, *Balanced allocations*, SIAM J. Comput. **29** (1999), no. 1, 180–200 (electronic).

[4] I. Benjamini and Y. Makarychev, *Balanced allocation: memory performance tradeoffs*. preprint.

[5] D. L. Burkholder, *Sharp inequalities for martingales and stochastic integrals*, Astérisque (1988), no. 157-158, 75–94. Colloque Paul Lévy sur les Processus Stochastiques (Palaiseau, 1987).

[6] W. Feller, *An introduction to probability theory and its applications. Vol. I*, Third edition, John Wiley & Sons Inc., New York, 1968.

[7] G. H. Gonnet, *Expected length of the longest probe sequence in hash code searching*, J. Assoc. Comput. Mach. **28** (1981), no. 2, 289–304.

[8] N. L. Johnson and S. Kotz, *Urn models and their application*, John Wiley & Sons, New York-London-Sydney, 1977. An approach to modern discrete probability theory; Wiley Series in Probability and Mathematical Statistics.

[9] C. McDiarmid, *Concentration*, Probabilistic methods for algorithmic discrete mathematics, Algorithms Combin., vol. 16, Springer, Berlin, 1998, pp. 195–248.

[10] M. Mitzenmacher, A. W. Richa, and R. Sitaraman, *The power of two random choices: a survey of techniques and results*, Handbook of randomized computing, Vol. I, II, Comb. Optim., vol. 9, Kluwer Acad. Publ., Dordrecht, 2001, pp. 255–312.

Noga Alon
School of Mathematics, Tel Aviv University, Tel Aviv, 69978, Israel.
E-mail address: nogaa@tau.ac.il

Ori Gurel-Gurevich
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA.
E-mail address: origurel@microsoft.com

Eyal Lubetzky
Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA.
E-mail address: eyal@microsoft.com