

Chapter 20

The power of two choices

By Sarel Har-Peled, April 26, 2022[Ⓓ]

The Peace of Olivia. How sweet and peaceful it sounds! There the great powers noticed for the first time that the land of the Poles lends itself admirably to partition.

The tin drum, Gunter Grass

Consider the problem of throwing n balls into n bins. It is well known that the maximum load is $\Theta(\log n / \log \log n)$ with high probability. Here we show that if one is allowed to pick d bins for each ball, and throw it into the bin that contains less balls, then the maximum load of a bin decreases to $\Theta(\log \log n / \log d)$. A variant of this approach leads to maximum load $\Theta((\log \log n) / d)$.

As a concrete example, for $n = 10^9$, this leads to maximum load 13 in the regular case, compared to maximum load of 4, with only two-choices – see [Figure 20.1](#).

20.1. Balls and bins with many rows

20.1.1. The game

Consider throwing n balls into n bins. Every bin can contain a single ball. As such, as we throw the balls, some balls would be rejected because their assigned bin already contains a ball. We collect all the rejected balls, and throw them again into a second row of n bins. We repeat this process till all the balls had found a good and loving home (i.e., an empty bin). How many rows one needs before this process is completed?

20.1.2. Analysis

Lemma 20.1.1. *Let $m = \alpha n$ balls be thrown into n bins. Let Y_{end} the number of bins that are not empty in the end of the process (here, we allow more than one ball into a bin).*

(A) *For $\alpha \in (0, 1]$, we have $\mu = \mathbb{E}[Y_{\text{end}}] \geq (m - \alpha) \exp(-\alpha)$.*

(B) *If $\alpha \geq 1$, then $\mu = \mathbb{E}[Y_{\text{end}}] \geq n(1 - \exp(-\alpha))$.*

(C) *We have $\mathbb{P}[|Y_{\text{end}} - \mu| > \sqrt{3cm \log n}] \leq 1/n^c$.*

Proof: (A) The probability of the i th ball to be the first ball in its bin, is $(1 - \frac{1}{n})^{i-1}$. To see this we use backward analysis – throw in the i th ball, and now throw in the earlier $i - 1$ balls. The probability that none of the earlier balls hit the same bin as the i th ball is as stated. Now, the expected number of non-empty bins is the number of balls that are first in their bins, which in turn is

$$\mu = \sum_{i=0}^{m-1} \left(1 - \frac{1}{n}\right)^i \geq (m - \alpha)(1 - 1/n)^{m-\alpha} = (m - \alpha)(1 - 1/n)^{\alpha(n-1)} \geq (m - \alpha) \exp(-\alpha) \geq \frac{m - \alpha}{e},$$

[Ⓓ]This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

using $m = \alpha n \leq n$, and $(1 - 1/n)^{n-1} \geq 1/e$, see [Lemma 20.6.3](#).

(B) We repeat the above analysis from the point of view of the bin. The probability of a bin to be empty is $(1 - 1/n)^{\alpha n}$. As such, we have that

$$\mu = \mathbb{E}[Y_{\text{end}}] = n(1 - (1 - 1/n)^{\alpha n}) \geq n(1 - \exp(-\alpha)),$$

using $1 - 1/n \leq \exp(-1/n)$.

(C) Let X_i be the index of the bin the i th ball picked. Let $Y_i = \mathbb{E}[Y_{\text{end}} \mid X_1, \dots, X_i]$. This is a Doob martingale, with $|Y_i - Y_{i-1}| \leq 1$. As such, [Azuma's inequality](#) implies, for $\lambda = \sqrt{3cm \ln n}$, that

$$\mathbb{P}[|Y_{\text{end}} - \mathbb{E}[Y_{\text{end}}]| \geq \lambda] \leq 2 \exp(-\lambda^2/2m) \leq 1/n^c. \quad \blacksquare$$

Remark. The reader might be confused by cases (A) and (B) of [Lemma 20.1.1](#) for $\alpha = 1$, as the two lower bounds are different. Observe that (A) is loose if α is relatively large and close to 1.

Back to the problem. Let $\alpha_1 = 1$ and $n_1 = \alpha_1 n$. For $i > 1$, inductively, assume that numbers of balls being thrown in the i th round is

$$n_i = \alpha_i n + O(\sqrt{\alpha_{i-1} n \log n}).$$

By [Lemma 20.1.1](#), with high probability, the number of balls stored in the i th row is

$$s_i = n_i \exp(-\alpha_i) \pm O(\sqrt{n_i \log n}).$$

As such, as long as the first term is significantly large than the second term, we have that $s_i = n\alpha_i \exp(-\alpha_i)(1 \pm o(1))$. For the time being, let us ignore the $o(1)$ term. We have that

$$n_{i+1} = n_i - s_i = n(\alpha_i - \alpha_i \exp(-\alpha_i)) \leq n(\alpha_i - \alpha_i(1 - \alpha_i)) = n\alpha_i^2,$$

since $\exp(-\alpha_i) \geq 1 - \alpha_i$.

Definition. For a number $x > 0$, we use $\lg x = \log_2 x$.

Observation 20.1.2. Consider the sequence $\alpha_1 = 1$, $c = \alpha_2 = 1 - 1/e$, and $\alpha_{i+1} = \alpha_i^2$, for $i > 2$. We have that $\alpha_{i+1} = c^{2^{i-2}}$. In particular, for

$$\Delta = 3 + \lg \log_{1/c} n = 3 + \lg \frac{\lg n}{\lg(1/c)} = 3 + \lg \lg n - \lg \lg \frac{1}{1 - 1/e} \leq 3 + \lg \lg n.$$

we have that $\alpha_\Delta = c^{2^{\Delta-2}} < 1/n$.

The above observation almost implies that we need Δ rows. The problem is that the above calculations (i.e., the high probability guarantee in [Lemma 20.1.1](#)) breaks down when $n_i = O(\log n)$ – that is, when $\alpha_i = O((\log n)/n)$. However, if one throws in $O(\log n)$ balls into n bins, the probability of a single collision is at most $O((\log n)^2/n)$. In particular, this implies that after roughly additional c rows, the probability of any ball left is $\leq 1/n^c$.

The above argumentation, done more carefully, implies the following – we omit the details because (essentially) the same analysis for a more involved case is done next (the lower bound stated follows also from the same argumentation).

Theorem 20.1.3. Consider the process of throwing n balls into n bins in several rounds. Here, a ball that can not be placed in a round, because their chosen bin is already occupied, are promoted to the next round. The next round throws all the rejected balls from the previous round into a new row of n empty bins. This process, with high probability, ends after $M = \lg \lg n + \Theta(1)$ rounds (i.e., after M rounds, all balls are placed in bins).

20.1.3. With only d rows

Lemma 20.1.4. For $\alpha \in (0, 1/4]$, let $\gamma_1 = \alpha$, and $\gamma_i = 2\gamma_{i-1}^2$. We have that $\gamma_{d+1} \leq \alpha^{(2^d+1)/2}$.

Proof: The proof, minimal as it may be, is by induction:

$$\gamma_{i+1} = 2\gamma_i^2 \leq 2\left(\alpha^{(2^{i-1}+1)/2}\right)^2 = 2\alpha^{(2^i+2)/2} \leq \alpha^{(2^i+1)/2},$$

since $2\sqrt{\alpha} \leq 1$. ■

Lemma 20.1.5. Let $m = \alpha n$ balls be thrown into n bins, with d rows, where $\alpha > 0$. Here every bin can contain only a single ball, and if inserting the ball into i th row failed, then we throw it in the next row, and so on, till it finds an empty bin, or it is rejected because it failed on the d th row. Let $Y(d, n, m)$ be the number of balls that did not get stored in this matrix of bins. We have

(A) For a constant $\alpha < 1/4$, we have $Y(d, n, \alpha n) \leq n\alpha^{(2^d+1)/2}$, with high probability.

(B) We $\mathbb{E}[Y(d, n, dn)] = O(n \log d)$.

(C) For a constant $c > 1$, we have $\mathbb{E}[Y(d, n, cn \log d)] = n/e^{-d/2}$, assuming d is sufficiently large.

Proof: (A) By [Lemma 20.1.1](#), in expectation, at least $s_1 = n\alpha \exp(-\alpha)$ balls are placed in the first row. As such, in expectation $n_2 = n\alpha(1 - \exp(-\alpha)) \leq n\alpha^2$ balls get thrown into the second row. Using Chernoff inequality, we get that $n_2 \leq 2\alpha^2 n$, with high probability. Setting $\gamma_1 = \alpha$, and $\gamma_i = 2\gamma_{i-1}^2$, we get the claim via [Lemma 20.1.4](#).

(B) As long as we throw $\Omega(n \log d)$ balls into a row, we expect by [Lemma 20.1.1](#) that at least $n(1 - 1/d^{O(1)})$ balls to get stored in this row. As such, let $D = O(\log d)$, and observe that the first $d - D$ rows in expectation contains $n(d - D)(1 - 1/d^{O(1)})$ balls. This implies that only $O(Dn)$ are not stored in these first $d - D$ rows, which implies the claim.

(C) Break the d rows into two groups. The first group of size

$$D = \lceil (c \log d - 1)/(1 - 1/e) \rceil + 1 = O(\log d),$$

and the second group is the remaining rows. As long as the number of balls arriving to a row is larger than n , we expect at least $n(1 - 1/e)$ of them to be stored in this row. As such, after the first D rows, we expect the number of remaining balls to be $\leq n$. Indeed, if we have i such rows, then the expected number of balls moving on to the $(i + 1)$ th row is at most

$$n_{i+1} = cn \log d - in(1 - 1/e).$$

Solving for $n_{i+1} \leq n$, we have $cn \log d - in(1 - 1/e) \leq n \implies i(1 - 1/e) \geq c \log d - 1 \implies i \geq (c \log d - 1)/(1 - 1/e) \geq D - 1$. As such, $n_D \leq n$, for $i \geq D$.

The same argumentation implies that the number of balls arriving to the $D+i$ row, in expectation, is at most n/e^i . In particular, we get that the number of balls failed to be placed is at most $n/e^{d-D} \leq n/e^{d/2}$. ■

20.2. The power of two choices

Making d choices. Let us throw n balls into n bins. For each ball, we first pick randomly $d \geq 2$ bins, and place the ball in the bin (among these d bins) that currently contains the smallest number of balls (here, a bin might contain an arbitrary number of balls). If there are several bins with the same minimum number of bins, we resolve it arbitrarily.

Here, we will show the surprising result that the maximum number of balls in any bin is bounded by $O\left(\frac{\log \log n}{\log d}\right)$ with high probability in the end of this process. For $d = 1$, which is the regular balls into bins setting, we already seen that this quantity is $\Theta\left(\frac{\log n}{\log \log n}\right)$, so this result is quite surprising.

20.2.1. Upper bound

Definition 20.2.1. The *load* of a bin is the number of balls in it. The *height* of a ball, is the load of the bin it was inserted into, just after it was inserted.

Some notations:

- (A) β_i : An upper bound on the number of bins that have load at least i by the end of the process.
- (B) $h(i)$: The height of the i th ball.
- (C) $\sqcup_{\geq i}(t)$: Number of bins with load at least i at time t .
- (D) $\circledast_{\geq i}(t)$: Number of balls with height at least i at time t .

Observation 20.2.2. $\sqcup_{\geq i}(t) \leq \circledast_{\geq i}(t)$.

Let $\uplus_{\geq i} = \sqcup_{\geq i}(n)$ be the number of bins, in the end of the process, that have load $\geq i$.

Observation 20.2.3. *Since every bin counted in $\uplus_{\geq i}$ contains at least i balls, and there are n balls, it follows that $\uplus_{\geq i} \leq n/i$. In particular, we have $\uplus_{\geq 4} \leq n/4$.*

Lemma 20.2.4. *Let $\beta_1 = n, \beta_2 = n/2, \beta_3 = n/3$, and $\beta_4 = n/4$, and let $\beta_{i+1} = 2n(\beta_i/n)^d$, for $i \geq 4$. Let I be the last iteration, such that $\beta_I \geq 16c \ln n$, where $c > 1$ is an arbitrary constant. Then, with probability $\geq 1 - 1/n^c$, we have that*

- (A) $\uplus_{\geq i} \leq \beta_i$, for $i = 1, \dots, I$.
- (B) $\uplus_{\geq I+1} \leq c' \log n$, for some constant c' .
- (C) For $j > 0$, and any constant $\varepsilon > 0$, we have $\mathbb{P}[\uplus_{\geq I+1+j} > 0] \leq O(1/n^{(d-1-\varepsilon)j})$.
- (D) With probability $\geq 1 - 1/n^c$, the maximum load of a bin is $I + O(c)$.

Proof: (A) The claim for $i = 1, 2, 3, 4$ follows readily from **Observation 20.2.3**.

Let \mathcal{B}_i be the bad event that $\uplus_{\geq i} > \beta_i$, for $i = 1, \dots, n$. The following analysis is conditioned on none of these bad events happening. Let $\mathcal{G}_k = \cap_{i=1}^k \overline{\mathcal{B}_i}$ be the good event. Let Y_i be an indicator variable that is one $\iff h(i) \geq i + 1$ conditioned on \mathcal{G}_{i-1} (for clarity, we omit mentioning this conditioning explicitly). We have that

$$\tau_j = \mathbb{P}[Y_j = 1] \leq p_i \quad \text{for} \quad p_i = (\beta_i/n)^d,$$

as all d probes must hit bins of height at least i , and there are at most β_i such bins. This readily implies that $\mathbb{E}[\circledast_{\geq i+1}(n)] \leq p_i n$. The variables Y_1, \dots, Y_n are not independent, but consider a variable Y'_j that is 1 if $Y_j = 1$, or if $Y_j = 0$, then Y'_j is 1 with probability $p - \tau_j$. Clearly, the variables Y'_1, \dots, Y'_n are independent, and $\sum_i Y'_i \geq \sum_i Y_i$. For $i < I$, setting

$$\beta_{i+1} = 2np_i = 2n(\beta_i/n)^d,$$

we have, by **Chernoff's inequality**, that

$$\begin{aligned} \alpha_{i+1} = \mathbb{P}[\mathcal{B}_{i+1}] &= \mathbb{P}[\circledast_{\geq i+1}(n) > \beta_{i+1}] = \mathbb{P}[\circledast_{\geq i+1}(n) > 2np_i] \leq \mathbb{P}\left[\sum_i Y'_i > (1+1)np_i\right] \\ &\leq \exp(-np_i/4) = \exp(-\beta_{i+1}/8) < 1/n^{2c}. \end{aligned}$$

(B) We have $\beta_{I+1} \leq 16c \log n$. Setting $\Delta = 2e \cdot 16c \log n$, and conditioning on the good event \mathcal{G}_1 , consider the sequence Y'_1, \dots, Y'_n as above, where the Y_i is the indicator that the i th ball has height $\geq I+1$. Arguing as above, for $Y' = \sum_i Y'_i$, we have $\mathbb{E}[Y'] \leq \beta_{I+1}$. As such, we have

$$\mathbb{P}[\uplus_{\geq I+1} > \Delta] \leq \mathbb{P}[\circledast_{\geq I+1}(n) > \Delta] \leq \mathbb{P}\left[Y' > \frac{\Delta}{\mathbb{E}[Y']} \mathbb{E}[Y']\right] \leq 2^{-\Delta} \leq \frac{1}{n^c},$$

by [Lemma 20.6.4](#), as $\mathbb{E}[Y'] \leq \beta_{I+1}$, and $\Delta/\beta_{I+1} > 2e$.

As for the conditioning used in the above, we have that

$$\mathbb{P}[\mathcal{G}_{I+1}] = \prod_{\ell=4}^{I+1} \mathbb{P}\left[\overline{\mathcal{B}_{\ell+1}} \mid \bigcap_{k=1}^{\ell} \overline{\mathcal{B}_k}\right] = \prod_i (1 - \alpha_i) \geq 1 - 1/n^{c-1},$$

since $I \leq n$.

(C) Observe that $\sqcup_{\geq i+1}(n) \leq \sqcup_{\geq i}(n)$. As such, for all $j > 0$, we have that $\sqcup_{\geq I+1+j}(n) \leq \mathbb{S}_{\geq I+1}(n) \leq \Delta = 2e \cdot 16c \log n$, by (B). As such, we have

$$\mathbb{E}\left[\mathbb{S}_{\geq I+1+j}(n)\right] \leq n(\Delta/n)^d = O(\log^d n/n^{d-1}) = O(1/n^{d-1-\varepsilon}) \ll 1,$$

for $\varepsilon > 0$ an arbitrary constant, and n sufficient large. Using Markov's inequality, we get that $q = \mathbb{P}\left[\mathbb{S}_{\geq I+1+j}(n) \geq 1\right] = O(1/n^{d-1-\varepsilon})$. The probability that the first j such rounds fail (i.e., that $\mathbb{S}_{\geq I+1+j}(n) > 0$) is at most q^j , as claimed.

(D) This follows immediately by picking $\varepsilon = 1/2$, and then using (C) with $j = O(c)$. ■

Lemma 20.2.5. *For $i = 4, \dots, I$, we have that $\beta_i \leq n/2^{d^{i-4}+1}$.*

Proof: The proof is by induction. For $i = 4$, we have $\beta_4 \leq n/4$, as claimed. Otherwise, we have

$$\beta_{i+1} = 2n(\beta_i/n)^d \leq 2n\left(1/2^{d^{i-4}+1}\right)^d = n/2^{d^{i+1-4}+d-1} \leq n/2^{d^{i+1-4}+1}. \quad \blacksquare$$

Theorem 20.2.6. *When throwing n balls into n bins, with d choices, with probability $\geq 1 - 1/n^{O(1)}$, we have that the maximum load of a bin is $O(1) + \frac{\lg \lg n}{\lg d}$*

Proof: By [Lemma 20.2.4](#), with the desired probability the β_i s bound the load in the bins for $i \leq I$. By [Lemma 20.2.5](#), it follows that for $I = O(1) + \frac{\lg \lg n}{\lg d}$, we have that $\beta_I \leq o(\log n)$. Thus giving us the desired bound. ■

It is not hard to verify that our upper bounds (i.e., β_i) are not too badly off, and as such the maximum load in the worst case is (up to additive constant) the same. We state the result without proof.

Theorem 20.2.7. *When throwing n balls into n bins, with d choices (where the ball is placed with the bin with the least load), with probability $\geq 1 - o(1/n)$, we have that the maximum load of a bin is at least $\frac{\lg \lg n}{\lg d} - O(1)$.*

20.2.2. Applications

As a direct application, we can use this approach for open hashing, where we use two hash functions, and place an element in the bucket of the hash table with fewer elements. By the above, this improves the worst case search time from $O(\log n / \log \log n)$ to $O(\log \log n)$. This comes at the cost of doubling the time it takes to do lookup on average.

20.2.3. The power of restricted d choices: Always go left

The always go left rule. Consider throwing a ball into n bins (which might already have some balls in them) as follows – you pick uniformly a number $X_i \in \llbracket n/d \rrbracket$, and you try locations Y_1, \dots, Y_d , where $Y_j = X_j + j(n/d)$, for $j = 1, \dots, d$. Let L_j be the load of bin Y_j , for $j = 1, \dots, d$, and let $L = \min_j L_j$ be the minimum load of any bin. Let τ be the minimum index such that $L_j = L$. We throw the ball into Y_τ .

What the above scheme does, is to partition the n bins into d groups each of size n/d , placed from left to right. We pick a bin uniformly from each group, and always throw the ball in the leftmost location that realizes the minimum load.

The following proof is informal for the sake of simplicity.

Theorem 20.2.8. *When throwing n balls into n bins, using the always-go-left rule, with d groups of size n/d , the maximum load of a bin is $O(1) + \frac{\log \log n}{d}$, with high probability.*

Proof: (Sketch.) We consider each of the d groups to be a row in the matrix being filled. So each row has n/d entries, and there are d rows. We can now think about the above algorithm as first trying to place the ball in the first row (if there is an empty bin), otherwise, trying the new row and so on. If all the d locations are full, in the row filling game we fail to place this ball. By [Lemma 20.1.5 \(B\)](#), we have that the number of unplaced balls is $\mathbb{E}[Y(d, n/d, (n/d)d)] = O((n/d) \log d)$. Thus, we have that the number of balls that get placed as the first ball in their bin is

$$\geq n \left(1 - \frac{O(\log d)}{d} \right),$$

and the height of these balls is one.

We now use the same argumentation for balls of height 2 – [Lemma 20.1.5 \(C\)](#) implies that at most $dn/e^{-d/2}$ balls have height strictly larger than 2.

[Lemma 20.1.5 \(A\)](#) implies that now we can repeat the same analysis as the power of two choices, the critical difference is that every one of the d groups, behaves like a separate height. Since there are $O(\log \log n)$ maximum height in the regular analysis, this implies that we get $O((\log \log n)/d)$ maximum load, with high probability. ■

20.3. Avoiding terrible choices

Interestingly, one can prove that two choices are not really necessary. Indeed, consider the variant where the i th ball randomly chooses a random location r_i . The ball then is placed in the bin with least load among the bins r_i and r_{i-1} (the first ball inspects only a single bin – r_1). It is not difficult to show that the above analysis applies in this settings, and the maximum load is $O(\log \log n)$ – despite making only n choices for n balls. Intuitively, what is going on is that the power of two choices lies in the ability to avoid following a horrible, no good, terrible choice, by having an alternative. This alternative choice does not have to be quite of the same quality as the original choice – it can be stolen from the previous ball, etc.

20.4. Escalated choices

A variant that seems to work even better in practice, is the following *escalated choices* algorithm: The idea is to try more than one bin only if you need to. To this end, try a random bin. If it is empty,

# balls in bin	Regular	2-choices	2-choices+go left
0	369,899,815	240,525,897	228,976,604
1	365,902,266	528,332,061	546,613,797
2	182,901,437	221,765,420	219,842,639
3	61,604,865	9,369,389	4,566,915
4	15,760,559	7,233	45
5	3,262,678		
6	568,919		
7	86,265		
8	11,685		
9	1,347		
10	143		
11	17		
12	2		
13	2		

Figure 20.1: Simulation of the three schemes described here. This was done with $n = 1,000,000,000$ balls thrown into n bins. Since $\log \log n$ is so small (i.e., ≈ 3 in this case, there does not seem to be any reasonable cases where there is a significant difference between d -choices and the go-left variant. In the simulations, the *go-left* variant always has a somewhat better distribution, as shown above.

then the algorithm stores the ball in it. Otherwise, the algorithm tries harder. In the j th iteration, for $j > 1$, the algorithm picks a random location. If any of the j locations have load $< \lceil j/2 \rceil$, then the algorithm places the ball in the min-load bin among these. Otherwise, the algorithm continues to the next iteration.

Experiments show that on average, this algorithm only probes 1.96 bins per ball (thus, making less probes than 2-choices). In this settings, the experiments show that 4-choices with move left do better, but if one use the threshold $< \lceil j/3 \rceil$, then the average number of probes is 2.30179, while having again a better performance. The intuition is that a sequence of really bad choices are rare, and one can afford to try harder in such cases to get out of them.

A theoretical analysis of this variant should be interesting.

(I “invented” this variant, but it might already be known.)

20.5. Bibliographical notes

The multi-row balls into bins (Section 20.1) is from the work by Broder and Karlin [BK90]. The power of two choices (Section 2.2) is from Azar *et al.* [ABKU99].

The restricted d choices structure, the always go-left rule, described in Section 20.2.3, is from [Vöc03].

20.6. From previous lectures

Theorem 20.6.1 (Azuma’s Inequality - Stronger Form). *Let X_0, X_1, \dots , be a martingale sequence such that for each k , $|X_k - X_{k-1}| \leq c_k$, where c_k may depend on k . Then, for all $t \geq 0$, and any $\lambda > 0$,*

we have

$$\mathbb{P}[|X_t - X_0| \geq \lambda] \leq 2 \exp\left(-\frac{\lambda^2}{2 \sum_{k=1}^t c_k^2}\right).$$

Lemma 20.6.2. *Let X_1, \dots, X_n be n independent Bernoulli trials, where $\mathbb{P}[X_i = 1] = p_i$, and $\mathbb{P}[X_i = 0] = 1 - p_i$, for $i = 1, \dots, n$. Let $X = \sum_{i=1}^n X_i$, and $\mu = \mathbb{E}[X] = \sum_i p_i$. For $\delta \in (0, 4)$, we have*

$$\mathbb{P}[X > (1 + \delta)\mu] < \exp(-\mu\delta^2/4),$$

Lemma 20.6.3. *For any positive integer n , we have:*

- (i) $(1 + 1/n)^n \leq e$.
- (ii) $(1 - 1/n)^{n-1} \geq e^{-1}$.
- (iii) $n! \geq (n/e)^n$.
- (iv) For any $k \leq n$, we have: $\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{ne}{k}\right)^k$.

Lemma 20.6.4. *Let X_1, \dots, X_n be n independent Bernoulli trials, where $\mathbb{P}[X_i = 1] = p_i$, and $\mathbb{P}[X_i = 0] = 1 - p_i$, for $i = 1, \dots, n$. Let $X = \sum_{i=1}^n X_i$, and $\mu = \mathbb{E}[X] = \sum_i p_i$. For $\delta > 2e - 1$, we have $\mathbb{P}[X > (1 + \delta)\mu] < 2^{-\mu(1+\delta)}$.*

References

- [ABKU99] Y. Azar, A. Broder, A. Karlin, and E. Upfal. *Balanced allocations*. *SIAM Journal on Computing*, 29(1): 180–200, 1999.
- [BK90] A. Z. Broder and A. R. Karlin. *Multilevel adaptive hashing*. *Proc. 1th ACM-SIAM Sympos. Discrete Algs. (SODA)*, 43–53, 1990.
- [Vöc03] B. Vöcking. *How asymmetry helps load balancing*. *J. ACM*, 50(4): 568–589, 2003.