Balanced Allocations with Incomplete Information: The Power of Two Queries

Dimitrios Los
Department of Computer Science & Technology, University of Cambridge, UK

Thomas Sauerwald
Department of Computer Science & Technology, University of Cambridge, UK

Abstract

We consider the allocation of m balls into n bins with incomplete information. In the classical Two-Choice process a ball first queries the load of *two* randomly chosen bins and is then placed in the least loaded bin. In our setting, each ball also samples two random bins but can only estimate a bin's load by sending *binary queries* of the form "Is the load at least the median?" or "Is the load at least 100?".

For the lightly loaded case $m = \mathcal{O}(n)$, Feldheim and Gurel-Gurevich (2021) showed that with one query it is possible to achieve a maximum load of $\mathcal{O}(\sqrt{\log n / \log\log n})$, and they also pose the question whether a maximum load of $m/n + \mathcal{O}(\sqrt{\log n/\log\log n})$ is possible for any $m = \Omega(n)$. In this work, we resolve this open problem by proving a lower bound of $m/n + \Omega(\sqrt{\log n})$ for a fixed $m = \Theta(n\sqrt{\log n})$, and a lower bound of $m/n + \Omega(\log n/\log\log n)$ for some m depending on the used strategy.

We complement this negative result by proving a positive result for multiple queries. In particular, we show that with only *two* binary queries per chosen bin, there is an oblivious strategy which ensures a maximum load of $m/n + \mathcal{O}(\sqrt{\log n})$ for any $m \geq 1$. Further, for any number of $k = \mathcal{O}(\log\log n)$ binary queries, the upper bound on the maximum load improves to $m/n + \mathcal{O}(k(\log n)^{1/k})$ for any $m \geq 1$.

This result for k queries has several interesting consequences: (i) it implies new bounds for the $(1+\beta)$-process introduced by Peres, Talwar and Wieder (2015), (ii) it leads to new bounds for the graphical balanced allocation process on dense expander graphs, and (iii) it recovers and generalizes the bound of $m/n + \mathcal{O}(\log\log n)$ on the maximum load achieved by the Two-Choice process, including the heavily loaded case $m = \Omega(n)$ which was derived in previous works by Berenbrink et al. (2006) as well as Talwar and Wieder (2014).

One novel aspect of our proofs is the use of multiple super-exponential potential functions, which might be of use in future work.

2012 ACM Subject Classification Mathematics of computing → Probability and statistics; Mathematics of computing → Discrete mathematics; Theory of computation → Randomness, geometry and discrete structures; Theory of computation → Design and analysis of algorithms

Keywords and phrases power-of-two-choices, balanced allocations, potential functions, thinning

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.103

Related Version *Full Version*: https://arxiv.org/abs/2107.03916

Funding *Thomas Sauerwald*: The author was supported by the ERC grant "Dynamic March". Part of this work was done while visiting Hasso-Plattner Institute, Potsdam, Germany.

1 Introduction

We study balls-and-bins processes where the goal is to allocate m balls (jobs) sequentially into n bins (servers). The balls-and-bins framework a.k.a. balanced allocations [5] is a very popular and simple framework for various resource allocation and storage problems such as

load balancing, scheduling or hashing (see surveys [27, 34] for more details). In most of these settings, the goal is to find a simple allocation strategy that results in an allocation that is as balanced as possible.

It is a classical result that if each ball is placed in a random bin chosen independently and uniformly (called One-Choice), then the maximum load is $\Theta(\log n/\log\log n)$ w.h.p.[1] for $m = n$, and $m/n + \Theta(\sqrt{(m/n)\log n})$ w.h.p. for $m \gg n$. Azar et al. [5] (and implicitly Karp et al. [21]) proved that if each ball is placed in the lesser loaded of *two* randomly chosen bins, then the maximum load drops to $\log_2\log n + \mathcal{O}(1)$ w.h.p., if $m = n$. This dramatic improvement of Two-Choice is widely known as "power of two choices", and similar ideas have been applied to other problems including routing, hashing and randomized rounding [27].

While for $m = n$ a wide range of different proof techniques have been employed, the heavily loaded case $m \gg n$ turns out to be much more challenging. In a seminal paper [9], Berenbrink et al. proved a maximum load of $m/n + \log_2\log n + \mathcal{O}(1)$ w.h.p. using a sophisticated Markov chain analysis. A simpler and more self-contained proof was recently found by Talwar and Wieder [32], giving a slightly weaker upper bound of $m/n + \log_2\log n + \mathcal{O}(\log\log\log n)$ for the maximum load and at the cost of a larger error probability.

In light of the dramatic improvement of Two-Choice (or d-Choice) over One-Choice, it is important to understand the robustness of these processes. For example, in a concurrent environment, information about the load of a bin might quickly become outdated or communication with bins might be restricted. Also, acquiring always $d \geq 2$ uncorrelated choices might be costly in practice. Motivated by this, Peres et al. [28] introduced the $(1+\beta)$-process, in which two choices are available with probability β, and otherwise only one. Thus, the $(1+\beta)$-process interpolates nicely between Two-Choice and One-Choice, and surprisingly, a bound on the gap between maximum and average load of $\mathcal{O}(\log n/\beta)$ w.h.p. was shown, which also holds in the heavily loaded case where $m = \Omega(n)$. The $(1+\beta)$-process has been also connected to other processes, including population protocols [2], balls-and-bins with weights [31, 32] and, most notably, graphical balanced allocation [22, 28, 3, 6]. In this graphical model, bins correspond to vertices of a graph, and for each ball we sample an edge uniformly at random and place the ball in the lesser loaded bin of the two endpoints.

Our Model. In this work, we will investigate the following model. At each step, a ball is allowed to sample two random bins chosen independently and uniformly, however, the load comparison between the two bins will be performed under incomplete information. This may capture scenarios in which it is costly to communicate or maintain the exact load of a bin.

Specifically, we assume that each ball is allowed to send up to k binary queries to each of the two bins, inquiring about their current load. These queries can either be about the absolute load (i.e., is the load at least 100?), which we call *threshold processes*, or about the relative load (i.e., is the load at least the median?), which we call *quantile processes*.

We will distinguish between *oblivious* and *adaptive* allocation strategies. For an *adaptive* strategy, the queries may depend on the current load configuration (i.e., the full history of the process), whereas in the *oblivious* setting, queries may depend only on the current time-step.

Our Results. For the case of $k = 1$ query, Feldheim and Gurel-Gurevich [16] proved a bound of $\mathcal{O}(\sqrt{\log n/\log\log n})$ on the gap (between the maximum and average load) in the lightly loaded case $m = \mathcal{O}(n)$. In the same work, the authors suggest that the same bound

[1] In general, with high probability refers to probability of at least $1 - n^{-c}$ for some constant $c > 0$.

might be also true in the heavily loaded case [16, Problem 1.3]. In this work, we disprove this by showing a lower bound of $\Omega(\sqrt{\log n})$ on the gap for $m = \Theta(n\sqrt{\log n})$ (Theorem 4.4). We also prove a lower bound of $\Omega(\log n/\log\log n)$ on the gap, which holds for at least $\Omega(n \log n/\log\log n)$ of the time-steps in $[1, n\log^2 n]$ (Corollary 4.2). These two lower bounds hold even for the more general class of adaptive strategies.

It is natural to ask whether we can get an improved performance by allowing more, say *two* queries per bin. We prove that this is indeed the case, establishing a "power of two queries" result. Specifically, we show in Theorem 6.1 that for any $k = \mathcal{O}(\log\log n)$, there is an allocation process with k uniform quantiles (i.e., queries only depend on n, but not on the time t) that achieves for any $m \geq 1$:

$$\mathbf{Pr}\left[\,\mathrm{Gap}(m) = \mathcal{O}\Big(k \cdot (\log n)^{1/k}\Big)\,\right] \geq 1 - n^{-3}.$$

Comparing this for $k = 2$ to the lower bounds for $k = 1$, we indeed observe a "power of two queries" effect. For $k = \Theta(\log\log n)$, the gap even becomes $\mathcal{O}(\log\log n)$, which matches the Two-Choice result up to a multiplicative constant [9, 32]. Hence, for large values of k, the process approximates Two-Choice, whereas for $k = 1$ it resembles the $(1+\beta)$-process. Indeed, the same upper bound of $\mathcal{O}(\log n)$ follows from the analysis of the $(1+\beta)$-process (Theorem 5.2).

We also prove new upper bounds on the gap of the $(1+\beta)$ process with β close to 1 by relating it to a relaxed quantile process (Theorem 7.1). We show that these in turn imply new upper bounds on the graphical balanced allocation on dense expander graphs, making progress towards Open Question 2 in [28] (Corollary 7.2).

Our Upper Bound Techniques. We use the following two techniques in our upper bounds:

1. For upper bounding the gap for k queries, where $k \geq 2$, we use a series of k super-exponential potential functions of the form:

$$\Phi_j^{(s)} := \sum_{i=1}^{n} \exp\left(\alpha \cdot (\log n)^{j/k} \cdot \left(x_i^{(s)} - \frac{s}{n} - \kappa \cdot j(\log n)^{1/k}\right)^{+}\right),$$

   for $0 \leq j < k$ and some constants $\alpha, \kappa > 0$. Next, in the spirit of layered induction, we show that when $\Phi_j^{(s)} = \mathcal{O}(n)$, then $\Phi_{j+1}^{(s)}$ drops in expectation when large. Ultimately, for $j = k-1$, we obtain the desired bound on the gap. Similar to the analysis in [32] for Two-Choice, the base case of this induction follows by the $(1+\beta)$-process for constant β.
2. The techniques of [28] show that the drop in expectation implies that the expectation of Φ_j is $\mathcal{O}(n)$. From this, by Markov's inequality one can obtain that w.h.p. $\Phi_j^{(s)} = \mathrm{poly}(n)$. However, in the layered induction we need that w.h.p. $\Phi_j^{(s)} = \mathcal{O}(n)$. To obtain the high probability, we use a second instance Ψ_j of the potential function of the same form as Φ_j, but with larger (constant) $\widetilde{\alpha}$ instead of α. Then conditioning on $\Psi_j^{(s)} = \mathrm{poly}(n)$, the change $|\Phi_j^{(s+1)} - \Phi_j^{(s)}|$ is bounded and so we can apply a variant of the method of bounded differences (Theorem 2.1).

Applications and Implications on other Models. A direct implementation of the k-quantile protocol in practice requires to maintain some *global* information about the load configuration (that is, the exact, or at least the approximate, values of the quantiles). If this can be achieved, then the results of k-quantile for $k \geq 2$ demonstrate that a sub-logarithmic gap is possible – even with very limited *local* information about the individual bin loads.

In addition, our study of the k-quantile process also leads to new results for some previously studied allocation processes. We demonstrate that a $(1+\beta)$-process for β close to 1 is majorized by a (relaxed version of the) k-quantile process. For any $\beta = 1 - o(1)$, this leads to a sub-logarithmic bound on the gap, and if $\beta = 1 - 1/\operatorname{poly}(n)$, we recover the $\mathcal{O}(\log\log n)$ gap from the Two-Choice process. Secondly, we use a similar majorization argument to analyze graphical balanced allocation, which has been studied in several works on different graphs [28, 22, 3, 6]. Specifically, we prove for dense and strong expander graphs (including random d-regular graphs for $d = \operatorname{poly}(n)$) a gap of $\mathcal{O}(\log\log n)$. To the best of our knowledge, these are the first sub-logarithmic gap bounds in the heavily loaded case for the $(1+\beta)$-process and graphical balanced allocation (apart from $\beta = 1$ or the graph being a clique, both equivalent to Two-Choice).

Further Related Work. Our model for $k = 1$ is equivalent to the d-Thinning process for $d = 2$, where for each ball, a random bin is "suggested" and based on the bin's load, the ball is either allocated there or it is allocated to a second bin chosen uniformly and independently. Generalizing the results of [16] for $d = 2$, Feldheim and Li [18] also analyzed an extension of 2-Thinning, called d-Thinning. For $m = \mathcal{O}(n)$, they proved tight lower and upper bounds, resulting into an achievable gap of $(d + o(1)) \cdot (d \log n / \log\log n)^{1/d}$. Iwama and Kawachi [19] analyzed a special case of the threshold process for $m = n$ and for k equally-spaced thresholds, proving a gap of $(k + \mathcal{O}(1)) \sqrt[k+1]{(k+1)\frac{\log n}{\log((k+1)\log n)}}$. Mitzenmacher [26, Section 5] coined the term *weak threshold process* for the two threshold process in a queuing setting, where a customer chooses two queues uniformly at random and enters the first one iff it is shorter than T. This and previous work [14, 20, 35] analyze the case of a fixed threshold for queues and they do not directly imply results for the heavily loaded case.

In another related work, Alon et al. [4] established for the case $m = \Theta(n)$ a trade-off between the number of bits used for the representation of the load and the number of d bin choices. This is a more restricted case of having a fixed number of non-adaptive queries. For $d = 2$, Benjamini and Makarychev [7] obtained tight results for the gap, using a process very similar to the threshold process, but considering the case $m = \Theta(n)$ only.

Czumaj and Stemann [13] investigated general allocation processes, in which the decision whether to take a second (or further) sample depends on the load of the lightest sampled bin. They obtained strong and tight guarantees, but they assume the full information model and also $m = \mathcal{O}(n)$ (see [10] for some results for $m \geq n$). Other processes with inaccurate (or outdated) information about the load of a bin have been studied in an asynchronous environment [1] or a batch-based allocation [8]. However, the obtained bounds on the gap are only $\mathcal{O}(\log n)$. Other protocols that study the communication between balls and bins in more detail are [24, 23, 15, 30], but they assume that a ball can sample more than two bins.

After an earlier version of this paper was made available, Feldheim, Gurel-Gurevich and Li [17] extended the lower bounds for 2-Thinning when $m = \mathcal{O}(n \log^2 n)$ and also provided an adaptive thinning process that matches the $\Omega(\log n / \log\log n)$ lower bound proved in this paper. Also, Los, Sauerwald and Sylvester [25] proved that Threshold(m/n) (or equivalently 2-Thinning where the threshold is m/n) achieves w.h.p. a $\Theta(\log n)$ gap.

Organization. In Section 2, we introduce our model more formally in addition to some notation used in the analysis. In Section 4, we present our lower bounds on processes with one query. In Section 5, we present the upper bound for the quantile process with one query. In Section 6, we present a generalized upper bound for $k \geq 2$ queries. Section 7 contains our applications to $(1+\beta)$-process and graphical balanced allocations. We close in Section 8 by

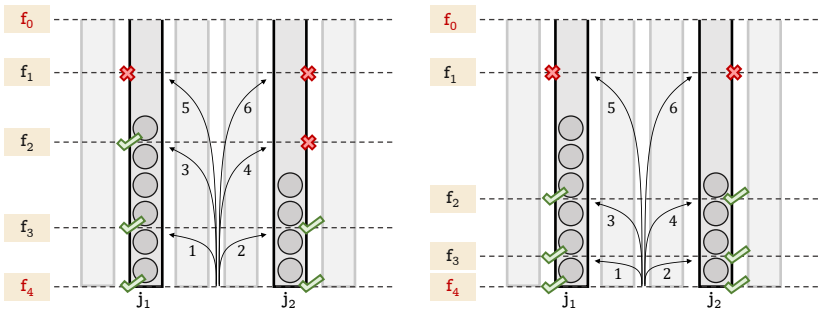


Figure 1 Example allocation using two 3-threshold processes (f_1, f_2, f_3). (**Left**): The ball is allocated in j_2, since $i_1 = 2$ and $i_2 = 3$. (**Right**): For a different choice of thresholds, the process may not be able to differentiate the two loaded bins, so the ball will be allocated at random.

summarizing our main results and pointing to some open problems. We also briefly present some experimental results in Section 9. In Section 3, we formally relate the new quantile (and threshold) processes to each other and to other processes studied before (see Figure 2 for an overview).

2 Notation, Definitions and Preliminaries

We sequentially allocate m balls (jobs) into n bins (servers). The load vector at step t is $x^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_n^{(t)})$ and in the beginning, $x_i^{(0)} = 0$ for $i \in [n]$. Also $y^{(t)} = (y_1^{(t)}, y_2^{(t)}, \ldots, y_n^{(t)})$ will be the permuted load vector, sorted decreasingly in load. This can be described by **ranks**, which form a permutation of $[n]$ that satisfies $r = \mathrm{Rank}^{(t)}(i) \Rightarrow y_r^{(t)} = x_i^{(t)}$. Following previous work, we analyze allocation processes in terms of the

$$\mathrm{Gap}(t) := \max_{1 \le i \le n} x_i^{(t)} - \frac{t}{n} = y_1^{(t)} - \frac{t}{n},$$

i.e., the difference between maximum and average load at time $t \ge 0$. It is well-known that even for Two-Choice, the gap between maximum and minimum load is $\Omega(\log n)$ for large m (e.g. [28]). Here our focus is on sequential allocation processes based on binary queries. That is, at each step t:

1. Sample *two* bins independently and uniformly at random (with replacement).
2. Send the same k binary queries to each of the two bins about their load.
3. Allocate the ball in the lesser loaded one of the two bins (based on the answers to the queries), breaking ties randomly.

We first describe **threshold-based processes**, where queries to each bin j are of the type "Is $x_j^{(t)} \ge f(t)$" for some function f that maps into $\mathbb{N}$. For example, we could ask whether the load of a bin is at least the average load. Formally, we denote such a process with two choices and k queries by Threshold$(f_1, f_2, \ldots, f_k)$, where $f_1 > f_2 > \ldots > f_k$ are k different load thresholds, that may depend on the time t, in which case we write $f_i(t)$. After sending all k queries to a bin j, we receive the correct answers to all these queries and then we determine the i $(0 \le i \le k)$ for which,

$$x_j^{(t)} \in (f_{i+1}(t), f_i(t)],$$

where $f_0(t) = +\infty$ and $f_{k+1}(t) = -\infty$ (see Figure 1). After having obtained two such numbers $i_1, i_2 \in \{0, 1, \ldots, k\}$, one for each bin j_1 and j_2, we will allocate the ball "greedily", i.e., into j_1 if $i_1 < i_2$ and into j_2 if $i_1 > i_2$. If $i_1 = i_2$, then we will break ties randomly.

We proceed to define **quantile-based processes**. In this process, queries to a bin j are of the type "Is $x_j^{(t)} \geq y_{\delta(t)\cdot n}^{(t)}$?", for some function δ that maps t into $\{1/n, 2/n, \ldots, 1\}$. For example if $\delta = 1/2$, we are querying whether the load of a bin is at most the median load. We denote such a process with two choices and k queries by $\textsc{Quantile}(\delta_1, \delta_2, \ldots, \delta_k)$, where $\delta_1 < \delta_2 < \ldots < \delta_k$ are k different quantiles, which may depend on the time t. After sending all k queries to a bin j in step t, we receive the correct answers and then we determine the i $(0 \leq i \leq k)$ for which,

$$\text{Rank}^{(t)}(j) \in (\delta_i(t) \cdot n, \delta_{i+1}(t) \cdot n],$$

where $\delta_0(t) = 0$ and $\delta_{k+1}(t) = 1$. As before, we allocate the ball to the bin with smaller i-value and break ties randomly.

Quantile and Threshold processes can be classified into oblivious processes and adaptive processes, depending on the type of queries. In an **oblivious process**, the queries $f_1, f_2, \ldots$ (or $\delta_1, \delta_2, \ldots$) may only depend on t (as well as n) – a special case is a **uniform process** where $\delta_1, \delta_2, \ldots$ are constants (independent of t), and the f_i's are of the form $t/n + f_i(n)$. In an **adaptive process**, queries in step t may depend on the full history of the process, i.e., the load vector $x^{(t-1)}$, so each query i involves a function $f_i(x^{(t-1)})$, but this must be specified before receiving any answers. In the adaptive setting, a k-quantile process can simulate any k-threshold process, by setting the quantile to the largest $\delta_i(t)$ such that $y_{\delta_i(t)\cdot n} \leq f_i(t)$ (Lemma 3.7).

The d**-Thinning process** [16] works as follows. For each ball to be allocated, an overseer can inspect up to d randomly sampled bins in an online fashion, and based on all previous history, can accept or reject each bin (however, one of the d proposed bins must be accepted).

The d**-Choice process** [5] (sometimes also called $\textsc{Greedy}[d]$) is the process where, for each ball, d bins are chosen uniformly at random and the ball is placed in the least loaded bin. We will refer to the special case $d = 1$ as the **One-Choice process**, and $d = 2$ as the **Two-Choice process**. The $(1+\beta)$**-process** [28] is the process where each ball is placed with probability β according to Two-Choice and with probability $1 - \beta$ according to One-Choice.

Finally, in **graphical balanced allocation** [22, 28], we are given an undirected graph G with n vertices corresponding to n bins. For each ball to be allocated, we select an edge $\{u, v\} \in E(G)$ uniformly at random, and place the ball in the lesser loaded bin among $\{u, v\}$.

Following [28] and generalizing the processes above, **an allocation process** can be described by a **probability vector** $p^{(t)} = (p_1^{(t)}, p_2^{(t)}, \ldots, p_n^{(t)})$ for step t, where $p_i^{(t)}$ is the probability for incrementing the load of the i-th most loaded bin. Following the idea of **majorization**, if two processes with (time-invariant) probability vectors p and q, for all $i \in [n]$ satisfy $\sum_{j \leq i} p_j \leq \sum_{j \leq i} q_j$, then there is a coupling between the allocation processes with sorted load vectors $y(p)$ and $y(q)$ such that $\sum_{j \leq i} y_i^{(t)}(p) \leq \sum_{j \leq i} y_i^{(t)}(q)$ for all $i \in [n]$ (q **majorizes** p).

Finally, we define the **height** of a ball as $i \geq 1$ if it is the i^{th} ball added to the bin.

Many statements in this work hold only for sufficiently large n, and several constants are chosen generously with the intention of making it easier to verify some technical inequalities.

2.1 Probabilistic Tools

In order to state the concentration inequality for supermartingales conditional on a bad event not occurring, we introduce the following definitions from [11]. Consider any r.v. X (in our case it will be the Φ_j and the Γ_1 potentials) that can be evaluated by a sequence of decisions

$Y_1, Y_2, \ldots, Y_N$ of finitely many outputs (the allocated balls). We can describe the process by a *decision tree* T, a complete rooted tree with depth n with vertex set $V(T)$. Each edge uv of T is associated with a probability p_{uv} depending on the decision made from u to v.

We say $f : V(T) \to \mathbb{R}$ satisfies an *admissible condition* P if $P = \{P_v\}$ holds for every vertex v. For an admissible condition P, the associated bad set B_i over the X_i is defined to be

$$B_i = \{v \mid \text{the depth of } v \text{ is } i, \text{ and } P_u \text{ does not hold for some ancestor } u \text{ of } v\}.$$

▶ **Theorem 2.1** (Theorem 8.5 from [11]). *For a filter $\mathcal{F}$, $\{\emptyset, \Omega\} = \mathcal{F}^{(0)} \subset \mathcal{F}^{(1)} \subset \ldots \subset \mathcal{F}^{(N)} = \mathcal{F}$, suppose that a random variable $X^{(s)}$ is $\mathcal{F}^{(s)}$-measurable, for $0 \leq s \leq N$. Let B be the bad set associated with the following admissible conditions:*

$$\begin{aligned} \mathbf{E}\left[X^{(s)} \mid \mathcal{F}^{(s-1)} \right] &\leq X^{(s-1)}, \\ \mathbf{Var}\left[X^{(s)} \mid \mathcal{F}^{(s-1)} \right] &\leq \sigma_s^2, \\ X^{(s)} - \mathbf{E}\left[X^{(s)} \mid \mathcal{F}^{(s-1)} \right] &\leq a_s + M, \end{aligned}$$

for fixed $\sigma_s > 0$ and $a_s > 0$. Then, we have for any $\lambda > 0$,

$$\mathbf{Pr}\left[X^{(N)} \geq X^{(0)} + \lambda \right] \leq \exp\left(-\frac{\lambda^2}{2(\sum_{s=1}^{N}(\sigma_s^2 + a_s^2) + M\lambda/3)} \right) + \mathbf{Pr}\left[B \right].$$

3 Basic Relations between Allocation Processes

In this section we collect several basic relations between allocation processes, following the notion of majorization [28]. Figure 2 gives a high-level overview of some of these relations, along the with the derived and implied gap bounds.

Recall that the Two-Choice probability vector is given by $p_i = \frac{2i-1}{n^2}$, for $i \in [n]$:

The $(1+\beta)$ probability vector [28] interpolates between those of One-Choice and Two-Choice, so for any $i \in [n]$, $p_i = (1-\beta) \cdot \frac{1}{n} + \beta \cdot \frac{2i-1}{n^2}$.

For the process Quantile$(\delta_1, \ldots, \delta_k)$, it is straightforward to verify that the probability vector satisfies for any $i \in [n]$:

$$p_i = \begin{cases} \frac{\delta_1}{n} & 1 \leq i \leq \delta_1 \cdot n, \\ \frac{\delta_1+\delta_2}{n} & \delta_1 \cdot n < i \leq \delta_2 \cdot n, \\ \vdots & \\ \frac{\delta_{k-1}+\delta_k}{n} & \delta_{k-1} \cdot n < i \leq \delta_k \cdot n, \\ \frac{1+\delta_k}{n} & \delta_k \cdot n < i. \end{cases} \tag{3.1}$$

We start by making some simple observations for the quantile processes:

▶ **Observation 3.1.** *For any $n \geq 0$, the Quantile$(\frac{1}{n}, \frac{2}{n} \ldots, \frac{n-1}{n})$ process is equivalent to the Two-Choice process.*

▶ **Observation 3.2.** *For $k < n-1$, for any $\delta', \delta_1, \ldots, \delta_k$ quantiles, the Quantile$(\delta_1, \ldots, \delta_k)$ process majorizes Quantile$(\delta_1, \ldots, \delta_i, \delta', \delta_{i+1}, \ldots, \delta_k)$.*

By combining Observation 3.1 and Observation 3.2, we get:

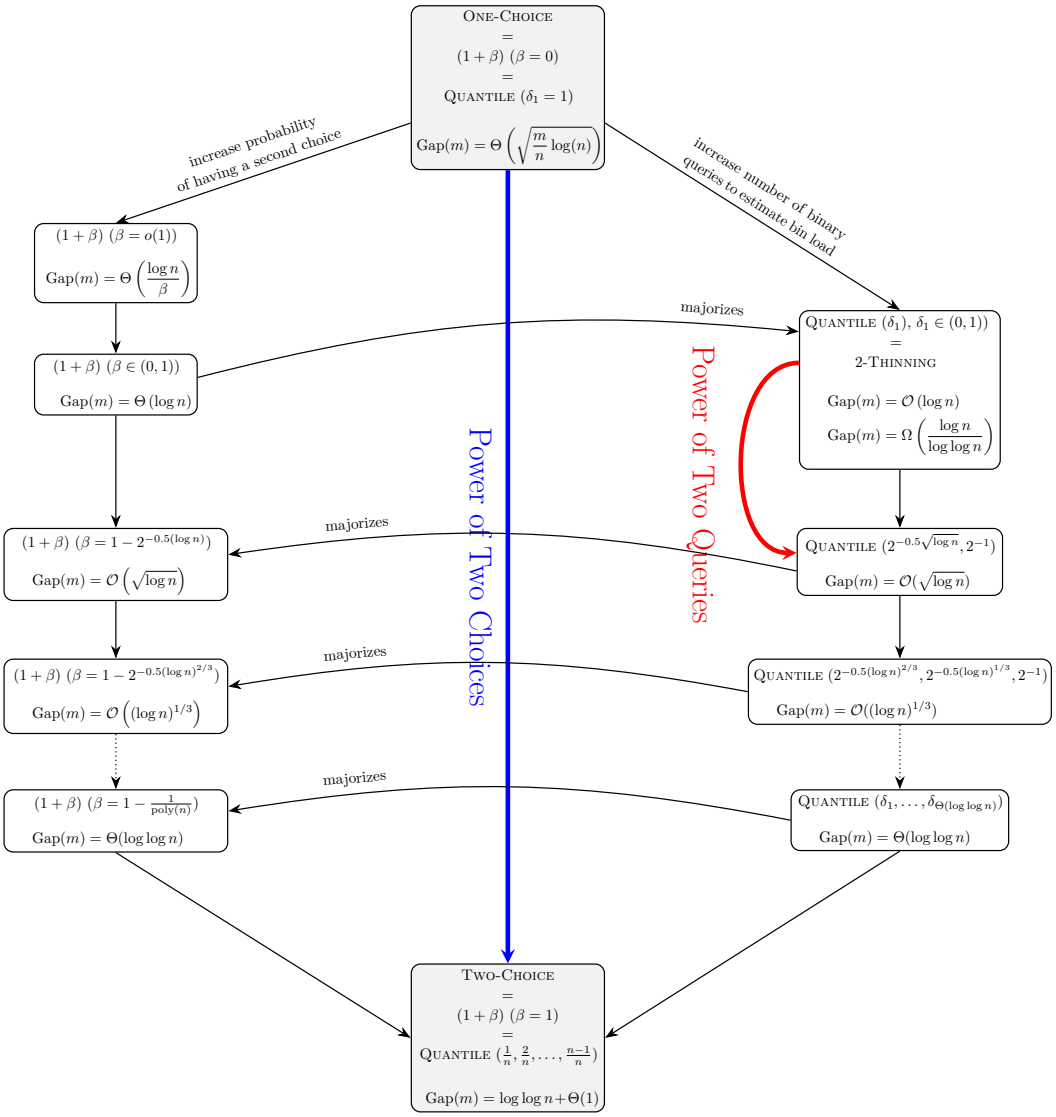


Figure 2 Overview of bounds on $\mathrm{Gap}(m)$ for various allocation processes that interpolate between One-Choice and Two-Choice. All stated upper bounds are valid for any $m \geq 1$, while lower bounds may only hold for certain ranges of m. Some of the majorization results in the figure only hold for a suitable Relaxed-Quantile process.

▶ **Corollary 3.3.** *Any* Quantile$(\delta_1, \ldots, \delta_k)$ *process majorizes* Two-Choice.

Further, we show that we can always execute the Quantile(δ) and Threshold(f) processes in the same way as 2-Thinning:

▶ **Lemma 3.4.** *Consider a quantile process* Quantile(δ) *with one query. This process can always be transformed into an equivalent instance of* 2-Thinning*: Sample a bin, if its rank is greater than* $n \cdot \delta(t)$*, then place the ball there; otherwise, place the ball in a randomly chosen bin.*

▶ **Lemma 3.5.** *Consider a threshold process* Threshold(f) *with one query. This process can be always transformed into the following equivalent process: For the first sampled bin* i*, if its load is smaller than* $f(t)$*, place the ball; otherwise, place the ball in another randomly chosen bin* j*.*

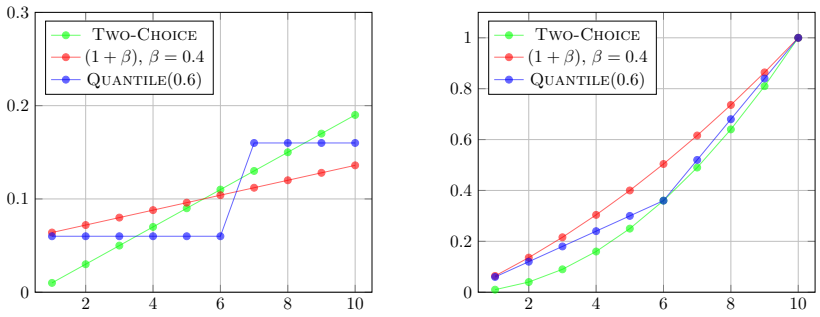


Figure 3 Illustration of the probability vector $(p_1, p_2, \ldots, p_{10})$ and cumulative probability distribution of Two-Choice, $(1+\beta)$ with $\beta = 0.4$ and Quantile (0.6), which is sandwiched by the other two processes.

▶ **Lemma 3.6.** *For any $\delta \in (0,1)$ and any $\beta \in (0,1)$ with $\beta \le \delta \le 1-\beta$, the process Quantile (δ) is majorized by a $(1+\beta)$-process. In particular, the gap of the quantile process is stochastically smaller than that of the $(1+\beta)$-process.*

Note that for any given $\delta \in (0,1)$, $\beta := \min\{\delta, 1-\delta\}$ always satisfies the precondition of the lemma. Conversely, for any given $\beta \le 1/2$, we have $\beta \le 1/2 \le (1-\beta)$, and thus we can set $\delta := 1/2$. The majorization results in Corollary 3.3 and Lemma 3.6 are illustrated in Figure 3 for $n = 10$.

Next, we establish that any Threshold$(f_1, \ldots, f_k)$ can be simulated by an adaptive Quantile process with k quantiles, and similarly, any Quantile$(\delta_1, \ldots, \delta_k)$ process can be simulated by an adaptive randomised Threshold process with k thresholds.

▶ **Lemma 3.7.** *Any Threshold$(f_1, \ldots, f_k)$ process can be simulated by an adaptive quantile process with k queries.*

▶ **Lemma 3.8.** *Any step t of a Quantile$(\delta_1, \ldots, \delta_k)$ process can be simulated by first choosing $f_1(t), f_2(t), \ldots, f_k(t)$ randomly (from a suitable distribution depending on $x^{(t)}$ and $\delta_1(t), \ldots, \delta_k(t)$) and then running Threshold$(f_1, f_2, \ldots, f_k)$.*

Finally, we establish the following relation between Quantile and $(2k)$-Thinning:

▶ **Lemma 3.9.** *For any $k \ge 1$, a Quantile $(\delta_1, \ldots, \delta_k)$ process can be simulated by an adaptive (and randomized) $(2k)$-Thinning process.*

4 Lower Bounds for One Quantile and One Threshold

In the lightly loaded case (i.e., $m = n$), [16] proved an upper bound of $(2 + o(1)) \cdot (\sqrt{2 \log n / \log\log n})$ on the maximum load for a uniform Threshold(f)-process with $f = \sqrt{2 \log n / \log\log n}$ ([18] extended this to $d > 2$). They also proved that this strategy is asymptotically optimal. In [16, Problem 1.3], the authors suggest that the $\mathcal{O}(\sqrt{\log n / \log\log n})$ bound on the gap extends to the heavily loaded case. Here we will disprove this, establishing a slightly larger lower bound of $\Omega(\sqrt{\log n})$ (Theorem 4.4). We also derive additional lower bounds (Theorem 4.1 and Corollary 4.2) that demonstrate that any Quantile or Threshold process will "frequently" attain a gap which is even as large as $\Omega(\log n / \log\log n)$.

Let us describe the intuition behind this bound in case of uniform quantiles, neglecting technicalities. Consider Quantile(δ) and the equivalent 2-Thinning instance where a ball is placed in the first bin if its load is among the $(1-\delta) \cdot n$ lightest bins, and otherwise it is placed in a new (second) bin chosen uniformly at random (Lemma 3.4). We have two cases:

Case 1: We choose most times a "large" δ. Then we allocate approximately $m \cdot \delta$ balls to their second bin choice which is uniform over all n bins. This will lead to a behavior close to One-Choice.
Case 2: We choose most times a "small" δ. Then we allocate approximately $m \cdot (1-\delta)$ balls with the first bin choice, which is a One-Choice process over the $n \cdot (1-\delta)$ lightest bins. For small δ there are simply "too many" light bins that will reach a high load level, so the process is again close to One-Choice.

▶ **Theorem 4.1.** *For any adaptive* Quantile(δ) *(or* Threshold(f)*) process,*

$$\mathbf{Pr}\left[\max_{t\in[0,n\log^2 n]} \mathrm{Gap}(t) \geq \frac{1}{8}\cdot\frac{\log n}{\log\log n}\right] \geq 1-o(n^{-2}).$$

Let us also observe a slightly stronger statement which follows directly from Theorem 4.1:

▶ **Corollary 4.2.** *Any adaptive process* Quantile(δ) *satisfies:*

$$\mathbf{Pr}\left[\bigcup_{t\in[0,n\log^2 n]} \min_{s\in\left[t,t+\frac{1}{16}n\frac{\log n}{\log\log n}\right)} \mathrm{Gap}(s) \geq \frac{1}{16}\cdot\frac{\log n}{\log\log n}\right] \geq 1-n^{-2}.$$

In other words, the corollary states that for at least $\Omega(n\log n/\log\log n)$ (consecutive) steps in $[1,\Theta(n\log^2 n)]$, the gap is $\Omega(\log n/\log\log n)$. This is in contrast to the behavior of the process Quantile(δ_1,δ_2), for which our result in Section 6 implies that with high probability the gap is *always* below $\mathcal{O}(\sqrt{\log n})$ during any time-interval of the same length.

Further for uniform Quantile(δ), we are always either in Case 1 or Case 2, so the following strengthened version of Theorem 4.1 holds:

▶ **Corollary 4.3.** *For any uniform* Quantile(δ) *process for $m = n\log^2 n$ balls,*

$$\mathbf{Pr}\left[\mathrm{Gap}(m) \geq \frac{1}{8}\cdot\frac{\log n}{\log\log n}\right] \geq 1-o(n^{-2}).$$

We also show a lower bound for fixed m, which is derived in a similar way as Theorem 4.1, but with a different parameterization of "large" and "small" quantiles:

▶ **Theorem 4.4.** *For any adaptive* Quantile(δ) *(or* Threshold(f)*) process, with $m = K\cdot n\sqrt{\log n}$ balls for $K = 1/10$, it holds that*

$$\mathbf{Pr}\left[\mathrm{Gap}(m) \geq \frac{1}{20}\sqrt{\log n}\right] \geq 1-o(n^{-2}).$$

5 Upper Bounds for One Quantile

In this section we study the Quantile(δ) process for constant $\delta \in (0,1)$. This analysis will also serve as the basis for the k-quantile case with $k > 1$ in Section 6. First, we define the following exponential potential function (similarly to [28]): For any time-step $s \geq 0$,

$$\Phi_0^{(s)} := \sum_{i=1}^{n} \exp\left(\alpha_2 \cdot (x_i^{(s)} - \frac{s}{n})^+\right),$$

where $z^+ = \max(z,0)$ and $\alpha_2 > 0$ to be specified later. We first remark that with the results in [28], a bound on the expected value of Φ_0 can be easily derived:

▶ **Theorem 5.1** (cf. Theorem 2.10 in [28]). *Consider any allocation process with probability vector p that is (i) non-decreasing in i, $p_i \le p_{i+1}$ and (ii) for some $0 < \epsilon < 1/4$,*

$$p_{n/3} \le \frac{1-4\epsilon}{n} \quad \text{and} \quad p_{2n/3} \ge \frac{1+4\epsilon}{n}.$$

Then, for $0 < \alpha_2 < \epsilon/6$, we have for any $s \ge 0$, $\mathbf{E}\left[\Phi_0^{(s)}\right] \le cn$, where $c = \frac{40 \cdot 128^3}{\epsilon^5}$.

In particular, by verifying the condition on the probability vector and applying Markov's inequality, we immediately obtain an upper bound of $\mathcal{O}(\log n)$ on the gap.

▶ **Theorem 5.2.** *For the quantile process QUANTILE(δ) with $\delta \in [1/3, 2/3]$ and any $m \ge 1$,*

$$\mathbf{Pr}\left[\,\mathrm{Gap}(m) \le 300 \log n\,\right] \ge 1 - \mathcal{O}(n^{-2}).$$

However, to analyze the process with more than one quantile in the next section, we will need a tighter analysis. We prove the following refined version of Theorem 5.1:

▶ **Theorem 5.3.** *Consider any probability vector p that is (i) non-decreasing in i, i.e., $p_i \le p_{i+1}$ and (ii) for $\epsilon = 1/12$,*

$$p_{n/3} \le \frac{1-4\epsilon}{n} \quad \text{and} \quad p_{2n/3} \ge \frac{1+4\epsilon}{n}.$$

Then, for any $t \ge 0$ and $\alpha_2 := 0.0002$, $c := c_{\epsilon,\alpha_2} := 2 \cdot 40 \cdot 128^3 \cdot \epsilon^{-7} \cdot 4 \cdot \alpha_2^{-1}$,

$$\mathbf{Pr}\left[\bigcap_{s \in [t, t+n\log^5 n]} \Phi_0^{(s)} \le 2cn\right] \ge 1 - n^{-3}.$$

Note that Theorem 5.3 not only implies a gap of $\mathcal{O}(\log n)$ using Markov's inequality (as Theorem 5.1), but also that for any fixed time s, the number of bins with load at least $s/n + \lambda$ is at most $2cn/\exp(\alpha_2 \cdot \lambda)$ for any $\lambda \ge 0$. In particular, for any $\lambda = \Theta(\log n)$, only a polynomially small fraction of all bins have load at least $s/n + \lambda$.

Proof Outline of Theorem 5.3. In order to prove that Φ_0 is small, we will reduce it to the potential function Γ used in [28]:

$$\Gamma^{(s)} := \sum_{i=1}^{n} \left(\exp\left(\alpha(x_i^{(s)} - s/n)\right) + \exp\left(-\alpha(x_i^{(s)} - s/n)\right)\right),$$

for some constant $0 < \alpha < 1/(6 \cdot 12)$. Note that if $\alpha = \alpha_2$, then $\Phi_0^{(s)} \le \Gamma^{(s)}$, so it suffices to upper bound $\Gamma^{(s)}$. It is crucial that this potential includes both the $\exp(\alpha(\cdot))$ and $\exp(-\alpha(\cdot))$ terms, as otherwise the potential may not decrease, even if it is large (see [28, Appendix]).

▶ **Lemma 5.4** (Theorem 2.9 and 2.10 in [28]). *For any process satisfying the conditions of Theorem 5.3, (i) for any $t \ge 0$,*

$$\mathbf{E}\left[\Gamma^{(t+1)} \mid \Gamma^{(t)}\right] \le \left(1 - \frac{\epsilon'_\alpha}{n}\right) \cdot \Gamma^{(t)} + c',$$

where $\epsilon'_\alpha := \frac{\alpha\epsilon}{4}$ and $c' := \frac{40 \cdot 128^3}{\epsilon^5}$. Furthermore, (ii) for any $t \ge 0$, $\mathbf{E}\left[\Gamma^{(t)}\right] \le cn$.

To obtain the stronger statement that $\Gamma^{(t)} = \mathcal{O}(n)$ w.h.p., we will be using two instances of the potential function: Γ_1 with $\alpha_1 = 0.01$ and Γ_2 with $\alpha_2 = 0.0002$; so $\Gamma_1 \geq \Gamma_2$. The interplay between these two potentials is shown in Figure 4. We pick α_1 such that $12.1 \cdot \frac{\alpha_1}{\alpha_2} < \frac{1}{3}$ and hence the additive change of Γ_2 (given Γ_1 is small) is $n^{1/3}$:

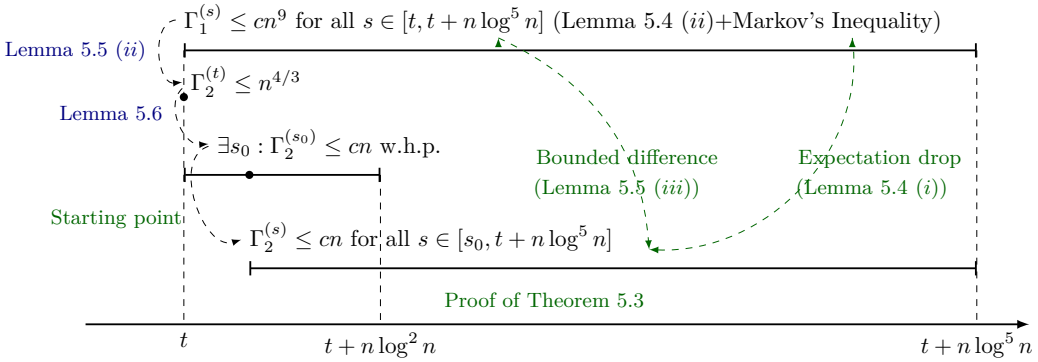


Figure 4 Outline for the proof of Theorem 5.3. Results in green are used in the application of the concentration inequality (Theorem 2.1) in Theorem 5.3.

▶ **Lemma 5.5.** *For any $t \geq 0$, if $\Gamma_1^{(t)} \leq cn^9$, then, (i) $\left|x_i^{(t)} - \frac{t}{n}\right| \leq \frac{9.1}{\alpha_1}\log n$ for all $i \in [n]$, (ii) $\Gamma_2^{(t)} \leq n^{4/3}$, and, (iii) $|\Gamma_2^{(t+1)} - \Gamma_2^{(t)}| \leq n^{1/3}$.*

The precondition of Lemma 5.5 is easy to satisfy thanks to Lemma 5.4 and Markov's inequality. The next lemma proves a weaker version of Theorem 5.3, in the sense that the potential $\Gamma_2^{(s)}$ is small in *at least* one step. Note that due to the choice of α_1 and α_2, we have $c > \frac{2c'}{\epsilon'_{\alpha_2}}$.

▶ **Lemma 5.6.** *For any $t \geq n\log^2 n$, for constants $c' > 0, \epsilon'_{\alpha_2} > 0$ defined as above,*

$$\mathbf{Pr}\left[\bigcup_{s\in[t-n\log^2 n,t]} \Gamma_2^{(s)} \leq \frac{2c'}{\epsilon'_{\alpha_2}} \cdot n\right] \geq 1 - 2cn^{-8}.$$

To prove the strong version that $\Gamma_2^{(s)}$ is small at *all* time-steps, we use Lemma 5.6 to obtain a starting point s_0. For the following time-steps, we bound the expected value of $\Gamma_2^{(s)}$ for $s \geq s_0$, using Lemma 5.4. Then we apply a concentration inequality for supermartingales (Theorem 2.1), and use the bounded difference $|\Gamma_2^{(s+1)} - \Gamma_2^{(s)}| \leq n^{1/3}$ for all $s \geq t$ (Lemma 5.5).

6 Upper Bounds for More Than One Quantile

6.1 Upper Bounds on the Original Quantile Process and Consequences

We now generalize the analysis from Section 5 for one quantile to $2 \leq k \leq \kappa \cdot \log\log n$ quantiles, where $\kappa := 1/\log(10^4)$. We emphasize that our chosen quantiles are oblivious and even uniform, i.e., independent of t (but dependent on n). Specifically, we define

$$\widetilde{\delta}_i = \begin{cases} \frac{1}{2} & \text{for } i = k, \\ 2^{-0.5(\log n)^{(k-i)/k}} & \text{for } 1 \leq i < k\ , \end{cases}$$

and let each δ_i be $\widetilde{\delta}_i$ rounded up to the nearest multiple of $\frac{1}{n}$. The intuition is that the largest quantile $\delta_k = \frac{1}{2}$ ensures that the load distribution is at least "coarsely" balanced, analogous to the $(1+\beta)$-process. All smaller quantiles $\delta_1, \delta_2, \ldots, \delta_{k-1}$ almost always return a negative answer, but they gradually reduce the probability of allocating to a heavy bin.

▶ **Theorem 6.1** (Theorem 6.5 simplified). *For any integer $2 \le k \le \kappa \log\log n$, consider the* Quantile$(\delta_1, \delta_2, \ldots, \delta_k)$ *process with the δ_i's defined above. Then for any $m \ge 1$,*

$$\mathbf{Pr}\left[\,\mathrm{Gap}(m) \le 1000 \cdot k \cdot (\log n)^{1/k}\,\right] \ge 1 - n^{-3}.$$

For $k = 2$ and $k = 3$, Theorem 6.1 directly implies the following corollary:

▶ **Corollary 6.2.** *For $k = 2$, the process* Quantile$(2^{-0.5\sqrt{\log n}}, \frac{1}{2})$ *satisfies for any $m \ge 1$,*

$$\mathbf{Pr}\left[\,\mathrm{Gap}(m) \le 2000 \cdot \sqrt{\log n}\,\right] \ge 1 - n^{-3}.$$

For $k = 3$ the process Quantile$(2^{-0.5(\log n)^{2/3}}, 2^{-0.5(\log n)^{1/3}}, \frac{1}{2})$ *satisfies for any $m \ge 1$,*

$$\mathbf{Pr}\left[\,\mathrm{Gap}(m) \le 3000 \cdot (\log n)^{1/3}\,\right] \ge 1 - n^{-3}.$$

Using the fact that any allocation process with k quantiles majorizes a suitable adaptive (and randomized) $2k$-Thinning process (Lemma 3.9), we also obtain:

▶ **Corollary 6.3.** *For any even $d \le \frac{2}{\kappa} \log\log n$, there is an (adaptive and randomized) d-Thinning process, satisfying for any $m \ge 1$,* $\mathbf{Pr}\left[\,\mathrm{Gap}(m) \le 2000 \cdot d \cdot (\log n)^{(2/d)}\,\right] \ge 1 - n^{-3}$.

This is an extension of [18, Theorem 1.1] to d-Thinning to the heavily-loaded case, but with an exponent of $2/d$ instead of $1/d$.

Finally, for $k = \Theta(\log\log n)$, the bound on the gap in Theorem 6.1 is $C \cdot \log\log n$ for some (large) constant $C > 0$. Surprisingly, this matches the gap of the full information setting (Two-Choice process), even though the Quantile process behaves quite differently. For instance, Quantile cannot discriminate among the $n/2$ most lightly loaded bins. Also since any Quantile process majorizes Two-Choice (see Corollary 3.3), we deduce:

▶ **Corollary 6.4.** *For Two-Choice, there is a constant $C > 0$ such that for any $m \ge 1$,* $\mathbf{Pr}\left[\,\mathrm{Gap}(m) \le C \log\log n\,\right] \ge 1 - n^{-3}$.

This result originally shown in [9] proved the tighter bound $\mathrm{Gap}(m) = \log_2 \log n \pm \mathcal{O}(1)$, w.h.p. However, their analysis combines sophisticated tools from Markov chain theory and computer-aided calculations. The simpler analysis in [32] derives the same gap bound up to an additive $\mathcal{O}(\log\log\log n)$ term, but the error probability is much larger, i.e., $\Theta((\log\log n)^{-4})$. In comparison to their bound, our result achieves a much smaller error probability of $\mathcal{O}(n^{-3})$, but it comes at the cost of a multiplicative constant in the gap bound.

6.2 Relaxed Quantile Process and Outline of the Inductive Step

We now define a class of processes Relaxed-Quantile$_\gamma(\delta_1, \ldots, \delta_k)$, which relaxes the definition of Quantile$(\delta_1, \ldots, \delta_k)$, with $1 \le k \le \kappa \log\log n$ and a relaxation factor $\gamma \ge 1$. The probability vector p of such a process satisfies four conditions: (i), for each $i \in [n]$,

$$p_i \le \begin{cases} \gamma \cdot \frac{\delta_1}{n} & 1 \le i \le \delta_1 \cdot n, \\ \gamma \cdot \frac{\delta_1 + \delta_2}{n} & \delta_1 \cdot n < i \le \delta_2 \cdot n, \\ \quad \vdots & \\ \gamma \cdot \frac{\delta_{k-1} + \delta_k}{n} & \delta_{k-1} \cdot n < i \le \delta_k \cdot n, \end{cases}$$

(*ii*) the probability vector p is non-decreasing in i, (*iii*) $p_{n/3} \leq \frac{1-4\epsilon}{n}$ and, (*iv*) $p_{2n/3} \geq \frac{1+4\epsilon}{n}$ for some $0 < \epsilon < 1/4$. Note that the process $\textsc{Quantile}(\delta_1, \delta_2, \ldots, \delta_k)$ with the δ_i's as defined above falls into this class with $\gamma = 1$ (cf. Equation (3.1)).

▶ **Theorem 6.5** (Theorem 6.1 generalized). *Consider a* $\textsc{Relaxed-Quantile}_\gamma(\delta_1, \delta_2, \ldots, \delta_k)$ *process with the* δ_i*'s above. Let* $2 \leq k \leq \kappa \log\log n$ *and* $1 \leq \gamma \leq 6$*. Then for any* $m \geq 1$*,*

$$\mathbf{Pr}\left[\,\mathrm{Gap}(m) \leq 1000 \cdot k \cdot (\log n)^{1/k}\,\right] \geq 1 - n^{-3}.$$

Reduction of Theorem 6.5 to Lemma 6.6. The proof of Theorem 6.5 employs some type of layered induction over k different, super-exponential potential functions. Generalizing the definition of $\Phi_0^{(s)}$ from Section 5, for any $0 \leq j \leq k-1$:

$$\Phi_j^{(s)} := \sum_{i=1}^{n} \exp\left(\alpha_2 \cdot (\log n)^{j/k} \cdot \left(x_i^{(s)} - \frac{s}{n} - \frac{2}{\alpha_2} j (\log n)^{1/k}\right)^+\right),$$

where $\alpha_2 = 0.0002$ (recall $z^+ = \max\{z, 0\}$). We will then employ this series of potential functions $j = 0, 1, \ldots, k-1$ to analyze the process over the time-interval $s \in [m - n\log^5 n, m]$.

The next lemma (Lemma 6.6) formalizes this inductive argument. It shows that if for all steps s within some suitable time-interval, the number of balls of height at least $\frac{s}{n} + \frac{2}{\alpha_2} j (\log n)^{1/k}$ is small, then the number of balls of height at least $\frac{s}{n} + \frac{2}{\alpha_2}(j+1)(\log n)^{1/k}$ is even smaller. This "even smaller" is encapsulated by the (non-constant) base of Φ_j, which increases in j; however, this comes at the cost of reducing the time-interval slightly by a $\Theta(n \log^3 n)$ term. Finally, for $j = k-1$, we can conclude that at step $s = m$, there are no balls of height $\frac{s}{n} + \frac{2}{\alpha_2} k (\log n)^{1/k}$. Hence we can infer that the gap is $\mathcal{O}(k \cdot (\log n)^{1/k})$.

▶ **Lemma 6.6 (Inductive Step).** *Assume that for some* $1 \leq j \leq k \leq \frac{1}{\log(10^4)} \log\log n$*, the process* $\textsc{Relaxed-Quantile}_\gamma(\delta_1, \ldots, \delta_k)$ *with the* δ_i*'s above, and* $\gamma \leq 6$ *and* $t \geq 0$ *satisfies:*

$$\mathbf{Pr}\left[\bigcap_{s \in [\beta_{j-1}, t + n\log^5 n]} \Phi_{j-1}^{(s)} \leq 2cn\right] \geq 1 - \frac{(\log n)^{8(j-1)}}{n^4},$$

where $\beta_j := t + 2jn\log^3 n$ *and* $c = c_{1/12,\alpha_2}$ *(see Theorem 5.3). Then, it also satisfies:*

$$\mathbf{Pr}\left[\bigcap_{s \in [\beta_j, t + n\log^5 n]} \Phi_j^{(s)} \leq 2cn\right] \geq 1 - \frac{(\log n)^{8j}}{n^4}.$$

As in Section 5, we will also use a second version of the potential function to extend an expected bound on the potential into a w.h.p. bound. Intuitively, we exploit the property that potential functions will have linear expectations for a range of coefficients. With this in mind, we define the following potential function for any $0 \leq j \leq k-1$,

$$\Psi_j^{(s)} := \sum_{i=1}^{n} \exp\left(\alpha_1 \cdot (\log n)^{j/k} \cdot \left(x_i^{(s)} - \frac{s}{n} - \frac{2}{\alpha_2} j (\log n)^{1/k}\right)^+\right),$$

where $\alpha_1 = 0.01$. Note that Ψ_j is defined in the same way as Φ_j with the only difference that α_1 is significantly larger α_2. The interplay between Ψ_j and Φ_j is similar to the interplay between Γ_1 and Γ_2 in the proof of Theorem 5.3, but some extra care is needed. In particular, while underloaded bins with load of $m/n - \Theta(\log n)$ contribute heavily to Γ_1 (or Γ_2), their contribution has to be eliminated here in order to derive a gap bound better than $\mathcal{O}(\log n)$.

6.3 Proof Outline of Lemma 6.6

We will now give a summary of the main technical steps in the proof of Lemma 6.6 (an illustration of the key steps is shown in Figure 5). On a high level, the proof mirrors the proof of Theorem 5.3; however, there are some differences, especially in the final part.

First, fix any $1 \leq j \leq k-1$. Then the inductive hypothesis ensures that $\Phi_{j-1}^{(r)}$ is small for $r \in [\beta_{j-1}, t + n\log^5 n]$. From that, it follows by a simple estimate that $\Psi_j^{(\beta_{j-1})} \leq e^{0.01\log^3 n}$ (Claim 6.13). Using a multiplicative drop (Lemma 6.8) repeatedly, it follows that there exists $u \in [\beta_{j-1}, \beta_{j-1} + n\log^3 n]$, $\mathbf{E}[\,\Psi_j^{(u)}\,] \leq cn$ (Lemma 6.10). Then by Lemma 6.11, this statement is extended to the time-interval $[\beta_{j-1} + n\log^3 n, t + n\log^5 n]$. By simply using Markov's inequality and a union bound, we can deduce that $\Psi_j^{(r)} \leq cn^{12}$ for all $r \in [\beta_{j-1} + n\log^3 n, t + n\log^5 n]$. By a simple relation between two potentials, this implies $\Phi_j^{(r)} \leq n^{4/3}$ (Claim 6.14 (ii)). Now using a multiplicative drop (Lemma 6.8) guarantees that this becomes $\Phi_j^{(r)} \leq cn$ w.h.p. for *a single* time-step $r \in [\beta_{j-1}, \beta_j]$ (Lemma 6.12).

To obtain the stronger statement which holds for all time-steps $r \in [\beta_{i-1}, \beta_j]$, we will use a concentration inequality. The key point is that whenever $\Psi_j^{(s)} \leq cn^{12}$, then the absolute difference $|\Phi_j^{(s+1)} - \Phi_j^{(s)}|$ is at most $n^{1/3}$, because $12.1\frac{\alpha_2}{\alpha_1} < 1/3$ (Claim 6.14 (ii)). This is crucial so that applying the supermartingale concentration bound Theorem 2.1 from [11] to Φ_j yields an $\mathcal{O}(n)$ guarantee for the entire time interval.

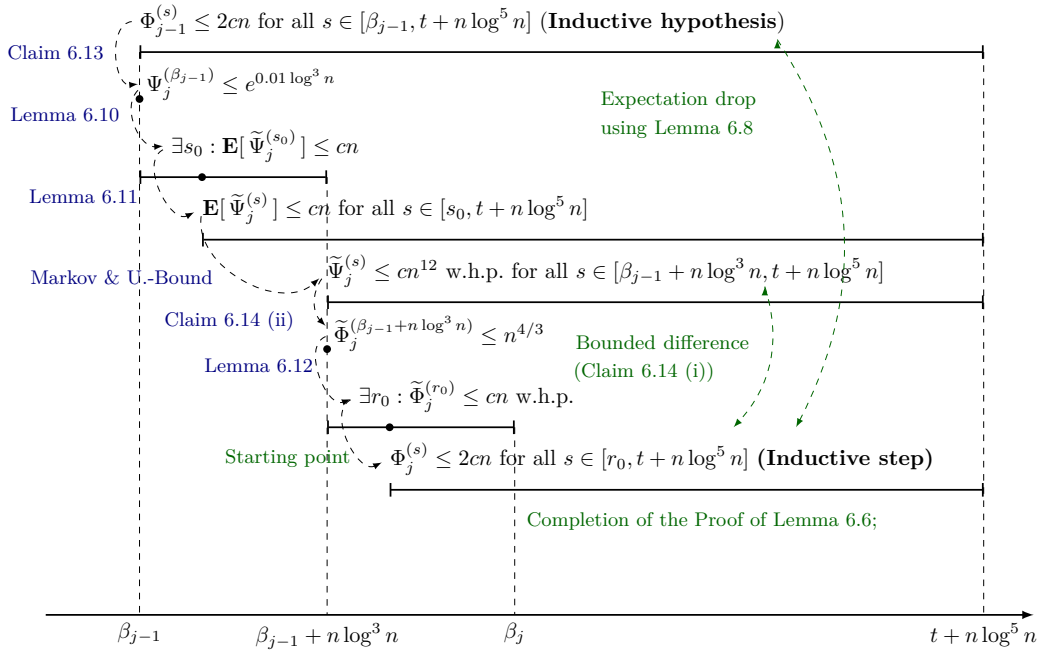


Figure 5 Outline for the proof of Lemma 6.6. Results in blue are given in Section 6.4, while results in green are used in the application of the concentration inequality (Theorem 2.1).

6.4 Auxiliary Definitions and Claims for the proof of Lemma 6.6

In the following, we will always implicitly assume that $1 \leq j \leq k-1$, as the case $j = 0$ has already been done. We define the following event, which will be used frequently in the proof:

$$\mathcal{E}_{j-1}^{(s)} := \left\{\Phi_{j-1}^{(s)} \leq 2cn\right\}.$$

Recall that the induction hypothesis asserts that $\mathcal{E}_{j-1}^{(s)}$ holds for all steps $s \in [\beta_{j-1}, t+n\log^5 n]$. In the following arguments we will be working frequently with the "killed" versions of the potentials, i.e., we condition on $\mathcal{E}_{j-1}^{(s)}$ holding on all time steps:

$$\widetilde{\Phi}_j^{(s)} := \Phi_j^{(s)} \cdot \mathbf{1}_{\cap_{r\in[\beta_{j-1},s]}\mathcal{E}_{j-1}^{(r)}} \text{ and } \widetilde{\Psi}_j^{(s)} := \Psi_j^{(s)} \cdot \mathbf{1}_{\cap_{r\in[\beta_{j-1},s]}\mathcal{E}_{j-1}^{(r)}}.$$

As the proof of Lemma 6.6 requires several claims and lemmas, the remainder of this section is divided further in:

1. Analysis of the (expected) drop of the potentials Φ_j and Ψ_j. (Section 6.4.1)
2. Auxiliary (Probabilistic) lemmas based on these drop results. (Section 6.4.2)
3. (Deterministic) inequalities that involve one or two potentials. (Section 6.4.3)

6.4.1 Analysis of the Drop of the Potentials Φ_j and Ψ_j

We define $\alpha_j^{(s)} := \frac{s}{n} + \frac{2}{\alpha_2}\cdot j(\log n)^{1/k}$, so that when $\mathcal{E}_{j-1}^{(s)}$ holds, then $y_{n\cdot\delta_{k-j}}^{(s)} \le \alpha_j^{(s)} - 1$; this will be established in the next lemma below.

▶ **Lemma 6.7.** *For any step $s \ge 1$, if $\mathcal{E}_{j-1}^{(s)}$ holds then $y_{n\cdot\delta_{k-j}}^{(s)} \le \alpha_j^{(s)} - 1$.*

▶ **Lemma 6.8.** *For any step $s \ge \beta_{j-1} = t + 2jn\log^3 n$, $\mathbf{E}\left[\Phi_j^{(s+1)} \mid \mathcal{E}_{j-1}^{(s)}, \Phi_j^{(s)}\right] \le \left(1-\frac{1}{n}\right)\cdot\Phi_j^{(s)} + 2$, and $\mathbf{E}\left[\Psi_j^{(s+1)} \mid \mathcal{E}_{j-1}^{(s)}, \Psi_j^{(s)}\right] \le \left(1-\frac{1}{n}\right)\cdot\Psi_j^{(s)} + 2$.*

▷ Claim 6.9. Let $\widetilde{\Phi}_j^{(s)}$, $\mathcal{E}_{j-1}^{(s)}$ and $\alpha_j^{(s)}$ be defined as in Lemma 6.8. Then for any bin $i \in [n]$ with $x_i^{(s)} \ge \alpha_j^{(s)}$, we get $\mathbf{Pr}\left[x_i^{(s+1)} = x_i^{(s)} + 1 \mid \widetilde{\Phi}_j^{(s)}, \mathcal{E}_{j-1}^{(s)}, x_i^{(s)} \ge \alpha_j^{(s)}\right] \le \frac{\gamma\delta}{n}$.

6.4.2 Auxiliary Probabilistic Lemmas on the Potential Functions

The first lemma proves that $\widetilde{\Psi}_j^{(s)}$ is small in expectation for at *at least one* time-step. It relies on the multiplicative drop (Lemma 6.8), and the fact that precondition $\cap_{r\in[\beta_{j-1},s]}\mathcal{E}_{j-1}^{(r)}$ holds due to the definition of the killed potential $\widetilde{\Psi}_{j-1}$.

▶ **Lemma 6.10.** *There exists $s \in [\beta_{j-1}, \beta_{j-1} + n\log^3 n]$ such that $\mathbf{E}[\,\widetilde{\Psi}_j^{(s)}\,] \le cn$.*

Generalizing the previous lemma, and again exploiting the conditioning on $\cap_{r\in[\beta_{j-1},s]}\mathcal{E}_{j-1}^{(r)}$ of $\Psi_j^{(s)}$, we know prove that $\widetilde{\Psi}_j^{(s)}$ is small in expectation for the entire time interval.

▶ **Lemma 6.11.** *For all $s \in [\beta_{j-1} + n\log^3 n, t + n\log^5 n]$, $\mathbf{E}[\,\widetilde{\Psi}_j^{(s)}\,] \le cn$.*

We now switch to the other potential function $\widetilde{\Phi}_j^{(s)}$, and prove that if it is polynomial in *at least one step*, then it is also linear in *at least one* step (not much later).

▶ **Lemma 6.12.** *For all $1 \le j < k$ it holds that,*

$$\mathbf{Pr}\left[\bigcup_{s\in[\beta_{j-1},\beta_j]}\{\widetilde{\Phi}_j^{(s)} \le cn\} \;\Big|\; \bigcup_{r\in[\beta_{j-1},\beta_{j-1}+n\log^3 n]}\{\widetilde{\Phi}_j^{(r)} \le n^{4/3}\}\right] \ge 1 - n^{-5}.$$

6.4.3 Deterministic Relations between the Potential Functions

We collect several basic facts about the potential functions $\Phi_j^{(s)}$ and $\Psi_j^{(s)}$.

▷ Claim 6.13. For any $s \geq 0$, $\Phi_j^{(s)} \leq 2cn$ implies $\Psi_{j+1}^{(s)} \leq \exp(0.01 \cdot \log^3 n)$.

The next claim is crucial for applying the concentration inequality, since the third statement bounds the maximum additive change of $\Phi^{(s)}$ (assuming $\Psi^{(s)}$ is small enough:

▷ Claim 6.14. For any $s \geq 0$, if $\Psi_j^{(s)} \leq cn^{12}$, then (i) $x_i^{(s)} \leq \frac{s}{n} + \frac{12.1}{\alpha_1} \cdot (\log n)^{\frac{k-j}{k}} + \frac{2}{\alpha_2} j (\log n)^{1/k}$ for all $i \in [n]$, (ii) $\Phi_j^{(s)} \leq n^{4/3}$ and (iii) $|\Phi_j^{(s+1)} - \Phi_j^{(s)}| \leq n^{1/3}$.

The next claim is a simple "smoothness" argument showing that the potential cannot decrease quickly within $n/\log^2 n$ steps. The derivation is elementary and relies on the fact that average load does not change by more than $1/\log^2 n$.

▷ Claim 6.15. For any $s \geq 0$ and any $r \in [s, s + n/\log^2 n]$, we have $\Phi_j^{(r)} \geq 0.99 \cdot \Phi_j^{(s)}$.

7 Applications of the Relaxed Quantile Process

In this section we present two implications of our analysis in Section 6, exploiting the flexibility of the *relaxed* version of the k-quantile process. The first implication is based on majorizing the $(1+\beta)$-process by a suitable relaxed k-quantile process, where k depends on β (see Lemma 7.3).

▶ **Theorem 7.1.** *Consider a $(1+\beta)$-process with $\beta \geq 1 - 2^{-0.5(\log n)^{(k-1)/k}}$ for some integer $1 \leq k \leq \kappa \cdot \log\log n$. Then for any $m \geq 1$,*

$$\mathbf{Pr}\left[\operatorname{Gap}(m) \leq 1000 \cdot k \cdot (\log n)^{1/k}\right] \geq 1 - n^{-3}.$$

In particular, if $\beta \geq 1 - n^{-c_1}$, for any (small) constant $c_1 > 0$, then there is a constant $c_2 = c_2(c_1) > 0$ such that the gap is at most $c_2 \cdot \log\log n$ w.h.p..

We can also derive an almost matching lower bound, showing that $1-\beta$ has to be (almost) polynomially small in order to achieve a gap of $\mathcal{O}(\log\log n)$ (Remark 7.4 in the appendix).

Our result for k quantiles can be also applied to graphical balanced allocations, where the graph is parameterized by its spectral expansion $\lambda \in [0, 1)$. Similar to the derivation of Theorem 7.1, the idea is to show that the graphical balanced allocation process can be majorized by a suitable relaxed k-quantile process.

▶ **Corollary 7.2** (special case of Theorem 7.8)**.** *Consider graphical balanced allocation on a d-regular graph with spectral expansion $\lambda \leq n^{-c_1}$ for a constant $c_1 > 0$. Then there is a constant $c_2 = c_2(c_1) > 0$ such that for any $m \geq 1$, $\mathbf{Pr}[\operatorname{Gap}(m) \leq c_2 \cdot \log\log n] \geq 1 - n^{-3}$.*

As shown in [33], for any $d = \operatorname{poly}(n)$, a random d-regular graph satisfies $\lambda = \mathcal{O}(1/\sqrt{d})$ w.h.p., and thus the gap bound above applies. For the lightly loaded case, [22] proved that any regular graph with degree at least $n^{\Omega(1/\log\log n)}$ achieves a gap $\mathcal{O}(\log\log n)$, and they also showed that this density is necessary. For the heavily loaded case, [28] proved a gap bound of $\mathcal{O}(\log n)$ for any expander. Hence Corollary 7.2 combines these lines of work, and establishes that the $\mathcal{O}(\log\log n)$ gap bound extends from complete graphs to dense and (strong) expanders.

7.1 $(1+\beta)$-Process for large β

We first relate the $(1+\beta)$-process to a relaxed quantile process.

▶ **Lemma 7.3.** *Consider a $(1+\beta)$-process with $\beta \geq 1 - 2^{-0.5(\log n)^{(k-1)/k}} = 1 - \widetilde{\delta}_1$ for some integer $k \geq 1$. Then this $(1+\beta)$-process is a $\textsc{Relaxed-Quantile}_\gamma(\delta_1, \ldots, \delta_k)$ process, where each δ_i is $\widetilde{\delta}_i$ being rounded up to the nearest multiple of $\frac{1}{n}$ and $\gamma = 3$.*

Using the above lemma, majorization and Theorem 6.5 yields immediately:

▶ **Theorem 7.1.** *Consider a $(1+\beta)$-process with $\beta \geq 1 - 2^{-0.5(\log n)^{(k-1)/k}}$ for some integer $1 \leq k \leq \kappa \cdot \log\log n$. Then for any $m \geq 1$,*

$$\mathbf{Pr}\left[\operatorname{Gap}(m) \leq 1000 \cdot k \cdot (\log n)^{1/k}\right] \geq 1 - n^{-3}.$$

In particular, if $\beta \geq 1 - n^{-c_1}$, for any (small) constant $c_1 > 0$, then there is a constant $c_2 = c_2(c_1) > 0$ such that the gap is at most $c_2 \cdot \log\log n$ w.h.p..

It is straightforward to derive an almost matching lower bound on the gap, showing that $1-\beta$ has to be (almost) polynomially small in order to achieve a gap of $\mathcal{O}(\log\log n)$:

▶ **Remark 7.4.** *Consider a $(1+\beta)$-process with $\beta \leq 1 - n^{-c_3/\log\log n}$ for some $c_3 > 0$ (not necessarily constant). Then, $\mathbf{Pr}\left[\operatorname{Gap}(n) \geq \frac{2}{c_3}\log\log n\right] \geq 1 - o(1)$.*

7.2 Graphical Balanced Allocation

We now analyze the *graphical balanced allocation process*, with a focus on dense expander graphs. To this end, we first recall some basic notation of spectral graph theory and expansion. For an undirected graph G, the normalized Laplacian Matrix of G is an $n \times n$-matrix defined by $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2} \cdot \mathbf{A} \cdot \mathbf{D}^{1/2}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{A}$ is the adjacency matrix and $\mathbf{D}$ is the diagonal matrix where $D_{u,u} = \deg(u)$ for any vertex $u \in V$. Further, let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the n eigenvalues of $\mathbf{L}$, and let $\lambda := \max_{i\in[2,n]} |1-\lambda_i|$ be the spectral expansion of G. Further, for any set $U \subseteq V$ define $\operatorname{vol}(U) := \sum_{v\in U}\deg(v)$. Note that for a d-regular graph, we have $\operatorname{vol}(U) = d\cdot|U|$ and $\operatorname{vol}(V) = dn$.

We now recall the following (stronger) version of the Expander Mixing Lemma (cf. [12]):

▶ **Lemma 7.5** (Expander Mixing Lemma)**.** *For any subsets $X, Y \subseteq V$,*

$$\left||E(X,Y)| - \frac{\operatorname{vol}(X)\cdot\operatorname{vol}(Y)}{\operatorname{vol}(V)}\right| \leq \lambda \cdot \frac{\sqrt{\operatorname{vol}(X)\cdot\operatorname{vol}(\overline{X})\cdot\operatorname{vol}(Y)\cdot\operatorname{vol}(\overline{Y})}}{\operatorname{vol}(V)},$$

where $\operatorname{vol}(\overline{X}) = \operatorname{vol}(V\setminus X)$.

In the following, we consider G to be a d-regular graph.

▶ **Proposition 7.6.** *Consider the probability vector p_i^t, $1 \leq i \leq n$ of a graphical balanced allocation process on a d-regular graph G with spectral expansion λ. Then this vector satisfies for any load configuration at any time t the following three inequalities.*

1. *For any $1 \leq j \leq \lambda\cdot n$, $\sum_{i=1}^{j} p_i^t \leq 2\lambda\cdot\frac{j}{n}$.*
2. *For any $\lambda\cdot n \leq j$, $\sum_{i=1}^{j} p_i^t \leq 2\cdot\left(\frac{j}{n}\right)^2$.*
3. *For any $1 \leq j \leq n$, $\sum_{i=1}^{j} p_i^t \leq \frac{j}{n}\cdot\left(1-(1-\lambda)\cdot\frac{n-j}{n}\right)$.*

▶ **Lemma 7.7.** *Consider a graphical balanced allocation process on a connected, d-regular graph on G with spectral expansion $\lambda \leq 1/2$. Further, let $2^{-0.5(\log n)^{(k-1)/k}} \geq \lambda$ for an integer $k \geq 1$. Then there exists a process in the class $\textsc{Relaxed-Quantile}_\gamma(\delta_1, \ldots, \delta_k)$, where each δ_i is $\widetilde{\delta}_i$ being rounded up to the nearest multiple of $\frac{1}{n}$ and $\gamma = 2$, which majorizes the probability vector of the graphical balanced allocation process in each round $t \geq 1$, for any possible load configuration.*

▶ **Theorem 7.8.** *Consider a graphical balanced allocation process on a connected, d-regular graph on G with spectral expansion $\lambda \leq 1/2$. Further, let $k \in [1, k \leq \kappa \cdot \log\log n]$ be the largest integer such that $2^{-0.5(\log n)^{(k-1)/k}} \geq \widetilde{\lambda} := \max\{\lambda, n^{-0.00005}\}$. Then for any $m \geq 1$,*

$$\mathbf{Pr}\left[\mathrm{Gap}(m) \leq 1000 \cdot k \cdot \left(\frac{\log n}{\log(1/\widetilde{\lambda})}\right)^{(k+1)/k}\right] \geq 1 - n^{-3}.$$

From the general bound in the above corollary, we can deduce the following two bounds:

▶ **Remark 7.9.** *Under the assumptions of Theorem 7.8, we have the following more explicit (but slightly weaker) bound for any $2 \leq k \leq \kappa \cdot \log\log n$,*

$$\mathbf{Pr}\left[\mathrm{Gap}(m) \leq 1000 \cdot \frac{\log\log n}{\log\log n - \log\log(1/\widetilde{\lambda}) + \log(0.5)} \cdot \left(\frac{\log n}{\log(1/\widetilde{\lambda})}\right)^{3/2}\right] \geq 1 - n^{-3}.$$

Also if $\lambda \leq 1/2^{(\log n)^{c_1}}$ for some constant $0 < c_1 < 1$, then

$$\mathbf{Pr}\left[\mathrm{Gap}(m) \leq 1000 \cdot \frac{1}{\log(10^4)} \log\log n \cdot (\log n)^{(3/2)\cdot(1-c_1)}\right] \geq 1 - n^{-3}.$$

Finally, let us consider the case where λ decays polynomially in n.

▶ **Corollary 7.2** (special case of Theorem 7.8)**.** *Consider graphical balanced allocation on a d-regular graph with spectral expansion $\lambda \leq n^{-c_1}$ for a constant $c_1 > 0$. Then there is a constant $c_2 = c_2(c_1) > 0$ such that for any $m \geq 1$, $\mathbf{Pr}\left[\mathrm{Gap}(m) \leq c_2 \cdot \log\log n\right] \geq 1 - n^{-3}$.*

Note that $\lambda \leq n^{-c_1}$ captures a *relaxed*, *multiplicative* approximation of Ramanujan graphs (it is in fact more relaxed than the existing notion "weakly Ramanujan"). Recently, [33] proved that for any $\mathrm{poly}(n) \leq d \leq n/2$, a random d-regular graph satisfies the constraint on λ with probability at least $1 - n^{-1}$.

Further, we remark that the above result extends one of the main results of [22] which states that for any graph with degree $n^{1/\log\log n}$, graphical balanced allocation achieves a gap of at most $\Theta(\log\log n)$ in the lightly loaded case ($m = n$). Our result above also refines a previous result of [28] which states that for any expander graph, a gap bound of $\mathcal{O}(\log n)$ holds (even in the heavily loaded case $m \geq n$). In conclusion, we see that the gap bound of $\mathcal{O}(\log\log n)$ extends from the complete graph (which is the Two-Choice process) to other graphs, provided we have a strong expansion and high density.

8 Conclusions

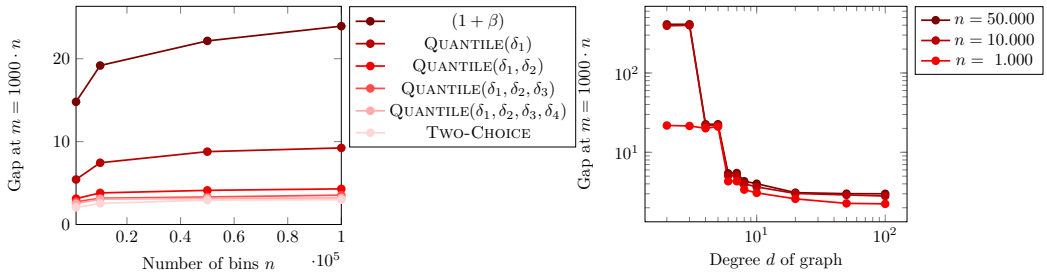
In this work, we introduced a new framework of balls-and-bins with incomplete information. The main contributions are as follows:

1. A lower bound of $\Omega(\sqrt{\log n})$ for a fixed $m = \Theta(n\sqrt{\log n})$ for one adaptive query (Theorem 4.4), disproving Problem 1.3 in [16]. Also, a stronger lower bound of $\Omega(\log n / \log\log n)$ for "many" time-steps in $[1, n\log^2 n]$ (Corollary 4.2), again for one adaptive query.
2. Design and analysis of an instance of the k-quantile process for any $k \geq 1$. This process performs well empirically (Section 9), and achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for any $m \geq 1$ and $k = \mathcal{O}(\log\log n)$ (Theorem 6.1). This theoretical result has several implications:
   - A "power of two queries" phenomenon: reduction of the gap from $\Omega(\log n/\log\log n)$ to $\mathcal{O}(\sqrt{\log n})$ by increasing the number of queries from one to two.
   - For $k = \Theta(\log\log n)$, a gap bound of $\mathcal{O}(\log\log n)$ which matches the gap of the process with full information (Two-Choice) up to multiplicative constants.
   - New upper bounds on the gap of the $(1+\beta)$ process with β close to 1 by relating it to a Relaxed-Quantile process (Theorem 7.1).
   - New upper bounds on the graphical balanced allocation on dense expander graphs, making progress towards Open Question 2 in [28] (Corollary 7.2).
3. Several majorizations and reductions between the processes Quantile, Threshold, Relaxed-Quantile, Thinning, $(1+\beta)$ and Two-Choice (see Figure 2 for a high-level outline, and Section 3 for more details).

One natural open question is whether we can prove matching lower bounds, in particular, the case $k \geq 2$ is wide open. Another interesting direction is to investigate other allocation processes with limited information, e.g., where a sampled bin reports its actual load perturbed by some random or deterministic noise function.

9 Experimental Results

In Table 1 and Figure 6a, we also recorded the empirical distribution of the gap for $m = 1000 \cdot n$ balls for the $(1+\beta)$ with $\beta = 1/2$, the k-Quantile (for $k = 1, 2, 3, 4$) of the form defined in Section 6, and the Two-Choice process. The experiments show a large improvement of $k = 2$ over $k = 1$ ("Power of Two Queries"). Figure 6b shows empirical evidence that the gap decreases (and approaching closely the Two-Choice gap) in regular graphs as the degree increases.



(a) Balanced allocation on complete graphs.

(b) Balanced allocation on random d-regular graphs.

Figure 6 (a) Average Gap vs. $n \in \{10^3, 10^4, 5 \cdot 10^4, 10^5\}$ for the experimental setup of Table 1 and (b) Average Gap vs. $n \in \{10^3, 10^4, 5 \cdot 10^4\}$ for regular graphs generated using [29].

Table 1 Summary of our Experimental Results ($m = 1000 \cdot n$).

n	$(1+\beta)$, for $\beta = 0.5$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	Two-Choice
10^3	**12** : 5% **13** : 15% **14** : 31% **15** : 21% **16** : 15% **17** : 5% **18** : 4% **19** : 2% **20** : 1% **21** : 1%	**3** : 1% **4** : 11% **5** : 46% **6** : 33% **7** : 6% **8** : 2% **10** : 1%	**2** : 4% **3** : 80% **4** : 16%	**2** : 24% **3** : 74% **4** : 2%	**2** : 50% **3** : 49% **4** : 1%	**2** : 93% **3** : 7%
10^4	**16** : 3% **17** : 21% **18** : 19% **19** : 10% **20** : 23% **21** : 11% **22** : 10% **23** : 2% **24** : 1%	**6** : 14% **7** : 42% **8** : 25% **9** : 15% **10** : 2% **11** : 1% **12** : 1%	**3** : 27% **4** : 65% **5** : 8%	**3** : 83% **4** : 17%	**3** : 95% **4** : 5%	**2** : 46% **3** : 54%
10^5	**20** : 2% **21** : 7% **22** : 9% **23** : 26% **24** : 27% **25** : 14% **26** : 6% **27** : 3% **28** : 4% **29** : 1% **34** : 1%	**8** : 28% **9** : 42% **10** : 18% **11** : 7% **12** : 3% **14** : 1% **15** : 1%	**4** : 72% **5** : 26% **6** : 2%	**3** : 46% **4** : 54%	**3** : 79% **4** : 21%	**3** : 100%

References

1 Dan Alistarh, Trevor Brown, Justin Kopinsky, Jerry Zheng Li, and Giorgi Nadiradze. Distributionally linearizable data structures. In *Proceedings of 30th on Symposium on Parallelism in Algorithms and Architectures (SPAA'18)*, pages 133–142, 2018. doi:10.1145/3210377.3210411.

2 Dan Alistarh, Rati Gelashvili, and Joel Rybicki. Fast graphical population protocols, 2021. arXiv:2102.08808.

3 Dan Alistarh, Giorgi Nadiradze, and Amirmojtaba Sabour. Dynamic averaging load balancing on cycles. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP'20)*, volume 168, pages 7:1–7:16, 2020. doi:10.4230/LIPIcs.ICALP.2020.7.

4 Noga Alon, Ori Gurel-Gurevich, and Eyal Lubetzky. Choice-memory tradeoff in allocations. *Ann. Appl. Probab.*, 20(4):1470–1511, 2010. doi:10.1214/09-AAP656.

5 Yossi Azar, Andrei Z. Broder, Anna R. Karlin, and Eli Upfal. Balanced allocations. *SIAM J. Comput.*, 29(1):180–200, 1999. doi:10.1137/S0097539795288490.

6 Nikhil Bansal and Ohad Feldheim. Well-balanced allocation on general graphs, 2021. arXiv:2106.06051.

7 Itai Benjamini and Yury Makarychev. Balanced allocation: memory performance tradeoffs. *Ann. Appl. Probab.*, 22(4):1642–1649, 2012. doi:10.1214/11-AAP804.

8 Petra Berenbrink, Artur Czumaj, Matthias Englert, Tom Friedetzky, and Lars Nagel. Multiple-choice balanced allocation in (almost) parallel. In *Proceedings of 16th International Workshop on Approximation, Randomization, and Combinatorial Optimization (RANDOM'12)*, pages 411–422, 2012. doi:10.1007/978-3-642-32512-0_35.

9 Petra Berenbrink, Artur Czumaj, Angelika Steger, and Berthold Vöcking. Balanced allocations: the heavily loaded case. *SIAM J. Comput.*, 35(6):1350–1385, 2006. doi:10.1137/S009753970444435X.

10 Petra Berenbrink, Kamyar Khodamoradi, Thomas Sauerwald, and Alexandre Stauffer. Balls-into-bins with nearly optimal load distribution. In *Proceedings of 25th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'13)*, pages 326–335, 2013. doi:10.1145/2486159.2486191.

11 Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Math.*, 3(1):79–127, 2006. doi:10.1080/15427951.2006.10129115.

12 Fan R. K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC; by the American Mathematical Society, Providence, RI, 1997. doi:10.1090/cbms/092.

13 Artur Czumaj and Volker Stemann. Randomized allocation processes. *Random Structures Algorithms*, 18(4):297–331, 2001. doi:10.1002/rsa.1011.

14 D. L. Eager, E. D. Lazowska, and J. Zahorjan. Adaptive load sharing in homogeneous distributed systems. *IEEE Transactions on Software Engineering*, SE-12(5):662–675, 1986. doi:10.1109/TSE.1986.6312961.

15 Guy Even and Moti Medina. Parallel randomized load balancing: a lower bound for a more general model. *Theoret. Comput. Sci.*, 412(22):2398–2408, 2011. doi:10.1016/j.tcs.2011.01.033.

16 Ohad N. Feldheim and Ori Gurel-Gurevich. The power of thinning in balanced allocation. *Electron. Commun. Probab.*, 26:Paper No. 34, 8, 2021. doi:10.1214/21-ecp400.

17 Ohad N. Feldheim, Ori Gurel-Gurevich, and Jiange Li. Long-term balanced allocation via thinning, 2021. arXiv:2110.05009.

18 Ohad Noy Feldheim and Jiange Li. Load balancing under d-thinning. *Electronic Communications in Probability*, 25:Paper No. 1, 13, 2020. doi:10.1214/19-ecp282.

19 Kazuo Iwama and Akinori Kawachi. Approximated two choices in randomized load balancing. In *Proceedings of 15th International Symposium on Algorithms and Computation (ISAAC'04)*, volume 3341, pages 545–557. Springer-Verlag, 2004. doi:10.1007/978-3-540-30551-4_48.

20 Y. Kanizo, D. Raz, and A. Zlotnik. Efficient use of geographically spread cloud resources. In *Proceedings of 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, pages 450–457, 2013. doi:10.1109/CCGrid.2013.18.

21 R. M. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. *Algorithmica*, 16(4-5):517–542, 1996. doi:10.1007/BF01940878.

22 Krishnaram Kenthapadi and Rina Panigrahy. Balanced allocation on graphs. In *Proceedings of 17th ACM-SIAM Symposium on Discrete Algorithms (SODA'06)*, pages 434–443, 2006. doi:10.1145/1109557.1109606.

23 Christoph Lenzen, Merav Parter, and Eylon Yogev. Parallel balanced allocations: The heavily loaded case. In *Proceedings of the 31st ACM on Symposium on Parallelism in Algorithms and Architectures (SPAA'19)*, pages 313–322. ACM, 2019. doi:10.1145/3323165.3323203.

24 Christoph Lenzen and Roger Wattenhofer. Tight bounds for parallel randomized load balancing [extended abstract]. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC'11)*, pages 11–20, 2011. doi:10.1145/1993636.1993639.

25 Dimitrios Los, Thomas Sauerwald, and John Sylvester. Balanced allocations: Caching and packing, twinning and thinning, 2021. arXiv:2110.10759.

26 M. Mitzenmacher. On the analysis of randomized load balancing schemes. *Theory Comput. Syst.*, 32(3):361–386, 1999. doi:10.1007/s002240000122.

27 Michael Mitzenmacher, Andréa W. Richa, and Ramesh Sitaraman. The power of two random choices: a survey of techniques and results. In *Handbook of randomized computing, Vol. I, II*, volume 9 of *Comb. Optim.*, pages 255–312. Kluwer Acad. Publ., Dordrecht, 2001. doi:10.1007/978-1-4615-0013-1_9.

28 Yuval Peres, Kunal Talwar, and Udi Wieder. Graphical balanced allocations and the $(1+\beta)$-choice process. *Random Structures Algorithms*, 47(4):760–775, 2015. doi:10.1002/rsa.20558.

29 A. Steger and N. C. Wormald. Generating random regular graphs quickly. *Combinatorics, Probability and Computing*, 8(4):377–396, 1999. doi:10.1017/S0963548399003867.

30 Volker Stemann. Parallel balanced allocations. In *Proceedings of the 8th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'96)*, pages 261–269, 1996. doi:10.1145/237502.237565.

31 Kunal Talwar and Udi Wieder. Balanced allocations: the weighted case. In *Proceedings of 39th ACM Symposium on Theory of Computing (STOC'07)*, pages 256–265, 2007. doi:10.1145/1250790.1250829.

32 Kunal Talwar and Udi Wieder. Balanced allocations: a simple proof for the heavily loaded case. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP'14)*, volume 8572, pages 979–990, 2014. doi:10.1007/978-3-662-43948-7_81.

33 Konstantin Tikhomirov and Pierre Youssef. The spectral gap of dense random regular graphs. *Ann. Probab.*, 47(1):362–419, 2019. doi:10.1214/18-AOP1263.

34 Udi Wieder. Hashing, load balancing and multiple choice. *Found. Trends Theor. Comput. Sci.*, 12(3-4):275–379, 2017. doi:10.1561/0400000070.

35 S. Zhou. A trace-driven simulation study of dynamic load balancing. *IEEE Transactions on Software Engineering*, 14(9):1327–1341, 1988. doi:10.1109/32.6176.