Balanced Allocations under Incomplete Information

Dimitrios Los[1], Thomas Sauerwald[1], John Sylvester[2]

[1]University of Cambridge, UK
[2]University of Glasgow, UK

Balanced allocations: Background

Balanced allocations setting

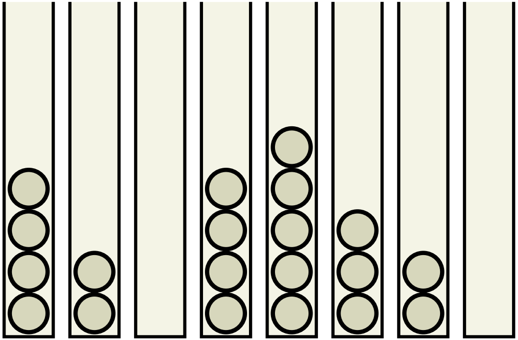
Allocate m tasks (balls) sequentially into n machines (bins).

Balanced allocations setting

Allocate m tasks (balls) sequentially into n machines (bins).
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.

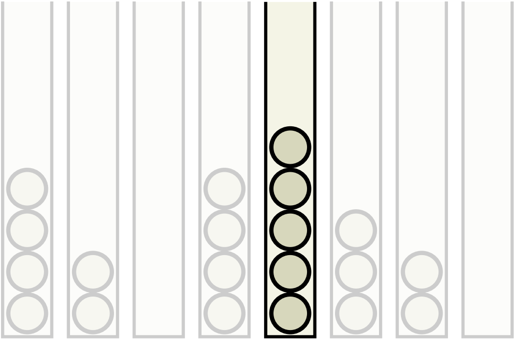
Balanced allocations setting

Allocate m tasks (balls) sequentially into n machines (bins).
Goal: minimise the **maximum load** $\max_{i\in[n]} x_i^m$, where x^t is the load vector after ball t.



Balanced allocations setting

Allocate m tasks (balls) sequentially into n machines (bins).
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.

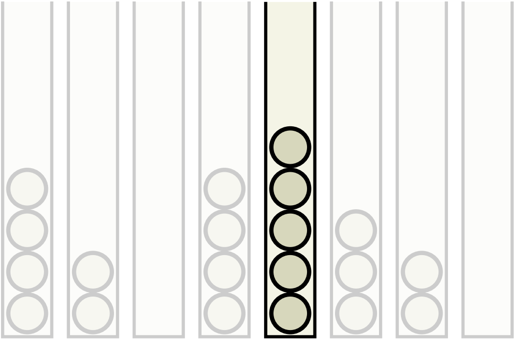


Balanced allocations setting

Allocate m tasks (balls) sequentially into n machines (bins).

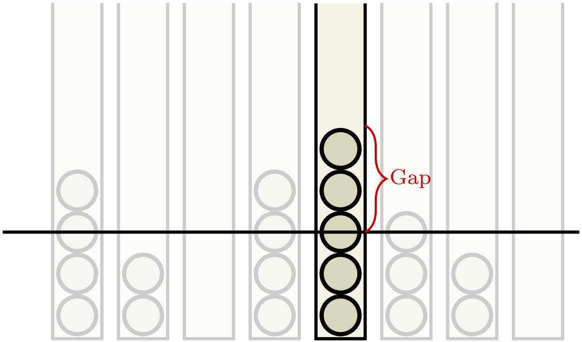
Goal: minimise the **maximum load** $\max_{i\in[n]} x_i^m$, where x^t is the load vector after ball t.

$\Leftrightarrow$ minimise the **gap**, where $\text{Gap}(m) = \max_{i\in[n]}(x_i^m - m/n)$.



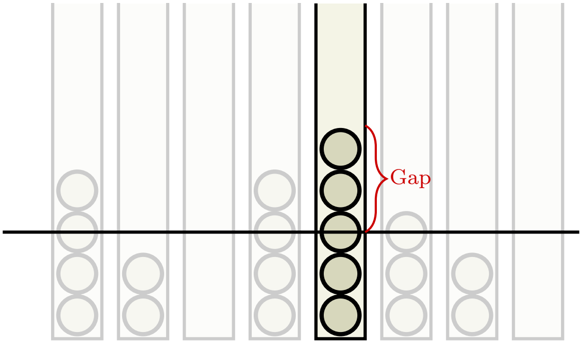
Balanced allocations setting

Allocate m tasks (balls) sequentially into n machines (bins).
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.
$\Leftrightarrow$ minimise the **gap**, where $\text{Gap}(m) = \max_{i \in [n]}(x_i^m - m/n)$.



Balanced allocations setting

Allocate m tasks (balls) sequentially into n machines (bins).
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.
$\Leftrightarrow$ minimise the **gap**, where $\text{Gap}(m) = \max_{i \in [n]}(x_i^m - m/n)$.



- Applications in hashing, load balancing and routing.

Outline of the presentation

- **Part A:** Definition of One-Choice, Two-Choice, and the $(1+\beta)$ process.
- **Part B:** The Quantile Process
- **Part C:** The Mean-Threshold Process
- **Part D:** Applications: Outdated information and Noise

One-Choice and Two-Choice processes

<u>One-Choice Process:</u>

Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

ONE-CHOICE and TWO-CHOICE processes

<u>ONE-CHOICE PROCESS:</u>
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\text{Gap}(n) = \Theta\left(\frac{\log n}{\log\log n}\right)$ [Gon81].

ONE-CHOICE and TWO-CHOICE processes

ONE-CHOICE PROCESS:
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case $(m = n)$, w.h.p. $\text{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].

Meaning with probability at least $1 - n^{-c}$ for constant $c > 0$.

One-Choice and Two-Choice processes

<u>One-Choice Process:</u>
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\text{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\text{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right)$ (e.g. [RS98]).

ONE-CHOICE and TWO-CHOICE processes

ONE-CHOICE PROCESS:
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\text{Gap}(n) = \Theta\left(\frac{\log n}{\log\log n}\right)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\text{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right)$ (e.g. [RS98]).

TWO-CHOICE PROCESS:
Iteration: For each $t \geq 0$, sample **two** bins independently u.a.r. and place the ball in the least loaded of the two.

One-Choice and Two-Choice processes

<u>One-Choice Process:</u>
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right)$ (e.g. [RS98]).

<u>Two-Choice Process:</u>
Iteration: For each $t \geq 0$, sample **two** bins independently u.a.r. and place the ball in the least loaded of the two.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \log_2 \log n + \Theta(1)$ [KLMadH96, ABKU99].

One-Choice and Two-Choice processes

<u>One-Choice Process:</u>
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right)$ (e.g. [RS98]).

<u>Two-Choice Process:</u>
Iteration: For each $t \geq 0$, sample **two** bins independently u.a.r. and place the ball in the least loaded of the two.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \log_2 \log n + \Theta(1)$ [KLMadH96, ABKU99].

ONE-CHOICE and TWO-CHOICE processes

ONE-CHOICE PROCESS:
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right)$ (e.g. [RS98]).

TWO-CHOICE PROCESS:
Iteration: For each $t \geq 0$, sample **two** bins independently u.a.r. and place the ball in the least loaded of the two.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \log_2 \log n + \Theta(1)$ [KLMadH96, ABKU99].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \log_2 \log n + \Theta(1)$ [BCSV06].

One-Choice and Two-Choice processes

One-Choice Process:
Iteration: For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \Theta\left(\frac{\log n}{\log\log n}\right)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right)$ (e.g. [RS98]).

Two-Choice Process:
Iteration: For each $t \geq 0$, sample **two** bins independently u.a.r. and place the ball in the least loaded of the two.

- In the lightly-loaded case ($m = n$), w.h.p. $\mathrm{Gap}(n) = \log_2 \log n + \Theta(1)$ [KLMadH96, ABKU99].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \log_2 \log n + \Theta(1)$ [BCSV06].

$(1+\beta)$ process: Definition

$(1+\beta)$ process:
Parameter: A probability $\beta \in (0,1]$.
Iteration: For each $t \geq 0$, with probability β allocate one ball via the Two-Choice process, otherwise allocate one ball via the One-Choice process.

$(1+\beta)$ process: Definition

$(1+\beta)$ process:
Parameter: A probability $\beta \in (0, 1]$.
Iteration: For each $t \geq 0$, with probability β allocate one ball via the Two-Choice process, otherwise allocate one ball via the One-Choice process.

- [Mit99] interpreted $(1-\beta)/2$ as the probability of making an erroneous comparison.

$(1+\beta)$ process: Definition

$(1+\beta)$ process:
Parameter: A probability $\beta \in (0,1]$.
Iteration: For each $t \geq 0$, with probability β allocate one ball via the Two-Choice process, otherwise allocate one ball via the One-Choice process.

- [Mit99] interpreted $(1-\beta)/2$ as the probability of making an erroneous comparison.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\Theta(\log n/\beta)$ for $\beta < 1-\epsilon$ for constant $\epsilon > 0$.

The Quantile Process

Quantile(δ) Process

Quantile(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } \mathrm{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{otherwise.} \end{cases}$$

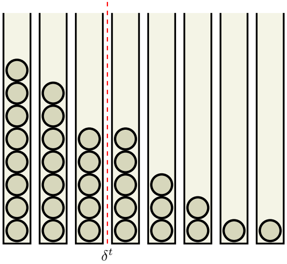
Quantile(δ) **Process**

Quantile(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^t + 1 & \text{if } \mathrm{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^t + 1 & \text{otherwise.} \end{cases}$$



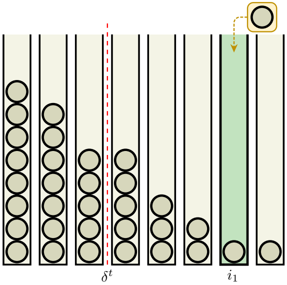
Quantile(δ) Process

Quantile(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } \mathrm{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{otherwise.} \end{cases}$$



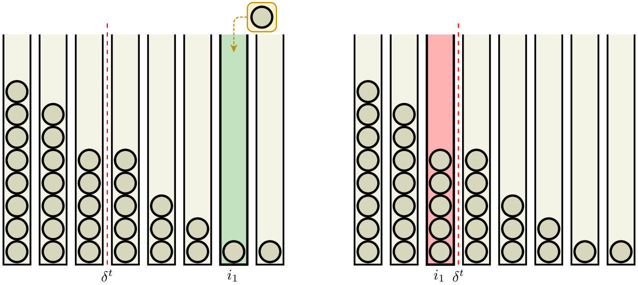
QUANTILE(δ) Process

QUANTILE(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^t + 1 & \text{if } \mathrm{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^t + 1 & \text{otherwise.} \end{cases}$$



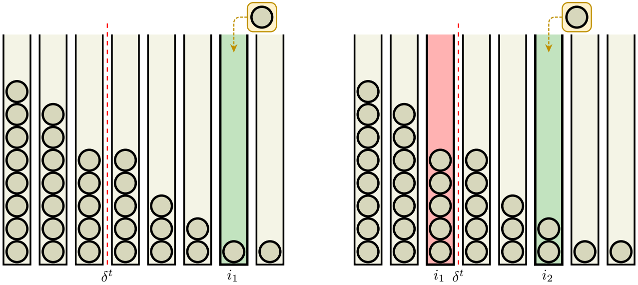
Quantile(δ) Process

Quantile(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^t + 1 & \text{if } \mathrm{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^t + 1 & \text{otherwise.} \end{cases}$$



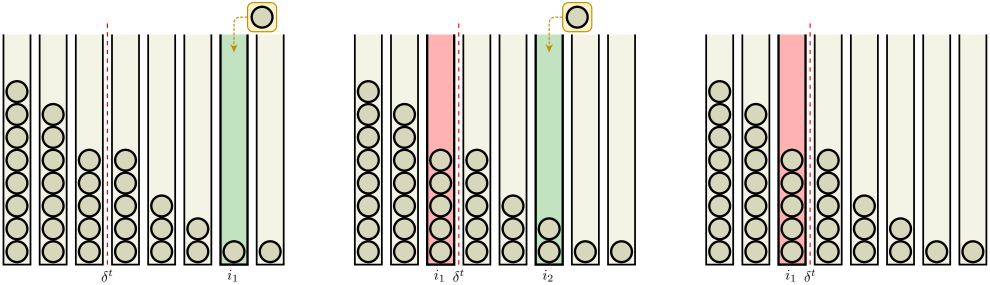
Quantile(δ) **Process**

Quantile(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } \text{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{otherwise.} \end{cases}$$



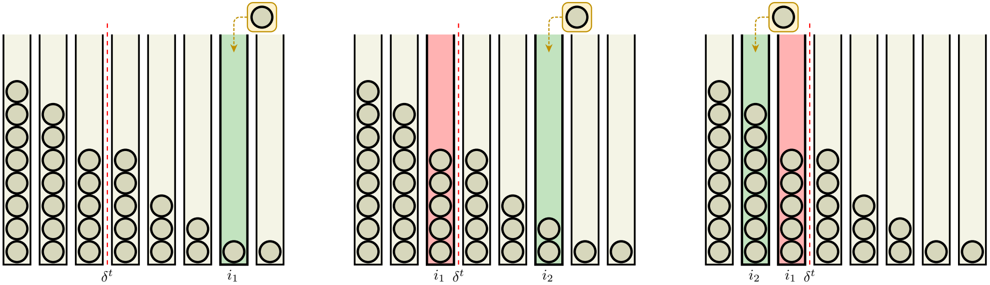
Quantile(δ) Process

Quantile(δ) Process:

Parameter: A quantile $\delta \in \{1/n, 2/n, \ldots, 1\}$.

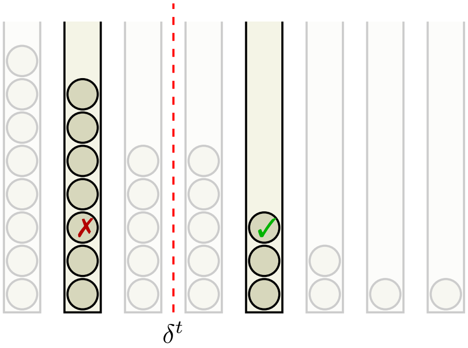
Iteration: For $t \geq 0$, sample two bins independently u.a.r. i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } \mathrm{Rank}(x^t, i_1) > \delta \cdot n, \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{otherwise.} \end{cases}$$



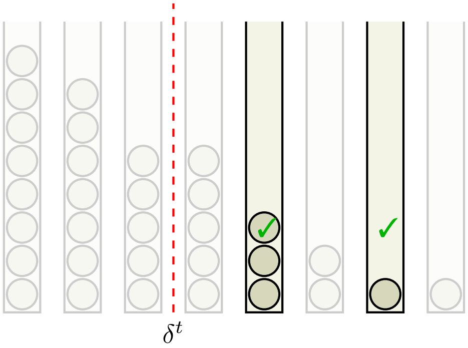
Quantile(δ) as Two-Choice with incomplete information

We can interpret Quantile(δ) as an instance of the Two-Choice process, where we are only able to compare the loads of the two sampled bins if one is above the quantile and one is below.



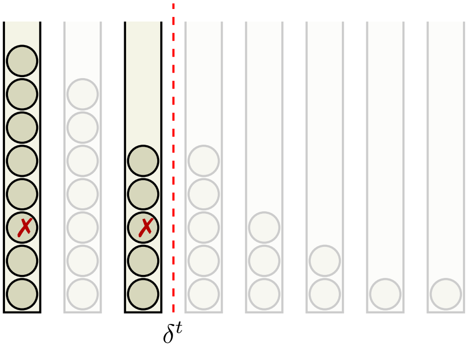
Quantile(δ) as Two-Choice with incomplete information

We can interpret Quantile(δ) as an instance of the Two-Choice process, where we are only able to compare the loads of the two sampled bins if one is above the quantile and one is below.



Quantile(δ) as Two-Choice with incomplete information

We can interpret Quantile(δ) as an instance of the Two-Choice process, where we are only able to compare the loads of the two sampled bins if one is above the quantile and one is below.

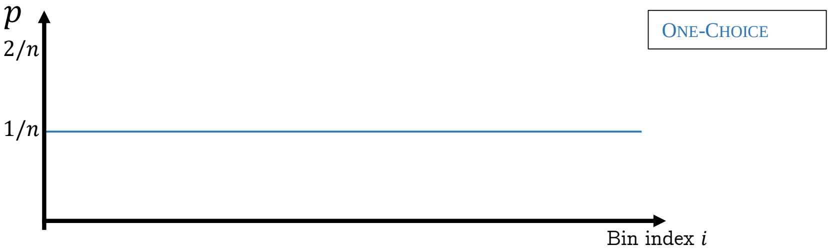


Probability allocation vectors

- **Probability allocation vector** p^t, where p_i^t is the prob. of allocating to i-th most loaded bin.

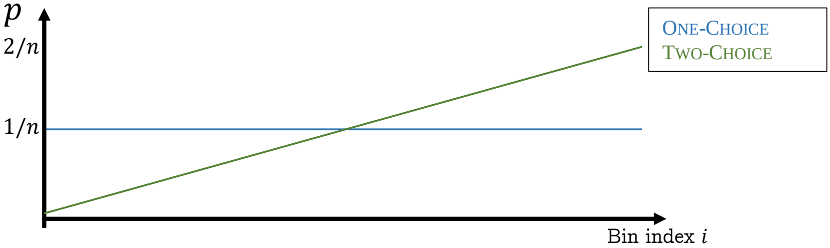
Probability allocation vectors

- **Probability allocation vector** p^t, where p_i^t is the prob. of allocating to i-th most loaded bin.
- For One-Choice, $p_{\text{One-Choice}} = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$.



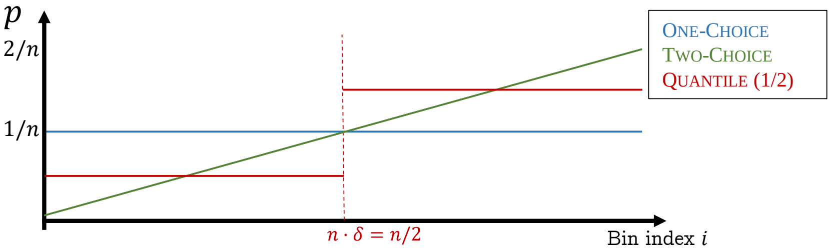
Probability allocation vectors

- **Probability allocation vector** p^t, where p_i^t is the prob. of allocating to i-th most loaded bin.
- For One-Choice, $p_{\textsc{One-Choice}} = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$.
- For Two-Choice, $p_{\textsc{Two-Choice}} = \left(\frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i-1}{n^2}, \ldots, \frac{2n-2}{n^2}\right)$.



Probability allocation vectors

- **Probability allocation vector** p^t, where p_i^t is the prob. of allocating to i-th most loaded bin.
- For One-Choice, $p_{\text{One-Choice}} = \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$.
- For Two-Choice, $p_{\text{Two-Choice}} = \left(\frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i-1}{n^2}, \ldots, \frac{2n-2}{n^2}\right)$.
- For Quantile(δ), $p_{\text{Quantile}(\delta)} = \Big(\underbrace{\frac{\delta}{n}, \ldots, \frac{\delta}{n}}_{\delta \cdot n \text{ entries}}, \underbrace{\frac{1+\delta}{n}, \ldots, \frac{1+\delta}{n}}_{(1-\delta) \cdot n \text{ entries}}\Big)$.



The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}} .$$

The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}} .$$

- [PTW15] show that $\mathbf{E}\left[\,\Gamma^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.

The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}} .$$

- [PTW15] show that $\mathbf{E}\left[\,\Gamma^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- This implies $\mathbf{E}\left[\,\Gamma^t\,\right] \leq c \cdot n$ for any $t \geq 0$.

The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}} .$$

- [PTW15] show that $\mathbf{E}\left[\, \Gamma^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- This implies $\mathbf{E}\left[\, \Gamma^t \,\right] \leq c \cdot n$ for any $t \geq 0$.
- By Markov's inequality, we get $\mathbf{Pr}\left[\, \Gamma^m \leq cn^3 \,\right] \geq 1 - n^{-2}$ which implies

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

The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}}.$$

- [PTW15] show that $\mathbf{E}\left[\, \Gamma^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- This implies $\mathbf{E}\left[\, \Gamma^t \,\right] \leq c \cdot n$ for any $t \geq 0$.
- By Markov's inequality, we get $\mathbf{Pr}\left[\, \Gamma^m \leq cn^3 \,\right] \geq 1 - n^{-2}$ which implies

$$\mathbf{Pr}\left[\text{Gap}(m) \leq \frac{1}{\alpha}(3 \cdot \log n + \log c) \right] \geq 1 - n^{-2}.$$

- For the $(1+\beta)$ process, $\alpha = \Theta(\beta)$.

The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}} .$$

- [PTW15] show that $\mathbf{E}\left[\, \Gamma^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- This implies $\mathbf{E}\left[\, \Gamma^t \,\right] \leq c \cdot n$ for any $t \geq 0$.
- By Markov's inequality, we get $\mathbf{Pr}\left[\, \Gamma^m \leq cn^3 \,\right] \geq 1 - n^{-2}$ which implies

$$\mathbf{Pr}\left[\operatorname{Gap}(m) \leq \frac{1}{\alpha}(3 \cdot \log n + \log c) \right] \geq 1 - n^{-2}.$$

- For the $(1 + \beta)$ process, $\alpha = \Theta(\beta)$.
- Same proof holds for the Quantile(δ) for constant $\delta \in (0, 1)$.

The exponential potential function

- [PTW15] used the two-sided **exponential potential**

$$\Gamma^t(x^t) := \underbrace{\sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)}}_{\text{Overload potential}} + \underbrace{\sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}}_{\text{Underload potential}} .$$

- [PTW15] show that $\mathbf{E}\left[\Gamma^{t+1} \mid \mathfrak{F}^t\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- This implies $\mathbf{E}\left[\Gamma^t\right] \leq c \cdot n$ for any $t \geq 0$.
- By Markov's inequality, we get $\mathbf{Pr}\left[\Gamma^m \leq cn^3\right] \geq 1 - n^{-2}$ which implies

$$\mathbf{Pr}\left[\text{Gap}(m) \leq \frac{1}{\alpha}(3 \cdot \log n + \log c)\right] \geq 1 - n^{-2}.$$

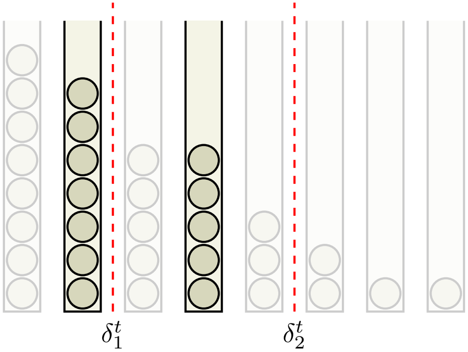
- For the $(1+\beta)$ process, $\alpha = \Theta(\beta)$.
- Same proof holds for the Quantile(δ) for constant $\delta \in (0,1)$.
- In [PTW15], $\alpha = \mathcal{O}(1)$ so the tightest gaps proved were $\mathcal{O}(\log n)$.

QUANTILE($\delta_1, \ldots, \delta_k$) process

- We can extend the QUANTILE(δ) process to k quantiles.

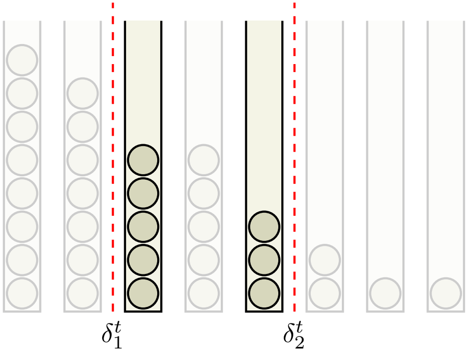
$\text{QUANTILE}(\delta_1, \ldots, \delta_k)$ process

- We can extend the $\text{QUANTILE}(\delta)$ process to k quantiles.
- We can only distinguish two bins if they are in different regions.



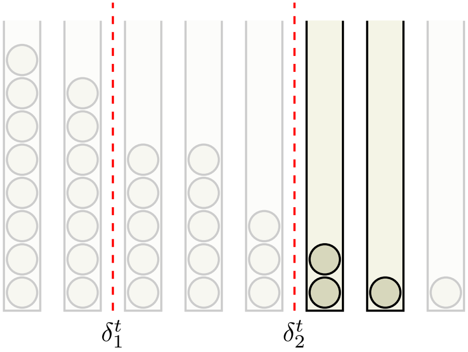
QUANTILE($\delta_1, \ldots, \delta_k$) process

- We can extend the QUANTILE(δ) process to k quantiles.
- We can only distinguish two bins if they are in different regions.



QUANTILE$(\delta_1, \ldots, \delta_k)$ **process**

- We can extend the QUANTILE(δ) process to k quantiles.
- We can only distinguish two bins if they are in different regions.

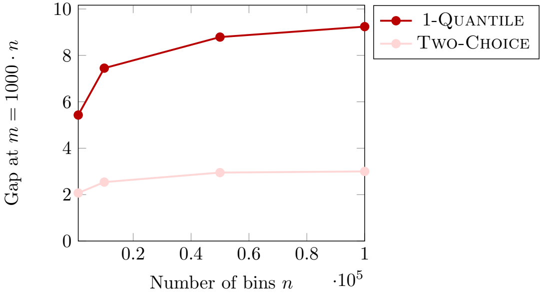


Our results

- A $\textsc{Quantile}(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.

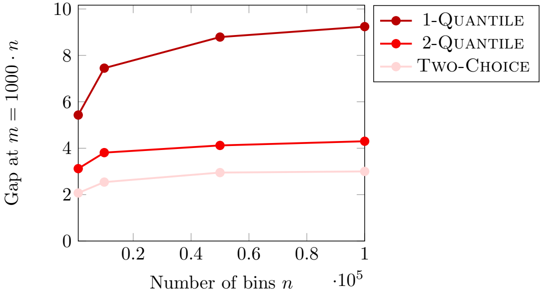
Our results

- A QUANTILE$(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.



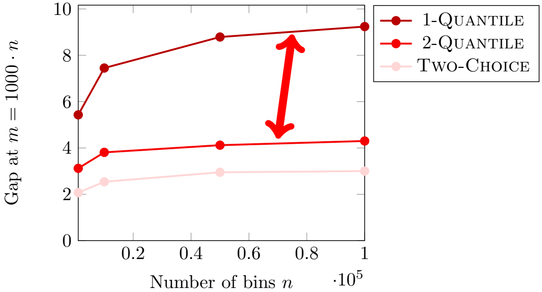
Our results

- A QUANTILE$(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.



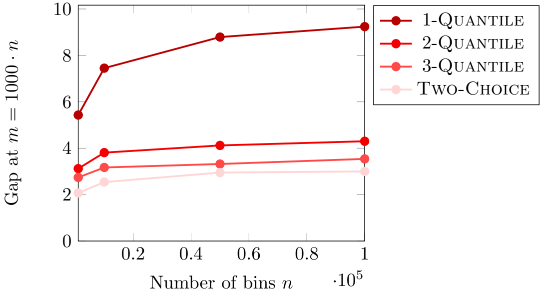
Our results

- A QUANTILE$(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.



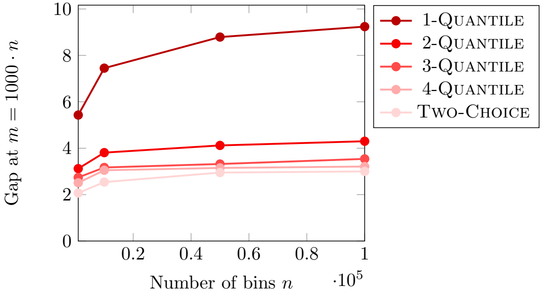
Our results

- A QUANTILE($\delta_1, \ldots, \delta_k$) process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.



Our results

- A Quantile($\delta_1, \ldots, \delta_k$) process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.



Our results

- A Quantile$(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.
- Implications:

Our results

- A QUANTILE($\delta_1, \ldots, \delta_k$) process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.
- Implications:
  - For $k = \Theta(\log \log n)$, we recover the TWO-CHOICE $\text{Gap}(m) = \mathcal{O}(\log \log n)$.

Our results

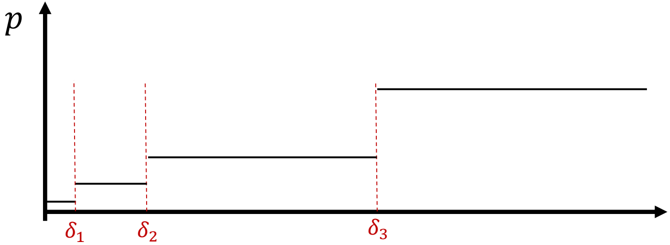
- A Quantile$(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.
- Implications:
  - For $k = \Theta(\log \log n)$, we recover the Two-Choice $\text{Gap}(m) = \mathcal{O}(\log \log n)$.
  - For $(1 + \beta)$ with $\beta = 1 - 2^{-0.5(\log n)^{(k-1)/k}}$, w.h.p. $\text{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$.

Our results

- A $\textsc{Quantile}(\delta_1, \ldots, \delta_k)$ process with uniform quantiles that achieves w.h.p. an $\mathcal{O}(k \cdot (\log n)^{1/k})$ gap for $k = \mathcal{O}(\log \log n)$.
- Implications:
  - For $k = \Theta(\log \log n)$, we recover the Two-Choice $\mathrm{Gap}(m) = \mathcal{O}(\log \log n)$.
  - For $(1 + \beta)$ with $\beta = 1 - 2^{-0.5(\log n)^{(k-1)/k}}$, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$.
  - Improvements for other processes (d-Thinning, graphical allocations).

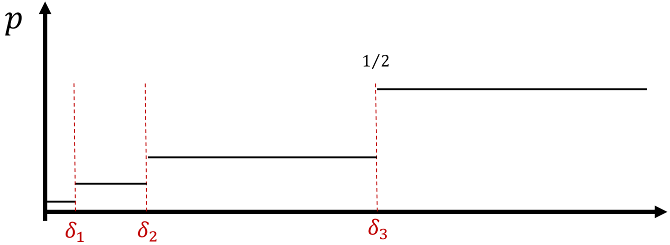
Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:



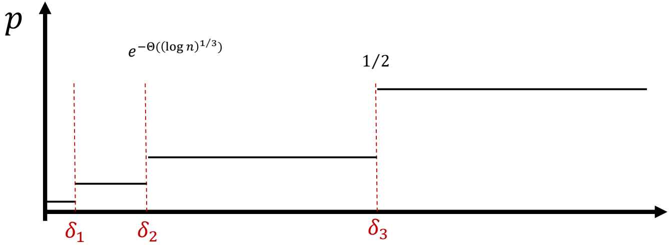
Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:



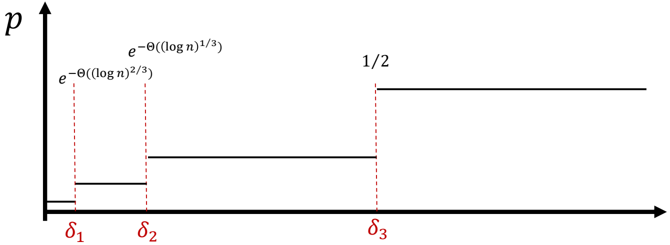
Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:



Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:



Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right),$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right),$$

$$\Phi_1^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right),$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right),$$

$$\Phi_1^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right),$$

$$\Phi_2^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{2/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{4}{\alpha'}(\log n)^{1/3}\right)^+\right).$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right), \qquad \mathcal{O}(\log n)$$

$$\Phi_1^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right),$$

$$\Phi_2^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{2/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{4}{\alpha'}(\log n)^{1/3}\right)^+\right).$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right), \qquad \color{green}{\mathcal{O}(\log n)}$$

$$\Phi_1^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right), \qquad \color{green}{\mathcal{O}((\log n)^{2/3})}$$

$$\Phi_2^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{2/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{4}{\alpha'}(\log n)^{1/3}\right)^+\right).$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^n \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right), \qquad \mathcal{O}(\log n)$$

$$\Phi_1^t := \sum_{i=1}^n \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right), \qquad \mathcal{O}((\log n)^{2/3})$$

$$\Phi_2^t := \sum_{i=1}^n \exp\left(\alpha' \cdot (\log n)^{2/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{4}{\alpha'}(\log n)^{1/3}\right)^+\right). \qquad \mathcal{O}((\log n)^{1/3})$$

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:
$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$
- We define the following exponential potential functions:
$$\Phi_0^t := \sum_{i=1}^n \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right), \qquad \mathcal{O}(\log n)$$
$$\Phi_1^t := \sum_{i=1}^n \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right), \qquad \mathcal{O}((\log n)^{2/3})$$
$$\Phi_2^t := \sum_{i=1}^n \exp\left(\alpha' \cdot (\log n)^{2/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{4}{\alpha'}(\log n)^{1/3}\right)^+\right). \qquad \mathcal{O}((\log n)^{1/3})$$
- When $\alpha = \Omega(1)$, the potential may not necessarily drop in expectation.

Analysis using super-exponential potential functions

- For $k = 3$, we define the quantile process which achives $\mathcal{O}((\log n)^{1/3})$ gap:

$$\delta_1 := e^{-\Theta((\log n)^{1/3})}, \quad \delta_2 := e^{-\Theta((\log n)^{2/3})}, \quad \delta_3 := 1/2$$

- We define the following exponential potential functions:

$$\Phi_0^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot \left(x_i^t - \frac{t}{n}\right)^+\right), \qquad \mathcal{O}(\log n)$$

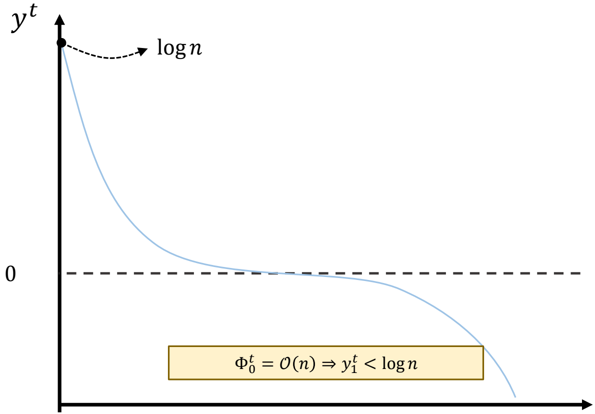
$$\Phi_1^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{1/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{2}{\alpha'}(\log n)^{1/3}\right)^+\right), \qquad \mathcal{O}((\log n)^{2/3})$$

$$\Phi_2^t := \sum_{i=1}^{n} \exp\left(\alpha' \cdot (\log n)^{2/3} \cdot \left(x_i^t - \frac{t}{n} - \frac{4}{\alpha'}(\log n)^{1/3}\right)^+\right). \qquad \mathcal{O}((\log n)^{1/3})$$

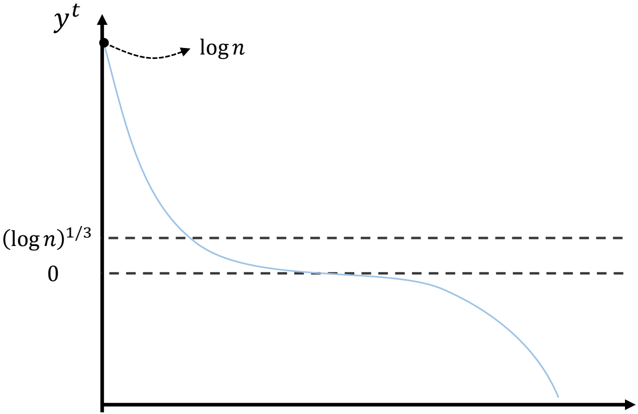
- When $\alpha = \Omega(1)$, the potential may not necessarily drop in expectation.
- We prove that when $y_{\delta_{3-j} \cdot n}^t < \frac{2}{\alpha'} j (\log n)^{1/3}$, then

$$\mathbf{E}\left[\, \Phi_j^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_j^t \cdot \left(1 - \frac{1}{n}\right) + 2.$$

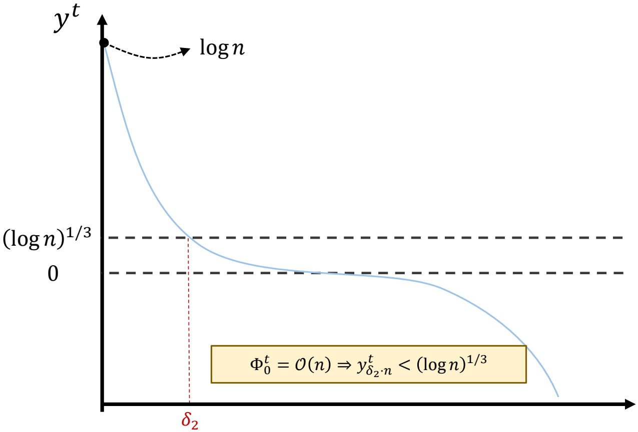
Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



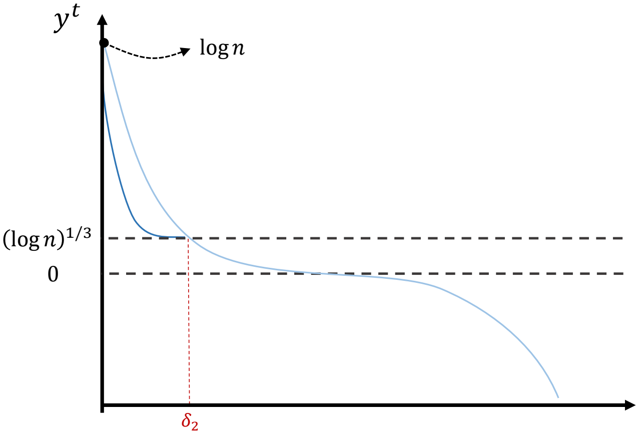
Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



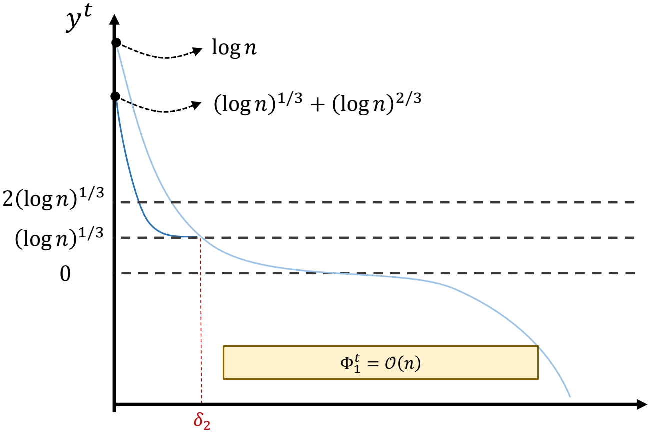
Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



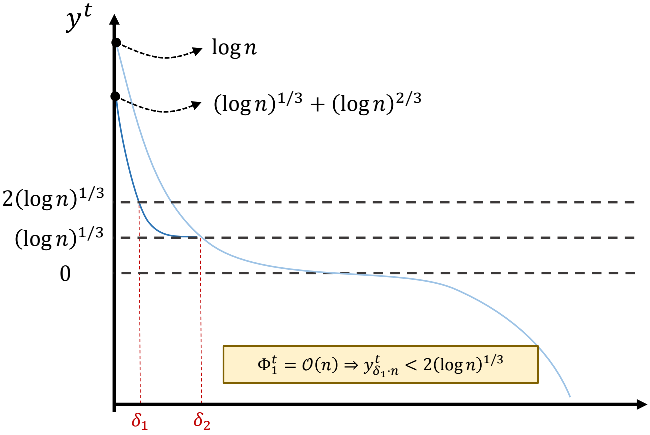
Proving $\text{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



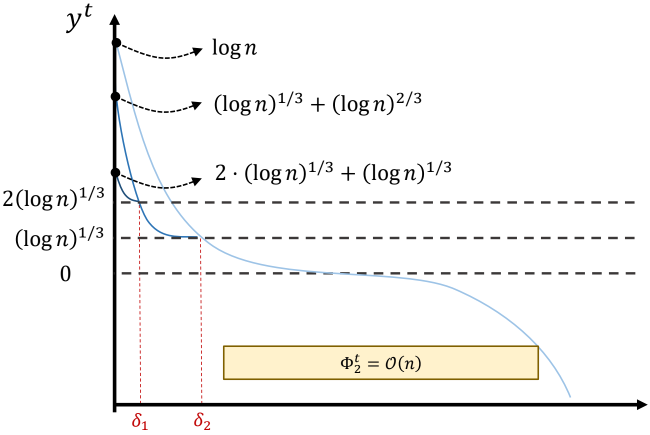
Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



Proving $\text{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot (\log n)^{1/k})$



Mean-Threshold

THRESHOLD process

THRESHOLD($f(n)$) Process:

Parameter: A threshold function $f(n) \geq 0$.

Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

Threshold process

Threshold$(f(n))$ Process:
Parameter: A threshold function $f(n) \geq 0$.
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

Decentralised

Threshold **process**

Threshold($f(n)$) Process:
Parameter: A threshold function $f(n) \geq 0$.
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

- Mean-Threshold has $f(n) = 0$.

THRESHOLD process

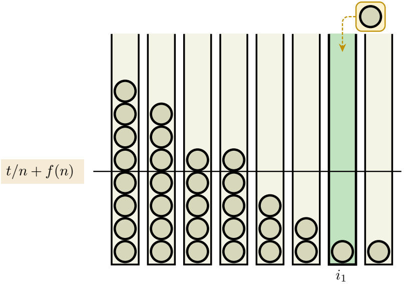
THRESHOLD($f(n)$) Process:

Parameter: A threshold function $f(n) \geq 0$.

Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

- MEAN-THRESHOLD has $f(n) = 0$.



Threshold process

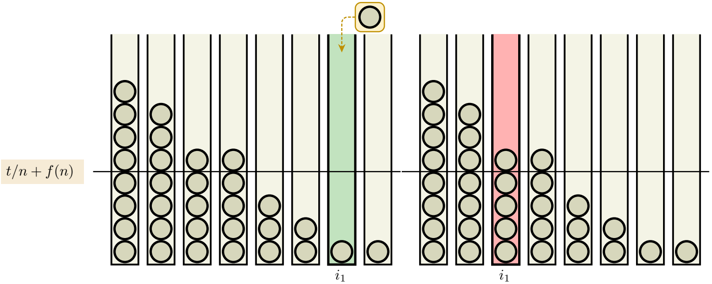
Threshold($f(n)$) Process:

Parameter: A threshold function $f(n) \geq 0$.

Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^t + 1 & \text{if } x_{i_1}^t < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^t + 1 & \text{if } x_{i_1}^t \geq \frac{t}{n} + f(n). \end{cases}$$

- Mean-Threshold has $f(n) = 0$.



THRESHOLD process

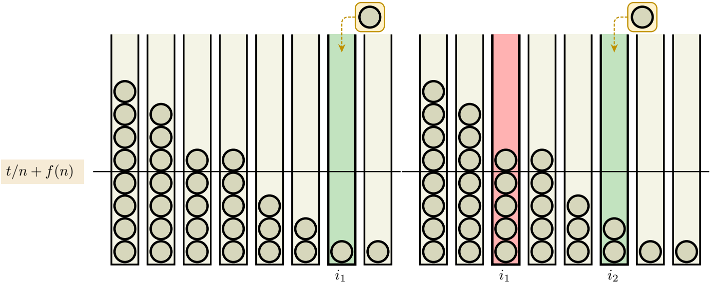
THRESHOLD$(f(n))$ Process:

Parameter: A threshold function $f(n) \geq 0$.

Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^t + 1 & \text{if } x_{i_1}^t < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^t + 1 & \text{if } x_{i_1}^t \geq \frac{t}{n} + f(n). \end{cases}$$

- MEAN-THRESHOLD has $f(n) = 0$.



THRESHOLD process

THRESHOLD($f(n)$) Process:
Parameter: A threshold function $f(n) \geq 0$.
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

- MEAN-THRESHOLD has $f(n) = 0$.
- [FGG21] found the asymptotically optimal threshold in the lightly-loaded case.

THRESHOLD **process**

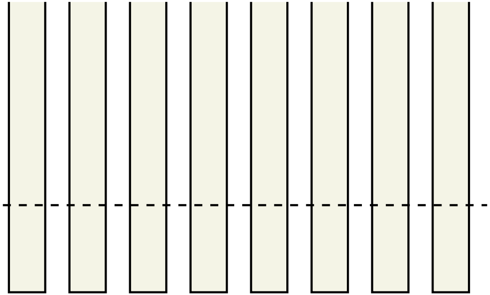
THRESHOLD($f(n)$) Process:
Parameter: A threshold function $f(n) \geq 0$.
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

- MEAN-THRESHOLD has $f(n) = 0$.
- [FGG21] found the asymptotically optimal threshold in the lightly-loaded case.
- [IK05, FL20] analysed a d-sample version for the lightly-loaded case.

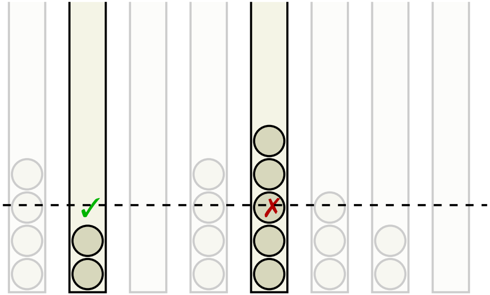
Threshold as Two-Choice with incomplete information

We can interpret Threshold as an instance of the Two-Choice process, where we are only able to compare the loads of the two sampled bins if one is above the threshold and one is below.



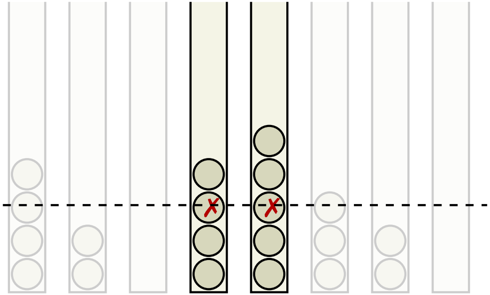
Threshold as Two-Choice with incomplete information

We can interpret Threshold as an instance of the Two-Choice process, where we are only able to compare the loads of the two sampled bins if one is above the threshold and one is below.



Threshold as Two-Choice with incomplete information

We can interpret Threshold as an instance of the Two-Choice process, where we are only able to compare the loads of the two sampled bins if one is above the threshold and one is below.



Mean-Threshold: Message/sample efficiency

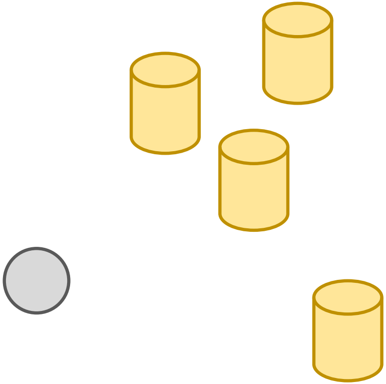
- Mean-Threshold does not require communication of the load.

Mean-Threshold: Message/sample efficiency

- Mean-Threshold does not require communication of the load.
- Requires 1-bit responses.

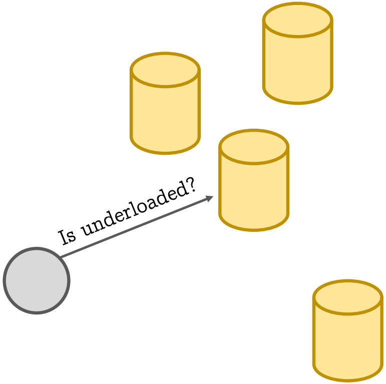
Mean-Threshold: Message/sample efficiency

- Mean-Threshold does not require communication of the load.
- Requires 1-bit responses.



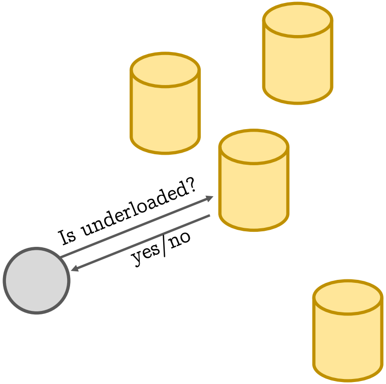
Mean-Threshold: Message/sample efficiency

- Mean-Threshold does not require communication of the load.
- Requires 1-bit responses.



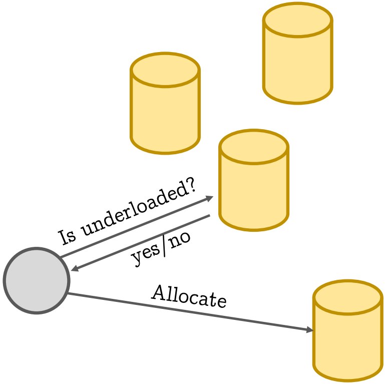
MEAN-THRESHOLD: Message/sample efficiency

- MEAN-THRESHOLD does not require communication of the load.
- Requires 1-bit responses.



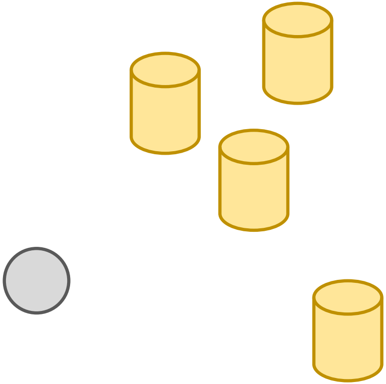
Mean-Threshold: Message/sample efficiency

- Mean-Threshold does not require communication of the load.
- Requires 1-bit responses.



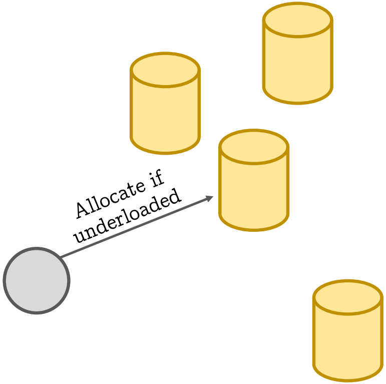
Mean-Threshold: Message/sample efficiency

- Mean-Threshold does not require communication of the load.
- Requires 1-bit responses.
- Or we can completely avoid responses to the allocator.



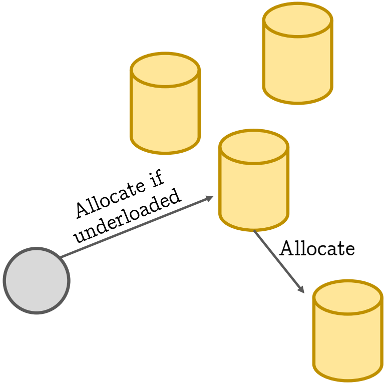
Mean-Threshold: Message/sample efficiency

- Mean-Threshold does not require communication of the load.
- Requires 1-bit responses.
- Or we can completely avoid responses to the allocator.



MEAN-THRESHOLD: Message/sample efficiency

- MEAN-THRESHOLD does not require communication of the load.
- Requires 1-bit responses.
- Or we can completely avoid responses to the allocator.



Mean-Threshold: Our results

- For heavily-loaded case, Mean-Threshold achieves w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.

Mean-Threshold: **Our results**

- For heavily-loaded case, Mean-Threshold achieves w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.

- For sufficiently large m, Mean-Threshold achieves w.h.p. $\text{Gap}(m) = \Omega(\log n)$.

Mean-Threshold: **Our results**

- For heavily-loaded case, Mean-Threshold achieves w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.
- For sufficiently large m, Mean-Threshold achieves w.h.p. $\text{Gap}(m) = \Omega(\log n)$.
- Mean-Threshold uses w.h.p. $2 - \epsilon$ samples per allocation.

Mean-Threshold: Why the analysis is tricky

- Let δ^t be the quantile position of the mean.

Mean-Threshold: Why the analysis is tricky

- Let δ^t be the quantile position of the mean.
- If δ^t is very large, say $\delta^t = 1 - 1/n$, then p becomes very close to the One-Choice vector :

$$p_{\text{Mean-Threshold}}(x^t) = \Big(\underbrace{\frac{1}{n} - \frac{1}{n^2}, \ldots, \frac{1}{n} - \frac{1}{n^2}}_{(n-1)\text{ entries}}, \frac{2}{n} - \frac{1}{n^2} \Big).$$

MEAN-THRESHOLD: Why the analysis is tricky

- Let δ^t be the quantile position of the mean.
- If δ^t is very large, say $\delta^t = 1 - 1/n$, then p becomes very close to the ONE-CHOICE vector :

$$p_{\text{MEAN-THRESHOLD}}(x^t) = \Big(\underbrace{\frac{1}{n} - \frac{1}{n^2}, \ldots, \frac{1}{n} - \frac{1}{n^2}}_{(n-1)\text{ entries}}, \frac{2}{n} - \frac{1}{n^2}\Big).$$

- With this worst-case probability vector, we can only obtain w.h.p. a gap of $\mathcal{O}(n \log n)$ using Γ^t with $\alpha = \Theta(1/n)$.

Mean-Threshold: Why the analysis is tricky

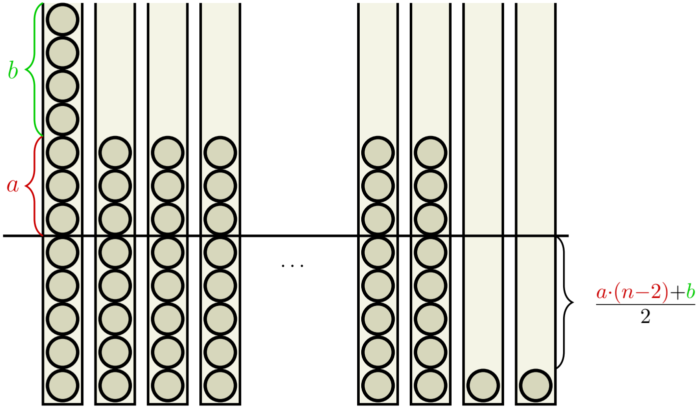
- Let δ^t be the quantile position of the mean.
- If δ^t is very large, say $\delta^t = 1 - 1/n$, then p becomes very close to the One-Choice vector :

$$p_{\textsc{Mean-Threshold}}(x^t) = \Big(\underbrace{\frac{1}{n} - \frac{1}{n^2}, \ldots, \frac{1}{n} - \frac{1}{n^2}}_{(n-1)\text{ entries}}, \frac{2}{n} - \frac{1}{n^2}\Big).$$

- With this worst-case probability vector, we can only obtain w.h.p. a gap of $\mathcal{O}(n \log n)$ using Γ^t with $\alpha = \Theta(1/n)$.

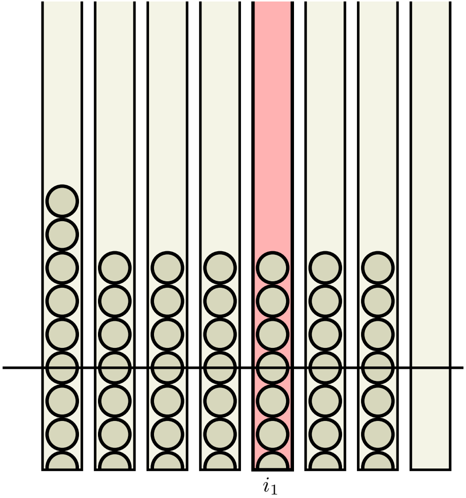
But what happens for Γ^t with constant α?

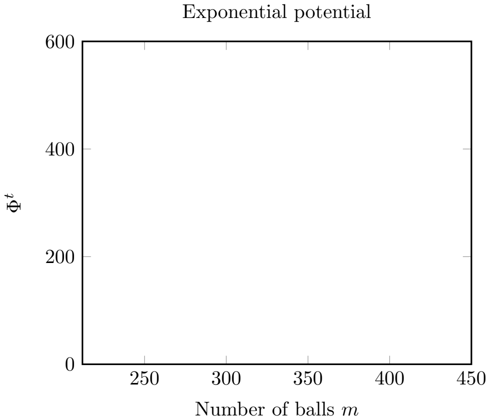
Mean-Threshold: **Bad configuration**



- There is a very small bias away from overloaded bins.
- **The exponential potential for constant α increases in expectation.**

MEAN-THRESHOLD: Recovery from a bad configuration





A closer look at Γ^t

- An analysis similar to [PTW15] shows that

A closer look at Γ^t

- An analysis similar to [PTW15] shows that
  - (**Good step**) If $\delta^t \in (\epsilon, 1-\epsilon)$ for const $\epsilon > 0$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \{\delta^t \in (\epsilon, 1-\epsilon)\}, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\alpha}{n}\right)\right).$$

A closer look at Γ^t

- An analysis similar to [PTW15] shows that
  - (**Good step**) If $\delta^t \in (\epsilon, 1-\epsilon)$ for const $\epsilon > 0$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \{\delta^t \in (\epsilon, 1-\epsilon)\}, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\alpha}{n}\right)\right).$$

  - (**Bad step**) If $\delta^t \notin (\epsilon, 1-\epsilon)$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 + \Theta\left(\frac{\alpha^2}{n}\right)\right).$$

A closer look at Γ^t

- An analysis similar to [PTW15] shows that
  - (**Good step**) If $\delta^t \in (\epsilon, 1-\epsilon)$ for const $\epsilon > 0$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \{\delta^t \in (\epsilon, 1-\epsilon)\}, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\alpha}{n}\right)\right).$$

  - (**Bad step**) If $\delta^t \notin (\epsilon, 1-\epsilon)$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 + \Theta\left(\frac{\alpha^2}{n}\right)\right).$$

- A properly tweaked potential function drops in expectation for any interval with constant fraction of good steps.

A closer look at Γ^t

- An analysis similar to [PTW15] shows that
  - (**Good step**) If $\delta^t \in (\epsilon, 1-\epsilon)$ for const $\epsilon > 0$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \{\delta^t \in (\epsilon, 1-\epsilon)\}, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\alpha}{n}\right)\right).$$

  - (**Bad step**) If $\delta^t \notin (\epsilon, 1-\epsilon)$, then

$$\mathbf{E}[\,\Gamma^{t+1} \mid \mathfrak{F}^t, \Gamma^t \geq c \cdot n\,] \leq \Gamma^t \cdot \left(1 + \Theta\left(\frac{\alpha^2}{n}\right)\right).$$

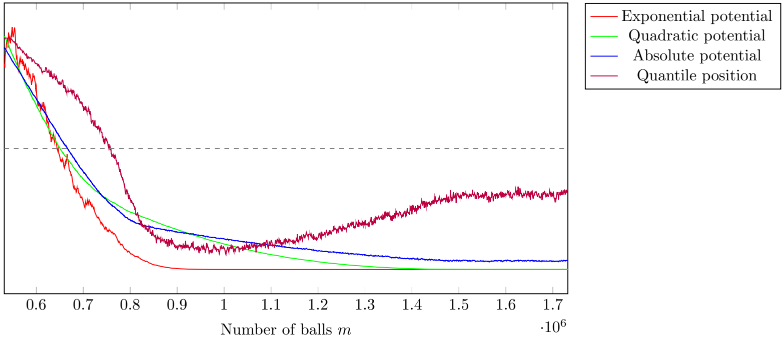
- A properly tweaked potential function drops in expectation for any interval with constant fraction of good steps.

How can we prove that there is a constant fraction of good steps?

Recovery from a bad configuration ($n = 1000$)

- Consider the **absolute value** and **quadratic** potentials,

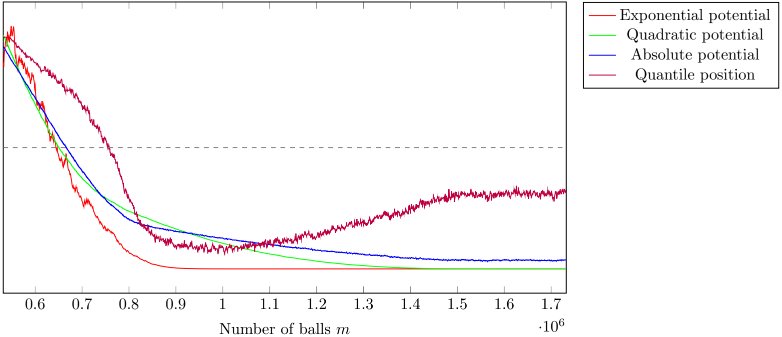
$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right| \quad \text{and} \quad \Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2 .$$



Recovery from a bad configuration ($n = 1000$)

- Consider the **absolute value** and **quadratic** potentials,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right| \quad \text{and} \quad \Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2 .$$

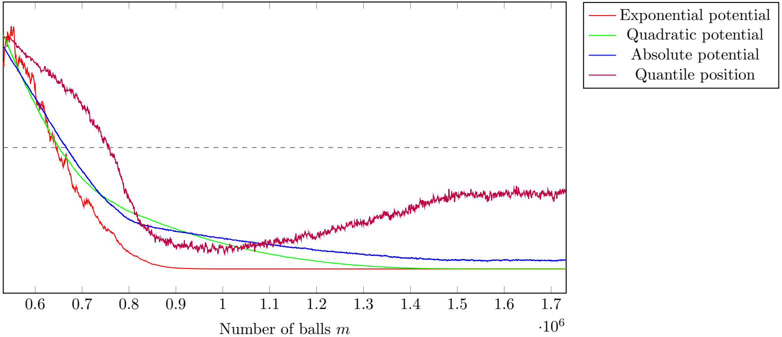


- As long as $\Delta^t = \Omega(n)$, Υ^t drops in expectation.

Recovery from a bad configuration ($n = 1000$)

- Consider the **absolute value** and **quadratic** potentials,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right| \quad \text{and} \quad \Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2 .$$

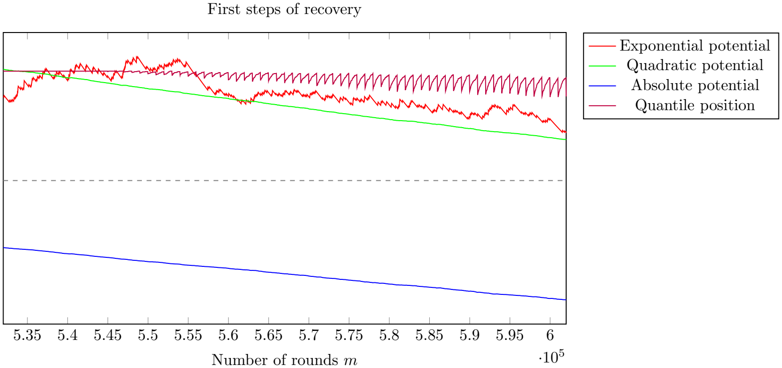


- As long as $\Delta^t = \Omega(n)$, Υ^t drops in expectation.
- As Δ^t becomes smaller, δ^t improves and Γ^t drops in expectation.

Recovery from a bad configuration ($n = 1000$)

- Consider the **absolute value** and **quadratic** potentials,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right| \quad \text{and} \quad \Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2 .$$



- As long as $\Delta^t = \Omega(n)$, Υ^t drops in expectation.
- As Δ^t becomes smaller, δ^t improves and Γ^t drops in expectation.

Conclusion

Summary of results:

- Analysed various processes including k-QUANTILE, MEAN-THRESHOLD and $(1 + \beta)$ for β close to 1.

Conclusion

Summary of results:

- Analysed various processes including k-QUANTILE, MEAN-THRESHOLD and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:

Conclusion

Summary of results:

- Analysed various processes including k-Quantile, Mean-Threshold and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.

Conclusion

Summary of results:

- Analysed various processes including k-Quantile, Mean-Threshold and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.

Conclusion

Summary of results:

- Analysed various processes including k-Quantile, Mean-Threshold and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.
- Further applications of the presented techniques:
  - Balls allocated in batches of $b \geq n$ balls : $\mathcal{O}(b/n + \log n)$ gap for Two-Choice, $(1+\beta)$ and Quantile(δ) [LS22a].

Conclusion

Summary of results:

- Analysed various processes including k-Quantile, Mean-Threshold and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.
- Further applications of the presented techniques:
  - Balls allocated in batches of $b \geq n$ balls : $\mathcal{O}(b/n + \log n)$ gap for Two-Choice, $(1+\beta)$ and Quantile(δ) [LS22a].
  - Two-Choice with adversarial noise $g \leq \log n$: $\mathcal{O}(\frac{g}{\log g} \cdot \log \log n)$ [LS22b].

Conclusion

Summary of results:

- Analysed various processes including k-QUANTILE, MEAN-THRESHOLD and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.
- Further applications of the presented techniques:
  - Balls allocated in batches of $b \geq n$ balls : $\mathcal{O}(b/n + \log n)$ gap for TWO-CHOICE, $(1+\beta)$ and QUANTILE(δ) [LS22a].
  - TWO-CHOICE with adversarial noise $g \leq \log n$: $\mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$ [LS22b].

Several directions for future work:

Conclusion

Summary of results:

- Analysed various processes including k-QUANTILE, MEAN-THRESHOLD and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.
- Further applications of the presented techniques:
  - Balls allocated in batches of $b \geq n$ balls : $\mathcal{O}(b/n + \log n)$ gap for TWO-CHOICE, $(1+\beta)$ and QUANTILE(δ) [LS22a].
  - TWO-CHOICE with adversarial noise $g \leq \log n$: $\mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$ [LS22b].

Several directions for future work:

- Investigating the robustness of MEAN-THRESHOLD.

Conclusion

Summary of results:

- Analysed various processes including k-Quantile, Mean-Threshold and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.
- Further applications of the presented techniques:
  - Balls allocated in batches of $b \geq n$ balls : $\mathcal{O}(b/n + \log n)$ gap for Two-Choice, $(1+\beta)$ and Quantile(δ) [LS22a].
  - Two-Choice with adversarial noise $g \leq \log n$: $\mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$ [LS22b].

Several directions for future work:

- Investigating the robustness of Mean-Threshold.
- Loosening the fully random hash function assumption.

Conclusion

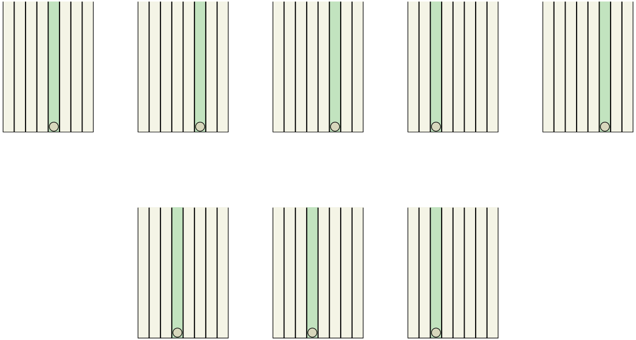
Summary of results:

- Analysed various processes including k-Quantile, Mean-Threshold and $(1+\beta)$ for β close to 1.
- Introduced two techniques for analysing balanced allocation processes:
  - Layered induction over super-exponential potentials.
  - Interplay between the absolute value and quadratic potentials.
- Further applications of the presented techniques:
  - Balls allocated in batches of $b \geq n$ balls : $\mathcal{O}(b/n + \log n)$ gap for Two-Choice, $(1+\beta)$ and Quantile(δ) [LS22a].
  - Two-Choice with adversarial noise $g \leq \log n$: $\mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$ [LS22b].

Several directions for future work:

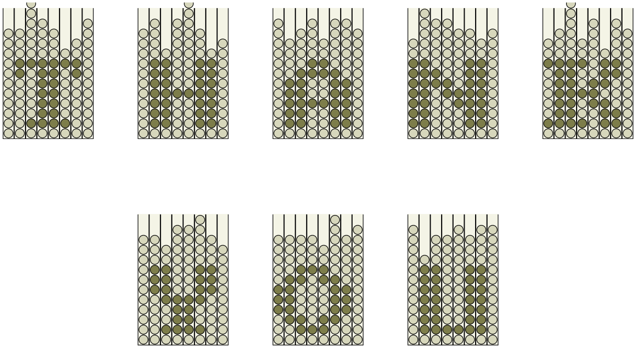
- Investigating the robustness of Mean-Threshold.
- Loosening the fully random hash function assumption.
- Analysing these processes in the graphical setting.

Questions?



More visualisations: `tinyurl.com/lss21-visualisations`

Questions?



More visualisations: `tinyurl.com/lss21-visualisations`

Appendix

Appendix A: Table of results

Process	Lightly Loaded Case $m = \mathcal{O}(n)$		Heavily Loaded Case $m = \omega(n)$	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound
$(1+\beta)$, const $\beta \in (0,1)$	$\frac{\log n}{\log \log n}$ [PTW15]		$\log n$	
Caching	$\log \log n$ [MPS02]		–	$\log n$
Packing	$\frac{\log n}{\log \log n}$		$\log n$	
Twinning	$\frac{\log n}{\log \log n}$		$\log n$	
Mean-Threshold	$\frac{\log n}{\log \log n}$		$\log n$	
2-Thinning $\left(\Theta(\sqrt{\frac{\log n}{\log \log n}})\right)$	$\sqrt{\frac{\log n}{\log \log n}}$ [FL20]		$\frac{\log n}{\log \log n}$ [LS21]	$\log n$
Adaptive-2-Thinning	$\sqrt{\frac{\log n}{\log \log n}}$ [FL20]		$\frac{\log n}{\log \log n}$ [LS21]	$\frac{\log n}{\log \log n}$ [FGGL21]

Table: Overview of the Gap achieved (with probability at least $1 - n^{-1}$), by different allocation processes considered in this work (and related works).

Appendix B: Detailed experimental results (I)

n	Mean-Threshold	Twinning	Packing	Caching
10^5	**8** : 3% **9** : 32% **10** : 38% **11** : 15% **12** : 6% **13** : 3% **14** : 3%	**14** : 2% **15** : 5% **16** : 25% **17** : 28% **18** : 17% **19** : 10% **20** : 8% **21** : 1% **22** : 1% **23** : 3%	**12** : 2% **13** : 16% **14** : 20% **15** : 28% **16** : 23% **17** : 5% **18** : 3% **19** : 1% **20** : 2%	**3** : 100%

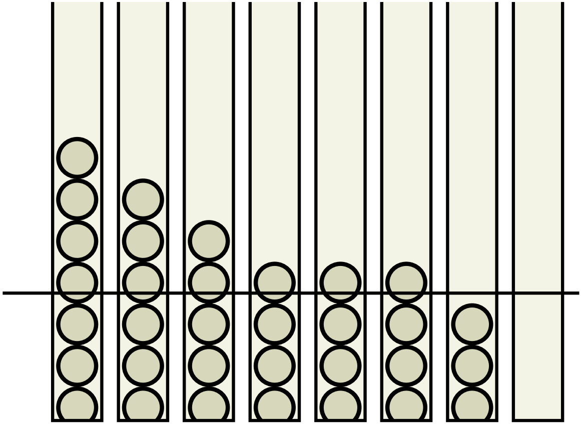
Table: Summary of observed gaps for $n \in \{10^3, 10^4, 10^5\}$ bins and $m = 1000 \cdot n$ number of balls, for 100 repetitions. The observed gaps are in bold and next to that is the % of runs where this was observed.

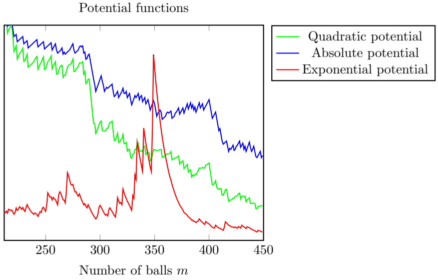
Appendix B: Detailed experimental results (II)

n	$(1+\beta)$, for $\beta = 0.5$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	Two-Choice
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%

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

Appendix C: Recovery from a bad configuration





Appendix D: Filling framework

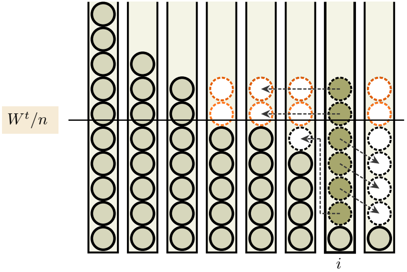
- We analyze a more general framework that includes Packing and Caching [MPS02].

Appendix D: Filling framework

- We analyze a more general framework that includes Packing and Caching [MPS02].
- We prove an $\mathcal{O}(\log n)$ gap for these processes.

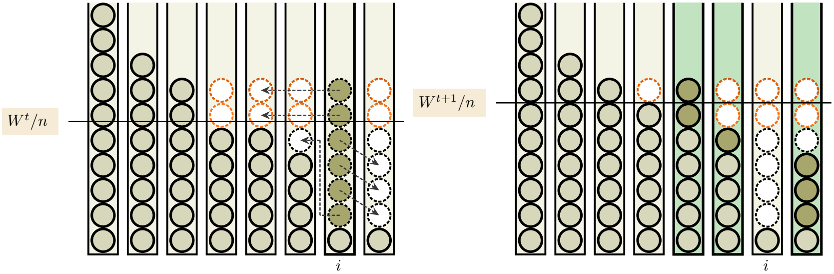
Appendix D: Filling framework

- We analyze a more general framework that includes Packing and Caching [MPS02].
- We prove an $\mathcal{O}(\log n)$ gap for these processes.

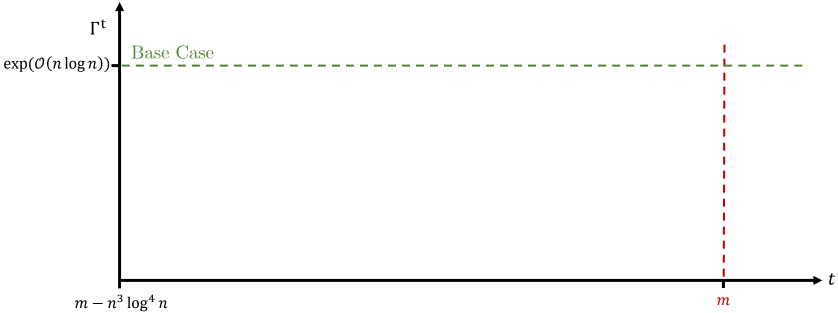


Appendix D: Filling framework

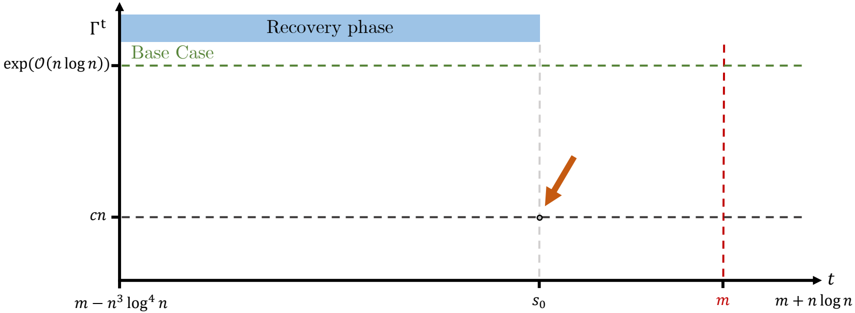
- We analyze a more general framework that includes PACKING and CACHING [MPS02].
- We prove an $\mathcal{O}(\log n)$ gap for these processes.



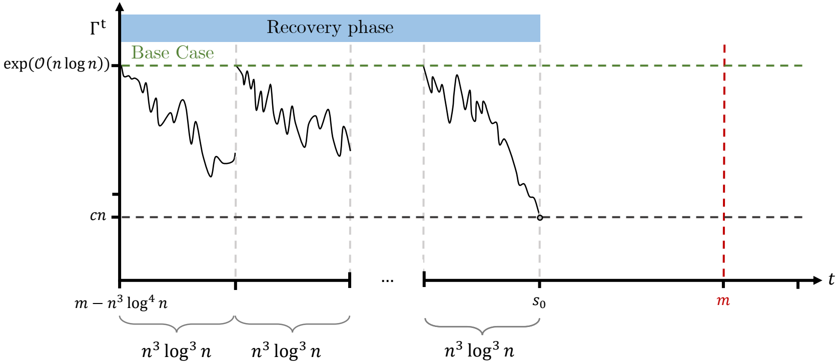
Appendix E: Completing the Mean-Threshold analysis



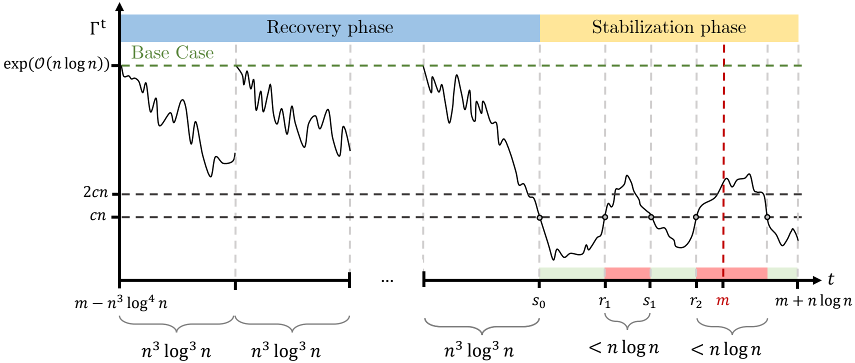
Appendix E: Completing the MEAN-THRESHOLD analysis



Appendix E: Completing the Mean-Threshold analysis



Appendix E: Completing the Mean-Threshold analysis



Appendix F: Outdated information

In "Balanced Allocations in Batches: Simplified and Generalized" [LS22a], we study process with outdated information:

Appendix F: Outdated information

In "Balanced Allocations in Batches: Simplified and Generalized" [LS22a], we study process with outdated information:

- In the *batched setting* balls arrive in batches of size b.

Appendix F: Outdated information

In “Balanced Allocations in Batches: Simplified and Generalized” [LS22a], we study process with outdated information:

- In the *batched setting* balls arrive in batches of size b.
- For $b = n$, [BCE+12] proved that Two-Choice has w.h.p. an $\mathcal{O}(\log n)$ gap.

Appendix F: Outdated information

In "Balanced Allocations in Batches: Simplified and Generalized" [LS22a], we study process with outdated information:

- In the *batched setting* balls arrive in batches of size b.
- For $b = n$, [BCE$^+$12] proved that Two-Choice has w.h.p. an $\mathcal{O}(\log n)$ gap.
- We show that a large class of proceses (including $(1+\beta)$ and Quantile(δ) for const β and δ) have an $\mathcal{O}(b/n \cdot \log n)$ gap for any $b \geq n$.

Appendix F: Outdated information

In "Balanced Allocations in Batches: Simplified and Generalized" [LS22a], we study process with outdated information:

- In the *batched setting* balls arrive in batches of size b.
- For $b = n$, [BCE$^+$12] proved that Two-Choice has w.h.p. an $\mathcal{O}(\log n)$ gap.
- We show that a large class of proceses (including $(1+\beta)$ and Quantile(δ) for const β and δ) have an $\mathcal{O}(b/n \cdot \log n)$ gap for any $b \geq n$.
- The proof follows by looking at Γ with $\alpha = \Theta(n/b)$.

Appendix F: Outdated information

In "Balanced Allocations in Batches: Simplified and Generalized" [LS22a], we study process with outdated information:

- In the *batched setting* balls arrive in batches of size b.
- For $b = n$, [BCE$^+$12] proved that Two-Choice has w.h.p. an $\mathcal{O}(\log n)$ gap.
- We show that a large class of proceses (including $(1+\beta)$ and Quantile(δ) for const β and δ) have an $\mathcal{O}(b/n \cdot \log n)$ gap for any $b \geq n$.
- The proof follows by looking at Γ with $\alpha = \Theta(n/b)$.
- By using a second potential $\tilde{\Gamma}$ with $\tilde{\alpha} = \Theta(\min(1/\log n, n/b))$ and conditioning on $\Gamma = \mathcal{O}(n)$, we prove an $\mathcal{O}(n/b + \log n)$ gap for $b \geq n$.

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

- In the *adversarial noise setting*, an adversary can perturb the observed loads by some amount g.

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

- In the *adversarial noise setting*, an adversary can perturb the observed loads by some amount g.
- Using an interplay between Δ^t and Υ, we prove an $\mathcal{O}(g + \log n)$ gap.

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

- In the *adversarial noise setting*, an adversary can perturb the observed loads by some amount g.
- Using an interplay between Δ^t and Υ, we prove an $\mathcal{O}(g + \log n)$ gap.
- Using layered induction of super-exponential potentials we get $\mathcal{O}(\frac{g}{\log g} \cdot \log \log n)$ for $g \leq \log n$, which is tight.

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

- In the *adversarial noise setting*, an adversary can perturb the observed loads by some amount g.
- Using an interplay between Δ^t and Υ, we prove an $\mathcal{O}(g + \log n)$ gap.
- Using layered induction of super-exponential potentials we get $\mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$ for $g \leq \log n$, which is tight.
- Implies tight bounds for random noise from sub-exponential distributions.

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

- In the *adversarial noise setting*, an adversary can perturb the observed loads by some amount g.
- Using an interplay between Δ^t and Υ, we prove an $\mathcal{O}(g + \log n)$ gap.
- Using layered induction of super-exponential potentials we get $\mathcal{O}(\frac{g}{\log g} \cdot \log \log n)$ for $g \leq \log n$, which is tight.
- Implies tight bounds for random noise from sub-exponential distributions.
- Implies tight upper bounds for Two-Choice with batch sizes of $b = \mathcal{O}(n)$.

Appendix G: Two-Choice with Adversarial Noise

In "Balanced Allocations with the Choice of Noise" [LS22b], we study Two-Choice with noise:

- In the *adversarial noise setting*, an adversary can perturb the observed loads by some amount g.
- Using an interplay between Δ^t and Υ, we prove an $\mathcal{O}(g + \log n)$ gap.
- Using layered induction of super-exponential potentials we get $\mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$ for $g \leq \log n$, which is tight.
- Implies tight bounds for random noise from sub-exponential distributions.
- Implies tight upper bounds for Two-Choice with batch sizes of $b = \mathcal{O}(n)$.
- In particular, implies $\text{Gap}(n) = \Theta(\log n / \log\log n)$ for $b = n$.
- And for the setting where the load of a bin is chosen adversarially from the last b steps.

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right|.$$

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right|.$$

- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\epsilon, 1 - \epsilon)$ w.h.p. for a constant fraction of the next $\Theta(n)$ steps.

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right|.$$

- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\epsilon, 1 - \epsilon)$ w.h.p. for a constant fraction of the next $\Theta(n)$ steps.
- Consider the **quadratic potential**,

$$\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2.$$

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right|.$$

- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\epsilon, 1 - \epsilon)$ w.h.p. for a constant fraction of the next $\Theta(n)$ steps.
- Consider the **quadratic potential**,

$$\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2.$$

- We prove that

$$\mathbf{E}\left[\Upsilon^{t+1} \mid \mathfrak{F}^t \right] \leq \Upsilon^t - \Delta^t + 1.$$

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right| .$$

- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\epsilon, 1-\epsilon)$ w.h.p. for a constant fraction of the next $\Theta(n)$ steps.
- Consider the **quadratic potential**,

$$\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2 .$$

- We prove that

$$\mathbf{E}\left[\Upsilon^{t+1} \mid \mathfrak{F}^t \right] \leq \Upsilon^t - \Delta^t + 1.$$

- By induction we get,

$$\mathbf{E}[\Upsilon^{t+k+1} \mid \mathfrak{F}^t] \leq \Upsilon^t - \frac{1}{n} \cdot \sum_{r=t}^{t+k} \mathbf{E}[\Delta^r \mid \mathfrak{F}^t] + (k+1).$$

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right|.$$

- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\epsilon, 1-\epsilon)$ w.h.p. for a constant fraction of the next $\Theta(n)$ steps.
- Consider the **quadratic potential**,

$$\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2.$$

- We prove that

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t - \Delta^t + 1.$$

- By induction we get,

$$\mathbf{E}[\, \Upsilon^{t+k+1} \mid \mathfrak{F}^t \,] \leq \Upsilon^t - \frac{1}{n} \cdot \sum_{r=t}^{t+k} \mathbf{E}[\, \Delta^r \mid \mathfrak{F}^t \,] + (k+1).$$

Appendix H: Mean quantile stabilisation

- Consider the **absolute value potential**,

$$\Delta^t := \sum_{i=1}^{n} \left| x_i^t - \frac{t}{n} \right|.$$

- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\epsilon, 1-\epsilon)$ w.h.p. for a constant fraction of the next $\Theta(n)$ steps.
- Consider the **quadratic potential**,

$$\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n} \right)^2.$$

- We prove that

$$\mathbf{E}\left[\Upsilon^{t+1} \mid \mathfrak{F}^t \right] \leq \Upsilon^t - \Delta^t + 1.$$

For $k = \Theta(\Upsilon^t)$, for constant fraction of steps $r \in [t, t+k]$, $\mathbf{E}\left[\Delta^r \mid \mathfrak{F}^t \right] = \mathcal{O}(n)$.

- By induction we get,

$$\mathbf{E}[\Upsilon^{t+k+1} \mid \mathfrak{F}^t] \leq \Upsilon^t - \frac{1}{n} \cdot \sum_{r=t}^{t+k} \mathbf{E}[\Delta^r \mid \mathfrak{F}^t] + (k+1).$$

Bibliography I

- Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, *Balanced allocations*, SIAM J. Comput. **29** (1999), no. 1, 180–200. MR 1710347
- Petra Berenbrink, Artur Czumaj, Matthias Englert, Tom Friedetzky, and Lars Nagel, *Multiple-choice balanced allocation in (almost) parallel*, Proceedings of 16th International Workshop on Approximation, Randomization, and Combinatorial Optimization (RANDOM'12) (Berlin Heidelberg), Springer-Verlag, 2012, pp. 411–422.
- P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, *Balanced allocations: the heavily loaded case*, SIAM J. Comput. **35** (2006), no. 6, 1350–1385. MR 2217150
- Ohad N. Feldheim and Ori Gurel-Gurevich, *The power of thinning in balanced allocation*, Electron. Commun. Probab. **26** (2021), Paper No. 34, 8. MR 4275960
- O. N. Feldheim, O. Gurel-Gurevich, and J. Li, *Long-term balanced allocation via thinning*, 2021, arXiv:2110.05009.
- O. N. Feldheim and J. Li, *Load balancing under d-thinning*, Electron. Commun. Probab. **25** (2020), Paper No. 1, 13. MR 4053904

Bibliography II

- G. H. Gonnet, *Expected length of the longest probe sequence in hash code searching*, J. Assoc. Comput. Mach. **28** (1981), no. 2, 289–304. MR 612082
- K. Iwama and A. Kawachi, *Approximated two choices in randomized load balancing*, Algorithms and Computation (Berlin, Heidelberg) (Rudolf Fleischer and Gerhard Trippen, eds.), Springer Berlin Heidelberg, 2005, pp. 545–557.
- R. M. Karp, M. Luby, and F. Meyer auf der Heide, *Efficient PRAM simulation on a distributed memory machine*, Algorithmica **16** (1996), no. 4-5, 517–542. MR 1407587
- D. Los and T. Sauerwald, *Balanced allocations with incomplete information: The power of two queries*, 2021, arXiv:2107.03916.
- Dimitrios Los and Thomas Sauerwald, *Balanced allocations in batches: Simplified and generalized*, 2022.
- ———, *Balanced allocations with the choice of noise*, 2022.
- ———, *Tight bounds for repeated balls-into-bins*, 2022.

Bibliography III

- Dimitrios Los, Thomas Sauerwald, and John Sylvester, *The power of filling bins*, 2022.
- M. Mitzenmacher, *On the analysis of randomized load balancing schemes*, Theory Comput. Syst. **32** (1999), no. 3, 361–386. MR 1678304
- M. Mitzenmacher, B. Prabhakar, and D. Shah, *Load balancing with memory*, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings., IEEE, 2002, pp. 799–808.
- Y. Peres, K. Talwar, and U. Wieder, *Graphical balanced allocations and the $(1+\beta)$-choice process*, Random Structures Algorithms **47** (2015), no. 4, 760–775. MR 3418914
- M. Raab and A. Steger, *"Balls into bins"—a simple and tight analysis*, Proceedings of 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'98), vol. 1518, Springer, 1998, pp. 159–170. MR 1729169