Balanced Allocations: Caching and Packing, Twinning and Thinning

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

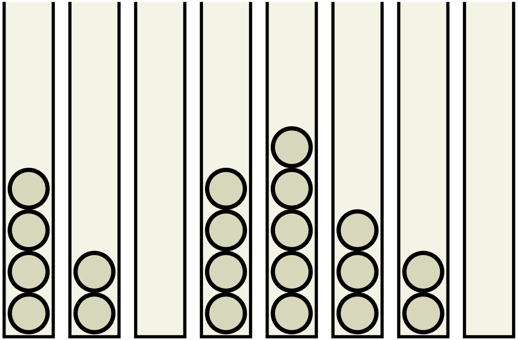
Allocate m tasks (balls) sequentially into n machines (bins).

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.

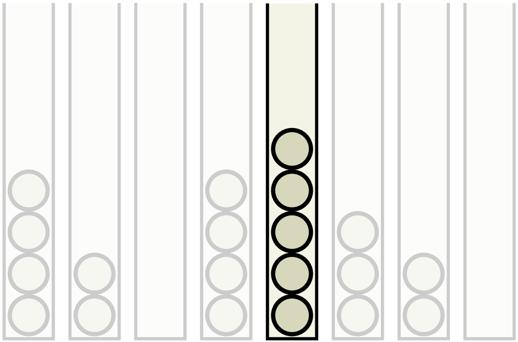
Balanced allocations setting

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.

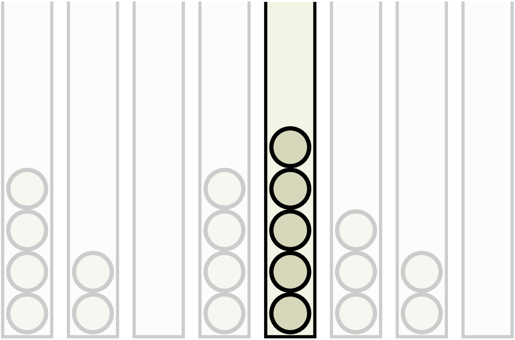


Balanced allocations setting

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

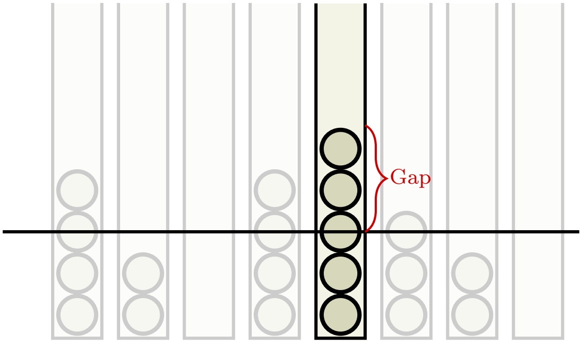
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.

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



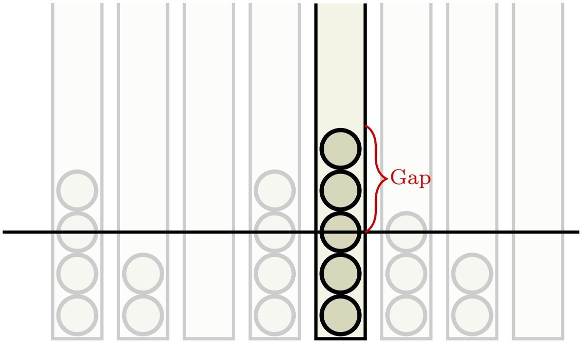
Balanced allocations setting

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 $\mathrm{Gap}(m) = \max_{i \in [n]}(x_i^m - m/n)$.



- Applications in hashing, load balancing and routing.

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. $\mathrm{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [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

<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.

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].

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.

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

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. $\mathrm{Gap}(n) = \Theta\Big(\frac{\log n}{\log\log n}\Big)$ [Gon81].
- In the heavily-loaded case ($m \gg n$), w.h.p. $\mathrm{Gap}(m) = \Theta\Big(\sqrt{\frac{m}{n} \cdot \log n}\Big)$ (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].
- 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 *mixing factor* $\beta \in (0,1]$.
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

<u>$(1+\beta)$ Process:</u>
Parameter: A *mixing factor* $\beta \in (0,1]$.
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.

- [Mit96] interpreted $(1-\beta)/2$ as the probability of making an erroneous comparison.

$(1+\beta)$ process: Definition

$(1+\beta)$ Process:
Parameter: A *mixing factor* $\beta \in (0,1]$.
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.

- [Mit96] 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 $1/n \leq \beta < 1-\epsilon$ for constant $\epsilon > 0$.

Two-Thinning **and** Twinning

Two-Thinning **with relative thresholds**

Relative-Threshold($f(n)$) Process:
Parameter: An *offset function* $f(n) \geq 0$.
Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., and update:

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

Two-Thinning **with relative thresholds**

Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.

Two-Thinning with relative thresholds

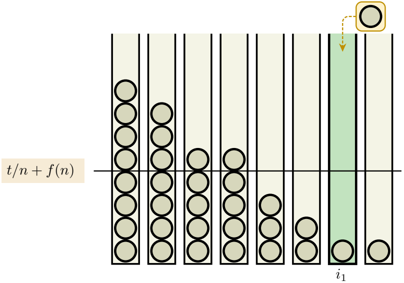
Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.



Two-Thinning **with relative thresholds**

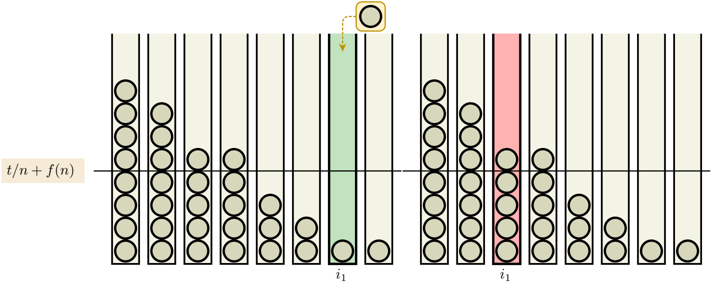
Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.



TWO-THINNING with relative thresholds

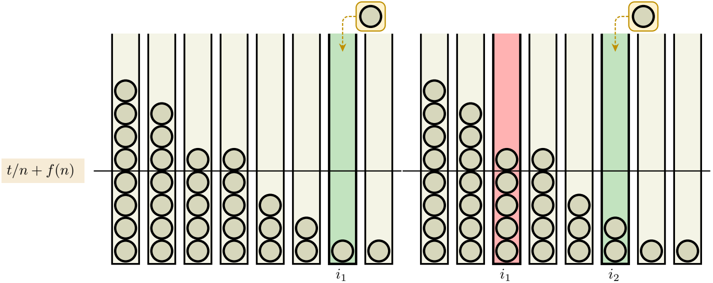
RELATIVE-THRESHOLD($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-THINNING has $f(n) = 0$.



Two-Thinning with relative thresholds

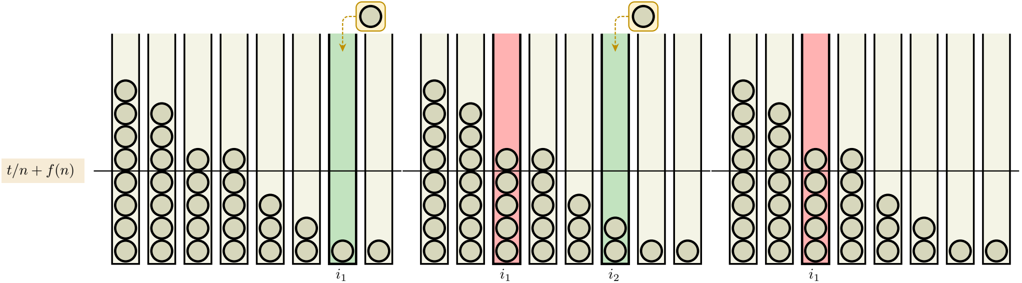
Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.



Two-Thinning with relative thresholds

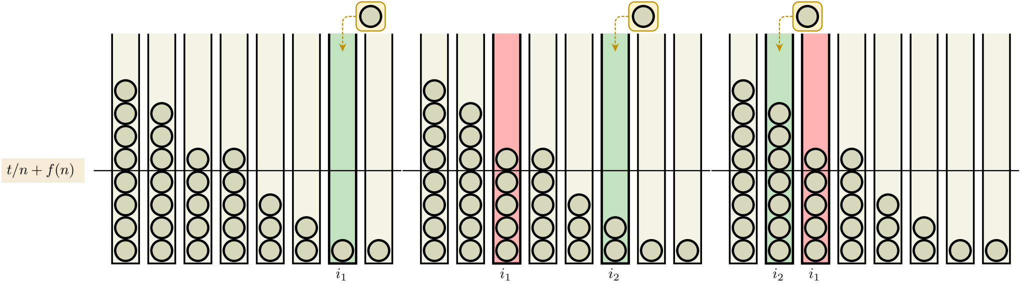
Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.



Two-Thinning with relative thresholds

Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.
- [FGG21] identified the optimal *adaptive* Two-Thinning, in the lightly-loaded case.

Two-Thinning with relative thresholds

Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.
- [FGG21] identified the optimal *adaptive* Two-Thinning, in the lightly-loaded case.
- [FGGL21, LS22] analyse Two-Thinning without relative thresholds, in the heavily-loaded case.

Two-Thinning with relative thresholds

Relative-Threshold($f(n)$) Process:
Parameter: An *offset function* $f(n) \geq 0$.
Iteration: For each $t \geq 0$, sample two bins i_1 and i_2 independently u.a.r., 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-Thinning has $f(n) = 0$.
- [FGG21] identified the optimal *adaptive* Two-Thinning, in the lightly-loaded case.
- [FGGL21, LS22] analyse Two-Thinning without relative thresholds, in the heavily-loaded case.
- [IK04, FL20, LS22] analyse d-Thinning processes.

Two-Thinning with relative thresholds

Relative-Threshold($f(n)$) Process:

Parameter: An *offset function* $f(n) \geq 0$.

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

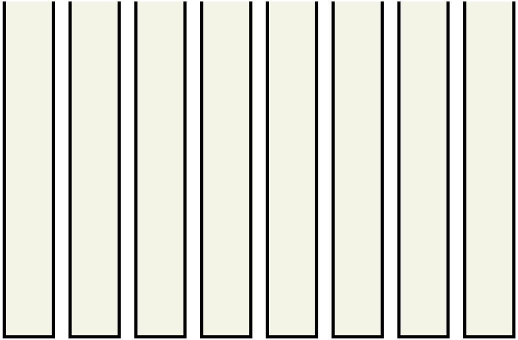
$$\begin{cases} x_{i_1}^{t+1} = x_{i_1}^{t} + 1 & \text{if } x_{i_1}^{t} < \frac{t}{n} + f(n), \\ x_{i_2}^{t+1} = x_{i_2}^{t} + 1 & \text{if } x_{i_1}^{t} \geq \frac{t}{n} + f(n). \end{cases}$$

- Mean-Thinning has $f(n) = 0$.
- [FGG21] identified the optimal *adaptive* Two-Thinning, in the lightly-loaded case.
- [FGGL21, LS22] analyse Two-Thinning without relative thresholds, in the heavily-loaded case.
- [IK04, FL20, LS22] analyse d-Thinning processes.

Open in Visualiser.

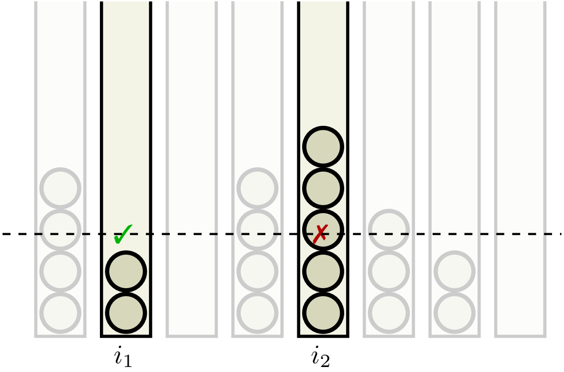
Two-Thinning as Two-Choice with incomplete information

We can interpret Two-Thinning 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.



Two-Thinning as Two-Choice with incomplete information

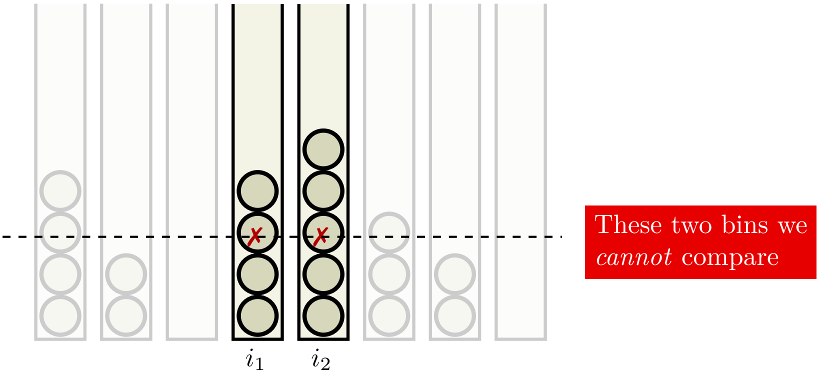
We can interpret Two-Thinning as an instance of the Two-Choice process, where we are only able to *compare* the loads of the two sampled bins if one is above the threshold and one is below.



These two bins we *can* compare

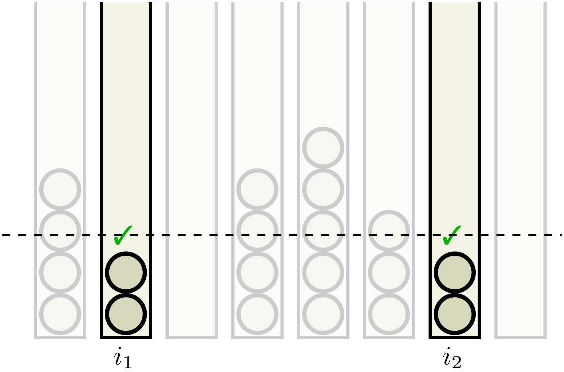
Two-Thinning as Two-Choice with incomplete information

We can interpret Two-Thinning 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.



Two-Thinning as Two-Choice with incomplete information

We can interpret Two-Thinning 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.



And, these two bins we *cannot* compare

Two-Thinning: Our results

Two-Thinning: **Our results**

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

TWO-THINNING: Our results

- For heavily-loaded case, MEAN-THINNING has w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\log n)$.

- For sufficiently large m, MEAN-THINNING has w.h.p. $\mathrm{Gap}(m) = \Omega(\log n)$.

Two-Thinning: **Our results**

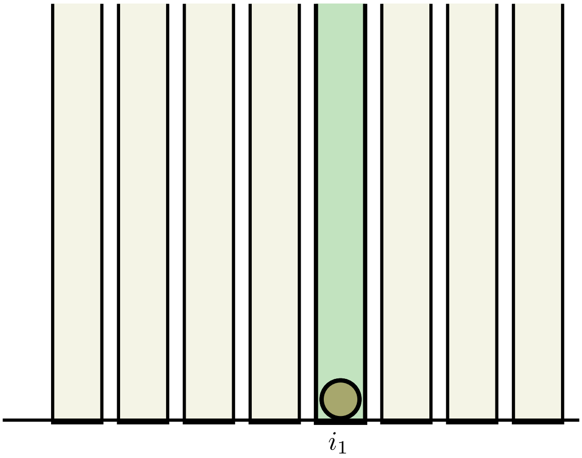
- For heavily-loaded case, Mean-Thinning has w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.

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

- By a coupling argument, Relative-Threshold$(f(n))$ with $f(n) \geq 0$ has w.h.p.

$$\text{Gap}(m) = f(n) + \mathcal{O}(\log n).$$

Mean-Thinning: **Visualisation**



Twinning: Definition

Twinning Process:

Iteration: For each $t \geq 0$, sample a bin i u.a.r., and update its load:

$$x_i^{t+1} = \begin{cases} x_i^t + 2 & \text{if } x_i^t < \frac{W^t}{n}, \\ x_i^t + 1 & \text{if } x_i^t \geq \frac{W^t}{n}. \end{cases}$$

TWINNING: **Definition**

TWINNING PROCESS:

Iteration: For each $t \geq 0$, sample a bin i u.a.r., and update its load:

$$x_i^{t+1} = \begin{cases} x_i^t + 2 & \text{if } x_i^t < \frac{W^t}{n}, \\ x_i^t + 1 & \text{if } x_i^t \geq \frac{W^t}{n}. \end{cases}$$

where $W^t := \sum_{i=1}^n x_i^t$

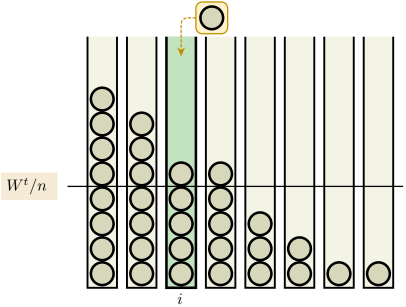
Twinning: Definition

<u>Twinning Process:</u>

Iteration: For each $t \geq 0$, sample a bin i u.a.r., and update its load:

$$x_i^{t+1} = \begin{cases} x_i^t + 2 & \text{if } x_i^t < \frac{W^t}{n}, \\ x_i^t + 1 & \text{if } x_i^t \geq \frac{W^t}{n}. \end{cases}$$

where $W^t := \sum_{i=1}^n x_i^t$



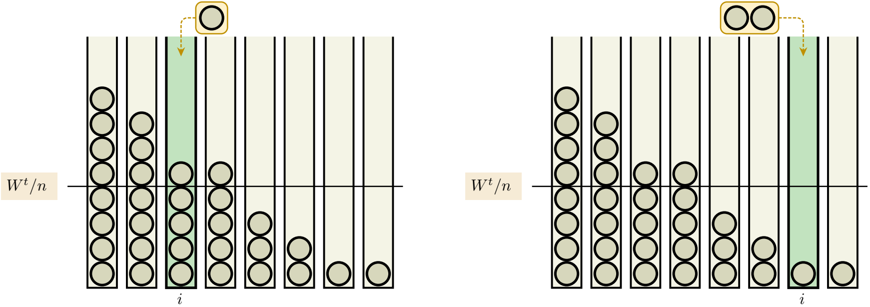
TWINNING: Definition

TWINNING PROCESS:

Iteration: For each $t \geq 0$, sample a bin i u.a.r., and update its load:

$$x_i^{t+1} = \begin{cases} x_i^t + 2 & \text{if } x_i^t < \frac{W^t}{n}, \\ x_i^t + 1 & \text{if } x_i^t \geq \frac{W^t}{n}. \end{cases}$$

where $W^t := \sum_{i=1}^n x_i^t$



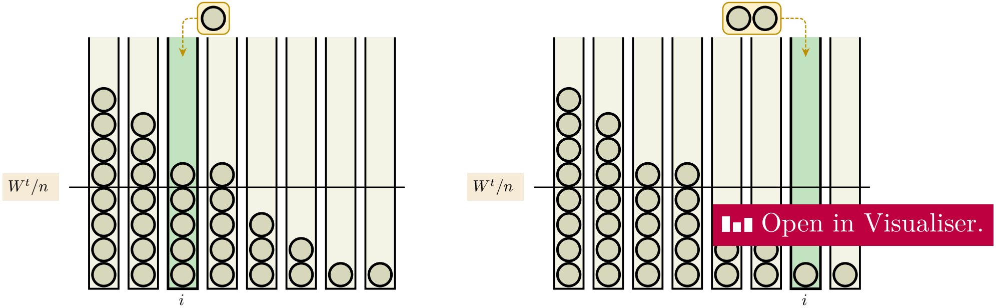
TWINNING: Definition

TWINNING PROCESS:

Iteration: For each $t \geq 0$, sample a bin i u.a.r., and update its load:

$$x_i^{t+1} = \begin{cases} x_i^t + 2 & \text{if } x_i^t < \frac{W^t}{n}, \\ x_i^t + 1 & \text{if } x_i^t \geq \frac{W^t}{n}. \end{cases}$$

where $W^t := \sum_{i=1}^n x_i^t$



TWINNING: Properties

- For the heavily loaded case, TWINNING has w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\log n)$.

Twinning: Properties

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

- Twinning w.h.p. uses $1 - \epsilon$ samples per allocatied ball, for const $\epsilon > 0$.

TWINNING: Properties

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

- TWINNING w.h.p. uses $1 - \epsilon$ samples per allocatied ball, for const $\epsilon > 0$.

- However, the twinning operation *may not* always be implementable in practice.

Probability allocation vectors

Probability allocation vectors

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

Probability allocation vectors

- **Probability allocation vector** p^t, where p_i^t is the prob. of allocating to i-th heaviest bin.
- For One-Choice and Two-Choice, p is *time-independent*,

$$p_{\textsc{One-Choice}} = \Big(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\Big),$$

$$p_{\textsc{Two-Choice}} = \Big(\frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i-1}{n^2}, \ldots, \frac{2n-2}{n^2}\Big).$$

Probability allocation vectors

- **Probability allocation vector** p^t, where p_i^t is the prob. of allocating to i-th heaviest bin.
- For ONE-CHOICE and TWO-CHOICE, p is *time-independent*,

$$p_{\textsc{One-Choice}} = \Big(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\Big),$$

$$p_{\textsc{Two-Choice}} = \Big(\frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i-1}{n^2}, \ldots, \frac{2n-2}{n^2}\Big).$$

- However, for MEAN-THINNING, p^t depends on the load distribution,

$$p^t_{\textsc{Mean-Thinning}}(x^t) = \Big(\underbrace{\frac{\delta^t}{n}, \frac{\delta^t}{n}, \ldots, \frac{\delta^t}{n}}_{\delta \cdot n \text{ entries}}, \underbrace{\frac{1+\delta^t}{n}, \ldots, \frac{1+\delta^t}{n}}_{(1-\delta^t)\cdot n \text{ entries}}\Big),$$

where $\delta^t \in [1/n, 1]$ is the **quantile of the mean**.

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition** $\mathcal{P}$: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition** $\mathcal{P}$: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

  - (**Underloaded bins**) For each bin i with $x_i^t < t/n$,

$$p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-^t.$$

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition** $\mathcal{P}$: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

  - (**Underloaded bins**) For each bin i with $x_i^t < t/n$,

$$p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-^t.$$

- **Condition** $\mathcal{W}$: When bin i is chosen for allocation,
  - (**Overloaded bins**) If $x_i^t \geq W^t/n$, then allocate w_+ balls,
  - (**Underloaded bins**) If $x_i^t < W^t/n$, then allocate w_- balls,

  where w_+, w_- are positive integer constants.

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition** $\mathcal{P}$: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

  - (**Underloaded bins**) For each bin i with $x_i^t < t/n$,

$$p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-^t.$$

- **Condition** $\mathcal{W}$: When bin i is chosen for allocation,
  - (**Overloaded bins**) If $x_i^t \geq W^t/n$, then allocate w_+ balls,
  - (**Underloaded bins**) If $x_i^t < W^t/n$, then allocate w_- balls,

  where w_+, w_- are positive integer constants.

	$k_1, k_2 \geq 0$	$k_1, k_2 > 0$
$w_+ \leq w_-$		
$w_+ < w_-$		

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition** $\mathcal{P}$: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

  - (**Underloaded bins**) For each bin i with $x_i^t < t/n$,

$$p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-^t.$$

- **Condition** $\mathcal{W}$: When bin i is chosen for allocation,
  - (**Overloaded bins**) If $x_i^t \geq W^t/n$, then allocate w_+ balls,
  - (**Underloaded bins**) If $x_i^t < W^t/n$, then allocate w_- balls,

  where w_+, w_- are positive integer constants.

One-Choice

	$k_1, k_2 \geq 0$	$k_1, k_2 > 0$
$w_+ \leq w_-$		
$w_+ < w_-$		

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition** $\mathcal{P}$: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

  - (**Underloaded bins**) For each bin i with $x_i^t < t/n$,

$$p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-^t.$$

- **Condition** $\mathcal{W}$: When bin i is chosen for allocation,
  - (**Overloaded bins**) If $x_i^t \geq W^t/n$, then allocate w_+ balls,
  - (**Underloaded bins**) If $x_i^t < W^t/n$, then allocate w_- balls,

where w_+, w_- are positive integer constants.

	$k_1, k_2 \geq 0$	$k_1, k_2 > 0$
$w_+ \leq w_-$		
$w_+ < w_-$		

One-Choice

Twinning

Framework: Probability and weight bias

For processes with probability vector p^t such that for each round $t \geq 0$:

- **Condition $\mathcal{P}$**: There exist constants k_1, k_2, such that
  - (**Overloaded bins**) For each bin i with $x_i^t \geq t/n$,

$$p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+^t.$$

  - (**Underloaded bins**) For each bin i with $x_i^t < t/n$,

$$p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-^t.$$

- **Condition $\mathcal{W}$**: When bin i is chosen for allocation,
  - (**Overloaded bins**) If $x_i^t \geq W^t/n$, then allocate w_+ balls,
  - (**Underloaded bins**) If $x_i^t < W^t/n$, then allocate w_- balls,

  where w_+, w_- are positive integer constants.

	$k_1, k_2 \geq 0$	$k_1, k_2 > 0$
$w_+ \leq w_-$		
$w_+ < w_-$		

One-Choice

Mean-Thinning, $(1+\beta)$,
Two-Choice

Twinning

Outline of the analysis

The hyperbolic cosine potential function

The hyperbolic cosine potential function

- [PTW15] used the **hyperbolic cosine potential**

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

The hyperbolic cosine potential function

- [PTW15] used the **hyperbolic cosine potential**

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

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

The hyperbolic cosine potential function

- [PTW15] used the **hyperbolic cosine potential**

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

- For the $(1+\beta)$ process, $\gamma = \Theta(\beta)$.
- [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 hyperbolic cosine potential function

- [PTW15] used the **hyperbolic cosine potential**

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

- For the $(1+\beta)$ process, $\gamma = \Theta(\beta)$.
- [PTW15] show that $\mathbf{E}\left[\,\Gamma^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- By *induction*, this implies $\mathbf{E}\left[\,\Gamma^t\,\right] \leq cn$ for any $t \geq 0$.

The hyperbolic cosine potential function

- [PTW15] used the **hyperbolic cosine potential**

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

- For the $(1+\beta)$ process, $\gamma = \Theta(\beta)$.
- [PTW15] show that $\mathbf{E}\left[\Gamma^{t+1} \mid \mathfrak{F}^t\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2$.
- By *induction*, this implies $\mathbf{E}\left[\Gamma^t\right] \leq cn$ 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

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

Mean-Thinning: **Why the analysis is tricky**

- If δ^t is very large, say $\delta^t = 1 - 1/n$, then p^t becomes *very close* to the One-Choice vector :

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

Mean-Thinning: **Why the analysis is tricky**

- If δ^t is very large, say $\delta^t = 1 - 1/n$, then p^t becomes *very close* to the One-Choice vector :

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

- 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 $\gamma = \Theta(1/n)$.

Mean-Thinning: Why the analysis is tricky

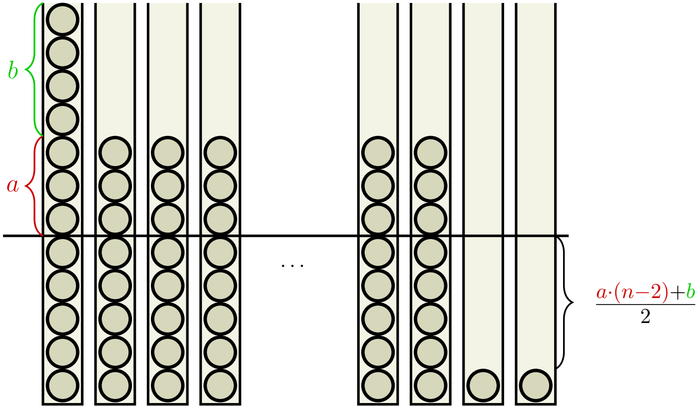
- If δ^t is very large, say $\delta^t = 1 - 1/n$, then p^t becomes *very close* to the One-Choice vector :

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

- 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 $\gamma = \Theta(1/n)$.

But what happens for Γ^t with constant γ?

Mean-Thinning: Bad configuration



- There is a very small bias to allocate *away* from overloaded bins.
- **The potential $\Gamma := \Gamma(\gamma)$ for constant γ *increases* in expectation.**

A closer look at Γ^t

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

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

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

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

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

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

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

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

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

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

- A properly *adjusted 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

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

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

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

- A properly *adjusted 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?

Mean quantile stabilisation

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

- 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.

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

- 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**,

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

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

- 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**,

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

- We prove that

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t - (p_-^t \cdot w_- - p_+^t \cdot w_+) \cdot \Delta^t + 4 \cdot (w_-)^2.$$

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

- 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**,

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

- We prove that

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \Delta^t + \kappa_2.$$

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

- 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**,

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

- We prove that

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \Delta^t + \kappa_2.$$

- By *induction* we get,

$$\mathbf{E}[\, \Upsilon^{t+k+1} \mid \mathfrak{F}^t \,] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \sum_{r=t}^{t+k} \mathbf{E}[\, \Delta^r \mid \mathfrak{F}^t \,] + \kappa_2 \cdot (k+1).$$

Mean quantile stabilisation

- Consider the **absolute value potential**,

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

- 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**,

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

- We prove that

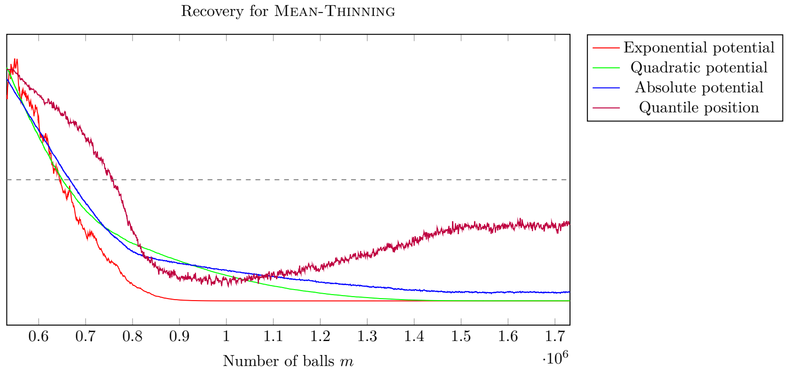
$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \Delta^t + \kappa_2.$$

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

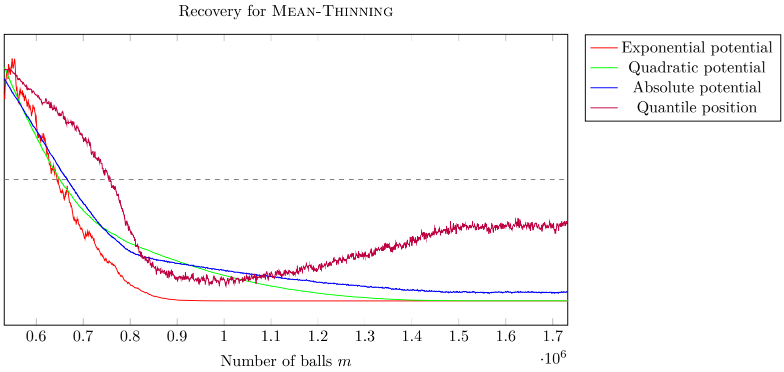
- By *induction* we get,

$$\mathbf{E}[\, \Upsilon^{t+k+1} \mid \mathfrak{F}^t \,] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \sum_{r=t}^{t+k} \mathbf{E}[\, \Delta^r \mid \mathfrak{F}^t \,] + \kappa_2 \cdot (k+1).$$

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

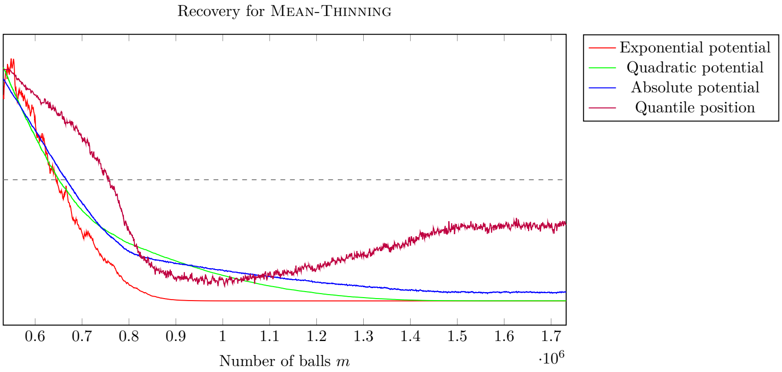


Recovery from a bad configuration ($n = 1000$)



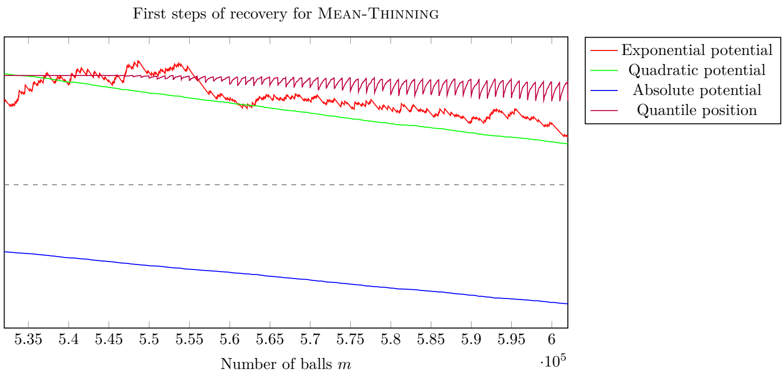
- As long as $\Delta^t = \Omega(n)$, Υ^t drops in expectation.

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



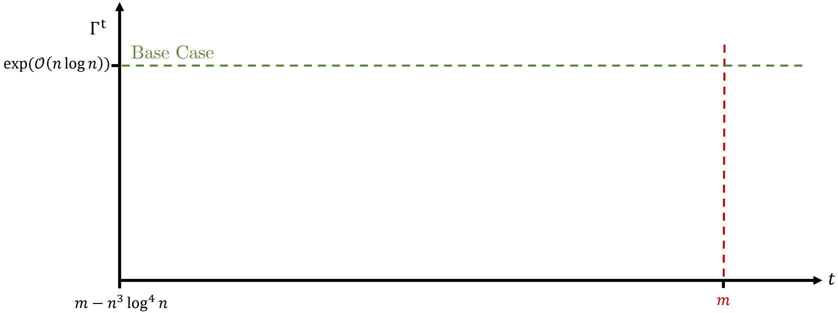
- As long as $\Delta^t = \Omega(n)$, Υ^t drops in expectation.
- As Δ^t becomes smaller, δ^t improves and Γ^t drops in expectation.

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

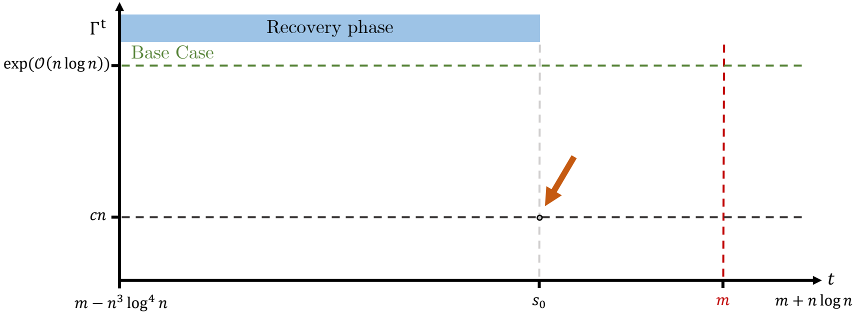


- As long as $\Delta^t = \Omega(n)$, Υ^t drops in expectation.
- As Δ^t becomes smaller, δ^t improves and Γ^t drops in expectation.

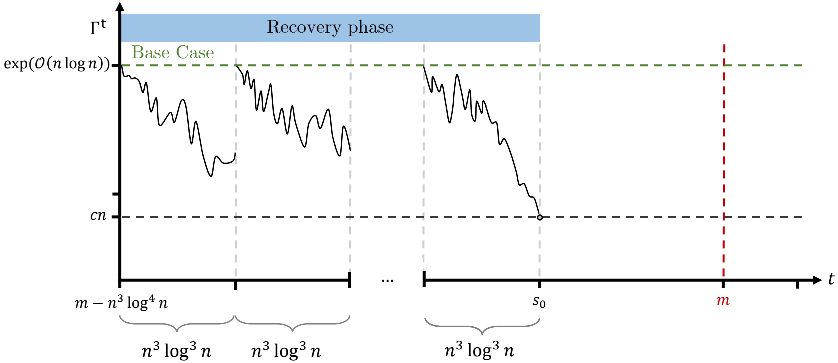
Completing the analysis



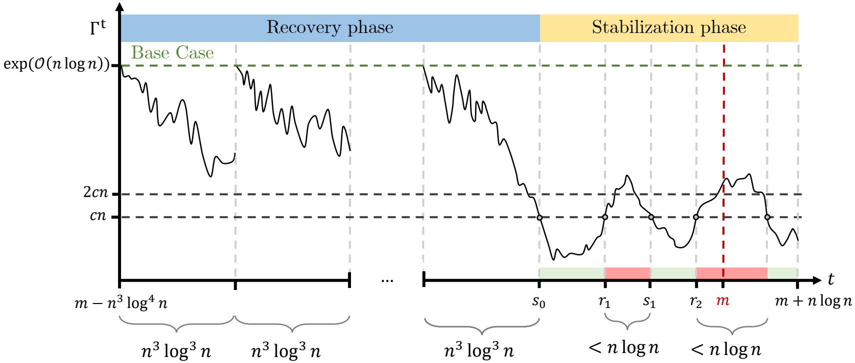
Completing the analysis



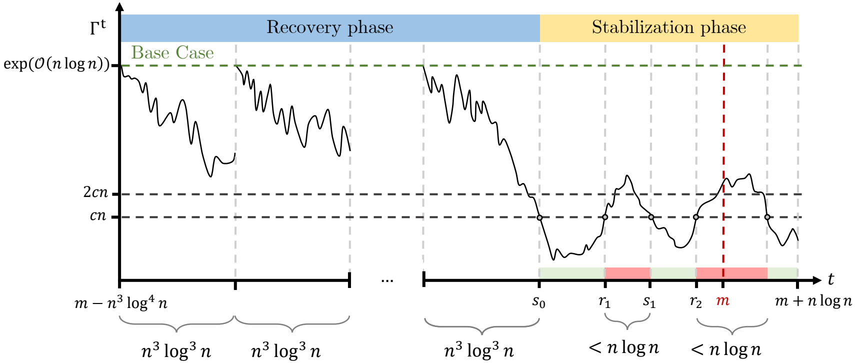
Completing the analysis



Completing the analysis

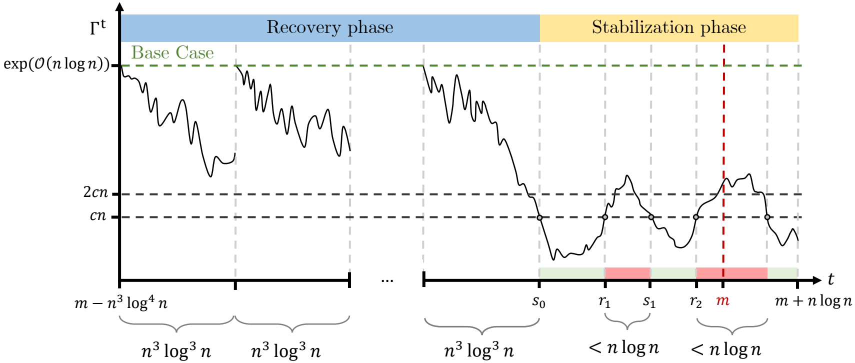


Completing the analysis



Other applications of *quantile stabilisation*:

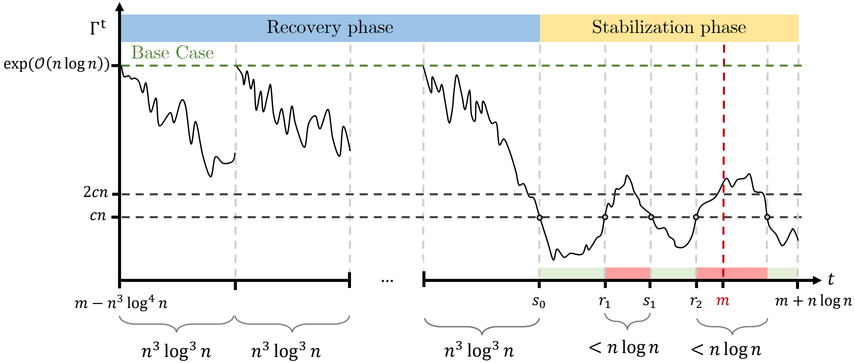
Completing the analysis



Other applications of *quantile stabilisation*:

- Sample efficiency: $2 - \epsilon$ for Mean-Thinning and $1 - \epsilon$ for Twinning.

Completing the analysis

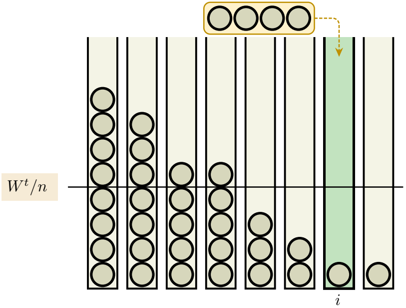


Other applications of *quantile stabilisation*:

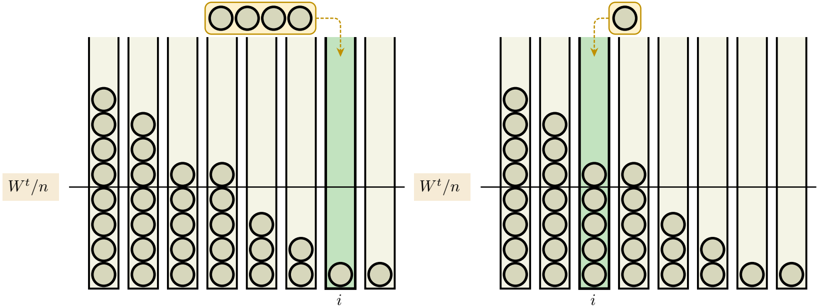
- Sample efficiency: $2 - \epsilon$ for Mean-Thinning and $1 - \epsilon$ for Twinning.
- Lower bound of $\Omega(\log n)$ for Mean-Thinning and Twinning.

Packing **(and** Caching)

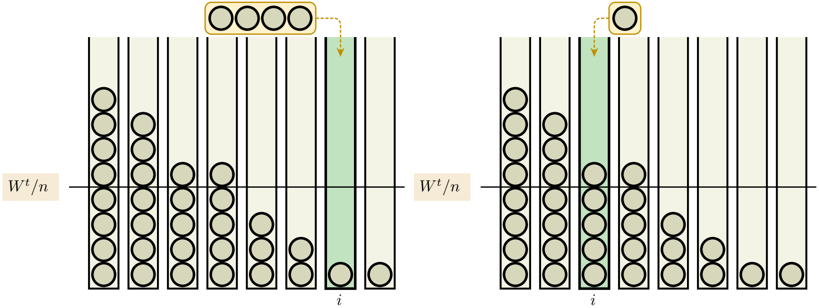
Packing: Definition



Packing: Definition



Packing: Definition

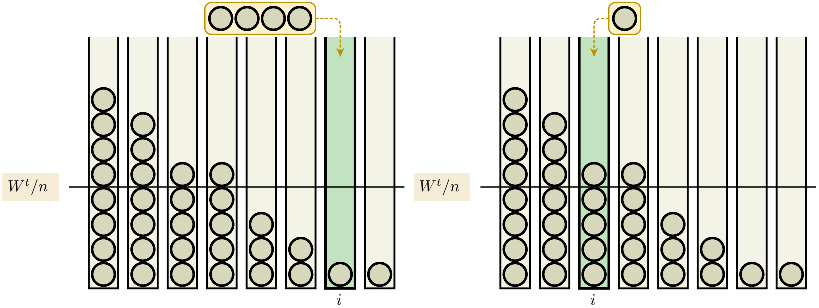


Packing Process:

Iteration: For each $t \geq 0$, sample bin i u.a.r., and update its load:

$$x_i^{t+1} = \begin{cases} \left\lceil \frac{W^t}{n} \right\rceil + 1 & \text{if } x_i^t < \frac{W^t}{n}, \\ x_i^t + 1 & \text{if } x_i^t \geq \frac{W^t}{n}. \end{cases}$$

Packing: Definition



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

Conclusion

Conclusion

Summary of results:

Conclusion

Summary of results:

- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including MEAN-THINNING and TWINNING.

Conclusion

Summary of results:

- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Mean-Thinning and Twinning.
- Proved a matching lower bound for Mean-Thinning and Twinning.

Conclusion

Summary of results:

- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Mean-Thinning and Twinning.
- Proved a matching lower bound for Mean-Thinning and Twinning.
- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Packing and Caching.

Conclusion

Summary of results:

- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including MEAN-THINNING and TWINNING.
- Proved a matching lower bound for MEAN-THINNING and TWINNING.
- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including PACKING and CACHING.

Future work:

Conclusion

Summary of results:

- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Mean-Thinning and Twinning.
- Proved a matching lower bound for Mean-Thinning and Twinning.
- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Packing and Caching.

Future work:

- Extend the framework to *non-constant* probability and weight biases.

Conclusion

Summary of results:

- Proved $\mathrm{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including MEAN-THINNING and TWINNING.
- Proved a matching lower bound for MEAN-THINNING and TWINNING.
- Proved $\mathrm{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including PACKING and CACHING.

Future work:

- Extend the framework to *non-constant* probability and weight biases.
- Find a natural framework that implies $o(\log n)$ gap bounds.

Conclusion

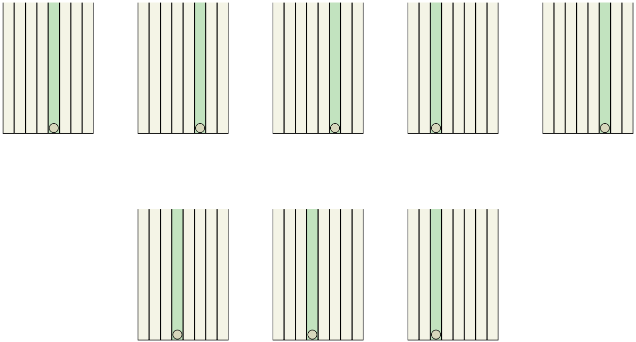
Summary of results:

- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Mean-Thinning and Twinning.
- Proved a matching lower bound for Mean-Thinning and Twinning.
- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including Packing and Caching.

Future work:

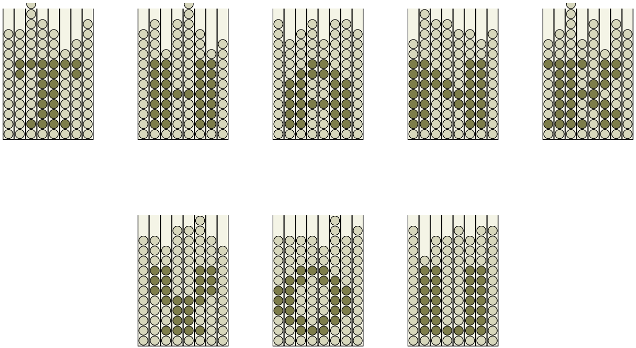
- Extend the framework to *non-constant* probability and weight biases.
- Find a natural framework that implies $o(\log n)$ gap bounds.
- Investigate Mean-Thinning with *outdated information* and *noise*.

Questions?



Visualisations: `dimitrioslos.com/soda22`

Questions?



Visualisations: `dimitrioslos.com/soda22`

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]		$\log n$	$\log n$
Caching	$\log \log n$ [MPS02]		–	$\log n$
Packing	$\frac{\log n}{\log \log n}$		$\frac{\log n}{\log \log n}$	$\log n$
Twinning	$\frac{\log n}{\log \log n}$		$\log n$	
Mean-Thinning	$\frac{\log n}{\log \log n}$		$\log n$	
Relative-Threshold$(f(n))$	$\sqrt{\frac{\log n}{\log \log n}}$ [FL20]	$\frac{\log n}{\log \log n}$	$\frac{\log n}{\log \log n}$ [LS22]	$f(n) + \log n$
Adaptive-Two-Thinning	$\sqrt{\frac{\log n}{\log \log n}}$ [FL20]		$\frac{\log n}{\log \log n}$ [LS22]	$\frac{\log n}{\log \log n}$ [FGGL21]

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

Appendix B: Experimental results

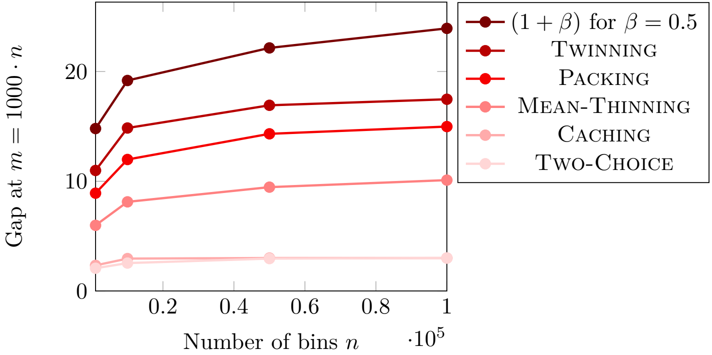


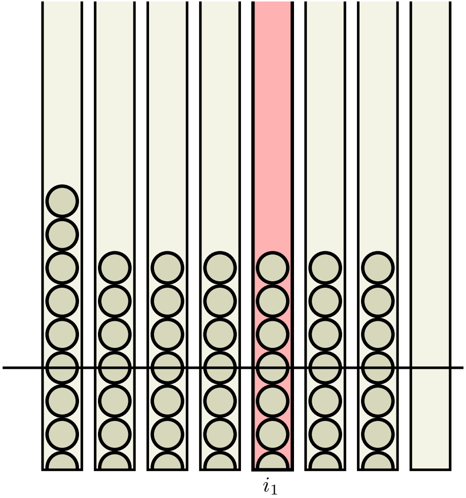
Figure: Average Gap vs. $n \in \{10^3, 10^4, 5 \cdot 10^4, 10^5\}$ and $m = 1000 \cdot n$.

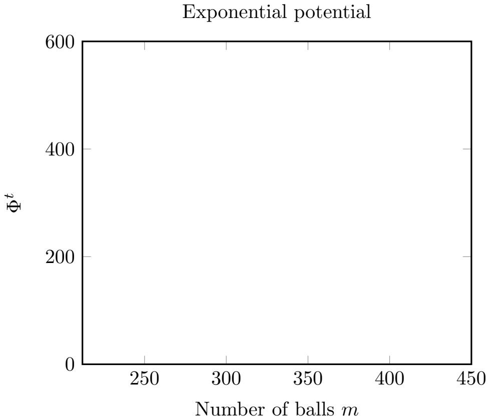
Appendix C: Detailed experimental results

n	Mean-Thinning	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%

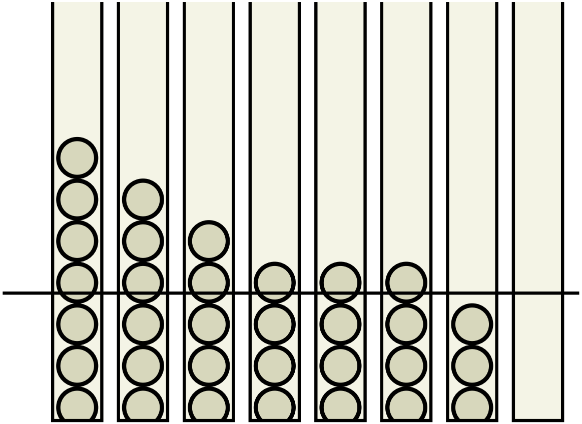
Table: Summary of observed gaps for $n \in \{10^3, 10^4, 10^5\}$ bins and $m = 1000 \cdot n$ number of balls, for 100 repetitions. The observed gaps are in bold and next to that is the % of runs where this *gap value* was observed.

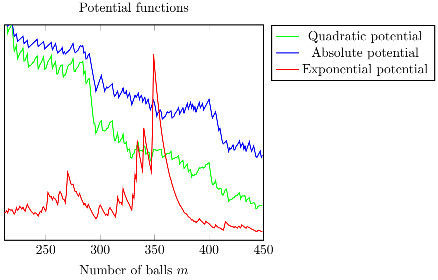
Appendix D1: Recovery from a bad configuration





Appendix D2: Recovery from a bad configuration





Bibliography I

- Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, *Balanced allocations*, SIAM J. Comput. **29** (1999), no. 1, 180–200.
- P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, *Balanced allocations: the heavily loaded case*, SIAM J. Comput. **35** (2006), no. 6, 1350–1385.
- O. N. Feldheim and O. Gurel-Gurevich, *The power of thinning in balanced allocation*, Electron. Commun. Probab. **26** (2021), Paper No. 34, 8.
- 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.
- G. H. Gonnet, *Expected length of the longest probe sequence in hash code searching*, J. Assoc. Comput. Mach. **28** (1981), no. 2, 289–304.

Bibliography II

- Kazuo Iwama and Akinori Kawachi, *Approximated two choices in randomized load balancing*, 15th International Symposium on Algorithms and Computation (ISAAC'04), Lecture Notes in Computer Science, vol. 3341, Springer, 2004, 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.
- Dimitrios Los and Thomas Sauerwald, *Balanced allocations with incomplete information: The power of two queries*, 13th Innovations in Theoretical Computer Science Conference (ITCS'22), vol. 215, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 103:1–103:23.
- M. Mitzenmacher, *The power of two choices in randomized load balancing*, Ph.D. thesis, University of California at Berkeley, 1996.
- M. Mitzenmacher, B. Prabhakar, and D. Shah, *Load balancing with memory*, 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'02), IEEE, 2002, pp. 799–808.

Bibliography III

- 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.
- M. Raab and A. Steger, *"Balls into bins"—a simple and tight analysis*, 2nd International Workshop on Randomization and Computation (RANDOM'98), vol. 1518, Springer, 1998, pp. 159–170.