Balanced Allocations: Caching and Packing, Twinning and Thinning

Dimitrios Los\textsuperscript{1}, Thomas Sauerwald\textsuperscript{1}, John Sylvester\textsuperscript{2}

\textsuperscript{1}University of Cambridge, UK
\textsuperscript{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 $t$. 

![Image of balls distributed among machines](image-url)
Balanced allocations setting

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

**Goal:** minimise the maximum load $\max_{i \in [n]} x^m_i$, 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 $t$.

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

![Diagram showing the gap between load vectors and the maximum load](image-url)
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.
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.

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

- 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**

**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].

**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. $\text{Gap}(n) = \log \frac{2}{2} \log n + \Theta(1)$ [KLMadH96, ABKU99].

- In the heavily-loaded case ($m \gg n$), w.h.p. $\text{Gap}(m) = \log \frac{2}{2} \log n + \Theta(1)$ [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. $\text{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].

**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].

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


**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]).
**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 ($m \gg n$), w.h.p. $\text{Gap}(m) = \log_2 \log n + \Theta(1)$ [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

**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].
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].
## 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**

**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].
\((1 + \beta)\) process: Definition

(1 + \beta) Process:
Parameter: A mixing factor \( \beta \in (0, 1] \).
Iteration: For each \( t \geq 0 \), with probability \( \beta \) allocate one ball via the **TWO-CHOICE** process, otherwise allocate one ball via the **ONE-CHOICE** process.
(1 + β) process: Definition

(1 + β) Process:
Parameter: A mixing factor β ∈ (0, 1].
Iteration: For each t ≥ 0, with probability β allocate one ball via the TWO-CHOICE process, otherwise allocate one ball via the ONE-CHOICE process.

[Mit96] interpreted (1 − β)/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 $\beta$ 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:

\[
\begin{align*}
    x_{i_1}^{t+1} &= x_{i_1}^t + 1 & \text{if } x_{i_1}^t < \frac{t}{n} + f(n), \\
    x_{i_2}^{t+1} &= x_{i_2}^t + 1 & \text{if } x_{i_1}^t \geq \frac{t}{n} + f(n).
\end{align*}
\]
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:

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

- **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:

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

- **Mean-Thinning** has $f(n) = 0$. 

---

\[ t/n + f(n) \]

\[ i_1 \]
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:

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

**Mean-Thinning** has $f(n) = 0$. 

---

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

---

Diagram showing the process of Two-Thinning with relative thresholds.
**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:

\[
\begin{align*}
x_{i_1}^{t+1} &= x_{i_1}^t + 1 \quad \text{if } x_{i_1}^t < \frac{t}{n} + f(n), \\
x_{i_2}^{t+1} &= x_{i_2}^t + 1 \quad \text{if } x_{i_1}^t \geq \frac{t}{n} + f(n).
\end{align*}
\]

- **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:

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

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 $f(n) = 0$.

[Diagram of the process showing two bins being sampled and updated based on the relative thresholds.]
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:

$$
\begin{align*}
    x^{t+1}_{i_1} &= x^t_{i_1} + 1 & \text{if } x^t_{i_1} < \frac{t}{n} + f(n), \\
    x^{t+1}_{i_2} &= x^t_{i_2} + 1 & \text{if } x^t_{i_1} \geq \frac{t}{n} + f(n).
\end{align*}
$$

- **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:

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

- **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:

\[
\begin{align*}
    x_{i_1}^{t+1} &= x_{i_1}^t + 1 \quad \text{if } x_{i_1}^t < \frac{t}{n} + f(n), \\
    x_{i_2}^{t+1} &= x_{i_2}^t + 1 \quad \text{if } x_{i_1}^t \geq \frac{t}{n} + f(n).
\end{align*}
\]

- **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 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.
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 cannot 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: Our results

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

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

By a coupling argument, \( \text{Relative-Threshold} (f(n)) \) with \( f(n) \geq 0 \) has w.h.p. \( \text{Gap}(m) = f(n) + O(\log n) \).
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. $\text{Gap}(m) = f(n) + \mathcal{O}(\log n)$. 

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. $\text{Gap}(m) = \Omega(\log n)$. 
**Two-Thinning: Our results**

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

- 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.

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

**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: Definition**

**TWINNING Process:**

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

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

where $W_{i}^{t} := \sum_{i=1}^{n} x_{i}^{t}$.
For the heavily loaded case, \textsc{Twinning} has w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.
**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 allocated ball, for const $\epsilon > 0$. 
**TWINNING: Properties**

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

- **TWINNING** w.h.p. uses \( 1 - \epsilon \) samples per allocated ball, for const \( \epsilon > 0 \).

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

- Probability allocation vector $p_t$, where $p_t[i]$ is the probability of allocating to the $i$-th heaviest bin.

- For One-Choice and Two-Choice, $p$ is time-independent: $p_{One-Choice} = (1/n, 1/n, \ldots, 1/n)$, $p_{Two-Choice} = (1/n^2, 3/n^2, \ldots, 2i/n^2, \ldots, 2n/n^2)$.

- However, for Mean-Thinning, $p_t$ depends on the load distribution: $p_{t,\text{Mean-Thinning}}(x_t) = (\delta_{t,n}, \delta_{t,n}, \ldots, \delta_{t,n}) \cdot n$ entries, $1 + \delta_{t,n}, \ldots, 1 + \delta_{t,n}$ entries, $(1 - \delta_{t,n}) \cdot n$ entries, where $\delta_{t,n} \in [1/n, 1]$ is the quantile of the mean.
Probability allocation vectors

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

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

- For **One-Choice and Two-Choice**, $p$ is *time-independent*,

  $$p_{\text{One-Choice}} = \left( \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \right),$$

  $$p_{\text{Two-Choice}} = \left( \frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i-1}{n^2}, \ldots, \frac{2n-2}{n^2} \right).$$
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*,

  \[
  p_{\text{One-Choice}} = \left( \frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n} \right),
  \]

  \[
  p_{\text{Two-Choice}} = \left( \frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i - 1}{n^2}, \ldots, \frac{2n - 2}{n^2} \right).
  \]

- However, for **Mean-Thinning**, $p^t$ depends on the load distribution,

  \[
  p^t_{\text{Mean-Thinning}}(x^t) = \left( \frac{\delta^t}{n}, \frac{\delta^t}{n}, \ldots, \frac{\delta^t}{n}, \frac{1 + \delta^t}{n}, \ldots, \frac{1 + \delta^t}{n} \right),
  \]

  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 $x^t_i \geq t/n$,

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

  ▶ **(Underloaded bins)** For each bin $i$ with $x^t_i < t/n$,
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 $x_i^t \geq t/n$,
    \[
    p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+.
    \]
  - **(Underloaded bins)** For each bin $i$ with $x_i^t < t/n$,
    \[
    p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-.
    \]
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 $x^t_i \geq t/n$,
    
    $$p^t_i \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p^+_t.$$

  - **(Underloaded bins)** For each bin $i$ with $x^t_i < t/n$,
    
    $$p^t_i \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p^-_t.$$

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

<table>
<thead>
<tr>
<th>$w_+$</th>
<th>$w_-$</th>
<th>$k_1, k_2 \geq 0$</th>
<th>$k_1, k_2 &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_+ \leq w_-$</td>
<td>Red</td>
<td>Green</td>
<td>Green</td>
</tr>
<tr>
<td>$w_+ &lt; w_-$</td>
<td>Green</td>
<td>Green</td>
<td>Green</td>
</tr>
</tbody>
</table>
Framework: Probability and weight bias

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

- **Condition $P$:** 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 $W$:** 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.

\[
\begin{array}{|c|c|c|}
\hline
\text{ } & k_1, k_2 \geq 0 & k_1, k_2 > 0 \\
\hline
w_+ \leq w_- & \text{Red} & \text{Green} \\
\hline
w_+ < w_- & \text{Green} & \text{Green} \\
\hline
\end{array}
\]

**ONE-CHOICE**
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 $x_i^t \geq t/n$,
    \[
    p_i^t \leq \frac{1}{n} - \frac{k_1 \cdot (1 - \delta^t)}{n} =: p_+.
    \]
  - (Underloaded bins) For each bin $i$ with $x_i^t < t/n$,
    \[
    p_i^t \geq \frac{1}{n} + \frac{k_2 \cdot \delta^t}{n} =: p_-.
    \]

- **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,

\[
\begin{array}{c | c c}
  & k_1, k_2 \geq 0 & k_1, k_2 > 0 \\
\hline
w_+ \leq w_- & \text{Red} & \text{Green} \\
\hline
w_+ < w_- & \text{Green} & \text{Green}
\end{array}
\]

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.

<table>
<thead>
<tr>
<th></th>
<th>$k_1, k_2 \geq 0$</th>
<th>$k_1, k_2 &gt; 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_+ \leq w_-$</td>
<td><strong>Red</strong></td>
<td><strong>Green</strong></td>
</tr>
<tr>
<td>$w_+ &lt; w_-$</td>
<td><strong>Green</strong></td>
<td><strong>Green</strong></td>
</tr>
</tbody>
</table>

- **One-Choice**

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

- **Twinning**
Outline of the analysis
The hyperbolic cosine potential function

\[ \Gamma_t = \sum_{i=1}^{\infty} e^{\gamma (x_t - W_t/n)} + \sum_{i=1}^{\infty} e^{-\gamma (x_t - W_t/n)} \]

For the \((1 + \beta)^2\) process, \(\gamma = \Theta(\beta)\).

\[ E[\Gamma_t + 1 | F_t] \leq \Gamma_t \cdot (1 - c_1 n) + c_2. \]

By induction, this implies \(E[\Gamma_t] \leq c n\) for any \(t \geq 0\).

By Markov's inequality, we get \(\Pr[\Gamma_m \leq c n^3] \geq 1 - n^{-2}\) which implies \(\Pr[\text{Gap}(m) \leq 1/\gamma (3 \log n + \log c)] \geq 1 - n^{-2}\).
The hyperbolic cosine potential function

- [PTW15] used the \textbf{hyperbolic cosine potential}

\[ \Gamma^t := \Gamma^t(\gamma) := \sum_{i=1}^{n} e^{\gamma(x_i^t-W^t/n)} + \sum_{i=1}^{n} e^{-\gamma(x_i^t-W^t/n)}. \]

Overload potential \hspace{1cm} Underload potential

By induction, this implies \( \mathbb{E}[\Gamma^t] \leq cn \) for any \( t \geq 0 \).

By Markov's inequality, we get \( \Pr[\Gamma^m \leq cn^3] \geq 1 - n^{-2} \) which implies \( \Pr[\text{Gap}(m) \leq 1 + 3 \cdot \log n + \log c] \geq 1 - n^{-2} \).
The hyperbolic cosine potential function

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

\[
\Gamma_t := \Gamma^t(\gamma) := \sum_{i=1}^{n} e^{\gamma(x_i^t - W_t^t/n)} + \sum_{i=1}^{n} e^{-\gamma(x_i^t - W_t^t/n)} .
\]

- Overload potential
- Underload potential

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

- [PTW15] used the hyperbolic cosine potential

\[ \Gamma^t := \Gamma^t(\gamma) := \sum_{i=1}^{n} e^{\gamma(x^t_i - W^t/n)} + \sum_{i=1}^{n} e^{-\gamma(x^t_i - W^t/n)}. \]

\text{Overload potential} \quad \text{Underload potential}

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

- [PTW15] show that \(E \left[ \Gamma^{t+1} \mid \mathcal{F}^t \right] \leq \Gamma^t \cdot (1 - \frac{c_1}{n}) + c_2.\)
The hyperbolic cosine potential function

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

\[ \Gamma^t := \Gamma^t(\gamma) := \sum_{i=1}^{n} e^{\gamma(x_i^t-W^t/n)} + \sum_{i=1}^{n} e^{-\gamma(x_i^t-W^t/n)}. \]

\( \Gamma^t \) is divided into **Overload potential** and **Underload potential**.

- For the \((1 + \beta)\) process, \(\gamma = \Theta(\beta)\).
- [PTW15] show that \( \mathbb{E}[\Gamma^{t+1} | \mathcal{F}^t] \leq \Gamma^t \cdot (1 - \frac{c_1}{n}) + c_2 \).
- By induction, this implies \( \mathbb{E}[\Gamma^t] \leq cn \) for any \( t \geq 0 \).
The hyperbolic cosine potential function

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

\[
\Gamma^t := \Gamma^t(\gamma) := \sum_{i=1}^{n} e^{\gamma(x_i^t - W^t/n)} + \sum_{i=1}^{n} e^{-\gamma(x_i^t - W^t/n)}. \]

- Overload potential
- Underload potential

- For the \((1 + \beta)\) process, \(\gamma = \Theta(\beta)\).
- [PTW15] show that \(\mathbb{E}\left[\Gamma^{t+1} \mid \mathcal{F}^t\right] \leq \Gamma^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2\).
- By induction, this implies \(\mathbb{E}\left[\Gamma^t\right] \leq cn\) for any \(t \geq 0\).
- By Markov's inequality, we get \(\mathbb{P}\left[\Gamma^m \leq cn^3\right] \geq 1 - n^{-2}\) which implies

\[
\mathbb{P}\left[\text{Gap}(m) \leq \frac{1}{\gamma} (3 \cdot \log n + \log c)\right] \geq 1 - n^{-2}.
\]
**Mean-Thinning: Why the analysis is tricky**

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

$$p_{\text{Mean-Thinning}}(x^t) = \left(\frac{1}{n} - \frac{1}{n^2}, \cdots, \frac{1}{n} - \frac{1}{n^2}, \frac{2}{n} - \frac{1}{n^2}\right).$$

  
  
  With this worst-case probability vector, we can only obtain w.h.p. a gap of $O(n \log n)$ using $\Gamma^t$ with $\gamma = \Theta(1/n)$. But what happens for $\Gamma^t$ with constant $\gamma$?
Mean-Thinning: Why the analysis is tricky

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

$$
p_{\text{Mean-Thinning}}(x^t) = \left( \frac{1}{n} - \frac{1}{n^2}, \ldots, \frac{1}{n} - \frac{1}{n^2}, 2 - \frac{1}{n^2} \right)\text{.}$$

$(n-1)$ entries

- With this worst-case probability vector, we can only obtain w.h.p. a gap of $\mathcal{O}(n \log n)$ using $\Gamma^t$ with $\gamma = \Theta(1/n)$. 
**Mean-Thinning: Why the analysis is tricky**

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

  $$p_{\text{Mean-Thinning}}(x^t) = \left(\frac{1}{n} - \frac{1}{n^2}, \cdots, \frac{1}{n} - \frac{1}{n^2}, \frac{2}{n} - \frac{1}{n^2}\right).$$

  $(n-1)$ entries

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

**But what happens for $\Gamma^t$ with constant $\gamma$?**
Mean-Thinning: Bad configuration

There is a very small bias to allocate away from overloaded bins.

The potential $\Gamma := \Gamma(\gamma)$ for constant $\gamma$ increases in expectation.
A closer look at $\Gamma^t$

An analysis similar to [PTW15] shows that

**(Good step)** If $\delta_t \in (\epsilon, 1 - \epsilon)$ for const $\epsilon > 0$, then $E[\Gamma^t_{t+1} | F^t, \{\delta_t \in (\epsilon, 1 - \epsilon)\}, \Gamma^t] \leq \Gamma^t \cdot (1 - \Theta(\gamma^n))$.

**(Bad step)** If $\delta_t / \in (\epsilon, 1 - \epsilon)$, then $E[\Gamma^t_{t+1} | F^t, \Gamma^t] \geq cn \leq \Gamma^t \cdot (1 + \Theta(\gamma^{2n}))$.

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?
A closer look at $\Gamma^t$

- An analysis similar to [PTW15] shows that

($\text{Good step}$) If $\delta_t \in (\epsilon, 1-\epsilon)$ for const $\epsilon > 0$, then

$$E[\Gamma_{t+1} | F_t, \{\delta_t \in (\epsilon, 1-\epsilon)\}, \Gamma_t \geq cn] \leq \Gamma_t \cdot (1 - \Theta(\gamma_n)).$$

($\text{Bad step}$) If $\delta_t \not\in (\epsilon, 1-\epsilon)$, then

$$E[\Gamma_{t+1} | F_t, \Gamma_t \geq cn] \leq \Gamma_t \cdot (1 + \Theta(\gamma_n^2)).$$

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?
A closer look at $\Gamma^t$

- An analysis similar to [PTW15] shows that
  - **(Good step)** If $\delta^t \in (\epsilon, 1 - \epsilon)$ for const $\epsilon > 0$, then
    \[
    \mathbf{E}[\Gamma^{t+1} | \tilde{\delta}^t, \{\delta^t \in (\epsilon, 1 - \epsilon)\}, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\gamma}{n}\right)\right).
    \]
A closer look at $\Gamma^t$

- An analysis similar to [PTW15] shows that
  - (Good step) If $\delta^t \in (\epsilon, 1 - \epsilon)$ for const $\epsilon > 0$, then
    \[
    E[\Gamma^{t+1} \mid \tilde{\delta}^t, \{\delta^t \in (\epsilon, 1 - \epsilon)\}, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\gamma}{n}\right)\right).
    \]
  - (Bad step) If $\delta^t \notin (\epsilon, 1 - \epsilon)$, then
    \[
    E[\Gamma^{t+1} \mid \tilde{\delta}^t, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 + \Theta\left(\frac{\gamma^2}{n}\right)\right).
    \]

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?
A closer look at $\Gamma^t$

- An analysis similar to [PTW15] shows that
  - (Good step) If $\delta^t \in (\epsilon, 1 - \epsilon)$ for const $\epsilon > 0$, then
    \[
    \mathbb{E}[\Gamma^{t+1} | \tilde{\delta}^t, \{\delta^t \in (\epsilon, 1 - \epsilon)\}, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\gamma}{n}\right)\right).
    \]
  - (Bad step) If $\delta^t \notin (\epsilon, 1 - \epsilon)$, then
    \[
    \mathbb{E}[\Gamma^{t+1} | \tilde{\delta}^t, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 + \Theta\left(\frac{\gamma^2}{n}\right)\right).
    \]

- A properly adjusted potential function drops in expectation for any interval with constant fraction of good steps.
A closer look at $\Gamma^t$

- An analysis similar to [PTW15] shows that
  - (Good step) If $\delta^t \in (\epsilon, 1 - \epsilon)$ for const $\epsilon > 0$, then
    \[
    E[\Gamma^{t+1} \mid \tilde{\delta}^t, \{\delta^t \in (\epsilon, 1 - \epsilon)\}, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 - \Theta\left(\frac{\gamma}{n}\right)\right).
    \]
  - (Bad step) If $\delta^t \notin (\epsilon, 1 - \epsilon)$, then
    \[
    E[\Gamma^{t+1} \mid \tilde{\delta}^t, \Gamma^t \geq cn] \leq \Gamma^t \cdot \left(1 + \Theta\left(\frac{\gamma^2}{n}\right)\right).
    \]

- 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

Consider the absolute value potential,
\[ \Delta_t = \sum_{i=1}^{n} |x_{ti} - W_{tn}|. \]

If \( \Delta_t = O(n) \), then \( \delta_t \in (\epsilon, 1 - \epsilon) \) w.h.p. for a constant fraction of the next \( \Theta(n) \) steps.

Consider the quadratic potential,
\[ \Upsilon_t = \sum_{i=1}^{n} (x_{ti} - W_{tn})^2. \]

We prove that
\[ E[\Upsilon_{t+k+1} \mid F_t] \leq \Upsilon_t - \kappa_1 n \cdot \Delta_t + \kappa_2 \cdot (k+1). \]

By induction we get,
\[ E[\Upsilon_{t+k+1} \mid F_t] \leq \Upsilon_t - \kappa_1 n \cdot t + \kappa_2 \cdot (k+1). \]
Mean quantile stabilisation

Consider the absolute value potential,

$$\Delta^t := \sum_{i=1}^{n} |x^t_i - \frac{W^t}{n}|.$$
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**,  
  \[ \Delta^t := \sum_{i=1}^{n} \left| x^t_i - \frac{W^t}{n} \right| . \]

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

- Consider the **quadratic potential**,  
  \[ \Upsilon^t := \sum_{i=1}^{n} (x^t_i - \frac{W^t}{n})^2 . \]
Mean quantile stabilisation

Consider the **absolute value potential**,

\[ \Delta^t := \sum_{i=1}^{n} \left| x^t_i - \frac{W^t}{n} \right| . \]

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

Consider the **quadratic potential**,

\[ \Upsilon^t := \sum_{i=1}^{n} \left( x^t_i - \frac{W^t}{n} \right)^2 . \]

We prove that

\[ \mathbb{E} \left[ \Upsilon^{t+1} \mid \mathcal{F}^t \right] \leq \Upsilon^t - (p_-^t \cdot w_- - p_+^t \cdot w_+) \cdot \Delta^t + 4 \cdot (w_-)^2 . \]
Mean quantile stabilisation

- Consider the absolute value potential,

\[ \Delta^t := \sum_{i=1}^{n} \left| x^t_i - \frac{W^t}{n} \right| . \]

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

- Consider the quadratic potential,

\[ \Upsilon^t := \sum_{i=1}^{n} \left( x^t_i - \frac{W^t}{n} \right)^2 . \]

- We prove that

\[ \mathbb{E} \left[ \Upsilon^{t+1} \mid \mathcal{F}^t \right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \Delta^t + \kappa_2. \]
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 
\[ \mathbb{E} \left[ \Upsilon^{t+1} \mid \hat{\delta}^t \right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \Delta^t + \kappa_2. \]

By *induction* we get,
\[ \mathbb{E} \left[ \Upsilon^{t+k+1} \mid \hat{\delta}^{t} \right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \sum_{r=t}^{t+k} \mathbb{E} \left[ \Delta^r \mid \hat{\delta}^t \right] + \kappa_2 \cdot (k + 1). \]
Mean quantile stabilisation

- Consider the **absolute value potential**, 
  \[
  \Delta^t := \sum_{i=1}^{n} \left| x^t_i - \frac{W^t}{n} \right|.
  \]

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

- Consider the **quadratic potential**, 
  \[
  \Upsilon^t := \sum_{i=1}^{n} \left( x^t_i - \frac{W^t}{n} \right)^2.
  \]

- We prove that 
  \[
  \mathbb{E} \left[ \Upsilon^{t+1} \mid \mathcal{F}^t \right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \Delta^t + \kappa_2.
  \]

- By induction we get, 
  \[
  \mathbb{E} \left[ \Upsilon^{t+k+1} \mid \mathcal{F}^t \right] \leq \Upsilon^t - \frac{\kappa_1}{n} \cdot \sum_{r=t}^{t+k} \mathbb{E} \left[ \Delta^r \mid \mathcal{F}^t \right] + \kappa_2 \cdot (k + 1).
  \]

For \( k = \Theta(\Upsilon^t) \), for constant fraction of steps \( r \in [t, t + k] \), \( \mathbb{E} \left[ \Delta^r \mid \mathcal{F}^t \right] = O(n) \).
Recovery from a bad configuration \((n = 1000)\)
Recovery from a bad configuration \((n = 1000)\)

As long as \(\Delta^t = \Omega(n)\), \(\Upsilon^t\) drops in expectation.
Recovery from a bad configuration \((n = 1000)\)

As long as \(\Delta^t = \Omega(n)\), \(\Upsilon^t\) drops in expectation.  
As \(\Delta^t\) becomes smaller, \(\delta^t\) improves and \(\Gamma^t\) drops in expectation.
Recovery from a bad configuration \((n = 1000)\)

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

As \(\Delta^t\) becomes smaller, \(\delta^t\) improves and \(\Gamma^t\) drops in expectation.
Completing the analysis

\[ \Gamma^t \]

\[ \exp(\mathcal{O}(n \log n)) \]

Base Case

\[ m - n^3 \log^4 n \]

\[ m \]
Completing the analysis

\[ \Gamma_t \]

\[ \exp(\Theta(n \log n)) \]

\[ c n \]

\[ m - n^3 \log^4 n \]

\[ s_0 \]

\[ m \]

\[ m + n \log n \]
Completing the analysis

\[ \Gamma^t \]

\[ \exp(O(n \log n)) \]

\[ cn \]

\[ m - n^3 \log^4 n \]

\[ n^3 \log^3 n \]

\[ n^3 \log^3 n \]

\[ n^3 \log^3 n \]

\[ \text{Recovery phase} \]

\[ \text{Base Case} \]

\[ \text{Sample efficiency: } 2^{-\epsilon} \text{ for Mean-Thinning and } 1 - \epsilon \text{ for Twinning.} \]

\[ \text{Lower bound of } \Omega(\log n) \text{ for Mean-Thinning and Twinning.} \]
Completing the analysis

\[ \Gamma^t \]

- **Recovery phase**
- **Stabilization phase**

Base Case

\[ \exp(O(n \log n)) \]

\[ 2cn \]

\[ cn \]

\[ m - n^3 \log^4 n \]

\[ n^3 \log^3 n \]

\[ n^3 \log^3 n \]

\[ \ldots \]

\[ t \]

\[ s_0 \]

\[ r_1 \]

\[ s_1 \]

\[ r_2 \]

\[ m \]

\[ m + n \log n \]

Other applications of quantile stabilisation:

- **Sample efficiency:**
  - \[ 2 - \epsilon \] for Mean-Thinning
  - \[ 1 - \epsilon \] for Twinning.

- **Lower bound of** \[ \Omega(\log n) \] for Mean-Thinning and 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.
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

\[ W^{t/n} \]

\[ W^{t/n} \]

\[ i \]
### Packing: Definition

**Packing Process:**

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

$$x_{i}^{t+1} = \begin{cases} 
\left\lceil \frac{W^t}{n} \right\rceil + 1 & \text{if } x_{i}^{t} < \frac{W^t}{n}, \\
 x_{i}^{t} + 1 & \text{if } x_{i}^{t} \geq \frac{W^t}{n}.
\end{cases}$$
We analyze another general framework that includes Packing and Caching [MPS02].

We prove an $\mathcal{O}(\log n)$ gap for these processes.
Conclusion

Summary of results:

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

■ Proved a matching lower bound for Mean-Thinning and Twinning.

■ Proved $\text{Gap}(m) = O(\log n)$ 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.
Conclusion

Summary of results:

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

Summary of results:

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

Future work:

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

Summary of results:

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

Future work:

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

Summary of results:

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

Future work:

- Extend the framework to \textit{non-constant} probability and weight biases.
Conclusion

Summary of results:

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

Future work:

- Extend the framework to \textit{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 \textsc{Mean-Thinning} and \textsc{Twinning}.
- Proved a matching lower bound for \textsc{Mean-Thinning} and \textsc{Twinning}.
- Proved $\text{Gap}(m) = \mathcal{O}(\log n)$ for a set of processes including \textsc{Packing} and \textsc{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 \textsc{Mean-Thinning} with outdated information and noise.
Questions?

Visualisations: dimitrioslos.com/soda22
Questions?

Visualisations: dimitrioslos.com/soda22
Appendix
### Appendix A: Table of results

<table>
<thead>
<tr>
<th>Process</th>
<th>Lightly Loaded Case ( m = \mathcal{O}(n) )</th>
<th>Heavily Loaded Case ( m = \omega(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower Bound</td>
<td>Upper Bound</td>
</tr>
<tr>
<td>((1 + \beta), \text{const } \beta \in (0, 1))</td>
<td>(\frac{\log n}{\log \log n})</td>
<td>[PTW15]</td>
</tr>
<tr>
<td>Caching</td>
<td>\log \log n</td>
<td>[MPS02]</td>
</tr>
<tr>
<td>Packing</td>
<td>(\frac{\log n}{\log \log n})</td>
<td>(\frac{\log n}{\log \log n})</td>
</tr>
<tr>
<td>Twinning</td>
<td>(\frac{\log n}{\log \log n})</td>
<td>\log n</td>
</tr>
<tr>
<td>Mean-Thinning</td>
<td>(\frac{\log n}{\log \log n})</td>
<td>\log n</td>
</tr>
<tr>
<td>Relative-Threshold((f(n)))</td>
<td>(\sqrt{\frac{\log n}{\log \log n}})</td>
<td>[FL20]</td>
</tr>
<tr>
<td>Adaptive-Two-Thinning</td>
<td>(\sqrt{\frac{\log n}{\log \log n}})</td>
<td>[FL20]</td>
</tr>
</tbody>
</table>

**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**).
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

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.

<table>
<thead>
<tr>
<th>$n$</th>
<th>Mean-Thinning</th>
<th>Twinning</th>
<th>Packing</th>
<th>Caching</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^5$</td>
<td>8 : 3%</td>
<td>14 : 2%</td>
<td>12 : 2%</td>
<td>3 : 100%</td>
</tr>
<tr>
<td></td>
<td>9 : 32%</td>
<td>15 : 5%</td>
<td>13 : 16%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 : 38%</td>
<td>16 : 25%</td>
<td>14 : 20%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11 : 15%</td>
<td>17 : 28%</td>
<td>15 : 28%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 : 6%</td>
<td>18 : 17%</td>
<td>16 : 23%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>13 : 3%</td>
<td>19 : 10%</td>
<td>17 : 5%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>14 : 3%</td>
<td>20 : 8%</td>
<td>18 : 3%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>21 : 1%</td>
<td>19 : 1%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>22 : 1%</td>
<td>20 : 2%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>23 : 3%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix D1: Recovery from a bad configuration
Appendix D2: Recovery from a bad configuration

Potential functions
- Quadratic potential
- Absolute potential
- Exponential potential

Number of balls $m$

Graph showing the behavior of different potential functions over varying numbers of balls.
Bibliography I


Bibliography II


