Balanced Allocations: Relaxing Two-Choice

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

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

Balanced allocations setting

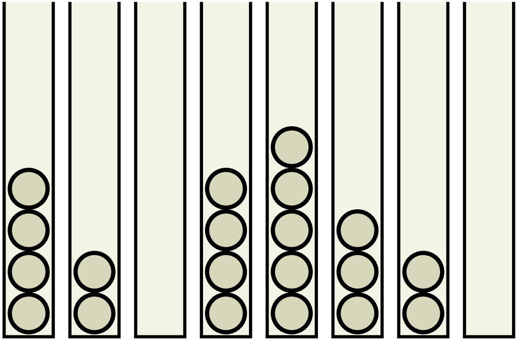
Allocate m tasks (balls) sequentially into n machines (bins).

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.

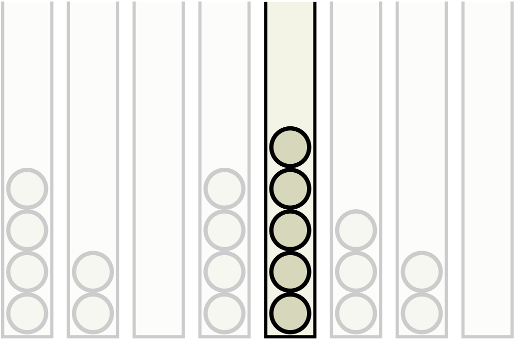
Balanced allocations setting

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.



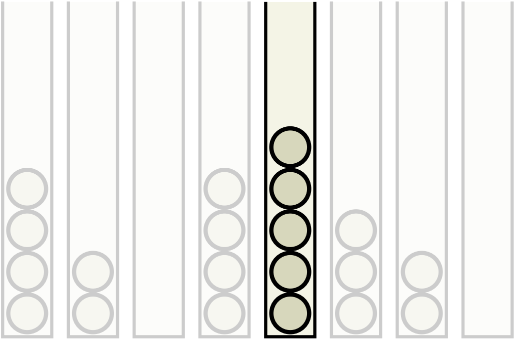
Balanced allocations setting

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



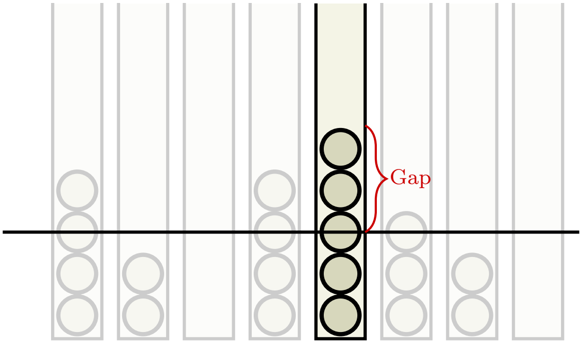
Balanced allocations setting

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



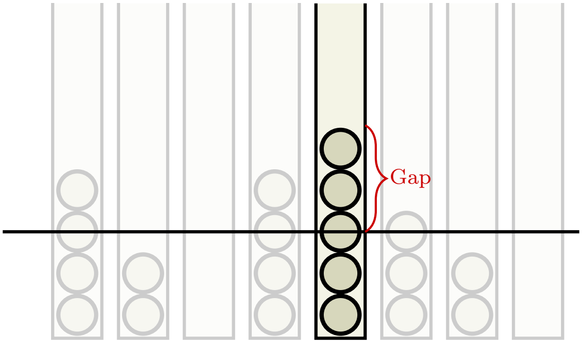
Balanced allocations setting

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



Balanced allocations setting

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



- Applications in hashing, load balancing and routing.

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. $\mathrm{Gap}(n) = \Theta\left(\frac{\log n}{\log \log n}\right)$ [Gon81].

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].

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

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]).

<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

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

<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]).

<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. $\text{Gap}(n) = \log_2 \log n + \Theta(1)$ [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

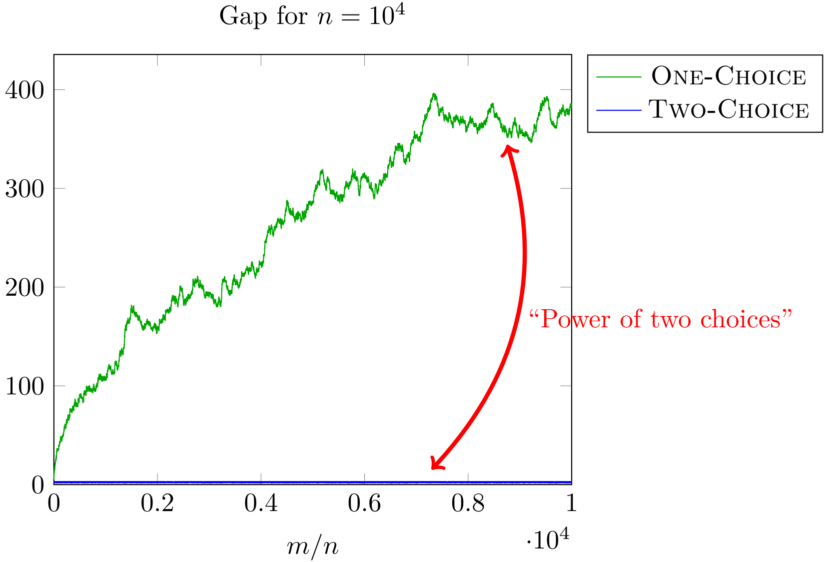
> ONE-CHOICE PROCESS:
> **Iteration:** For each $t \geq 0$, sample **one** bin uniformly at random (u.a.r.) and place the ball there.

- 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].

One-Choice and Two-Choice processes



Relaxing with incomplete information

<u>Mean-Thinning Process:</u>

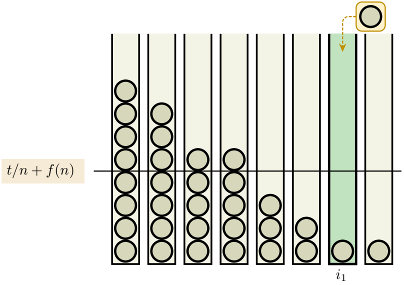
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

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

Relaxing with incomplete information

<u>MEAN-THINNING PROCESS:</u>
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

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

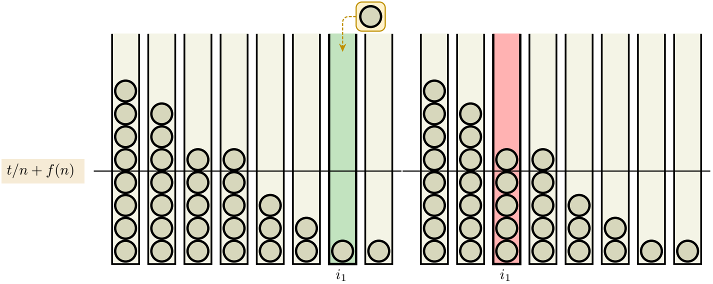


Relaxing with incomplete information

<u>MEAN-THINNING PROCESS:</u>

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

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

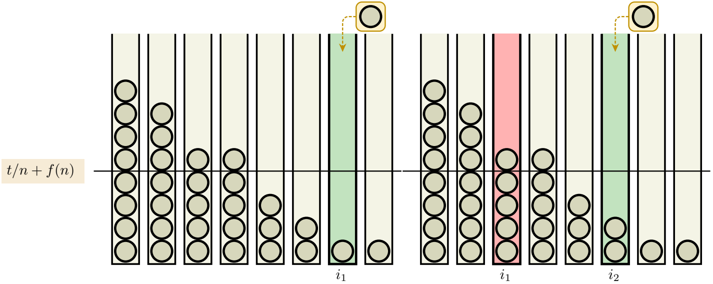


Relaxing with incomplete information

<u>MEAN-THINNING PROCESS:</u>

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

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

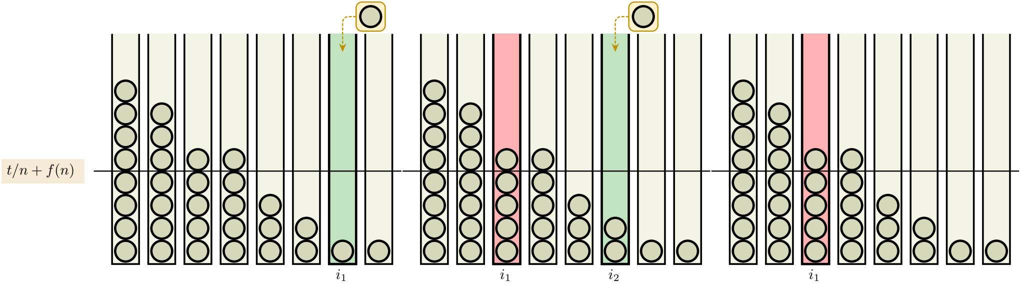


Relaxing with incomplete information

Mean-Thinning Process:

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

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

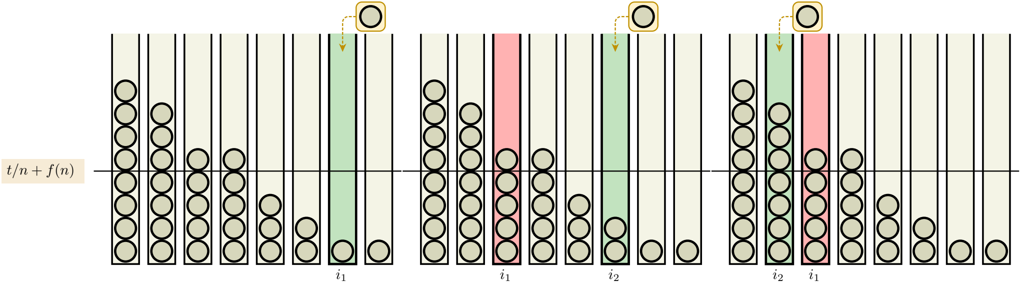


Relaxing with incomplete information

<u>MEAN-THINNING PROCESS:</u>

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

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

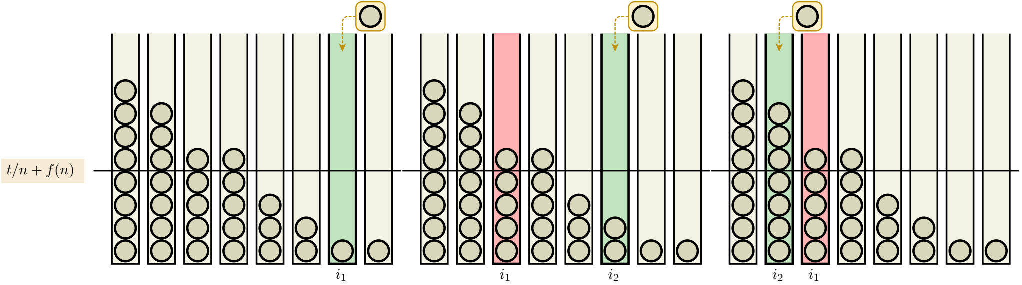


Relaxing with incomplete information

<u>MEAN-THINNING PROCESS:</u>

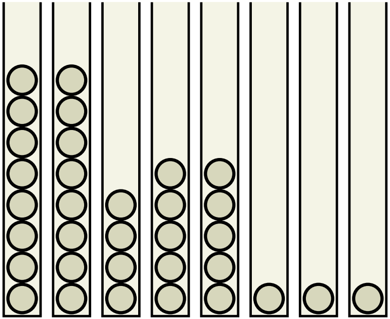
Iteration: For $t \geq 0$, sample two uniform bins i_1 and i_2 independently, and update:

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



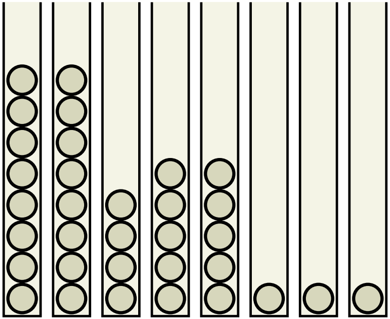
- Achieves $\text{Gap}(m) = \mathcal{O}(\log n)$ and uses $2 - \epsilon$ samples.

Relaxing with outdated information



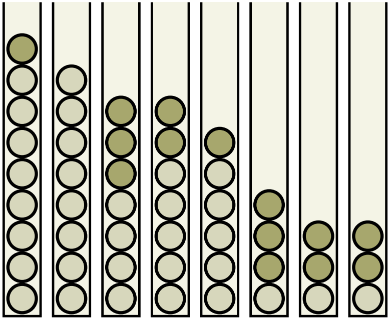
Relaxing with outdated information

- Allocate balls in batches of size b.



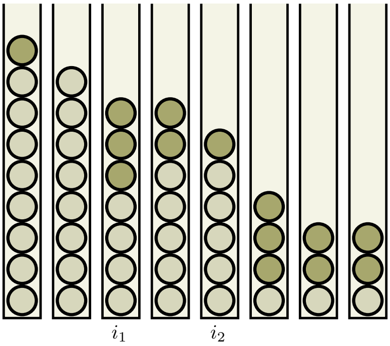
Relaxing with outdated information

- Allocate balls in batches of size b.



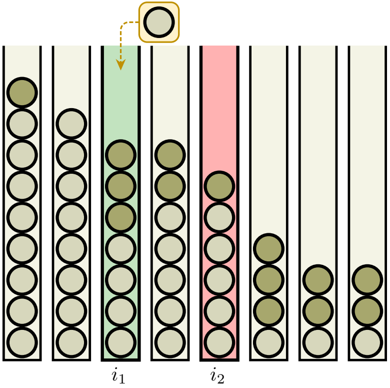
Relaxing with outdated information

- Allocate balls in batches of size b.



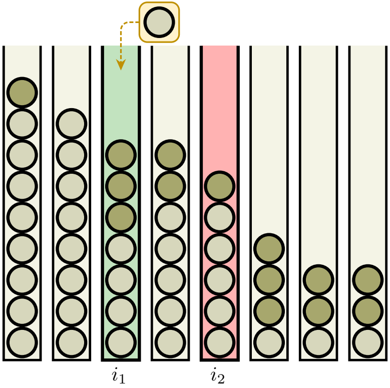
Relaxing with outdated information

- Allocate balls in batches of size b.



Relaxing with outdated information

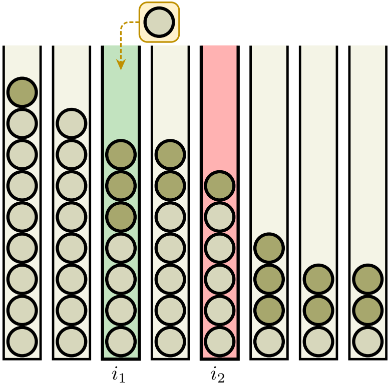
- Allocate balls in batches of size b.



- For $b = n$, improved $\text{Gap}(m) = \mathcal{O}(\log n)$ to $\text{Gap}(m) = \Theta(\log n / \log\log n)$.

Relaxing with outdated information

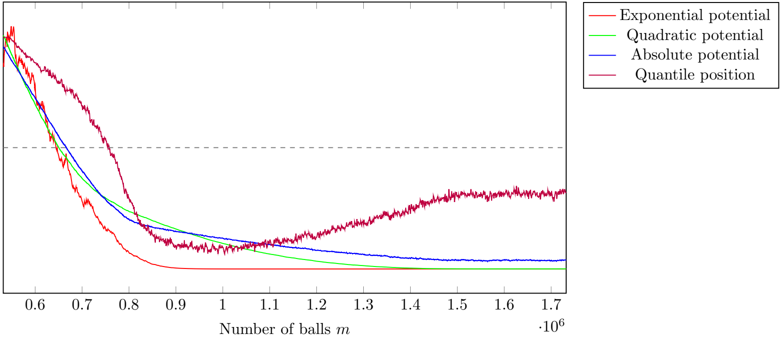
- Allocate balls in batches of size b.

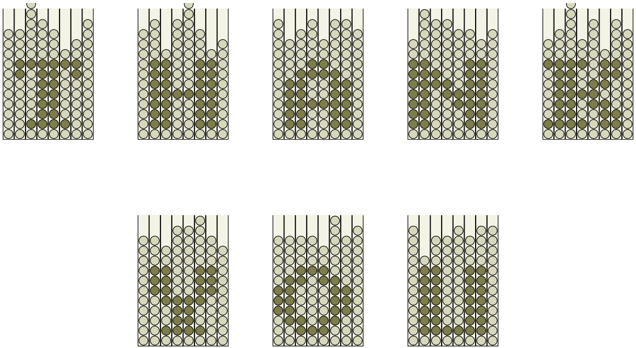


- For $b = n$, improved $\text{Gap}(m) = \mathcal{O}(\log n)$ to $\text{Gap}(m) = \Theta(\log n / \log \log n)$.
- For $b \geq n \log n$, $\text{Gap}(m) = \Theta(b/n)$.

Our techniques

- Interplay between (i) **linear**, (ii) **quadratic** and (iii) **exponential** potentials.





Visualisations: `tinyurl.com/lss21-visualisations`

Bibliography I

- Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, *Balanced allocations*, SIAM J. Comput. **29** (1999), no. 1, 180–200. MR 1710347
- P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, *Balanced allocations: the heavily loaded case*, SIAM J. Comput. **35** (2006), no. 6, 1350–1385. MR 2217150
- G. H. Gonnet, *Expected length of the longest probe sequence in hash code searching*, J. Assoc. Comput. Mach. **28** (1981), no. 2, 289–304. MR 612082
- R. M. Karp, M. Luby, and F. Meyer auf der Heide, *Efficient PRAM simulation on a distributed memory machine*, Algorithmica **16** (1996), no. 4-5, 517–542. MR 1407587
- M. Raab and A. Steger, *"Balls into bins"—a simple and tight analysis*, Proceedings of 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM'98), vol. 1518, Springer, 1998, pp. 159–170. MR 1729169