Balanced Allocations with the Choice of Noise

Dimitrios Los[1], Thomas Sauerwald[1]

[1]University of Cambridge, UK

Balanced allocations: Background

Balanced allocations setting

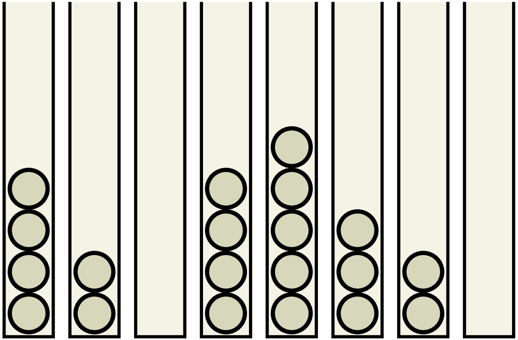
Allocate m tasks (balls) sequentially into n machines (bins).

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.

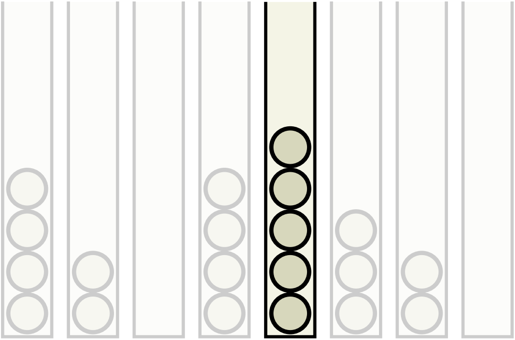
Balanced allocations setting

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.

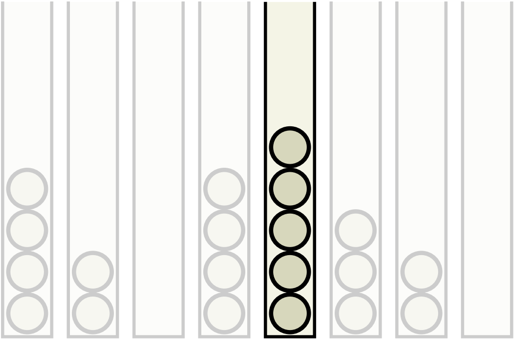


Balanced allocations setting

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

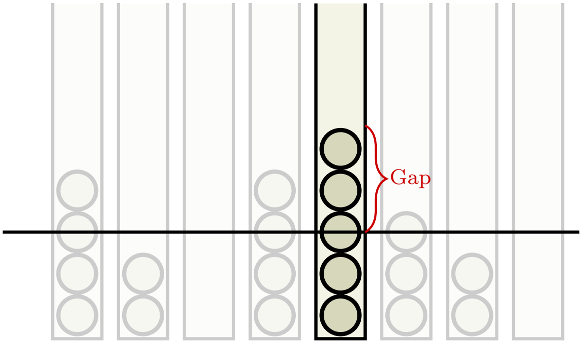
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.

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

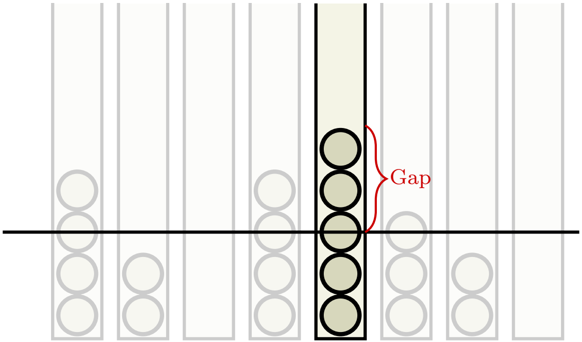


Balanced allocations setting

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 [PR01], load balancing [Wie16] and routing [GKK88].

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.

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

Noisy processes

Motivation

Motivation

1. What if the load information of a bin is **outdated**?

Motivation

1. What if the load information of a bin is **outdated**?

2. What if an **adversary** can **perturb** the load of a bin by some additive amount?

Motivation

1. What if the load information of a bin is **outdated**?

2. What if an **adversary** can **perturb** the load of a bin by some additive amount?

3. What about **random** (additive) perturbations?

Motivation

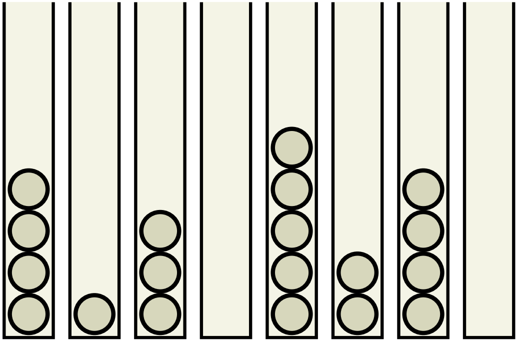
1. What if the load information of a bin is **outdated**?

2. What if an **adversary** can **perturb** the load of a bin by some additive amount?

3. What about **random** (additive) perturbations?

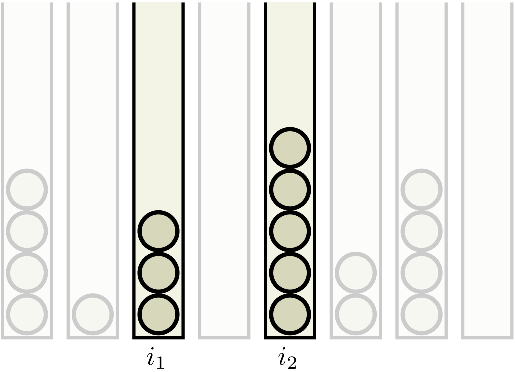
The Adversarial Comparison (g-ADV) setting

The Adversarial Comparison (g-ADV) setting



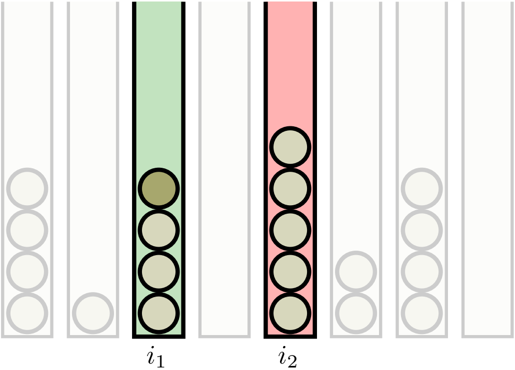
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2



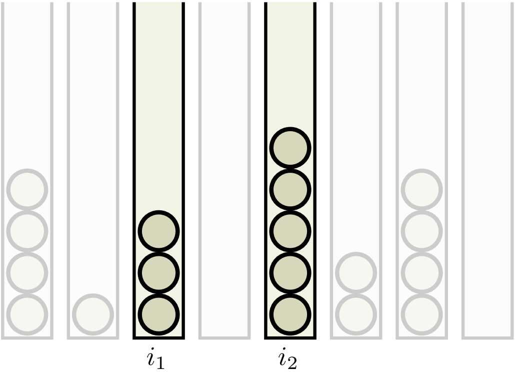
The Adversarial Comparison (g-ADV) setting

- In TWO-CHOICE, we sample two bins i_1 and i_2 and allocate to the least loaded bin.



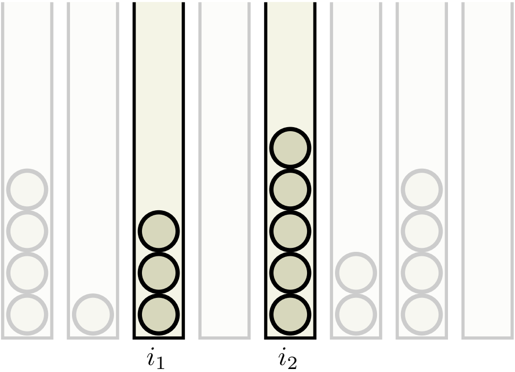
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:



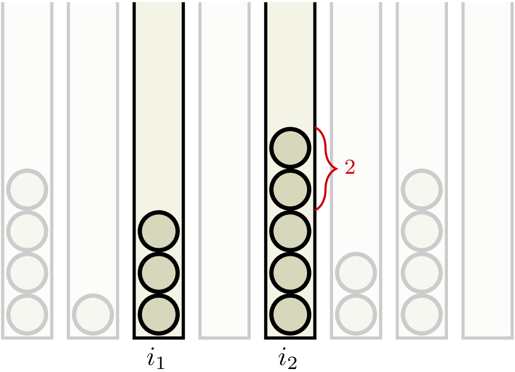
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.



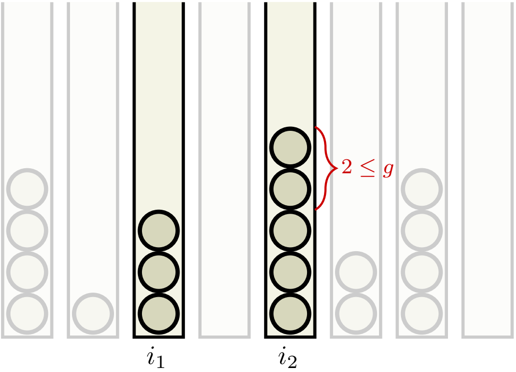
The Adversarial Comparison (g-ADV) setting

- In TWO-CHOICE, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-ADV process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.



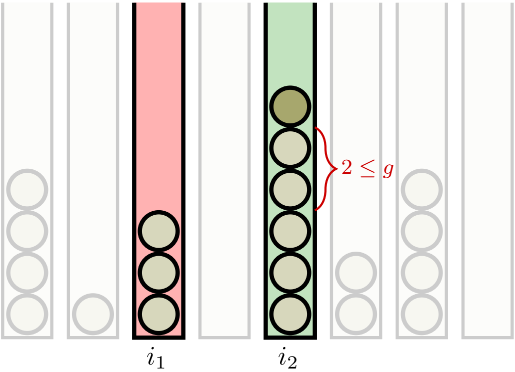
The Adversarial Comparison (g-ADV) setting

- In TWO-CHOICE, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-ADV process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.



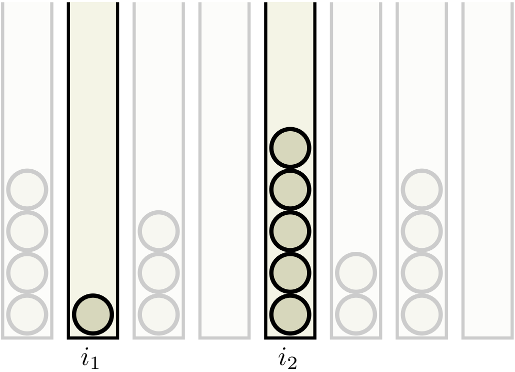
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.



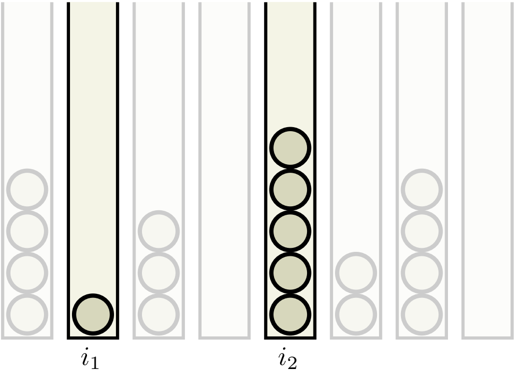
The Adversarial Comparison (g-ADV) setting

- In TWO-CHOICE, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-ADV process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.



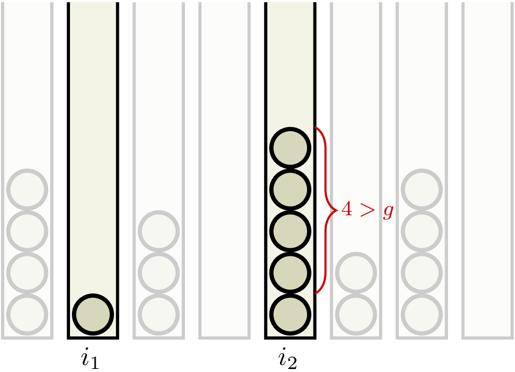
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:
  - If $|x^t_{i_1} - x^t_{i_2}| \leq g$, the adversary can allocate to **any** of the two bins.
  - Otherwise, allocate to the **lesser** loaded of the two.



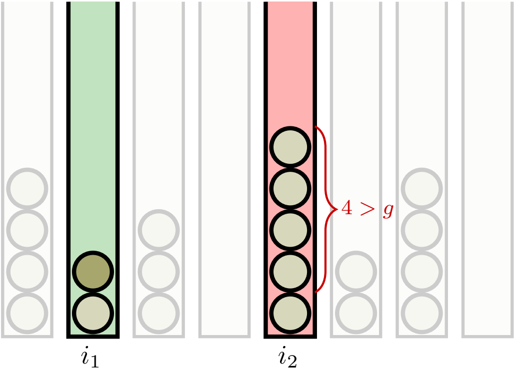
The Adversarial Comparison (g-ADV) setting

- In TWO-CHOICE, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-ADV process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.
  - Otherwise, allocate to the **lesser** loaded of the two.



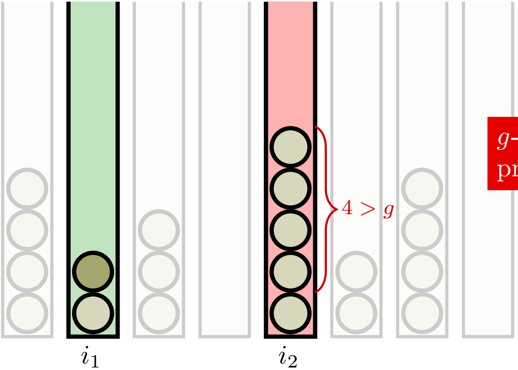
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.
  - Otherwise, allocate to the **lesser** loaded of the two.



The Adversarial Comparison (g-ADV) setting

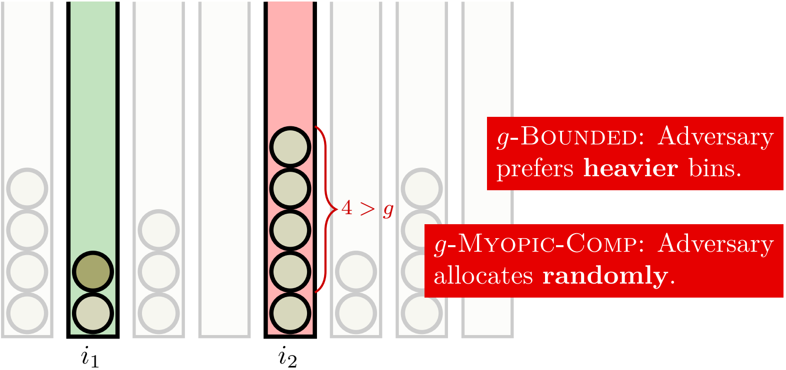
- In TWO-CHOICE, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-ADV process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.
  - Otherwise, allocate to the **lesser** loaded of the two.



g-BOUNDED: Adversary prefers **heavier** bins.

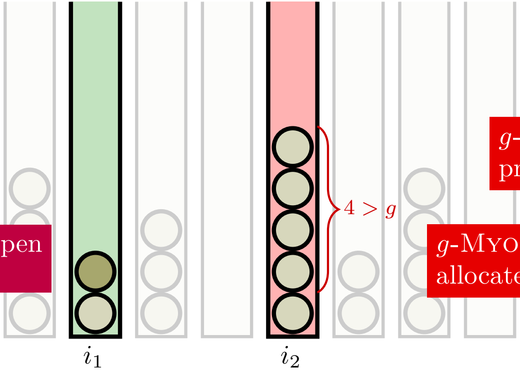
The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.
  - Otherwise, allocate to the **lesser** loaded of the two.



The Adversarial Comparison (g-Adv) setting

- In Two-Choice, we sample two bins i_1 and i_2 and allocate to the least loaded bin.
- In a g-Adv process (say for $g = 3$), again we sample two bins:
  - If $|x_{i_1}^t - x_{i_2}^t| \leq g$, the adversary can allocate to **any** of the two bins.
  - Otherwise, allocate to the **lesser** loaded of the two.



g-Bounded: Open in Visualiser.

g-Myopic-Comp: Open in Visualiser.

g-Bounded: Adversary prefers **heavier** bins.

g-Myopic-Comp: Adversary allocates **randomly**.

Result for the g-Adv setting (Main result of our work)

Result for the g-ADV setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-BOUNDED process.

Result for the g-ADV setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-BOUNDED process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.

Result for the g-Adv setting (Main result of our work)

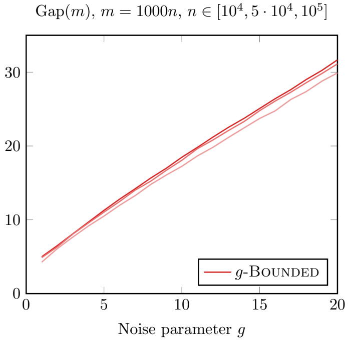
- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-Bounded process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-Adv process,

Result for the g-Adv setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-Bounded process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.

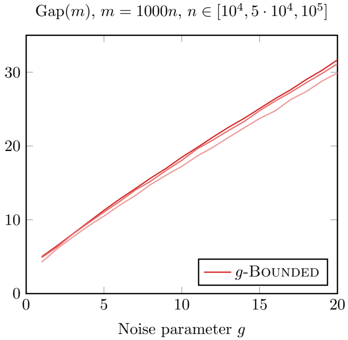
Result for the g-ADV setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK$^+$18] analyzed the g-BOUNDED process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-ADV process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.



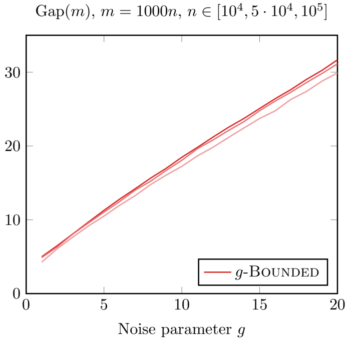
Result for the g-Adv setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-Bounded process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\frac{g}{\log g} \cdot \log \log n)$.



Result for the g-ADV setting (Main result of our work)

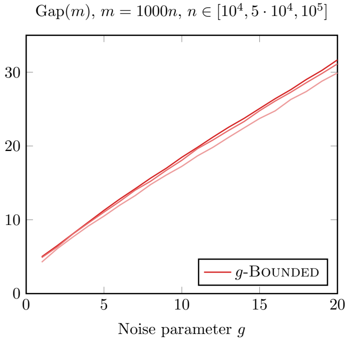
- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-BOUNDED process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-ADV process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$.



For $g = \mathcal{O}(1)$:
$\mathrm{Gap}(m) = \mathcal{O}(\log\log n)$.

Result for the g-ADV setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-BOUNDED process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-ADV process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\frac{g}{\log g} \cdot \log\log n)$.

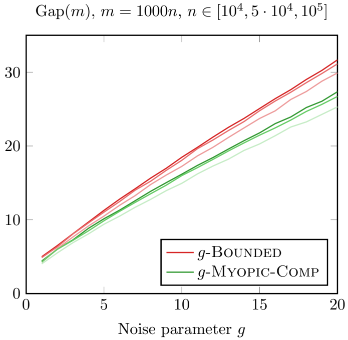


For $g = \mathcal{O}(1)$: $\mathrm{Gap}(m) = \mathcal{O}(\log\log n)$.

For $g = \Omega(\mathrm{polylog}(n))$: $\mathrm{Gap}(m) = \mathcal{O}(g)$.

Result for the g-ADV setting (Main result of our work)

- Alistarh, Brown, Kopinsky, Li and Nadiradze [ABK+18] analyzed the g-BOUNDED process.
- They proved that for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.
- We prove that for any g-ADV process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\frac{g}{\log g} \cdot \log \log n)$.
- For both cases, we prove a matching lower bound for g-MYOPIC-COMP.



For $g = \mathcal{O}(1)$:
$\mathrm{Gap}(m) = \mathcal{O}(\log \log n)$.

For $g = \Omega(\mathrm{polylog}(n))$:
$\mathrm{Gap}(m) = \mathcal{O}(g)$.

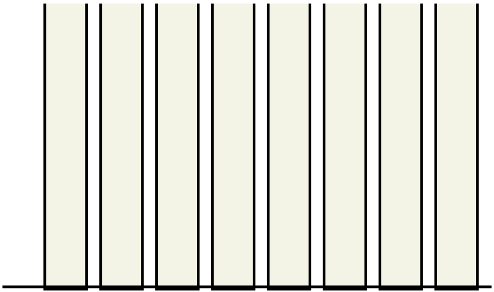
Two-Choice **with outdated information**

Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).

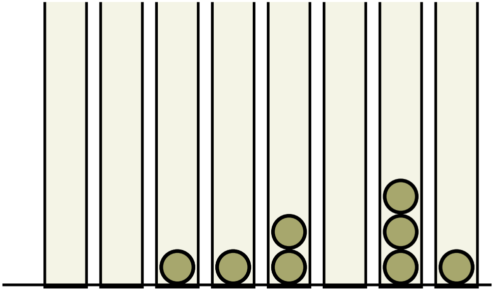
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



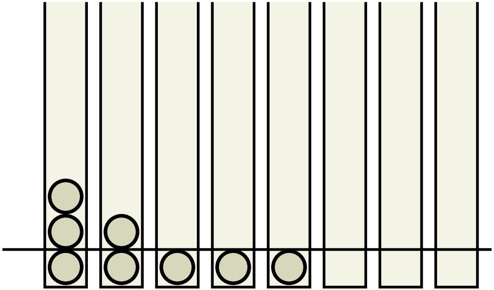
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



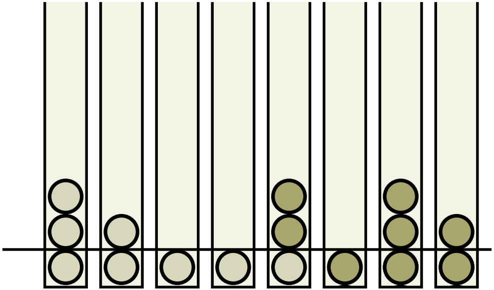
Two-Choice **with outdated information**

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



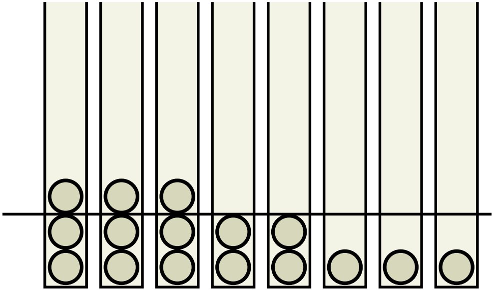
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



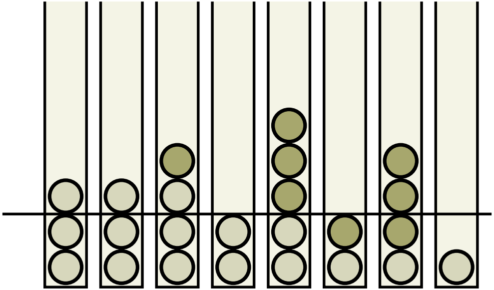
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



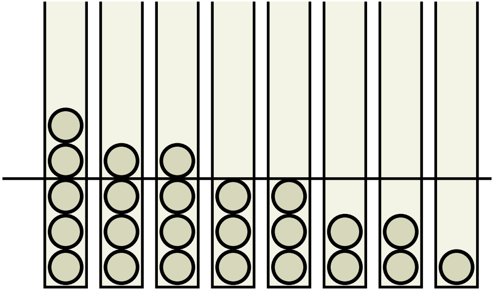
TWO-CHOICE with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied TWO-CHOICE where balls are allocated in *batches* of size b (b-BATCH).



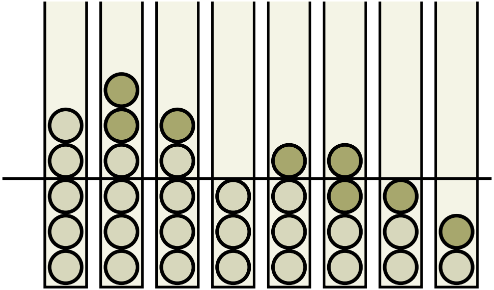
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



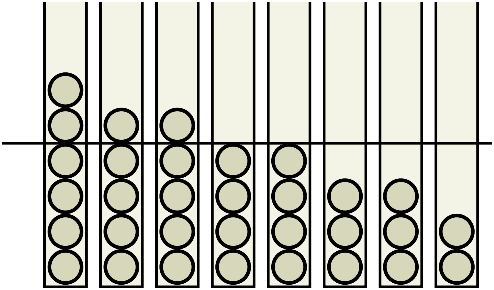
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



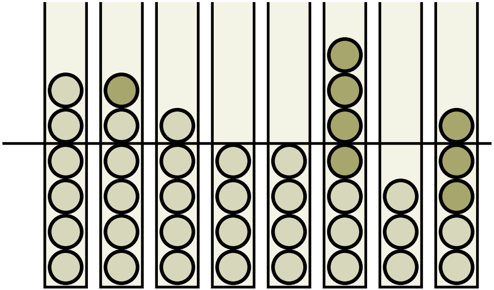
TWO-CHOICE with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied TWO-CHOICE where balls are allocated in *batches* of size b (b-BATCH).



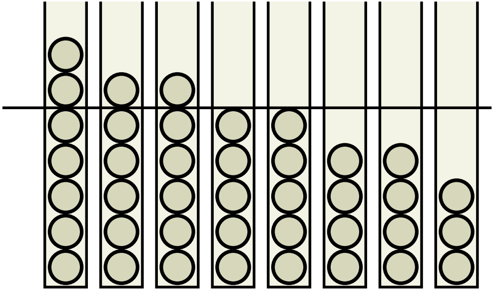
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



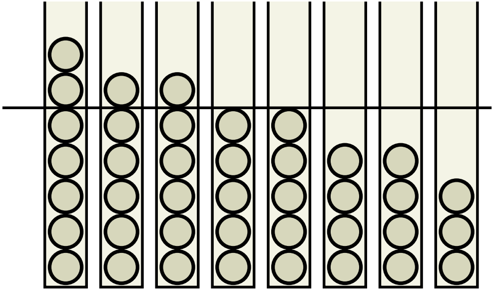
Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).



Two-Choice with outdated information

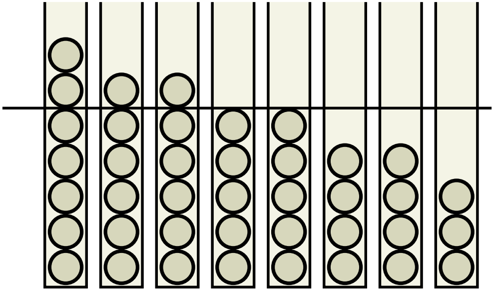
- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).
- For $b = n$, they showed that w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\log n)$.



Open in Visualiser.

Two-Choice with outdated information

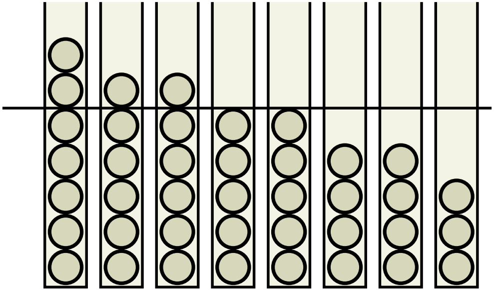
- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).
- For $b = n$, they showed that w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.
- For $b \in [n \log n, \text{poly}(n)]$, the authors [LS22a] showed that w.h.p. $\text{Gap}(m) = \Theta(b/n)$.



Open in Visualiser.

Two-Choice with outdated information

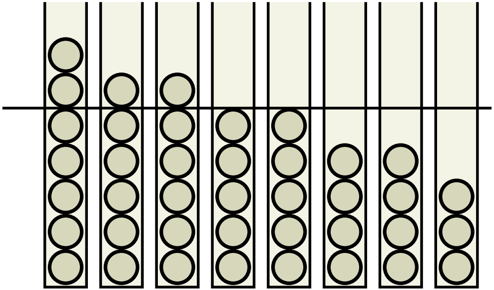
- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).
- For $b = n$, they showed that w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(\log n)$.
- For $b \in [n \log n, \mathrm{poly}(n)]$, the authors [LS22a] showed that w.h.p. $\mathrm{Gap}(m) = \Theta(b/n)$.
- For $b = n$, we show that w.h.p. $\mathrm{Gap}(m) = \Theta\left(\frac{\log n}{\log \log n}\right)$



Open in Visualiser.

Two-Choice with outdated information

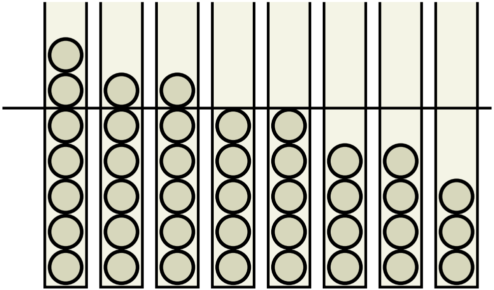
- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).
- For $b = n$, they showed that w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.
- For $b \in [n \log n, \text{poly}(n)]$, the authors [LS22a] showed that w.h.p. $\text{Gap}(m) = \Theta(b/n)$.
- For $b = n$, we show that w.h.p. $\text{Gap}(m) = \Theta\left(\frac{\log n}{\log \log n}\right)$, like One-Choice with n balls.



Open in Visualiser.

Two-Choice with outdated information

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batch).
- For $b = n$, they showed that w.h.p. $\text{Gap}(m) = \mathcal{O}(\log n)$.
- For $b \in [n \log n, \text{poly}(n)]$, the authors [LS22a] showed that w.h.p. $\text{Gap}(m) = \Theta(b/n)$.
- For $b = n$, we show that w.h.p. $\text{Gap}(m) = \Theta\left(\frac{\log n}{\log \log n}\right)$, like One-Choice with n balls.
- More generally, for $b \in \left[\frac{n}{\text{polylog}(n)}, n \log n\right]$ it follows One-Choice with b balls.



Open in Visualiser.

Two-Choice with outdated information: Reduction

Two-Choice with outdated information: Reduction

- For $b = n$, w.h.p. any bin can be selected at most $\mathcal{O}\left(\frac{\log n}{\log \log n}\right)$ times in a batch.

Two-Choice with outdated information: Reduction

- For $b = n$, w.h.p. any bin can be selected at most $\mathcal{O}\left(\frac{\log n}{\log \log n}\right)$ times in a batch.
- So, w.h.p. we can simulate b-Batch with a g-Adv process with $g = \Theta\left(\frac{\log n}{\log \log n}\right)$.

Two-Choice with outdated information: Reduction

- For $b = n$, w.h.p. any bin can be selected at most $\mathcal{O}\big(\frac{\log n}{\log \log n}\big)$ times in a batch.
- So, w.h.p. we can simulate b-Batch with a g-Adv process with $g = \Theta\big(\frac{\log n}{\log \log n}\big)$.
- Hence, w.h.p.

$$\mathrm{Gap}(m) = \mathcal{O}\Big(\frac{g}{\log g} \cdot \log \log n\Big).$$

Two-Choice with outdated information: Reduction

- For $b = n$, w.h.p. any bin can be selected at most $\mathcal{O}\big(\frac{\log n}{\log\log n}\big)$ times in a batch.
- So, w.h.p. we can simulate b-Batch with a g-Adv process with $g = \Theta\big(\frac{\log n}{\log\log n}\big)$.
- Hence, w.h.p.

$$\mathrm{Gap}(m) = \mathcal{O}\Big(\frac{g}{\log g} \cdot \log\log n\Big).$$

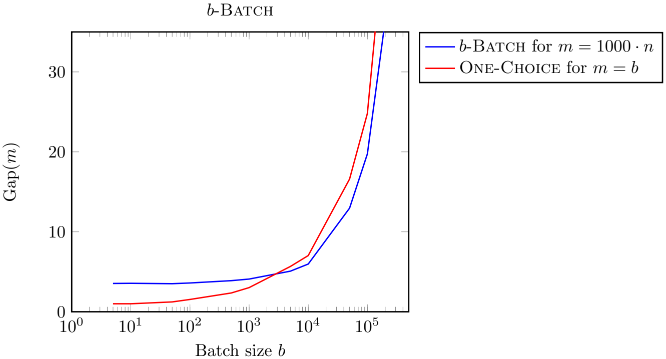
- For $b \in \big[\frac{n}{\mathrm{polylog}(n)}, n\log n\big]$, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.

Two-Choice with outdated information: Reduction

- For $b = n$, w.h.p. any bin can be selected at most $\mathcal{O}\left(\frac{\log n}{\log \log n}\right)$ times in a batch.
- So, w.h.p. we can simulate b-Batch with a g-Adv process with $g = \Theta\left(\frac{\log n}{\log \log n}\right)$.
- Hence, w.h.p.

$$\mathrm{Gap}(m) = \mathcal{O}\left(\frac{g}{\log g} \cdot \log \log n\right).$$

- For $b \in \left[\frac{n}{\mathrm{polylog}(n)}, n \log n\right]$, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.



TWO-CHOICE with outdated information: τ-DELAY

Two-Choice with outdated information: τ-Delay

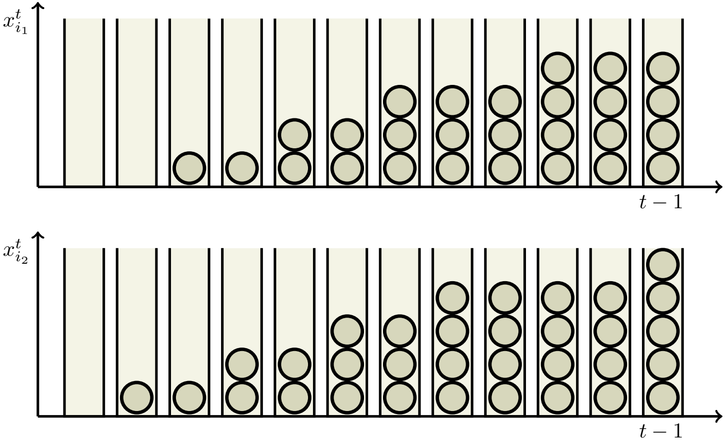
- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.

Two-Choice with outdated information: τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).

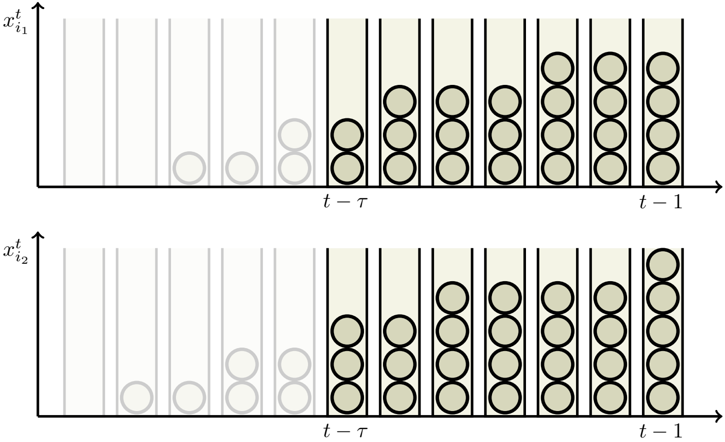
Two-Choice with outdated information: τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).



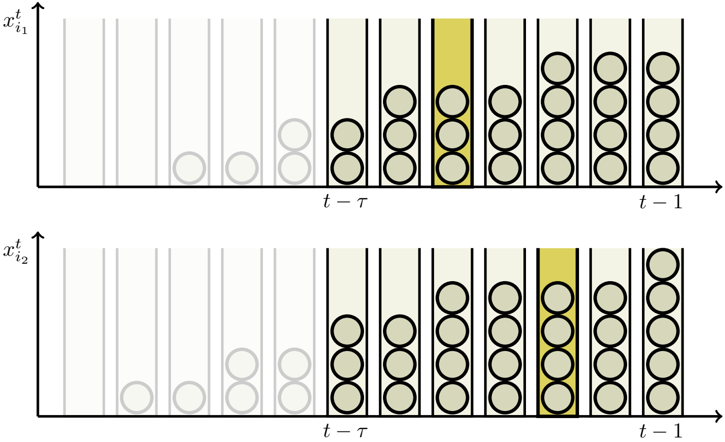
Two-Choice **with outdated information:** τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).



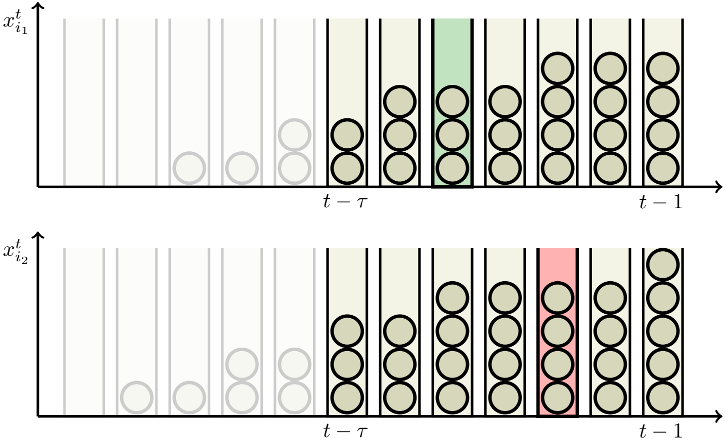
Two-Choice with outdated information: τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).



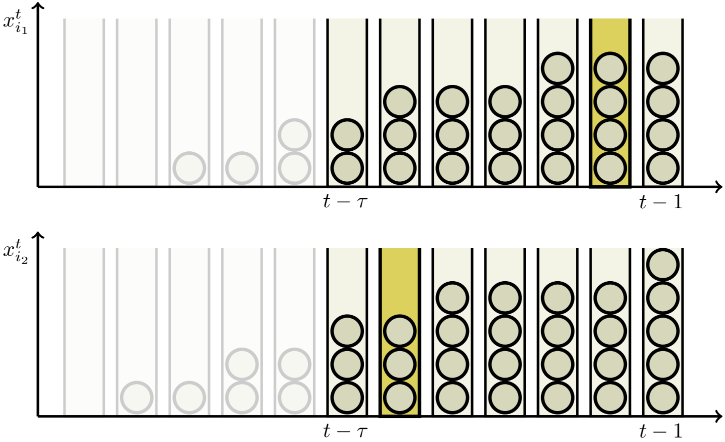
Two-Choice with outdated information: τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).



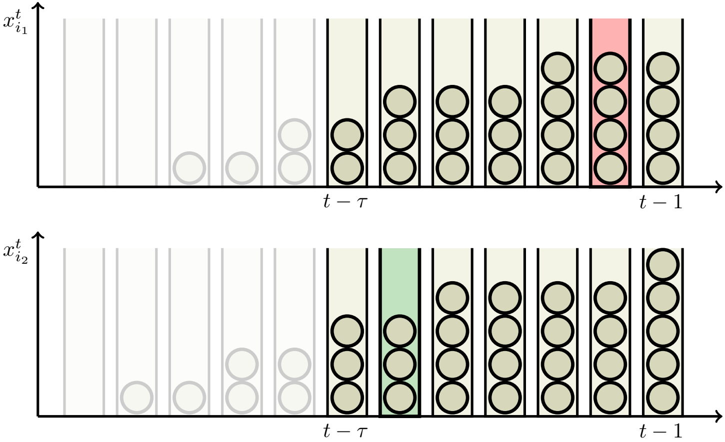
Two-Choice with outdated information: τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).



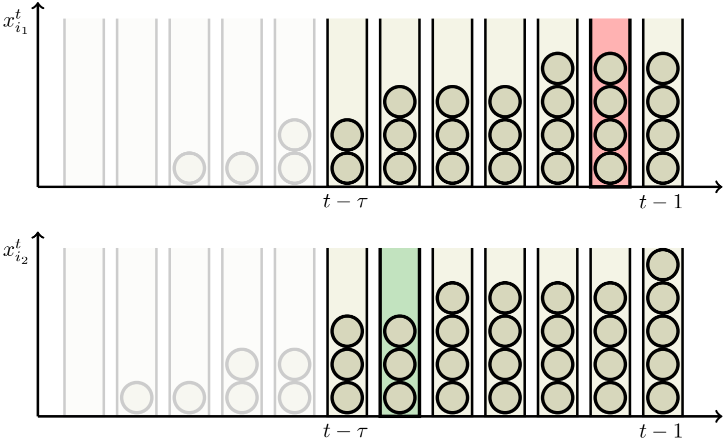
Two-Choice **with outdated information:** τ-Delay

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).



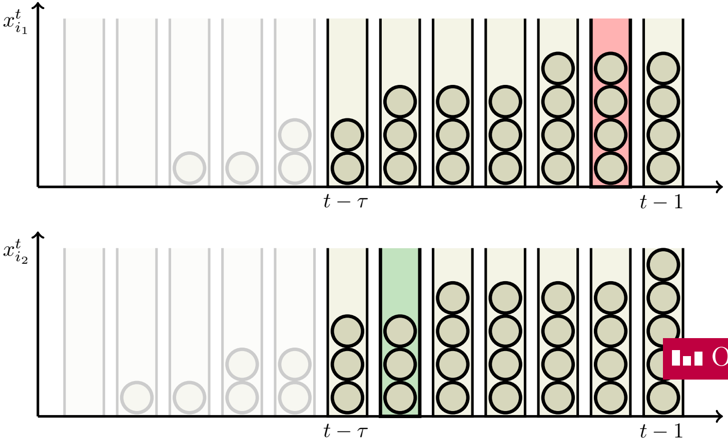
TWO-CHOICE with outdated information: τ-DELAY

- Same argument applies when the reported bin load $\widetilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-DELAY process ($\tau = b$).
- Same upper bounds apply here.



Two-Choice with outdated information: τ-Delay

- Same argument applies when the reported bin load $\tilde{x}_i^{t-1} \in [x_i^{t-\tau}, x_i^{t-1}]$.
- We call this the τ-Delay process ($\tau = b$).
- Same upper bounds apply here.



Open in Visualiser.

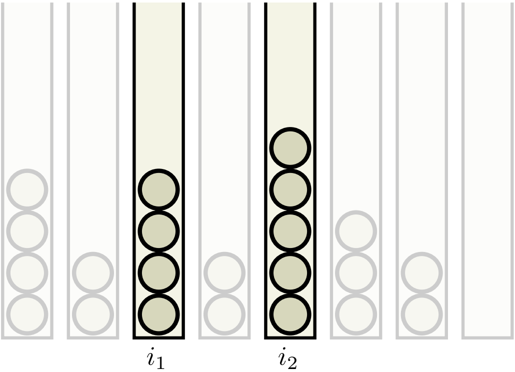
Two-Choice with random noise (I)

Two-Choice with random noise (I)

- Sample two random bins.

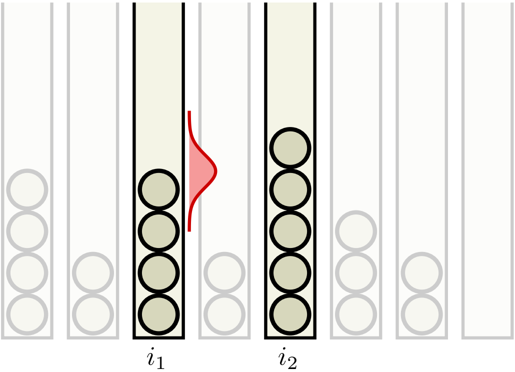
Two-Choice with random noise (I)

- Sample two random bins.



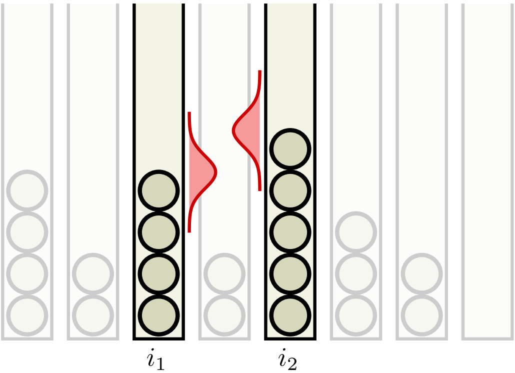
Two-Choice with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.



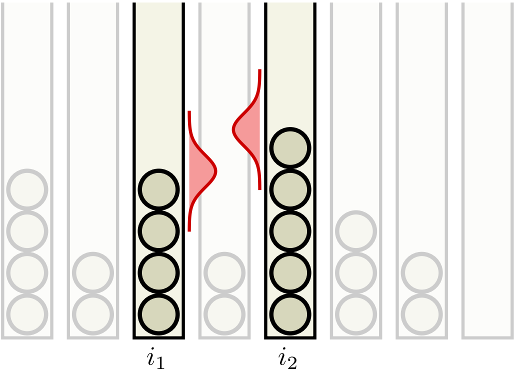
Two-Choice with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.



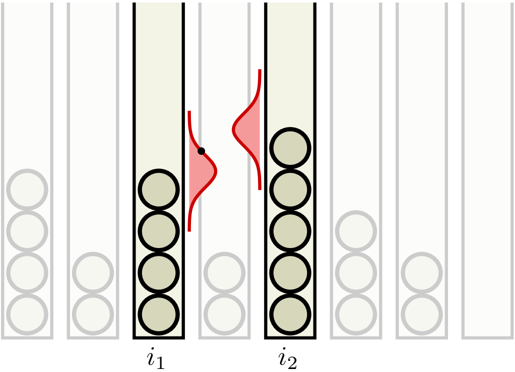
TWO-CHOICE with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise $\rightsquigarrow$ σ-NOISY-LOAD)



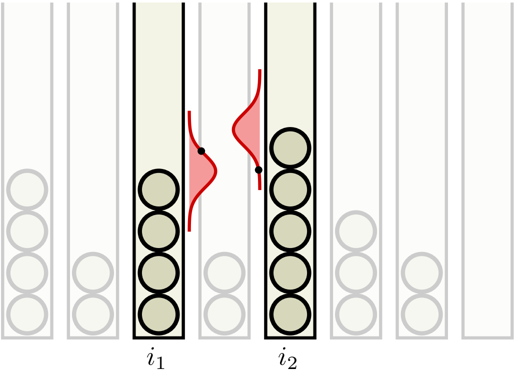
Two-Choice with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise ⇝ σ-Noisy-Load)



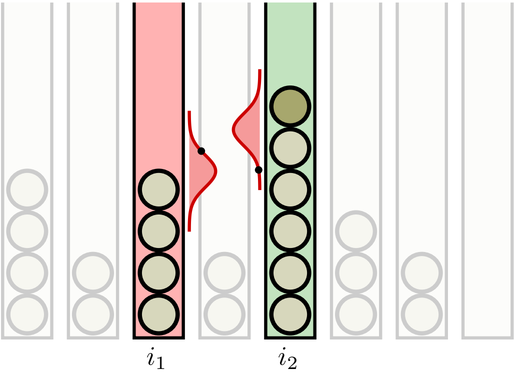
TWO-CHOICE with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise $\rightsquigarrow$ σ-NOISY-LOAD)



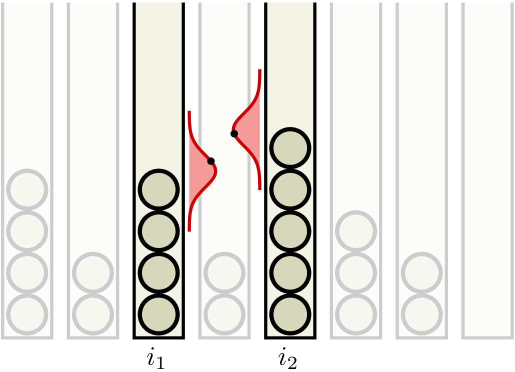
Two-Choice with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise $\rightsquigarrow$ σ-Noisy-Load)
- Allocate to the bin with smaller load estimate.



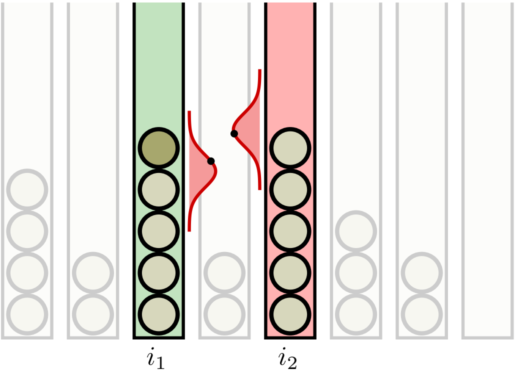
Two-Choice with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise $\rightsquigarrow$ σ-Noisy-Load)
- Allocate to the bin with smaller load estimate.



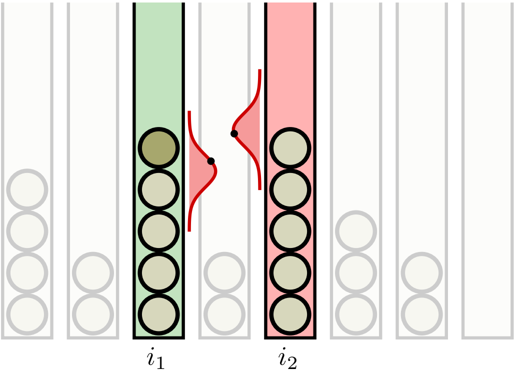
Two-Choice with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise $\rightsquigarrow$ σ-Noisy-Load)
- Allocate to the bin with smaller load estimate.



TWO-CHOICE with random noise (I)

- Sample two random bins.
- Obtain *load estimates* by adding noise to the bin loads.
  - (e.g., normal noise $\rightsquigarrow$ σ-NOISY-LOAD)
- Allocate to the bin with smaller load estimate.



Open in Visualiser.

TWO-CHOICE with random noise (II)

Two-Choice with random noise (II)

- We can further generalize this setting.

Two-Choice with random noise (II)

- We can further generalize this setting.
- Define the probability that the comparison between bins i_1 and i_2 is correct as
$$\rho(|x^t_{i_1} - x^t_{i_2}|).$$

Two-Choice with random noise (II)

- We can further generalize this setting.
- Define the probability that the comparison between bins i_1 and i_2 is correct as

$$\rho(|x_{i_1}^t - x_{i_2}^t|).$$

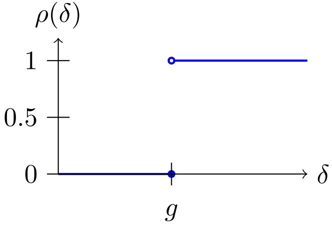
- Captures several processes:

Two-Choice with random noise (II)

- We can further generalize this setting.
- Define the probability that the comparison between bins i_1 and i_2 is correct as

$$\rho(|x_{i_1}^t - x_{i_2}^t|).$$

- Captures several processes: g-Bounded,

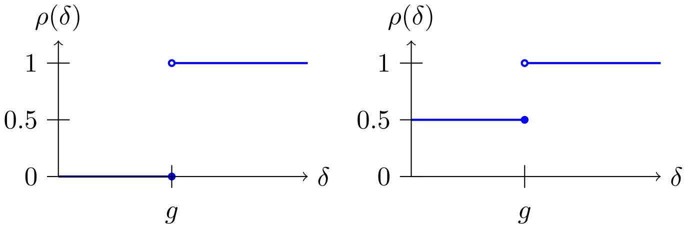


Two-Choice **with random noise (II)**

- We can further generalize this setting.
- Define the probability that the comparison between bins i_1 and i_2 is correct as

$$\rho(|x_{i_1}^t - x_{i_2}^t|).$$

- Captures several processes: g-Bounded, g-Myopic-Comp,

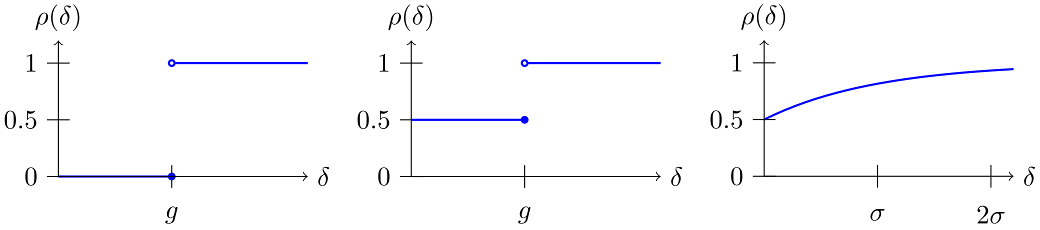


Two-Choice with random noise (II)

- We can further generalize this setting.
- Define the probability that the comparison between bins i_1 and i_2 is correct as

$$\rho(|x_{i_1}^t - x_{i_2}^t|).$$

- Captures several processes: g-Bounded, g-Myopic-Comp, σ-Noisy-Load

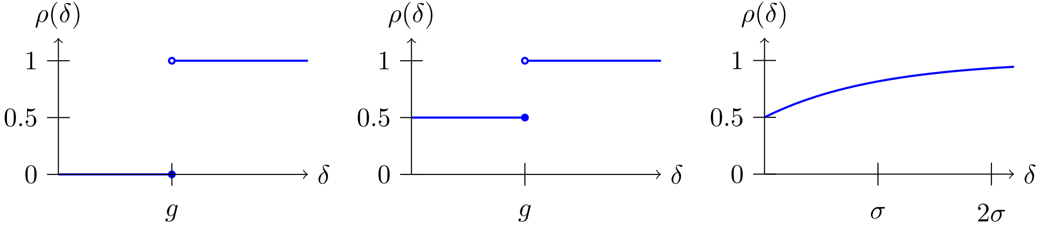


TWO-CHOICE with random noise (II)

- We can further generalize this setting.
- Define the probability that the comparison between bins i_1 and i_2 is correct as

$$\rho(|x^t_{i_1} - x^t_{i_2}|).$$

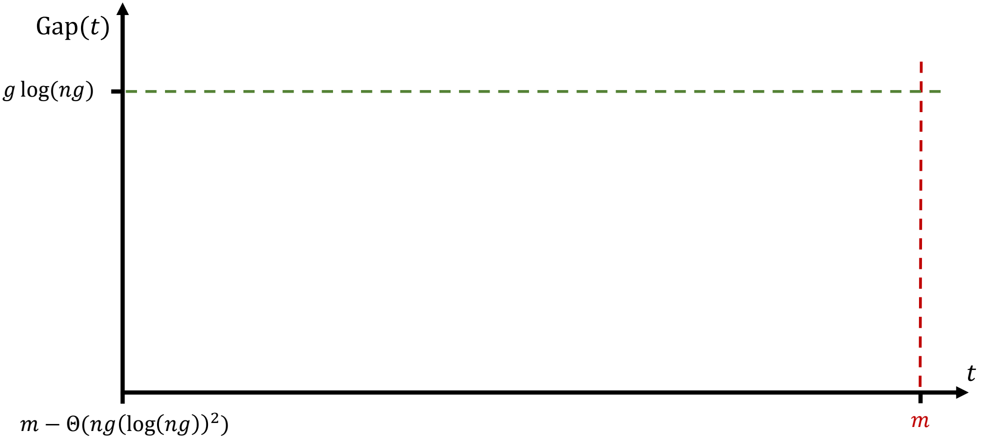
- Captures several processes: g-BOUNDED, g-MYOPIC-COMP, σ-NOISY-LOAD ...



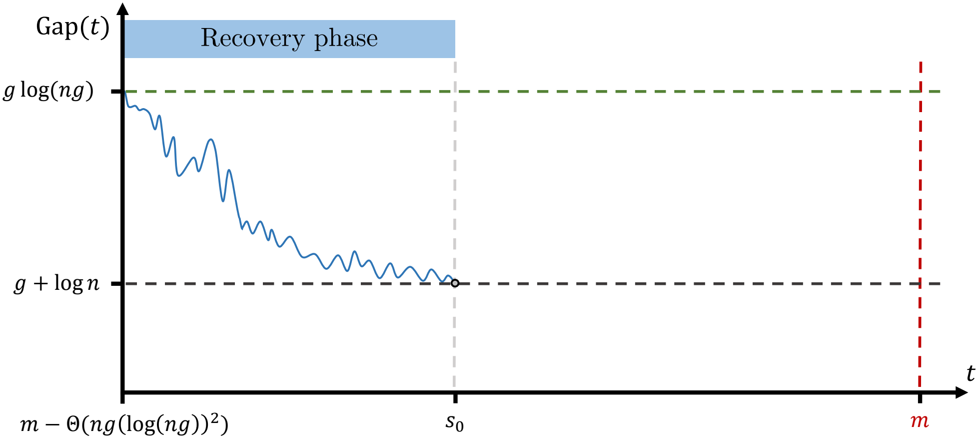
Techniques

Overview

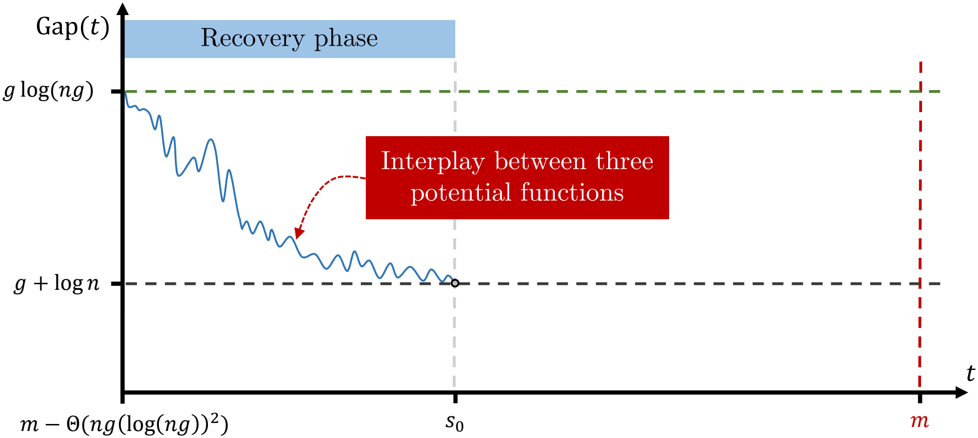
Overview



Overview

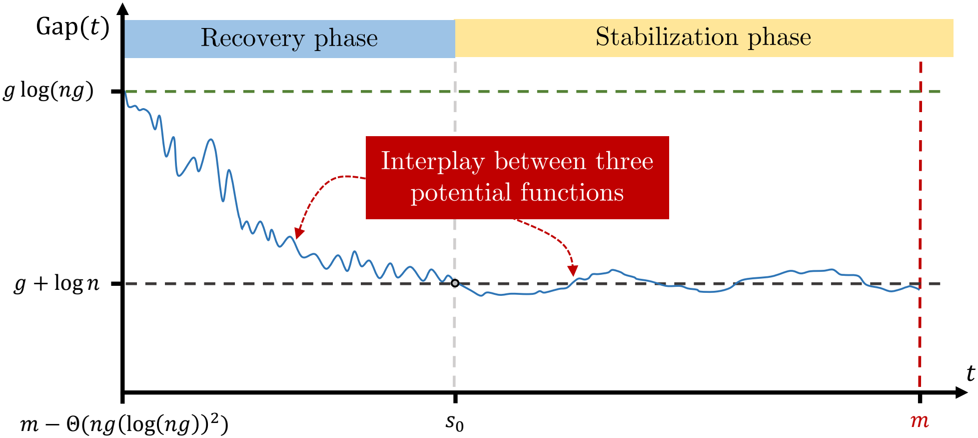


Overview



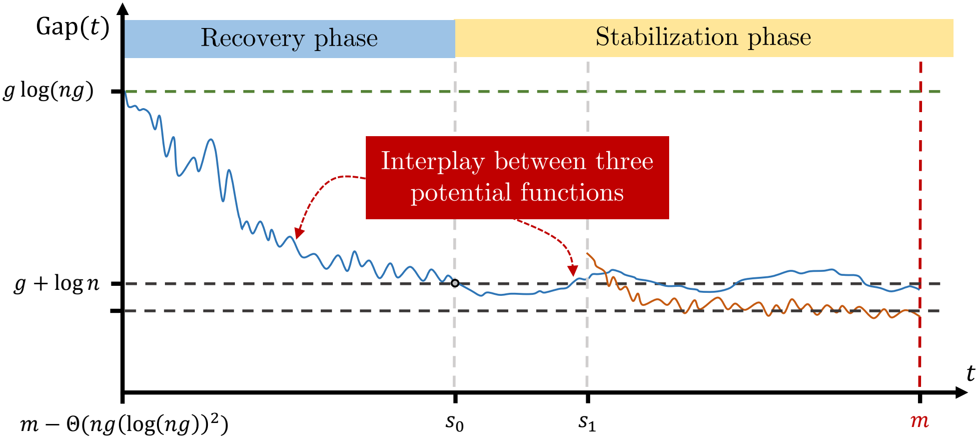
Gap(t)
Recovery phase
$g \log(ng)$
Interplay between three potential functions
$g + \log n$
t
$m - \Theta(ng(\log(ng))^2)$
s_0
m

Overview



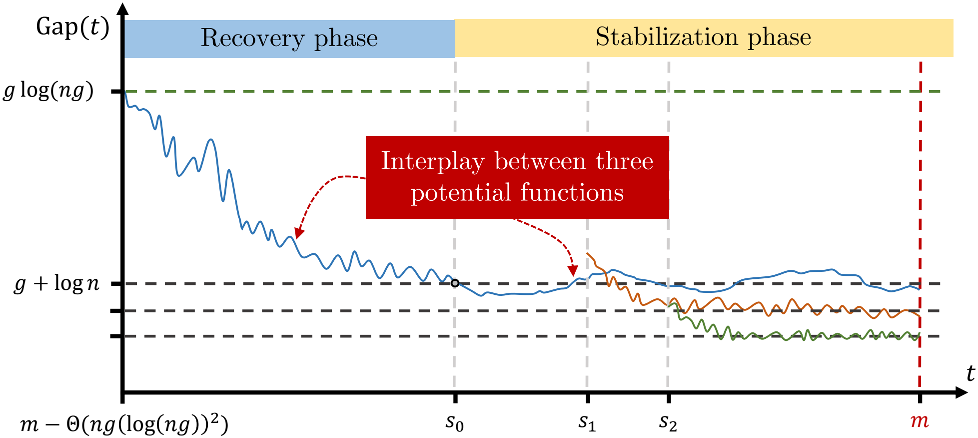
Gap(t)
Recovery phase
Stabilization phase
$g\log(ng)$
Interplay between three potential functions
$g + \log n$
t
$m - \Theta(ng(\log(ng))^2)$
s_0
m

Overview



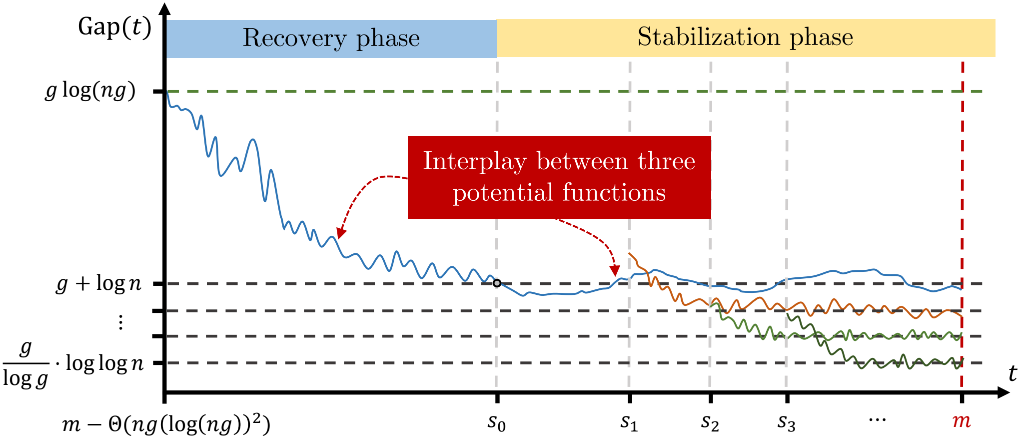
Gap(t)
Recovery phase
Stabilization phase
$g\log(ng)$
Interplay between three potential functions
$g + \log n$
t
$m - \Theta(ng(\log(ng))^2)$
s_0
s_1
m

Overview



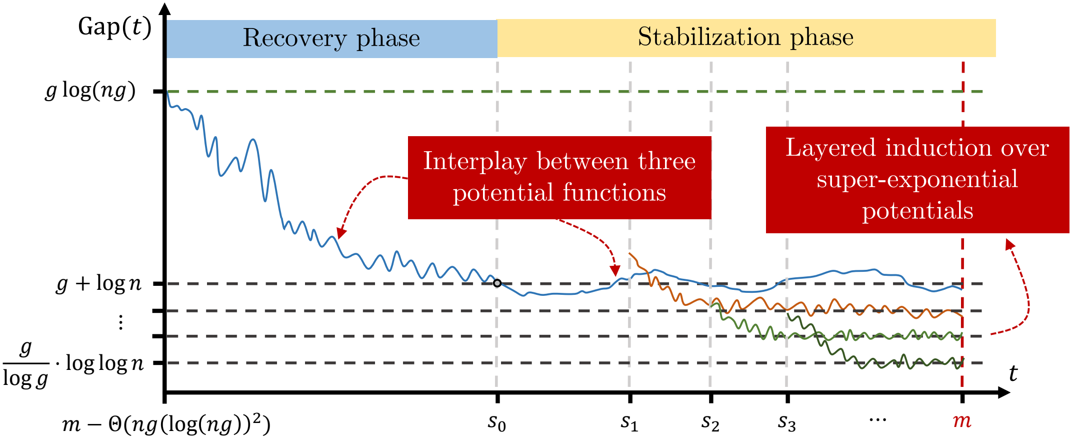
Gap(t)
Recovery phase
Stabilization phase
$g \log(ng)$
Interplay between three potential functions
$g + \log n$
t
$m - \Theta(ng(\log(ng))^2)$
s_0
s_1
s_2
m

Overview



Gap(t)
Recovery phase
Stabilization phase
$g \log(ng)$
Interplay between three potential functions
$g + \log n$
⋮
$\frac{g}{\log g} \cdot \log\log n$
t
$m - \Theta(ng(\log(ng))^2)$
s_0
s_1
s_2
s_3
⋯
m

Overview



Gap(t)
Recovery phase
Stabilization phase
$g\log(ng)$
Interplay between three potential functions
Layered induction over super-exponential potentials
$g+\log n$
⋮
$\frac{g}{\log g}\cdot\log\log n$
t
$m-\Theta(ng(\log(ng))^2)$
s_0
s_1
s_2
s_3
⋯
m

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

Probability allocation vectors

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

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

Probability allocation vectors

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

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

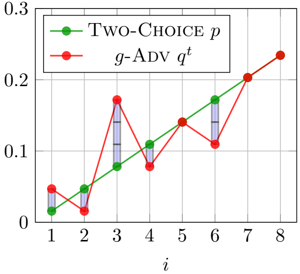
- For g-Adv, the probability vector q^t, is obtained from $p_{\text{Two-Choice}}$, by possibly moving $2/n^2$ probability between bins i_1, i_2 with loads $|x_{i_1}^t - x_{i_2}^t| \leq g$.

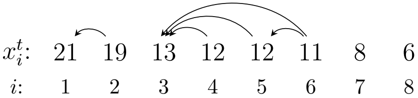
Probability allocation vectors

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

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

- For g-Adv, the probability vector q^t, is obtained from $p_{\text{Two-Choice}}$, by possibly moving $2/n^2$ probability between bins i_1, i_2 with loads $|x_{i_1}^t - x_{i_2}^t| \leq g$.





Upper Bound: $\mathcal{O}(g + \log n)$

Upper Bound: $\mathcal{O}(g + \log n)$

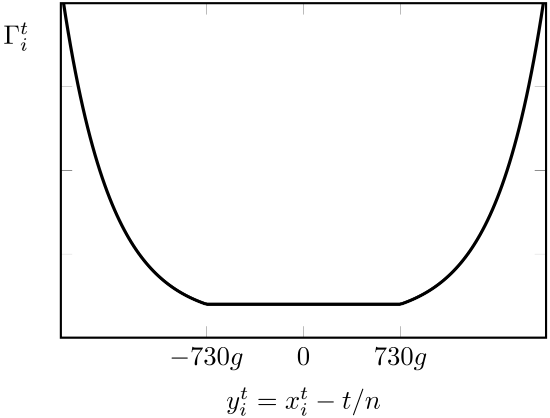
- We use the **hyperbolic cosine potential** [PTW15] with constant $\gamma > 0$:

$$\Gamma^t := \sum_{i=1}^{n} \left[\exp\left(\gamma(x_i^t - t/n - 730g)^+\right) + \exp\left(\gamma(-(x_i^t - t/n) - 730g)^+\right) \right].$$

Upper Bound: $\mathcal{O}(g + \log n)$

- We use the **hyperbolic cosine potential** [PTW15] with constant $\gamma > 0$:

$$\Gamma^t := \sum_{i=1}^{n} \left[\exp\left(\gamma(x_i^t - t/n - 730g)^+\right) + \exp\left(\gamma(-(x_i^t - t/n) - 730g)^+\right) \right].$$



Upper Bound: $\mathcal{O}(g + \log n)$

- We use the **hyperbolic cosine potential** [PTW15] with constant $\gamma > 0$:

$$\Gamma^t := \sum_{i=1}^{n} \Big[\exp\big(\gamma(x_i^t - t/n - 730g)^+\big) + \exp\big(\gamma(-(x_i^t - t/n) - 730g)^+\big) \Big].$$

- When $\Gamma^t = \mathcal{O}(n)$, then $\text{Gap}(t) = \mathcal{O}(g + \log n)$.

Upper Bound: $\mathcal{O}(g + \log n)$

- We use the **hyperbolic cosine potential** [PTW15] with constant $\gamma > 0$:

$$\Gamma^t := \sum_{i=1}^{n} \Big[\exp\big(\gamma(x_i^t - t/n - 730g)^+\big) + \exp\big(\gamma(-(x_i^t - t/n) - 730g)^+\big) \Big].$$

- When $\Gamma^t = \mathcal{O}(n)$, then $\mathrm{Gap}(t) = \mathcal{O}(g + \log n)$. **Goal:** Show w.h.p. $\Gamma^t = \mathcal{O}(n)$.

Upper Bound: $\mathcal{O}(g + \log n)$

- We use the **hyperbolic cosine potential** [PTW15] with constant $\gamma > 0$:

$$\Gamma^t := \sum_{i=1}^{n} \left[\exp\left(\gamma(x_i^t - t/n - 730g)^+\right) + \exp\left(\gamma(-(x_i^t - t/n) - 730g)^+\right) \right].$$

- When $\Gamma^t = \mathcal{O}(n)$, then $\text{Gap}(t) = \mathcal{O}(g + \log n)$. **Goal:** Show w.h.p. $\Gamma^t = \mathcal{O}(n)$.
- **Challenge:** For *some* configurations, Γ^t may *increase* in expectation, even when large. But, we always have the following loose upper bound:

$$\mathbf{E}\left[\, \Gamma^{t+1} \mid \mathfrak{F}^t, \Gamma^t \geq cn \,\right] \leq \Gamma^t \cdot \left(1 + \frac{3\gamma}{n}\right).$$

Upper Bound: $\mathcal{O}(g + \log n)$

- We use the **hyperbolic cosine potential** [PTW15] with constant $\gamma > 0$:

$$\Gamma^t := \sum_{i=1}^{n} \left[\exp\left(\gamma (x_i^t - t/n - 730g)^+\right) + \exp\left(\gamma(-(x_i^t - t/n) - 730g)^+\right) \right].$$

- When $\Gamma^t = \mathcal{O}(n)$, then $\text{Gap}(t) = \mathcal{O}(g + \log n)$. **Goal:** Show w.h.p. $\Gamma^t = \mathcal{O}(n)$.
- **Challenge:** For *some* configurations, Γ^t may *increase* in expectation, even when large. But, we always have the following loose upper bound:

$$\mathbf{E}\left[\, \Gamma^{t+1} \mid \mathfrak{F}^t, \Gamma^t \geq cn \,\right] \leq \Gamma^t \cdot \left(1 + \frac{3\gamma}{n}\right).$$

How can we prove that the potential drops in expectation over multiple steps when large?

Upper Bound: $\mathcal{O}(g + \log n)$

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **absolute value potential**:

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

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **absolute value potential**:

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

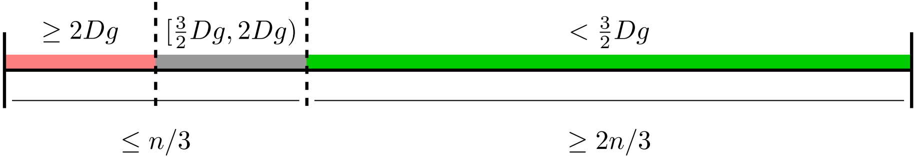
- When $\Delta^t \leq Dng$ $(D = 365)$, then at most $n/3$ bins i with load $\geq t/n + \frac{3}{2}Dg$.

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **absolute value potential**:

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

- When $\Delta^t \leq Dng$ $(D = 365)$, then at most $n/3$ bins i with load $\geq t/n + \frac{3}{2}Dg$. So, there is a bias to place away from bins with load $\geq t/n + 2Dg$.



Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **absolute value potential**:

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

- When $\Delta^t \leq Dng$ ($D = 365$), then at most $n/3$ bins i with load $\geq t/n + \frac{3}{2}Dg$. So, there is a bias to place away from bins with load $\geq t/n + 2Dg$.
- (**Good step**): This bias enough to prove that for some constant $\epsilon > 0$,

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

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **absolute value potential**:

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

- When $\Delta^t \leq Dng$ ($D = 365$), then at most $n/3$ bins i with load $\geq t/n + \frac{3}{2}Dg$. So, there is a bias to place away from bins with load $\geq t/n + 2Dg$.
- (**Good step**): This bias enough to prove that for some constant $\epsilon > 0$,

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

- A properly *adjusted* potential function drops in expectation in every step, for any interval with constant fraction of good steps.

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **absolute value potential**:

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

- When $\Delta^t \leq Dng$ $(D = 365)$, then at most $n/3$ bins i with load $\geq t/n + \frac{3}{2}Dg$. So, there is a bias to place away from bins with load $\geq t/n + 2Dg$.
- (**Good step**): This bias enough to prove that for some constant $\epsilon > 0$,

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

- A properly *adjusted* potential function drops in expectation in every step, for any interval with constant fraction of good steps.

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

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^{n} \left(y_i^t\right)^2$.

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^{n} 2p_i^t y_i^t + 1$$

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For TWO-CHOICE,

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

- For g-Adv,

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t + \sum_{i=1}^n 2q_i^t y_i^t + 1$$

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

- For g-Adv,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2q_i^t y_i^t + 1 \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 + 2g$$

  using the “probability transfer” argument.

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

- For g-Adv,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2q_i^t y_i^t + 1 \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 + 2g \leq \Upsilon^t - \frac{\Delta^t}{n} + 1 + 2g,$$

using the "probability transfer" argument.

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

- For g-Adv,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2q_i^t y_i^t + 1 \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 + 2g \leq \Upsilon^t - \frac{\Delta^t}{n} + 1 + 2g,$$

  using the "probability transfer" argument.
- By induction, we get

$$\mathbf{E}\left[\,\Upsilon^{t+k+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t - \sum_{r=t}^{t+k} \frac{\Delta^t}{n} + (1 + 2g) \cdot (k+1).$$

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

- For g-Adv,

$$\mathbf{E}\left[\, \Upsilon^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t + \sum_{i=1}^n 2q_i^t y_i^t + 1 \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 + 2g \leq \Upsilon^t - \frac{\Delta^t}{n} + 1 + 2g,$$

  using the "probability transfer" argument.
- By induction, we get

$$\mathbf{E}\left[\, \Upsilon^{t+k+1} \mid \mathfrak{F}^t \,\right] \leq \Upsilon^t - \sum_{r=t}^{t+k} \frac{\Delta^t}{n} + (1 + 2g) \cdot (k+1).$$

- When $k = \Omega(\Upsilon^t / g)$, then for a *constant fraction* of the steps $s \in [t, t+k]$ with $\mathbf{E}\left[\, \Delta^s \mid \mathfrak{F}^t \,\right] \leq Dng$.

Upper Bound: $\mathcal{O}(g + \log n)$

- **Solution:** Use the **quadratic potential**: $\Upsilon^t := \sum_{i=1}^n \left(x_i^t - \frac{t}{n}\right)^2 = \sum_{i=1}^n \left(y_i^t\right)^2$.
- For Two-Choice,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 \leq \Upsilon^t - \frac{\Delta^t}{n} + 1.$$

- For g-Adv,

$$\mathbf{E}\left[\,\Upsilon^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t + \sum_{i=1}^n 2q_i^t y_i^t + 1 \leq \Upsilon^t + \sum_{i=1}^n 2p_i^t y_i^t + 1 + 2g \leq \Upsilon^t - \frac{\Delta^t}{n} + 1 + 2g,$$

  using the "probability transfer" argument.
- By induction, we get

$$\mathbf{E}\left[\,\Upsilon^{t+k+1} \mid \mathfrak{F}^t\,\right] \leq \Upsilon^t - \sum_{r=t}^{t+k} \frac{\Delta^t}{n} + (1+2g)\cdot(k+1).$$

- When $k = \Omega(\Upsilon^t/g)$, then for a *constant fraction* of the steps $s \in [t, t+k]$ with $\mathbf{E}\left[\,\Delta^s \mid \mathfrak{F}^t\,\right] \leq Dng$.
- This concludes the $\mathcal{O}(g + \log n)$ bound.

Summary & Future Work

Summary of results:

Summary & Future Work

Summary of results:

- For any g-Adv process,

Summary & Future Work

Summary of results:

- For any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.

Summary & Future Work

Summary of results:

- For any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}\left(\frac{g}{\log g} \cdot \log\log n\right)$.

Summary & Future Work

Summary of results:

- For any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}\left(\frac{g}{\log g} \cdot \log\log n\right)$.
- Matching lower bound for the g-Myopic-Comp process.

Summary & Future Work

Summary of results:

- For any g-ADV process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}\big(\frac{g}{\log g} \cdot \log\log n\big)$.
- Matching lower bound for the g-MYOPIC-COMP process.
- Tight bounds for TWO-CHOICE with outdated information.

Summary & Future Work

Summary of results:

- For any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}\left(\frac{g}{\log g} \cdot \log\log n\right)$.
- Matching lower bound for the g-Myopic-Comp process.
- Tight bounds for Two-Choice with outdated information.

Future work:

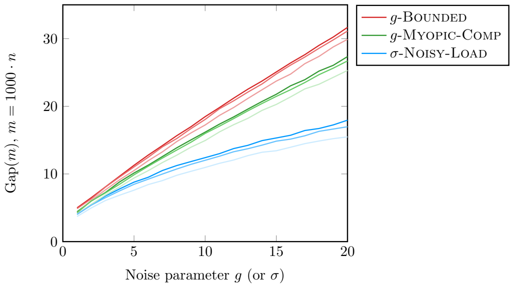
Summary & Future Work

Summary of results:

- For any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}\Big(\frac{g}{\log g} \cdot \log\log n\Big)$.
- Matching lower bound for the g-Myopic-Comp process.
- Tight bounds for Two-Choice with outdated information.

Future work:

- Improve the bounds for σ-Noisy-Load (or other distributions ρ).



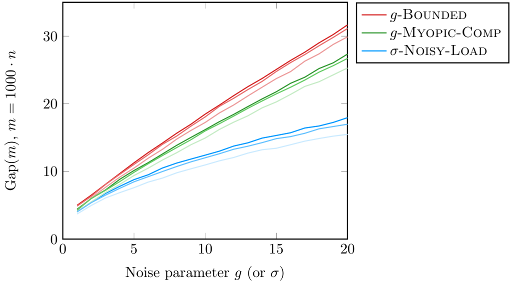
Summary & Future Work

Summary of results:

- For any g-Adv process,
  - If $g \geq \log n$, then for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\mathrm{Gap}(m) = \mathcal{O}\left(\frac{g}{\log g} \cdot \log\log n\right)$.
- Matching lower bound for the g-Myopic-Comp process.
- Tight bounds for Two-Choice with outdated information.

Future work:

- Improve the bounds for σ-Noisy-Load (or other distributions ρ).
- Analyze the noisy and outdated setting for other processes.



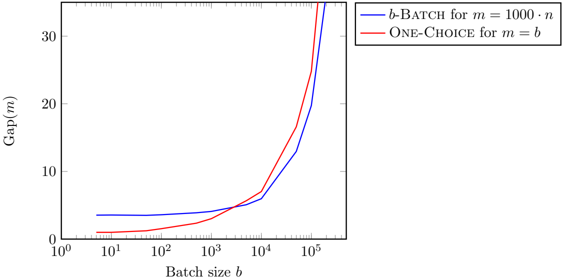
Summary & Future Work

Summary of results:

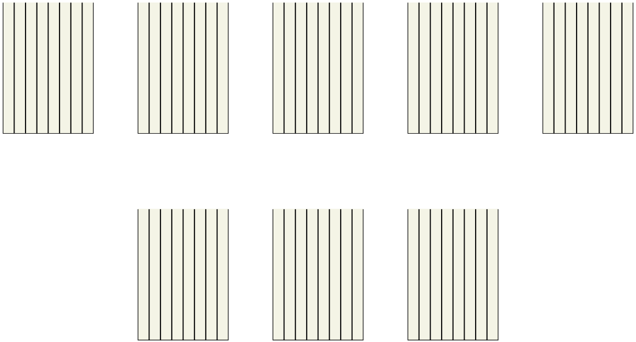
- For any g-ADV process,
  - If $g \geq \log n$, then for any m, w.h.p. $\text{Gap}(m) = \mathcal{O}(g)$.
  - Otherwise, for any m, w.h.p. $\text{Gap}(m) = \mathcal{O}\Big(\frac{g}{\log g} \cdot \log\log n\Big)$.
- Matching lower bound for the g-MYOPIC-COMP process.
- Tight bounds for TWO-CHOICE with outdated information.

Future work:

- Improve the bounds for σ-NOISY-LOAD (or other distributions ρ).
- Analyze the noisy and outdated setting for other processes.

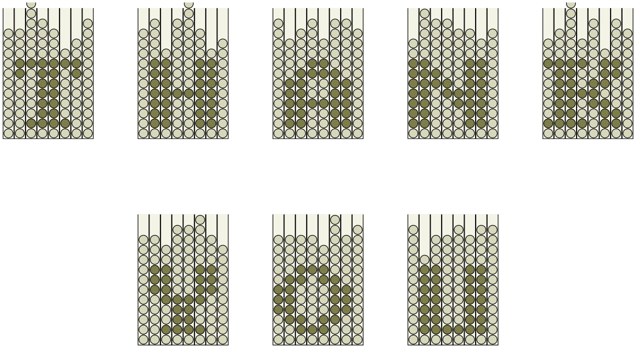


Questions?



Visualisations: `dimitrioslos.com/podc22`

Questions?



Visualisations: `dimitrioslos.com/podc22`

Appendix A: Detailed results for noise models

Model	Range	Lower Bound	Upper Bound
g-Bounded	$1 \leq g$	–	$\mathcal{O}(g \cdot \log(ng))$
g-Adv	$1 \leq g$	–	$\mathcal{O}(g + \log n)$
g-Adv	$1 < g \leq \log n$	–	$\mathcal{O}\big(\frac{g}{\log g} \cdot \log \log n\big)$
g-Myopic-Comp	$\frac{\log n}{\log \log n} \leq g$	$\Omega(g)$	—
g-Myopic-Comp	$1 < g \leq \frac{\log n}{\log \log n}$	$\Omega\big(\frac{g}{\log g} \cdot \log \log n\big)$	–
σ-Noisy-Load	$1 \leq \sigma$	–	$\mathcal{O}(\sigma \sqrt{\log n} \cdot \log(n\sigma))$
σ-Noisy-Load	$2 \cdot (\log n)^{-1/3} \leq \sigma$	$\Omega(\min\{1, \sigma\} \cdot (\log n)^{1/3})$	–
σ-Noisy-Load	$32 \leq \sigma$	$\Omega(\min\{\sigma^{4/5}, \sigma^{2/5} \cdot \sqrt{\log n}\})$	–

Table: Overview of the lower and upper bounds for Two-Choice with noisy information derived in previous works (rows in Gray) and in this work (rows in Green). Upper bounds hold for all values of $m \geq n$, while lower bounds may only hold for a suitable value of m.

Appendix A: Detailed results for outdated information

Model	Range	Lower Bound	Upper Bound
b-Batch	$b = \Omega(n \log n)$	$\Omega(b/n)$	$\mathcal{O}(b/n)$
b-Batch	$b = n$	$\Omega(\frac{\log n}{\log \log n})$	$\mathcal{O}(\log n)$
τ-Delay	$\tau = n$	–	$\mathcal{O}(\frac{\log n}{\log \log n})$
τ-Delay	$\tau \in \left[n \cdot e^{-(\log n)^c}, n \log n\right]$	–	$\mathcal{O}\left(\frac{\log n}{\log((4n/\tau) \log n)}\right)$
τ-Delay	$\tau = n^{1-\epsilon}$	–	$\mathcal{O}(\log \log n)$
b-Batch	$b = n$	$\Omega(\frac{\log n}{\log \log n})$	–
b-Batch	$b \in \left[n \cdot e^{-(\log n)^c}, n \log n\right]$	$\Omega\left(\frac{\log n}{\log((4n/b) \log n)}\right)$	–
b-Batch	$b = n^{1-\epsilon}$	$\Omega(\log \log n)$	–

Table: Overview of the lower and upper bounds for Two-Choice with outdated information, derived in previous works (rows in Gray) and in this work (rows in Green). Upper bounds hold for all values of $m \geq n$, while lower bounds may only hold for a suitable value of m.

Appendix B: Analysis outline for outdated information

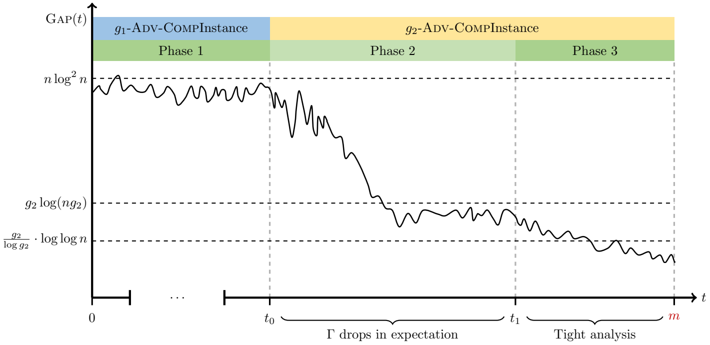


Figure: τ-Delay (and b-Batch) can be exactly simulated using a g_1-Adv-Comp process with $g_1 = \tau \leq n \log n$. This gives the $\mathcal{O}(n \log^2 n)$ gap (since $\tau \leq n \log n$). Then w.h.p. for n^3 steps it can be simulated using a g_2-Adv-Comp process where g_2 is the One-Choice gap for 2τ balls.

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

- [PTW15] used the **hyperbolic cosine potential** (with no offset)

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

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

- [PTW15] used the **hyperbolic cosine potential** (with no offset)

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

- [PTW15] showed that for Two-Choice, for small enough γ,

$$\begin{aligned}\mathbf{E}\left[\, \Gamma^{t+1} \mid y^t \,\right] &\leq \Gamma^t + \sum_{i=1}^{n} p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n}\Gamma^t\right) \\ &\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{48n}\right) + c.\end{aligned}$$

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

- [PTW15] used the **hyperbolic cosine potential** (with no offset)

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

- [PTW15] showed that for Two-Choice, for small enough γ,

$$\begin{aligned}\mathbf{E}\left[\,\Gamma^{t+1} \,\middle|\, y^t\,\right] &\leq \Gamma^t + \sum_{i=1}^n p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n}\Gamma^t\right) \\ &\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{48n}\right) + c.\end{aligned}$$

- Implies that $\mathbf{E}\left[\,\Gamma^m\,\right] \leq \frac{48c}{\gamma} \cdot n$.

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

- [PTW15] used the **hyperbolic cosine potential** (with no offset)

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

- [PTW15] showed that for Two-Choice, for small enough γ,

$$\begin{aligned}\mathbf{E}\left[\, \Gamma^{t+1} \,\middle|\, y^t \,\right] &\leq \Gamma^t + \sum_{i=1}^{n} p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n}\Gamma^t\right) \\ &\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{48n}\right) + c.\end{aligned}$$

- Implies that $\mathbf{E}\left[\, \Gamma^m \,\right] \leq \frac{48c}{\gamma} \cdot n$.
- By *Markov's inequality*, we get $\mathbf{Pr}\left[\Gamma^m \leq \frac{n^3}{\gamma}\right] \geq 1 - n^{-2}$

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

- [PTW15] used the **hyperbolic cosine potential** (with no offset)

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

- [PTW15] showed that for Two-Choice, for small enough γ,

$$\begin{aligned} \mathbf{E}\left[\, \Gamma^{t+1} \,\middle|\, y^t \,\right] &\leq \Gamma^t + \sum_{i=1}^{n} p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n} \Gamma^t\right) \\ &\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{48n}\right) + c. \end{aligned}$$

- Implies that $\mathbf{E}\left[\, \Gamma^m \,\right] \leq \frac{48c}{\gamma} \cdot n$.
- By *Markov's inequality*, we get $\mathbf{Pr}\left[\Gamma^m \leq \frac{n^3}{\gamma}\right] \geq 1 - n^{-2}$ which implies

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

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (I)

- [PTW15] used the **hyperbolic cosine potential** (with no offset)

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

- [PTW15] showed that for Two-Choice, for small enough γ,

$$\begin{aligned}\mathbf{E}\left[\, \Gamma^{t+1} \,\middle|\, y^t \,\right] &\leq \Gamma^t + \sum_{i=1}^{n} p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n}\Gamma^t\right) \\ &\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{48n}\right) + c.\end{aligned}$$

- Implies that $\mathbf{E}\left[\, \Gamma^m \,\right] \leq \frac{48c}{\gamma} \cdot n$.
- By *Markov's inequality*, we get $\mathbf{Pr}\left[\Gamma^m \leq \frac{n^3}{\gamma}\right] \geq 1 - n^{-2}$ which implies

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

- This gives that $\mathrm{Gap}(m) = \mathcal{O}(\frac{\log(n/\gamma)}{\gamma})$.

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (II)

- For g-ADV, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x^t_{i_1} - x^t_{i_2}| \leq g$.

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (II)

- For g-Adv, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x_{i_1}^t - x_{i_2}^t| \leq g$.
- Each transfer increases the bound by at most
$$\frac{2}{n^2} \cdot \gamma \cdot \left(e^{\gamma(x_i^t - t/n + g)} - e^{\gamma(x_i^t - t/n)}\right)$$

Appendix C: Upper bound of $\mathcal{O}(g\log(ng))$ (II)

- For g-ADV, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x^t_{i_1} - x^t_{i_2}| \leq g$.
- Each transfer increases the bound by at most $\frac{2}{n^2} \cdot \gamma \cdot \left(e^{\gamma(x^t_i - t/n + g)} - e^{\gamma(x^t_i - t/n)}\right) = \frac{2}{n^2} \cdot \gamma \cdot e^{\gamma(x^t_i - t/n)} \cdot (e^{\gamma g} - 1)$

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (II)

- For g-ADV, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x_{i_1}^t - x_{i_2}^t| \le g$.
- Each transfer increases the bound by at most
$$\frac{2}{n^2} \cdot \gamma \cdot \left(e^{\gamma(x_i^t - t/n + g)} - e^{\gamma(x_i^t - t/n)}\right) = \frac{2}{n^2} \cdot \gamma \cdot e^{\gamma(x_i^t - t/n)} \cdot (e^{\gamma g} - 1) \le \frac{2}{n^2} \cdot \gamma^2 \cdot e^{\gamma(x_i^t - t/n)},$$
by choosing $\gamma = \Theta(1/g)$.

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (II)

- For g-Adv, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x_{i_1}^t - x_{i_2}^t| \leq g$.
- Each transfer increases the bound by at most
$$\frac{2}{n^2} \cdot \gamma \cdot \left(e^{\gamma(x_i^t - t/n + g)} - e^{\gamma(x_i^t - t/n)}\right) = \frac{2}{n^2} \cdot \gamma \cdot e^{\gamma(x_i^t - t/n)} \cdot (e^{\gamma g} - 1) \leq \frac{2}{n^2} \cdot \gamma^2 \cdot e^{\gamma(x_i^t - t/n)},$$
by choosing $\gamma = \Theta(1/g)$. Similarly for the underloaded component.

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (II)

- For g-Adv, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x_{i_1}^t - x_{i_2}^t| \leq g$.
- Each transfer increases the bound by at most
$$\frac{2}{n^2} \cdot \gamma \cdot \left(e^{\gamma(x_i^t - t/n + g)} - e^{\gamma(x_i^t - t/n)}\right) = \frac{2}{n^2} \cdot \gamma \cdot e^{\gamma(x_i^t - t/n)} \cdot (e^{\gamma g} - 1) \leq \frac{2}{n^2} \cdot \gamma^2 \cdot e^{\gamma(x_i^t - t/n)},$$
by choosing $\gamma = \Theta(1/g)$. Similarly for the underloaded component.
- Hence, on aggregate
$$\mathbf{E}\left[\, \Gamma^{t+1} \mid y^t \,\right] \leq \Gamma^t + \sum_{i=1}^{n} p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n}\Gamma^t\right) + \mathcal{O}\left(\frac{\gamma^2}{n}\Gamma^t\right)$$
$$\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{96n}\right) + c.$$

Appendix C: Upper bound of $\mathcal{O}(g \log(ng))$ (II)

- For g-Adv, the adversary can "transfer" $2/n^2$ probability from i_1 to i_2 if $|x_{i_1}^t - x_{i_2}^t| \leq g$.
- Each transfer increases the bound by at most
$$\frac{2}{n^2} \cdot \gamma \cdot \left(e^{\gamma(x_i^t - t/n + g)} - e^{\gamma(x_i^t - t/n)} \right) = \frac{2}{n^2} \cdot \gamma \cdot e^{\gamma(x_i^t - t/n)} \cdot (e^{\gamma g} - 1) \leq \frac{2}{n^2} \cdot \gamma^2 \cdot e^{\gamma(x_i^t - t/n)},$$
by choosing $\gamma = \Theta(1/g)$. Similarly for the underloaded component.
- Hence, on aggregate
$$\begin{aligned} \mathbf{E}\left[\Gamma^{t+1} \mid y^t \right] &\leq \Gamma^t + \sum_{i=1}^{n} p_i \cdot (\gamma + \gamma^2) \cdot e^{\gamma y_i^t} + p_i \cdot (-\gamma + \gamma^2) \cdot e^{-\gamma y_i^t} + \mathcal{O}\left(\frac{\gamma}{n} \Gamma^t \right) + \mathcal{O}\left(\frac{\gamma^2}{n} \Gamma^t \right) \\ &\leq \Gamma^t \cdot \left(1 - \frac{\gamma}{96n} \right) + c. \end{aligned}$$
- This implies that $\mathrm{Gap}(m) = \mathcal{O}(g \log(ng))$.

Appendix D: Upper bound of $\mathcal{O}(\frac{g}{\log g} \log \log n)$ for $g \leq \log n$

Appendix D: Upper bound of $\mathcal{O}(\frac{g}{\log g} \log\log n)$ for $g \leq \log n$

- We define the **super-exponential potentials**, for $1 \leq j \leq \frac{\log\log n}{\log g} := k$:

$$\Phi_j^t := \sum_{i=1}^{n} \exp\left(\gamma \cdot (\log n) \cdot g^{j-k} \cdot \left(x_i^t - \frac{t}{n} - z_j\right)^+\right),$$

where $z_j := \Theta(j \cdot g)$.

Appendix D: Upper bound of $\mathcal{O}(\frac{g}{\log g} \log\log n)$ for $g \leq \log n$

- We define the **super-exponential potentials**, for $1 \leq j \leq \frac{\log\log n}{\log g} := k$:

$$\Phi_j^t := \sum_{i=1}^{n} \exp\left(\gamma \cdot (\log n) \cdot g^{j-k} \cdot \left(x_i^t - \frac{t}{n} - z_j\right)^+\right),$$

  where $z_j := \Theta(j \cdot g)$.
- When $\Phi_{j-1}^t = \mathcal{O}(n)$, then the number of bins i with $x_i^t \geq \frac{t}{n} + z_j$ is at most

$$n \cdot e^{-(\log n) \cdot g^{j-k}} := n \cdot \delta_j.$$

- Hence, $q_i^t \leq \frac{2\delta_j}{n}$.

Appendix D: Upper bound of $\mathcal{O}(\frac{g}{\log g} \log\log n)$ for $g \leq \log n$

- We define the **super-exponential potentials**, for $1 \leq j \leq \frac{\log\log n}{\log g} := k$:

$$\Phi_j^t := \sum_{i=1}^{n} \exp\left(\gamma \cdot (\log n) \cdot g^{j-k} \cdot \left(x_i^t - \frac{t}{n} - z_j\right)^+\right),$$

  where $z_j := \Theta(j \cdot g)$.
- When $\Phi_{j-1}^t = \mathcal{O}(n)$, then the number of bins i with $x_i^t \geq \frac{t}{n} + z_j$ is at most

$$n \cdot e^{-(\log n) \cdot g^{j-k}} := n \cdot \delta_j.$$

- Hence, $q_i^t \leq \frac{2\delta_j}{n}$.
- Similarly to [LS22b]

$$\mathbf{E}\left[\, \Phi_j^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi_{j-1}^t = \mathcal{O}(n) \,\right] \leq \Phi_j^t \cdot \left(1 - \frac{1}{n}\right) + 2.$$

Appendix D: Upper bound of $\mathcal{O}(\frac{g}{\log g} \log \log n)$ for $g \leq \log n$

- We define the **super-exponential potentials**, for $1 \leq j \leq \frac{\log \log n}{\log g} := k$:

$$\Phi_j^t := \sum_{i=1}^{n} \exp\left(\gamma \cdot (\log n) \cdot g^{j-k} \cdot \left(x_i^t - \frac{t}{n} - z_j\right)^+\right),$$

where $z_j := \Theta(j \cdot g)$.
- When $\Phi_{j-1}^t = \mathcal{O}(n)$, then the number of bins i with $x_i^t \geq \frac{t}{n} + z_j$ is at most

$$n \cdot e^{-(\log n) \cdot g^{j-k}} := n \cdot \delta_j.$$

- Hence, $q_i^t \leq \frac{2\delta_j}{n}$.
- Similarly to [LS22b]

$$\mathbf{E}\left[\, \Phi_j^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi_{j-1}^t = \mathcal{O}(n) \,\right] \leq \Phi_j^t \cdot \left(1 - \frac{1}{n}\right) + 2.$$

- And so, after $s = n \cdot \text{polylog}(n)$ steps, we get

$$\mathbf{E}\left[\, \Phi_j^{t+s} \,\middle|\, \mathfrak{F}^t, \cap_{r \in [t,t+s)} \Phi_{j-1}^r = \mathcal{O}(n) \,\right] = \mathcal{O}(n).$$

Appendix D: Upper bound of $\mathcal{O}(\frac{g}{\log g} \log\log n)$ for $g \leq \log n$

- We define the **super-exponential potentials**, for $1 \leq j \leq \frac{\log\log n}{\log g} := k$:

$$\Phi_j^t := \sum_{i=1}^{n} \exp\left(\gamma \cdot (\log n) \cdot g^{j-k} \cdot \left(x_i^t - \frac{t}{n} - z_j\right)^+\right),$$

  where $z_j := \Theta(j \cdot g)$.
- When $\Phi_{j-1}^t = \mathcal{O}(n)$, then the number of bins i with $x_i^t \geq \frac{t}{n} + z_j$ is at most

$$n \cdot e^{-(\log n) \cdot g^{j-k}} := n \cdot \delta_j.$$

- Hence, $q_i^t \leq \frac{2\delta_j}{n}$.
- Similarly to [LS22b]

$$\mathbf{E}\left[\, \Phi_j^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi_{j-1}^t = \mathcal{O}(n) \,\right] \leq \Phi_j^t \cdot \left(1 - \frac{1}{n}\right) + 2.$$

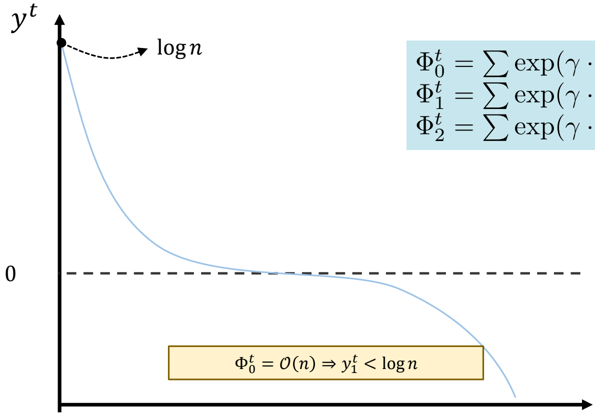
- And so, after $s = n \cdot \mathrm{polylog}(n)$ steps, we get

$$\mathbf{E}\left[\, \Phi_j^{t+s} \mid \mathfrak{F}^t, \cap_{r \in [t,t+s)} \Phi_{j-1}^r = \mathcal{O}(n) \,\right] = \mathcal{O}(n).$$

- Finally, when $\Phi_{k-1}^t = \mathcal{O}(n)$, we obtain that

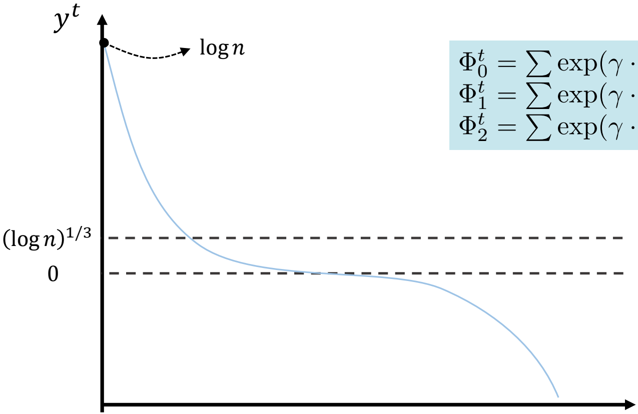
$$\mathrm{Gap}(t) = \mathcal{O}(k \cdot g) = \mathcal{O}\left(\frac{g}{\log g} \log\log n\right).$$

Appendix E: Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot g)$, for $g = (\log n)^{1/3}$



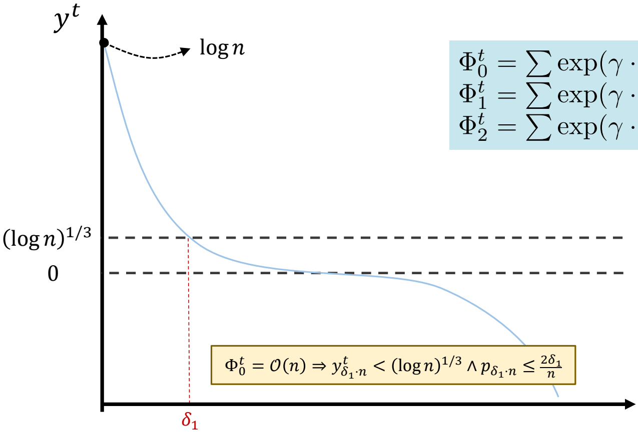
$$\Phi_0^t = \sum \exp(\gamma \cdot (\ldots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\ldots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\ldots))$$

Appendix E: Proving $\text{Gap}(m) = \mathcal{O}(k \cdot g)$, for $g = (\log n)^{1/3}$



$$\Phi_0^t = \sum \exp(\gamma \cdot (\ldots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\ldots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\ldots))$$

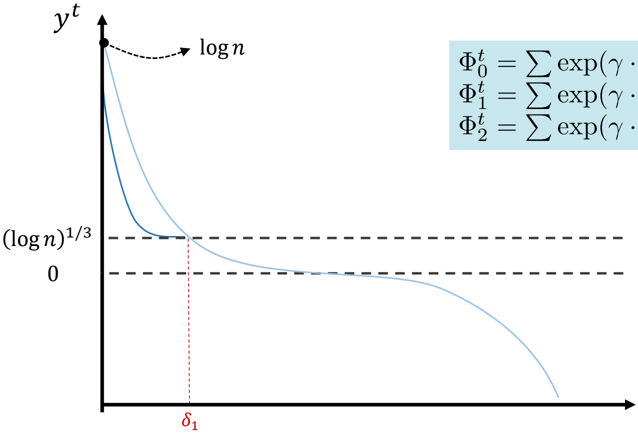
Appendix E: Proving $\text{Gap}(m) = \mathcal{O}(k \cdot g)$, for $g = (\log n)^{1/3}$



$$\Phi_0^t = \sum \exp(\gamma \cdot (\ldots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\ldots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\ldots))$$

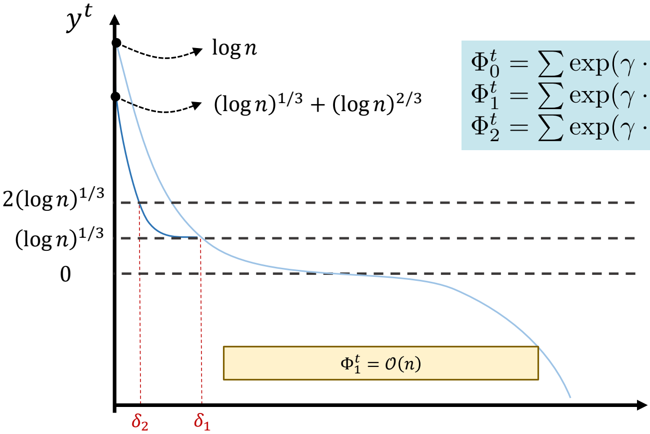
$\Phi_0^t = \mathcal{O}(n) \Rightarrow y_{\delta_1 \cdot n}^t < (\log n)^{1/3} \wedge p_{\delta_1 \cdot n} \leq \frac{2\delta_1}{n}$

Appendix E: Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot g)$**, for** $g = (\log n)^{1/3}$



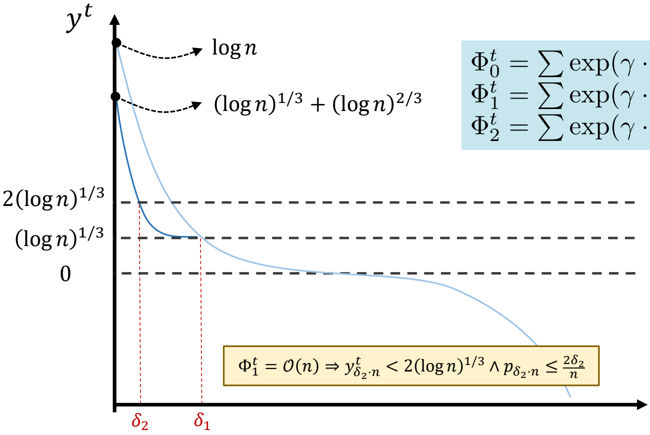
$$\Phi_0^t = \sum \exp(\gamma \cdot (\dots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\dots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\dots))$$

Appendix E: Proving $\mathrm{Gap}(m) = \mathcal{O}(k \cdot g)$, for $g = (\log n)^{1/3}$



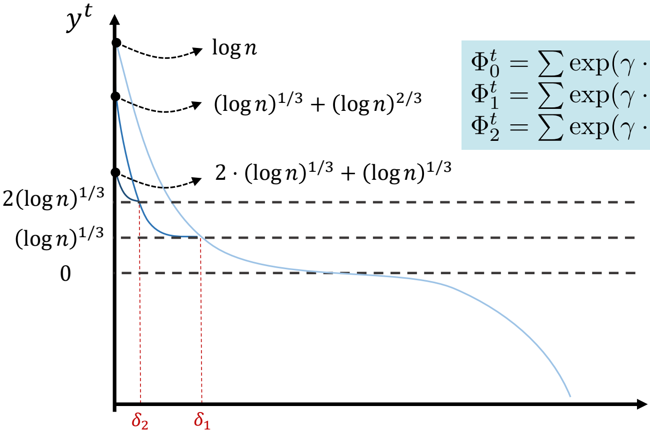
$$\Phi_0^t = \sum \exp(\gamma \cdot (\ldots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\ldots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\ldots))$$

Appendix E: Proving $\text{Gap}(m) = \mathcal{O}(k \cdot g)$, for $g = (\log n)^{1/3}$



$$\Phi_0^t = \sum \exp(\gamma \cdot (\ldots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\ldots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\ldots))$$

Appendix E: Proving $\text{Gap}(m) = \mathcal{O}(k \cdot g)$, for $g = (\log n)^{1/3}$



$$\Phi_0^t = \sum \exp(\gamma \cdot (\ldots))$$
$$\Phi_1^t = \sum \exp(\gamma \cdot (\log n)^{1/3} \cdot (\ldots))$$
$$\Phi_2^t = \sum \exp(\gamma \cdot (\log n)^{2/3} \cdot (\ldots))$$

Bibliography I

- D. Alistarh, T. Brown, J. Kopinsky, J. Z. Li, and G. Nadiradze, *Distributionally linearizable data structures*, 30th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'18), ACM, 2018, pp. 133–142.
- 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, M. Englert, T. Friedetzky, and L. Nagel, *Multiple-choice balanced allocation in (almost) parallel*, 16th International Workshop on Randomization and Computation (RANDOM'12) (Berlin Heidelberg), Springer-Verlag, 2012, pp. 411–422.
- P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, *Balanced allocations: the heavily loaded case*, SIAM J. Comput. **35** (2006), no. 6, 1350–1385.
- R.J. Gibbens, F.P. Kelly, and P.B. Key, *Dynamic alternative routing – modelling and behavior*, Proceedings of the 12 International Teletraffic Congress, Torino, Italy, Elsevier, Amsterdam, 1988.

Bibliography II

- G. H. Gonnet, *Expected length of the longest probe sequence in hash code searching*, J. Assoc. Comput. Mach. **28** (1981), no. 2, 289–304.
- 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.
- D. Los and T. Sauerwald, *Balanced allocations in batches: Simplified and generalized*, 34th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA'22), ACM, 2022, p. 389–400.
- ———, *Balanced Allocations with Incomplete Information: The Power of Two Queries*, 13th Innovations in Theoretical Computer Science Conference (ITCS'22) (Dagstuhl, Germany), vol. 215, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, pp. 103:1–103:23.
- R. Pagh and F. F. Rodler, *Cuckoo hashing*, Algorithms—ESA 2001 (Århus), Lecture Notes in Comput. Sci., vol. 2161, Springer, Berlin, 2001, pp. 121–133.

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.
- U. Wieder, *Hashing, load balancing and multiple choice*, Found. Trends Theor. Comput. Sci. **12** (2016), no. 3-4, front matter, 276–379.