Balanced Allocations: A refined drift theorem

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

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

Outline

Outline

- Balanced allocations (background and some highlights)
- The exponential and hyperbolic cosine potential functions
- The proof of the drift theorem
- The refinement and its applications
- Open problems

Balanced allocations: Background

Balanced allocations setting

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

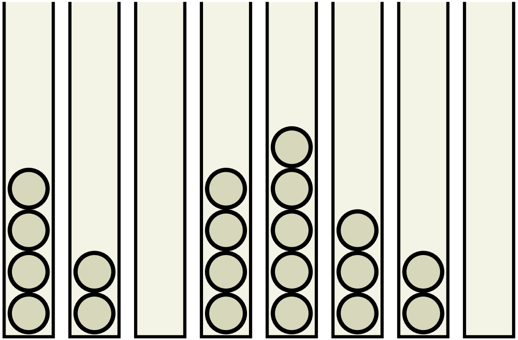
Balanced allocations setting

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.

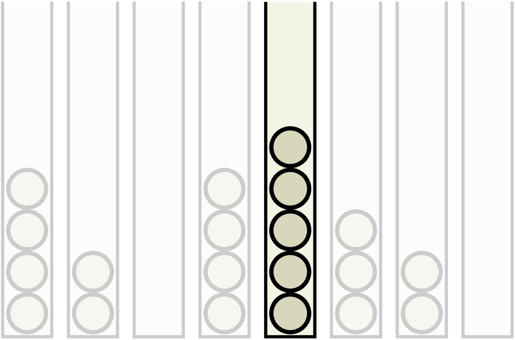
Balanced allocations setting

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

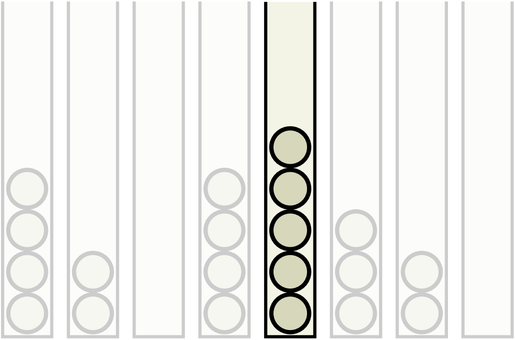


Balanced allocations setting

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

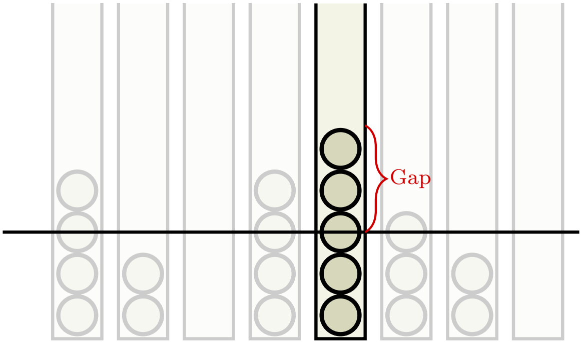
Goal: minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.

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



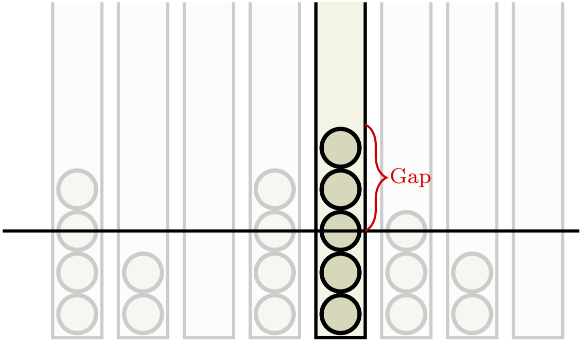
Balanced allocations setting

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



Balanced allocations setting

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



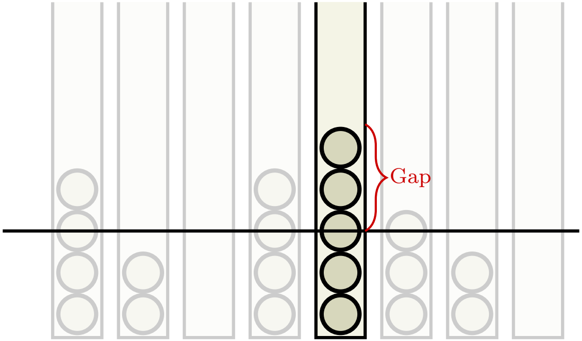
- Connections to

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.

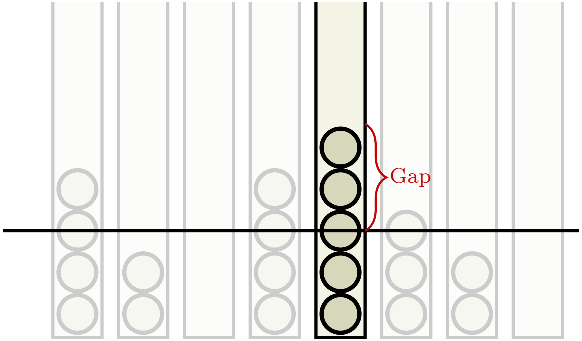
$\Leftrightarrow$ minimise the **gap**, where $\mathrm{Gap}(m) = \max_{i \in [n]}(x_i^m - m/n)$.



- Connections to *occupancy problems*,

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



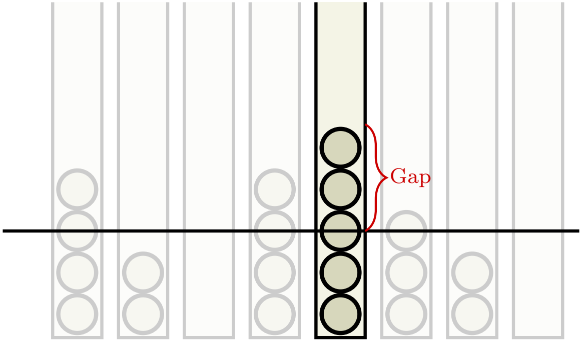
- Connections to *occupancy problems*, *urn processes*

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.

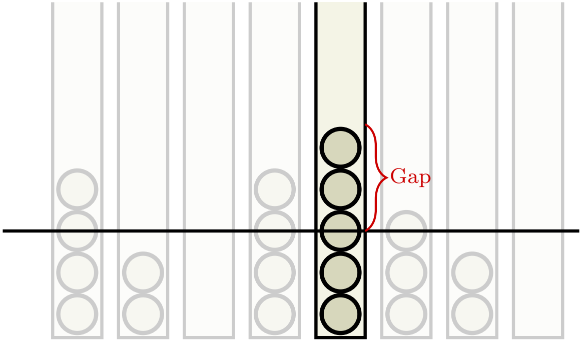
$\Leftrightarrow$ minimise the **gap**, where $\text{Gap}(m) = \max_{i \in [n]}(x_i^m - m/n)$.



- Connections to *occupancy problems*, *urn processes* and *queuing theory*.

Balanced allocations setting

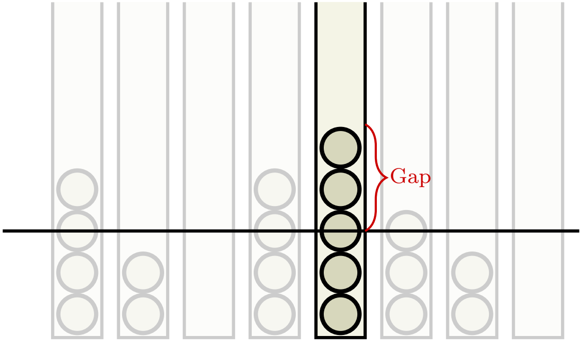
Allocate m tasks (balls) sequentially into n machines (bins).
Goal: minimise the **maximum load** $\max_{i\in[n]} x_i^m$, where x^t is the load vector after ball t.
$\Leftrightarrow$ minimise the **gap**, where $\mathrm{Gap}(m) = \max_{i\in[n]}(x_i^m - m/n)$.



- Connections to *occupancy problems*, *urn processes* and *queuing theory*. Applications in *hashing*,

Balanced allocations setting

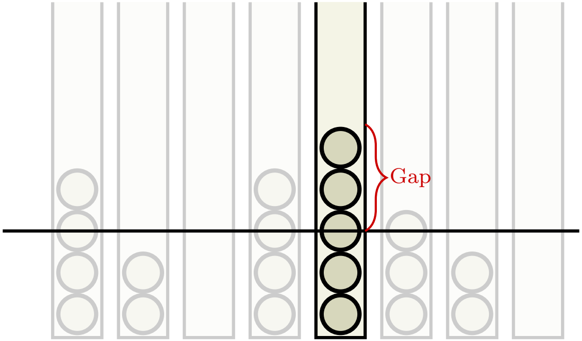
Allocate m tasks (balls) sequentially into n machines (bins).
<u>**Goal:**</u> minimise the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.
$\Leftrightarrow$ minimise the **gap**, where $\mathrm{Gap}(m) = \max_{i \in [n]}(x_i^m - m/n)$.



- Connections to *occupancy problems*, *urn processes* and *queuing theory*. Applications in *hashing*, *load balancing*

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



- Connections to *occupancy problems*, *urn processes* and *queuing theory*. Applications in *hashing*, *load balancing* and *routing*.

One-Choice and Two-Choice processes

One-Choice Process:

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

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

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

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

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

ONE-CHOICE and TWO-CHOICE processes

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

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

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

One-Choice and Two-Choice processes

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

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

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

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

One-Choice and Two-Choice processes

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

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

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

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

One-Choice and Two-Choice processes

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

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

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

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

One-Choice and Two-Choice processes

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

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

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

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

Is this a good process/model in practice?

Is this a good process/model in practice?

The practical significance of the "power of two choices" was recognised in the 2020 ACM Theory and Practice Award [ABK$^+$20]:

Is this a good process/model in practice?

The practical significance of the "power of two choices" was recognised in the 2020 ACM Theory and Practice Award [ABK$^+$20]:

> *"[...] it is not surprising that the power of two choices that requires only a local decision rather than global coordination has led to a wide range of practical applications. These include i-Google's web index, Akamai's overlay routing network, and highly reliable distributed data storage systems used by Microsoft and Dropbox, which are all based on variants of the power of two choices paradigm."*

Is this a good process/model in practice?

The practical significance of the "power of two choices" was recognised in the 2020 ACM Theory and Practice Award [ABK$^+$20]:

> *"[...] it is not surprising that the power of two choices that requires only a local decision rather than global coordination has led to a wide range of practical applications. These include i-Google's web index, Akamai's overlay routing network, and highly reliable distributed data storage systems used by Microsoft and Dropbox, which are all based on* ***variants*** *of the power of two choices paradigm."*

Is this a good process/model in practice?

The practical significance of the "power of two choices" was recognised in the 2020 ACM Theory and Practice Award [ABK+20]:

> *"[...] it is not surprising that the power of two choices that requires only a local decision rather than global coordination has led to a wide range of practical applications. These include i-Google's web index, Akamai's overlay routing network, and highly reliable distributed data storage systems used by Microsoft and Dropbox, which are all based on* ***variants*** *of the power of two choices paradigm."*

Question: Why *variants* and not vanilla TWO-CHOICE?

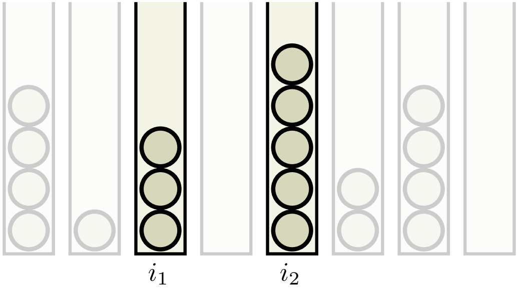
Motivation

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*.

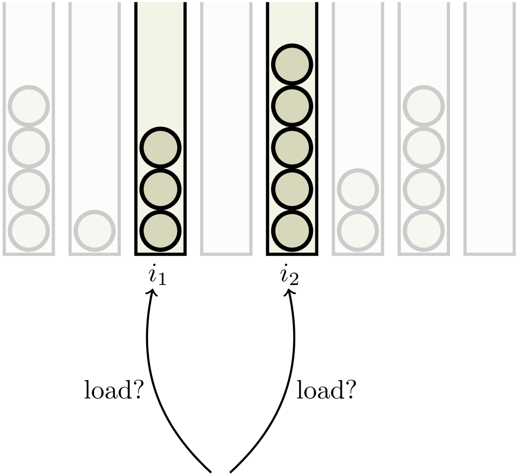
Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*.



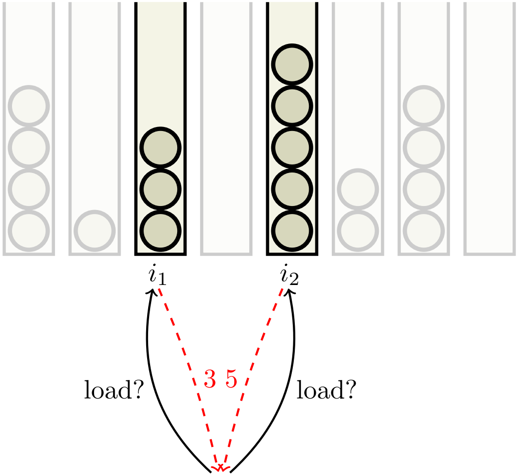
Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*.



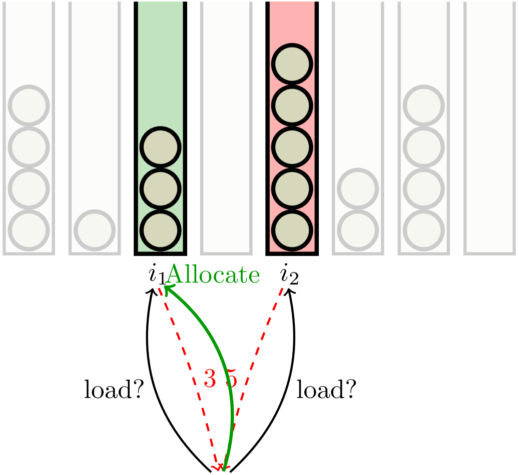
Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*.



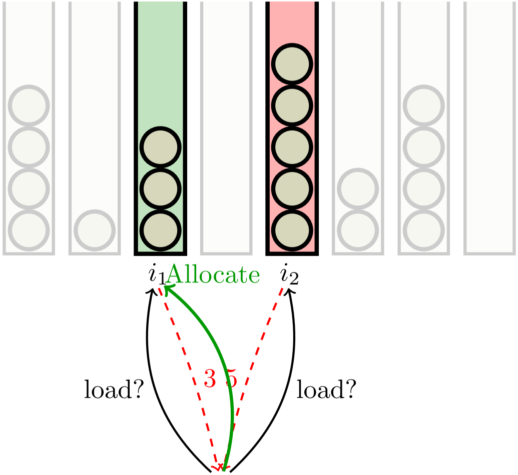
Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*.



Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon



Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of Two-Choice (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]).

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that TWO-CHOICE does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of TWO-CHOICE (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]). For `Sparrow` [OWZS13], they remark

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of Two-Choice (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]). For `Sparrow` [OWZS13], they remark

  > *The power of two choices suffers from two remaining performance problems: first, server queue length is a poor indicator of wait time, and second, due to messaging delays, multiple schedulers sampling in parallel may experience race conditions.*

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of Two-Choice (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]). For `Sparrow` [OWZS13], they remark

  > *The power of two choices suffers from two remaining performance problems: first, server queue length is a poor indicator of wait time, and second, due to messaging delays, multiple schedulers sampling in parallel may experience race conditions.*

- As noted in [LXK$^+$11], *communication* is a shortcoming of Two-Choice in some real-word load balancers:

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of Two-Choice (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]). For `Sparrow` [OWZS13], they remark

  > *The power of two choices suffers from two remaining performance problems: first, server queue length is a poor indicator of wait time, and second, due to messaging delays, multiple schedulers sampling in parallel may experience race conditions.*

- As noted in [LXK$^+$11], *communication* is a shortcoming of Two-Choice in some real-word load balancers:

  > *More importantly, the [Two-Choice] algorithm requires communication between dispatchers and processors at the time of job assignment. The communication time is on the critical path, hence contributes to the increase in response time.*

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of Two-Choice (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]). For `Sparrow` [OWZS13], they remark

  > *The power of two choices suffers from two remaining performance problems: first, server queue length is a poor indicator of wait time, and second, due to messaging delays, multiple schedulers sampling in parallel may experience race conditions.*

- As noted in [LXK$^+$11], *communication* is a shortcoming of Two-Choice in some real-word load balancers:

  > *More importantly, the [Two-Choice] algorithm requires communication between dispatchers and processors at the time of job assignment. The communication time is on the critical path, hence contributes to the increase in response time.*

- In the *queuing setting*, Whitt [Whi86] remarks:

Motivation

- Mitzenmacher [Mit00] and Dahlin [Dah00] empirically observed that Two-Choice does not perform well with *outdated information*. $\rightsquigarrow$ herd phenomenon
- Several *low-latency schedulers* use variants of Two-Choice (`Eagle` [DDDZ16], `Hawk` [DDKZ15], `Peacock` [KG18]). For `Sparrow` [OWZS13], they remark

  > *The power of two choices suffers from two remaining performance problems: first, server queue length is a poor indicator of wait time, and second, due to messaging delays, multiple schedulers sampling in parallel may experience race conditions.*

- As noted in [LXK+11], *communication* is a shortcoming of Two-Choice in some real-word load balancers:

  > *More importantly, the [Two-Choice] algorithm requires communication between dispatchers and processors at the time of job assignment. The communication time is on the critical path, hence contributes to the increase in response time.*

- In the *queuing setting*, Whitt [Whi86] remarks:

  > *We have shown that several natural selection rules are not optimal in various situations, but we have not identified any optimal rules. Identifying optimal rules in these situations would obviously be interesting, but appears to be difficult.*

An example of a variant of Two-Choice

$(1+\beta)$-Process

$(1+\beta)$-Process:

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

Iteration: For each $t \geq 0$, with probability β allocate one ball via the Two-Choice process, otherwise allocate one ball via the One-Choice process.

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for Two-Choice.

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for Two-Choice.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\beta}\right)$ for any $\beta \in (0, 1]$.

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for TWO-CHOICE.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\mathcal{O}\big(\frac{\log n}{\beta}\big)$ for any $\beta \in (0,1]$.
- It has been used to analyze

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for TWO-CHOICE.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\beta}\right)$ for any $\beta \in (0,1]$.
- It has been used to analyze *population protocols* [AAG18, AGR21],

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for Two-Choice.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\beta}\right)$ for any $\beta \in (0,1]$.
- It has been used to analyze *population protocols* [AAG18, AGR21], *distributed data structures* [ABK+18, AKLN17, Nad21]

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for Two-Choice.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\beta}\right)$ for any $\beta \in (0,1]$.
- It has been used to analyze *population protocols* [AAG18, AGR21], *distributed data structures* [ABK$^+$18, AKLN17, Nad21] and *online carpooling* [GKKS20].

$(1+\beta)$-Process

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

- Introduced by Mitzenmacher [Mit96] as a *faulty setting* for Two-Choice.
- In the heavily-loaded case, [PTW15] proved that the gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\beta}\right)$ for any $\beta \in (0,1]$.
- It has been used to analyze *population protocols* [AAG18, AGR21], *distributed data structures* [ABK$^+$18, AKLN17, Nad21] and *online carpooling* [GKKS20].

Question: Why choose a $\beta < 1$?

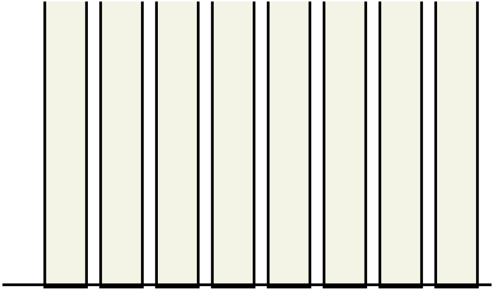
b-BATCHED **Setting**

b-Batched Setting

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

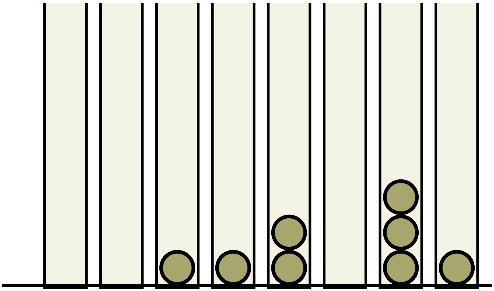
b-Batched Setting

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



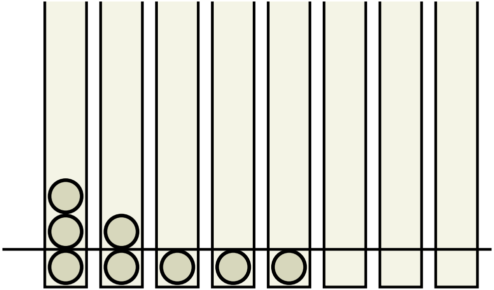
b-Batched Setting

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



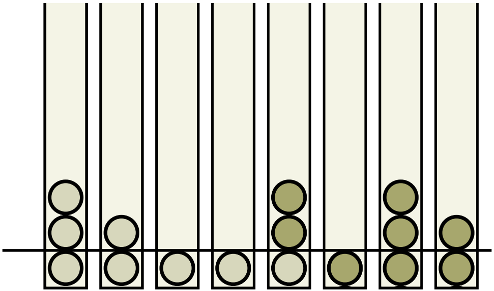
b-Batched Setting

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



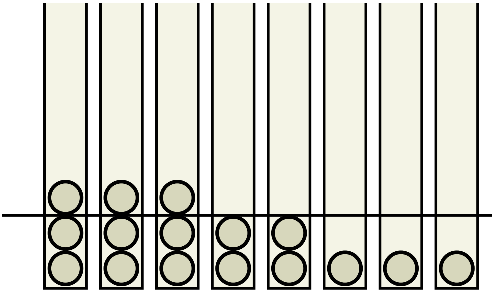
b-Batched Setting

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



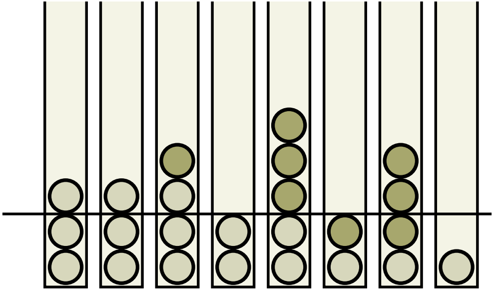
b-Batched Setting

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



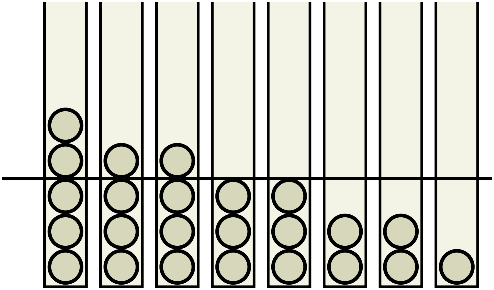
b-Batched Setting

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



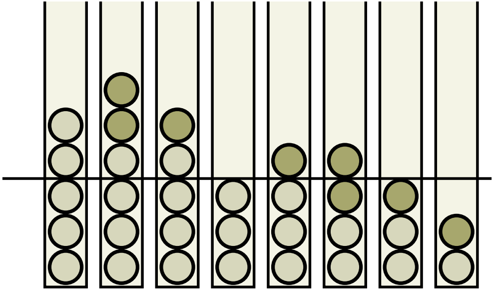
b-Batched Setting

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



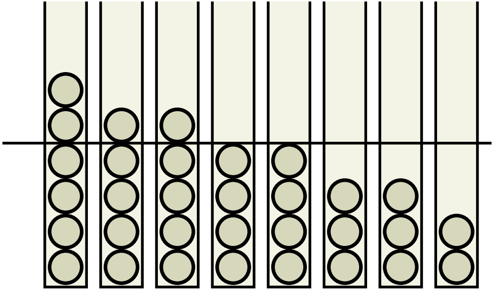
b-BATCHED Setting

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



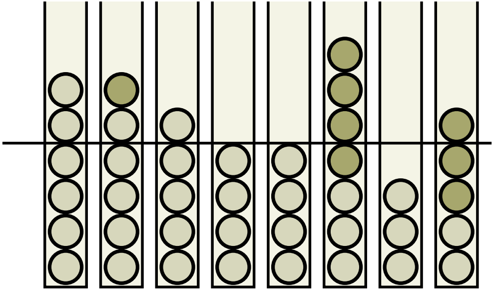
b-Batched Setting

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



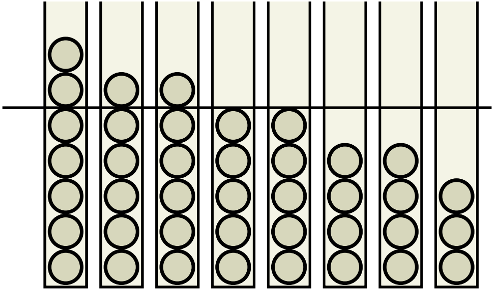
b-Batched Setting

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



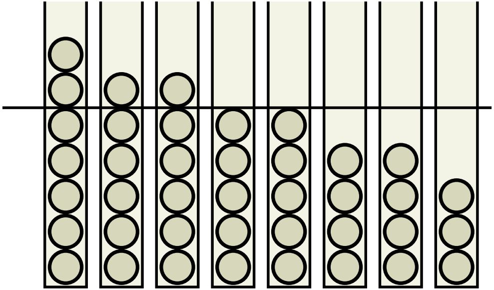
b-Batched Setting

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



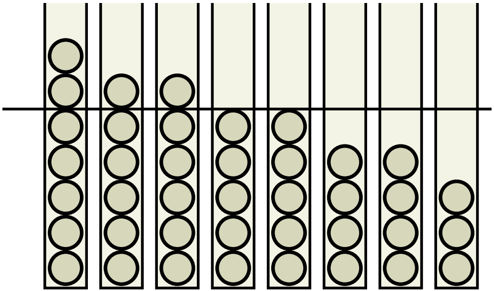
b-Batched Setting

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batched setting).
- For $b \geq n \log n$, we show that Two-Choice has w.h.p. $\mathrm{Gap}(m) = \Theta(\frac{b}{n})$.



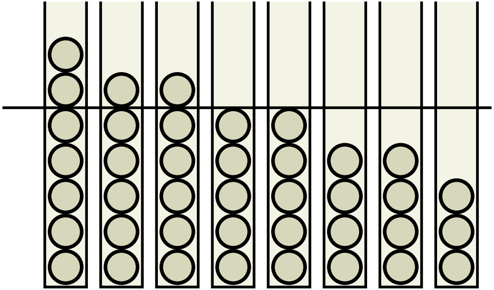
b-Batched Setting

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE$^+$12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batched setting).
- For $b \geq n \log n$, we show that Two-Choice has w.h.p. $\text{Gap}(m) = \Theta(\frac{b}{n})$.
- In constast to the case $b \ll n \log n$, where it is asymptotically optimal.



b-Batched Setting

- Berenbrink, Czumaj, Englert, Friedetzky and Nagel [BCE+12] studied Two-Choice where balls are allocated in *batches* of size b (b-Batched setting).
- For $b \geq n \log n$, we show that Two-Choice has w.h.p. $\mathrm{Gap}(m) = \Theta(\frac{b}{n})$.
- In constast to the case $b \ll n \log n$, where it is asymptotically optimal.
- For $b \geq n \log n$, we show that the $(1+\beta)$-process has w.h.p. $\mathrm{Gap}(m) = \Theta\left(\sqrt{\frac{b}{n} \cdot \log n}\right)$, for $\beta = \Theta(\sqrt{(n/b) \cdot \log n})$.



Probability allocation vectors

Probability allocation vectors

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

Probability allocation vectors

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

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

Probability allocation vectors

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

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

- For Two-Choice,

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

Probability allocation vectors

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

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

- For Two-Choice,

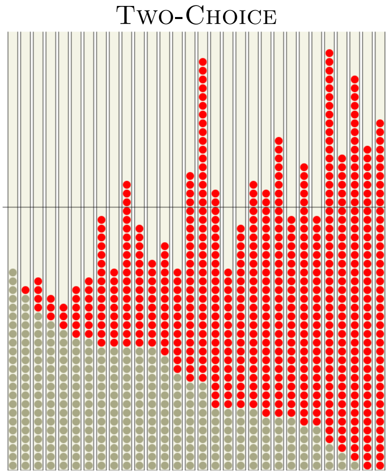
$$p_{\textsc{Two-Choice}} = \Big(\frac{1}{n^2}, \frac{3}{n^2}, \ldots, \frac{2i-1}{n^2}, \ldots, \frac{2n-2}{n^2}\Big).$$

- For $(1+\beta)$-process,

$$p_{(1+\beta)} = \Big(\ldots, \beta \cdot \frac{2i-1}{n^2} + (1-\beta) \cdot \frac{1}{n}, \ldots\Big).$$

A closer look at a single batch

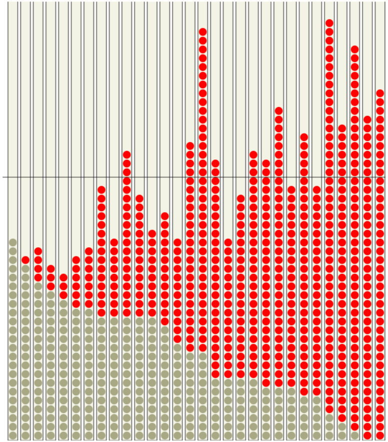
A closer look at a single batch



$$p_i = \frac{2i - 1}{n^2}$$

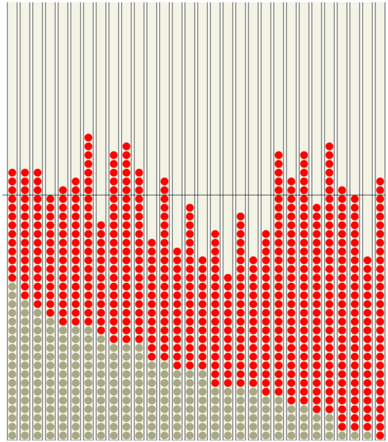
A closer look at a single batch

Two-Choice



$$p_i = \frac{2i-1}{n^2}$$

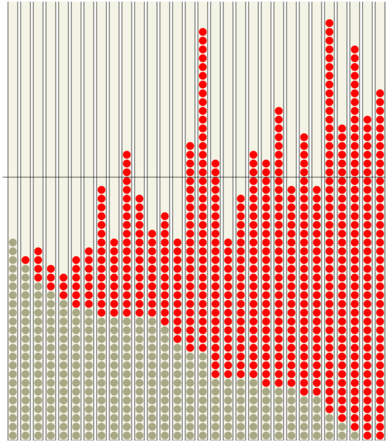
$(1+\beta)$-Process



$$p_i = \beta \cdot \frac{2i-1}{n^2} + (1-\beta) \cdot \frac{1}{n}$$

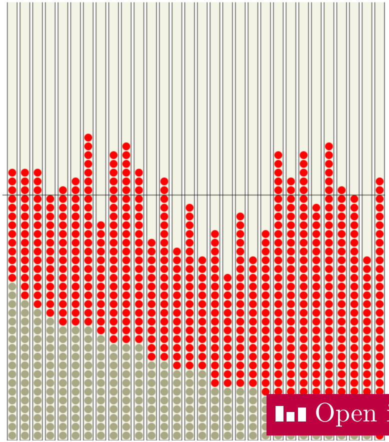
A closer look at a single batch

Two-Choice



$$p_i = \frac{2i-1}{n^2}$$

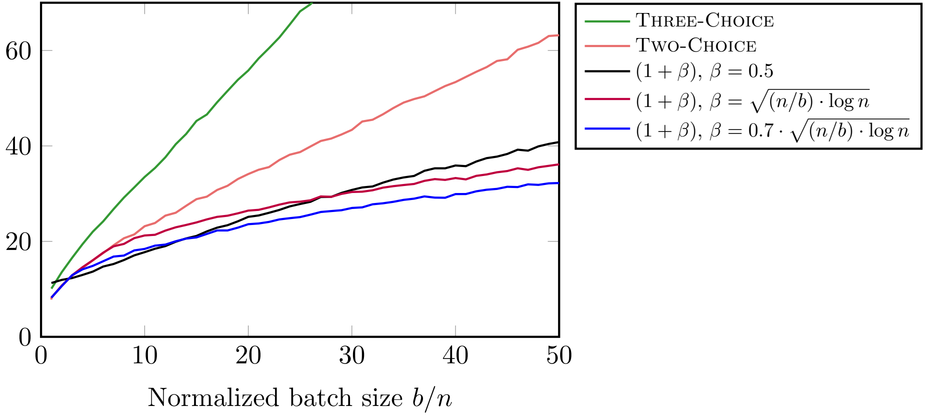
$(1+\beta)$-Process



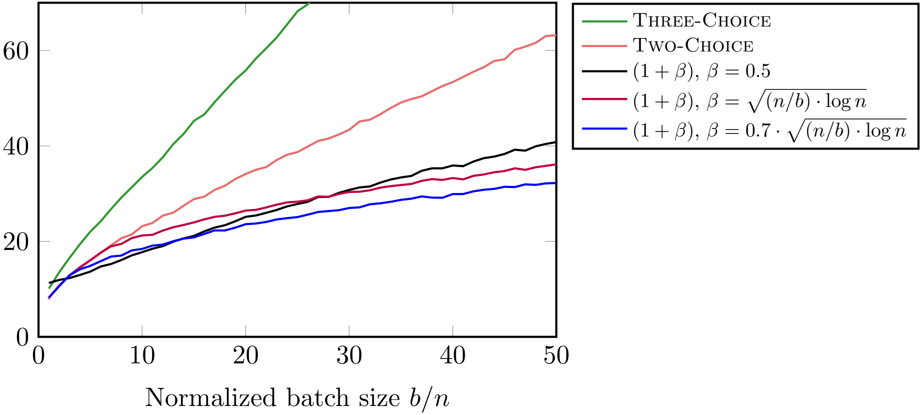
Open in Visualiser.

$$p_i = \beta \cdot \frac{2i-1}{n^2} + (1-\beta) \cdot \frac{1}{n}$$

Empirical results for different processes

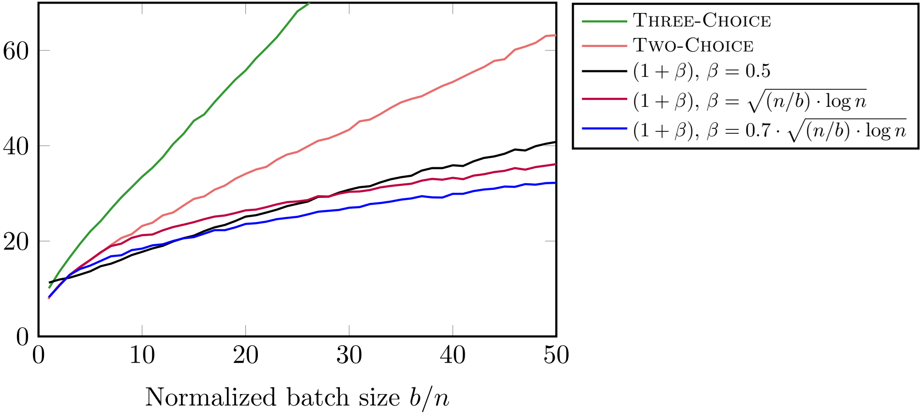


Empirical results for different processes



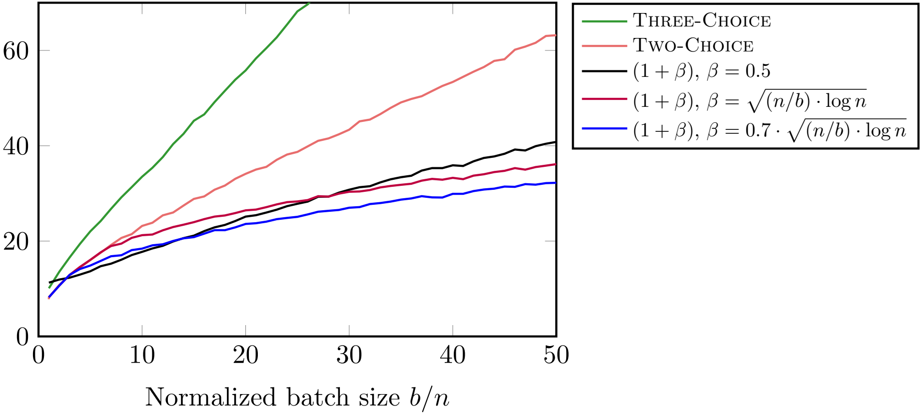
- The gaps are decreasingly ordered by p_n: $\approx \frac{3}{n}$ (for Three-Choice),

Empirical results for different processes



- The gaps are decreasingly ordered by p_n: $\approx \frac{3}{n}$ (for Three-Choice), $\approx \frac{2}{n}$ (for Two-Choice)

Empirical results for different processes

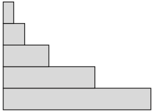


- The gaps are decreasingly ordered by p_n: $\approx \frac{3}{n}$ (for Three-Choice), $\approx \frac{2}{n}$ (for Two-Choice) and $\approx \frac{1+\beta}{n}$ (for the $(1+\beta)$-processes).

Potential functions

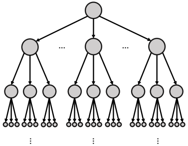
Techniques for analyzing balanced allocations

Layered induction



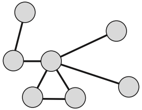
Two-Choice, Memory

Witness trees



Two-Choice, parallel allocations

Graphical processes



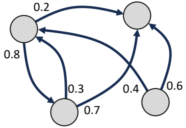
Two-Choice

Poissonisation

$$X_i \sim \mathsf{Poi}(\tfrac{m}{n})$$

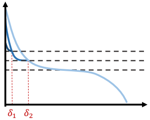
Unweighted, time-independent

Markov chains



Some weights, b-Batched, heterogeneous sampling

Potential functions



weights, b-Batched, outdated info, noise graphical, heterogeneous sampling

Processes

Processes

- Following [PTW15] we will study ϵ-biased processes that:

Processes

- Following [PTW15] we will study ϵ-biased processes that:
  - Have p is *non-decreasing*,

Processes

- Following [PTW15] we will study ϵ-biased processes that:
  - Have p is *non-decreasing*,
  - For some constant $\delta \in (0, 1)$, satisfy

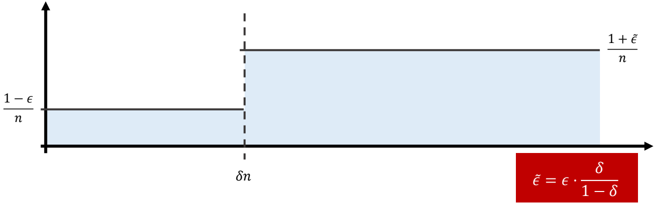
$$p_{\delta n} \leq \frac{1-\epsilon}{n}.$$

Processes

- Following [PTW15] we will study ϵ-biased processes that:
  - Have p is *non-decreasing*,
  - For some constant $\delta \in (0,1)$, satisfy

$$p_{\delta n} \leq \frac{1-\epsilon}{n}.$$

- **Key Observation 1:** We can assume the following *worst-case* allocation vector

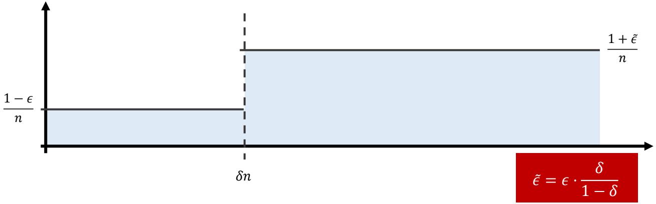


Processes

- Following [PTW15] we will study ϵ-biased processes that:
  - Have p is *non-decreasing*,
  - For some constant $\delta \in (0,1)$, satisfy

$$p_{\delta n} \leq \frac{1-\epsilon}{n}.$$

- **Key Observation 1:** We can assume the following *worst-case* allocation vector



- Our main aim will be to derive the w.h.p. $\mathcal{O}((\log n)/\epsilon)$ gap, for any $\epsilon \in (0,1)$.

The exponential potential

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as
$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)}$$

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as
$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as

$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$

- Our goal is to show that $\Phi^t = \mathcal{O}(\mathrm{poly}(n))$,

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as

$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$

- Our goal is to show that $\Phi^t = \mathcal{O}(\mathrm{poly}(n))$, as it implies that

$$y_i^t \leq \frac{1}{\alpha} \cdot \log \Phi^t$$

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as
$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$

- Our goal is to show that $\Phi^t = \mathcal{O}(\mathrm{poly}(n))$, as it implies that
$$y_i^t \leq \frac{1}{\alpha} \cdot \log \Phi^t = \mathcal{O}\left(\frac{1}{\alpha} \cdot \log n\right).$$

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as
$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$
- Our goal is to show that $\Phi^t = \mathcal{O}(\mathrm{poly}(n))$, as it implies that
$$y_i^t \leq \frac{1}{\alpha} \cdot \log \Phi^t = \mathcal{O}\left(\frac{1}{\alpha} \cdot \log n\right).$$
- Even better, if we show that $\Phi^t = \mathcal{O}(n)$,

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as
$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$

- Our goal is to show that $\Phi^t = \mathcal{O}(\mathrm{poly}(n))$, as it implies that
$$y_i^t \leq \frac{1}{\alpha} \cdot \log \Phi^t = \mathcal{O}\left(\frac{1}{\alpha} \cdot \log n\right).$$

- Even better, if we show that $\Phi^t = \mathcal{O}(n)$, then the bins with load $\geq t/n + z$ is at most $\mathcal{O}(n \cdot e^{-\alpha z})$.

The exponential potential

- At step $t \geq 0$, the *exponential potential* with smoothing parameter $\alpha > 0$ is defined as
$$\Phi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t}.$$
- Our goal is to show that $\Phi^t = \mathcal{O}(\mathrm{poly}(n))$, as it implies that
$$y_i^t \leq \frac{1}{\alpha} \cdot \log \Phi^t = \mathcal{O}\left(\frac{1}{\alpha} \cdot \log n\right).$$
- Even better, if we show that $\Phi^t = \mathcal{O}(n)$, then the bins with load $\geq t/n + z$ is at most $\mathcal{O}(n \cdot e^{-\alpha z})$.

Question: How can we prove this?

What is a drift theorem?

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\, \Phi^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\, \Phi^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form

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

  for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\, \Phi^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,
$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\, \Phi^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,
$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$
- It also implies that $\mathbf{E}\left[\, \Phi^t \,\right] \leq \frac{c_2}{c_1} \cdot n$,

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form

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

  for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,

$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$

- It also implies that $\mathbf{E}\left[\, \Phi^t \,\right] \leq \frac{c_2}{c_1} \cdot n$, since by induction,

$$\mathbf{E}\left[\, \Phi^{t+1} \,\right] = \mathbf{E}\left[\, \mathbf{E}\left[\, \Phi^{t+1} \mid \Phi^t \,\right] \,\right]$$

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\, \Phi^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,
$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$
- It also implies that $\mathbf{E}\left[\, \Phi^t \,\right] \leq \frac{c_2}{c_1} \cdot n$, since by induction,
$$\mathbf{E}\left[\, \Phi^{t+1} \,\right] = \mathbf{E}\left[\, \mathbf{E}\left[\, \Phi^{t+1} \mid \Phi^t \,\right] \,\right] \leq \mathbf{E}\left[\, \Phi^t \,\right] \cdot \left(1 - \frac{c_1}{n}\right) + c_2$$

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\,\Phi^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,
$$\mathbf{E}\left[\,\Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n\,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$
- It also implies that $\mathbf{E}\left[\,\Phi^t\,\right] \leq \frac{c_2}{c_1} \cdot n$, since by induction,
$$\mathbf{E}\left[\,\Phi^{t+1}\,\right] = \mathbf{E}\left[\,\mathbf{E}\left[\,\Phi^{t+1} \mid \Phi^t\,\right]\,\right] \leq \mathbf{E}\left[\,\Phi^t\,\right] \cdot \left(1 - \frac{c_1}{n}\right) + c_2 \leq \frac{c_2}{c_1} \cdot n - c_2 + c_2$$

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\, \Phi^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,
$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n \,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$
- It also implies that $\mathbf{E}\left[\, \Phi^t \,\right] \leq \frac{c_2}{c_1} \cdot n$, since by induction,
$$\mathbf{E}\left[\, \Phi^{t+1} \,\right] = \mathbf{E}\left[\, \mathbf{E}\left[\, \Phi^{t+1} \mid \Phi^t \,\right] \,\right] \leq \mathbf{E}\left[\, \Phi^t \,\right] \cdot \left(1 - \frac{c_1}{n}\right) + c_2 \leq \frac{c_2}{c_1} \cdot n - c_2 + c_2 = \frac{c_2}{c_1} \cdot n.$$

What is a drift theorem?

- A *drift theorem* (or *drift inequality*) has the form
$$\mathbf{E}\left[\,\Phi^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{n}\right) + c_2,$$
for any possible history of the process $\mathfrak{F}^t$ (or filtration).
- When Φ^t is *large* (i.e., $\Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n$), then it drops in expectation by a multiplicative factor,
$$\mathbf{E}\left[\,\Phi^{t+1} \,\middle|\, \mathfrak{F}^t, \Phi^t \geq 2 \cdot \frac{c_2}{c_1} \cdot n\,\right] \leq \Phi^t \cdot \left(1 - \frac{c_1}{2n}\right),$$
- It also implies that $\mathbf{E}\left[\,\Phi^t\,\right] \leq \frac{c_2}{c_1} \cdot n$, since by induction,
$$\mathbf{E}\left[\,\Phi^{t+1}\,\right] = \mathbf{E}\left[\,\mathbf{E}\left[\,\Phi^{t+1} \mid \Phi^t\,\right]\,\right] \leq \mathbf{E}\left[\,\Phi^t\,\right] \cdot \left(1 - \frac{c_1}{n}\right) + c_2 \leq \frac{c_2}{c_1} \cdot n - c_2 + c_2 = \frac{c_2}{c_1} \cdot n.$$
- Then, applying *Markov's inequality* we get that w.h.p. $\mathbf{E}\left[\,\Phi^t\,\right] = \text{poly}(n)$.

Deriving a drift theorem

Deriving a drift theorem

- Let us fix a bin $i \in [n]$.

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\mathbf{E}\left[\, \Phi_i^{t+1} \mid \Phi_i^t \,\right] = p_i \cdot e^{\alpha \cdot (y_i^t + 1 - 1/n)} + (1 - p_i) \cdot e^{\alpha \cdot (y_i^t - 1/n)}$$

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\,\Phi_i^{t+1} \mid \Phi_i^t\,\right] &= p_i \cdot e^{\alpha\cdot(y_i^t+1-1/n)} + (1-p_i)\cdot e^{\alpha\cdot(y_i^t-1/n)} \\ &\le \Phi_i^t \cdot \Big(p_i \cdot \Big(1+\alpha\cdot(1-1/n)+\alpha^2\Big) + (1-p_i)\cdot\Big(1-\alpha/n+\alpha^2/n^2\Big)\Big)\end{aligned}$$

using the Taylor estimate $e^z \le 1+z+z^2$ for sufficiently small z.

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\, \Phi_i^{t+1} \mid \Phi_i^t \,\right] &= p_i \cdot e^{\alpha \cdot (y_i^t + 1 - 1/n)} + (1 - p_i) \cdot e^{\alpha \cdot (y_i^t - 1/n)} \\ &\leq \Phi_i^t \cdot \left(p_i \cdot \left(1 + \alpha \cdot (1 - 1/n) + \alpha^2\right) + (1 - p_i) \cdot \left(1 - \alpha/n + \alpha^2/n^2\right)\right)\end{aligned}$$

using the Taylor estimate $e^z \leq 1 + z + z^2$ for sufficiently small z.

Applies also to weights $\mathcal{W}$ with unit expectation and finite MGF, i.e., $e^{\alpha \mathcal{W}} \leq 1 + \alpha + \alpha^2 \cdot S$.

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\, \Phi_i^{t+1} \mid \Phi_i^t \,\right] &= p_i \cdot e^{\alpha \cdot (y_i^t + 1 - 1/n)} + (1 - p_i) \cdot e^{\alpha \cdot (y_i^t - 1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1 + \alpha \cdot (1 - 1/n) + \alpha^2\big) + (1 - p_i) \cdot \big(1 - \alpha/n + \alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1 + \alpha \cdot \left(p_i - \frac{1}{n}\right) + \alpha^2 \cdot \left(p_i + \frac{1 - p_i}{n}\right)\right),\end{aligned}$$

using the Taylor estimate $e^z \leq 1 + z + z^2$ for sufficiently small z.

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\, \Phi_i^{t+1} \mid \Phi_i^t \,\right] &= p_i \cdot e^{\alpha \cdot (y_i^t + 1 - 1/n)} + (1 - p_i) \cdot e^{\alpha \cdot (y_i^t - 1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1 + \alpha \cdot (1 - 1/n) + \alpha^2\big) + (1 - p_i) \cdot \big(1 - \alpha/n + \alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1 + \alpha \cdot \left(p_i - \frac{1}{n}\right) + \alpha^2 \cdot \left(p_i + \frac{1 - p_i}{n}\right)\right),\end{aligned}$$

using the Taylor estimate $e^z \leq 1 + z + z^2$ for sufficiently small z.

- For bins with $p_i = \frac{1-\epsilon}{n}$,

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\, \Phi_i^{t+1} \mid \Phi_i^t \,\right] &= p_i \cdot e^{\alpha \cdot (y_i^t + 1 - 1/n)} + (1 - p_i) \cdot e^{\alpha \cdot (y_i^t - 1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1 + \alpha \cdot (1 - 1/n) + \alpha^2\big) + (1 - p_i) \cdot \big(1 - \alpha/n + \alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1 + \alpha \cdot \left(p_i - \frac{1}{n}\right) + \alpha^2 \cdot \left(p_i + \frac{1 - p_i}{n}\right)\right),\end{aligned}$$

  using the Taylor estimate $e^z \leq 1 + z + z^2$ for sufficiently small z.
- For bins with $p_i = \frac{1-\epsilon}{n}$, we have that ($\Delta\Phi_i^{t+1} := \Phi_i^{t+1} - \Phi_i^t$)

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(-\frac{\alpha\epsilon}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right)$$

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\,\Phi_i^{t+1} \mid \Phi_i^t\,\right] &= p_i \cdot e^{\alpha\cdot(y_i^t+1-1/n)} + (1-p_i)\cdot e^{\alpha\cdot(y_i^t-1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1+\alpha\cdot(1-1/n)+\alpha^2\big) + (1-p_i)\cdot\big(1-\alpha/n+\alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1 + \alpha\cdot\left(p_i - \frac{1}{n}\right) + \alpha^2\cdot\left(p_i + \frac{1-p_i}{n}\right)\right),\end{aligned}$$

  using the Taylor estimate $e^z \leq 1 + z + z^2$ for sufficiently small z.

- For bins with $p_i = \frac{1-\epsilon}{n}$, we have that $(\Delta\Phi_i^{t+1} := \Phi_i^{t+1} - \Phi_i^t)$

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(-\frac{\alpha\epsilon}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Good bin.}$$

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\,\Phi_i^{t+1} \mid \Phi_i^t\,\right] &= p_i \cdot e^{\alpha\cdot(y_i^t+1-1/n)} + (1-p_i)\cdot e^{\alpha\cdot(y_i^t-1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1+\alpha\cdot(1-1/n)+\alpha^2\big) + (1-p_i)\cdot\big(1-\alpha/n+\alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1+\alpha\cdot\left(p_i - \frac{1}{n}\right) + \alpha^2\cdot\left(p_i + \frac{1-p_i}{n}\right)\right),\end{aligned}$$

  using the Taylor estimate $e^z \leq 1+z+z^2$ for sufficiently small z.

- For bins with $p_i = \frac{1-\epsilon}{n}$, we have that ($\Delta\Phi_i^{t+1} := \Phi_i^{t+1} - \Phi_i^t$)

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(-\frac{\alpha\epsilon}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Good bin.}$$

- Otherwise, for bins with $p_i = \frac{1+\widetilde{\epsilon}}{n}$,

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\,\Phi_i^{t+1} \mid \Phi_i^t\,\right] &= p_i \cdot e^{\alpha\cdot(y_i^t+1-1/n)} + (1-p_i)\cdot e^{\alpha\cdot(y_i^t-1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1+\alpha\cdot(1-1/n)+\alpha^2\big) + (1-p_i)\cdot\big(1-\alpha/n+\alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1+\alpha\cdot\left(p_i-\frac{1}{n}\right)+\alpha^2\cdot\left(p_i+\frac{1-p_i}{n}\right)\right),\end{aligned}$$

  using the Taylor estimate $e^z \leq 1+z+z^2$ for sufficiently small z.

- For bins with $p_i = \frac{1-\epsilon}{n}$, we have that $(\Delta\Phi_i^{t+1} := \Phi_i^{t+1} - \Phi_i^t)$

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(-\frac{\alpha\epsilon}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Good bin.}$$

- Otherwise, for bins with $p_i = \frac{1+\widetilde{\epsilon}}{n}$,

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(+\frac{\alpha\widetilde{\epsilon}}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right)$$

Deriving a drift theorem

- Let us fix a bin $i \in [n]$. Then,

$$\begin{aligned}\mathbf{E}\left[\,\Phi_i^{t+1} \mid \Phi_i^t\,\right] &= p_i \cdot e^{\alpha\cdot(y_i^t+1-1/n)} + (1-p_i)\cdot e^{\alpha\cdot(y_i^t-1/n)} \\ &\leq \Phi_i^t \cdot \big(p_i \cdot \big(1+\alpha\cdot(1-1/n)+\alpha^2\big) + (1-p_i)\cdot\big(1-\alpha/n+\alpha^2/n^2\big)\big) \\ &= \Phi_i^t \cdot \left(1+\alpha\cdot\left(p_i - \frac{1}{n}\right) + \alpha^2\cdot\left(p_i + \frac{1-p_i}{n}\right)\right),\end{aligned}$$

  using the Taylor estimate $e^z \leq 1+z+z^2$ for sufficiently small z.

- For bins with $p_i = \frac{1-\epsilon}{n}$, we have that $(\Delta\Phi_i^{t+1} := \Phi_i^{t+1} - \Phi_i^t)$

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(-\frac{\alpha\epsilon}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Good bin.}$$

- Otherwise, for bins with $p_i = \frac{1+\widetilde{\epsilon}}{n}$,

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(+\frac{\alpha\widetilde{\epsilon}}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Bad bin.}$$

What could go wrong?

What could go wrong?

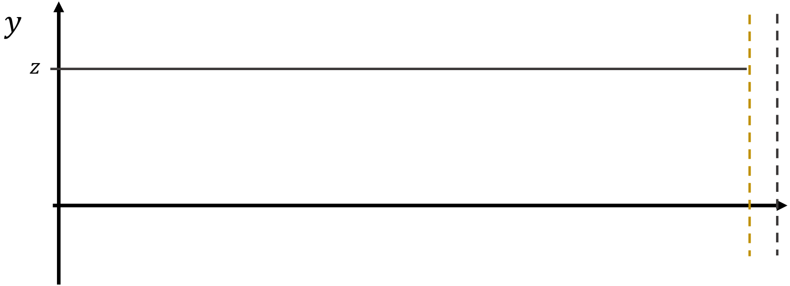
- There could be *too many overloaded bins*.

What could go wrong?

- There could be *too many overloaded bins*.
- Consider a step $t \geq 0$, where all but one bins have the same load:

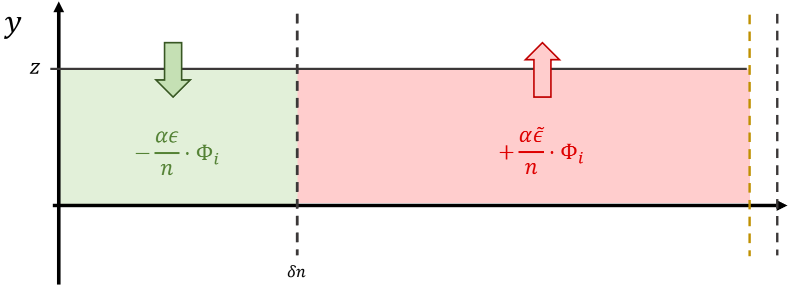
What could go wrong?

- There could be *too many overloaded bins*.
- Consider a step $t \geq 0$, where all but one bins have the same load:



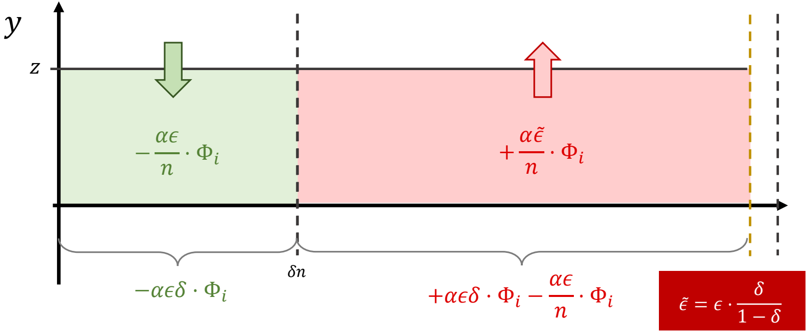
What could go wrong?

- There could be *too many overloaded bins*.
- Consider a step $t \geq 0$, where all but one bins have the same load:



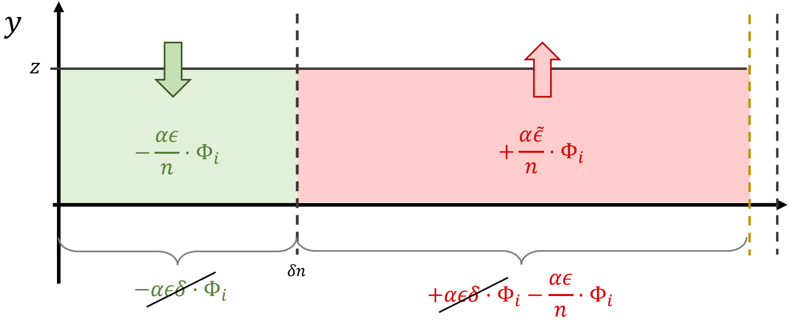
What could go wrong?

- There could be *too many overloaded bins.*
- Consider a step $t \geq 0$, where all but one bins have the same load:



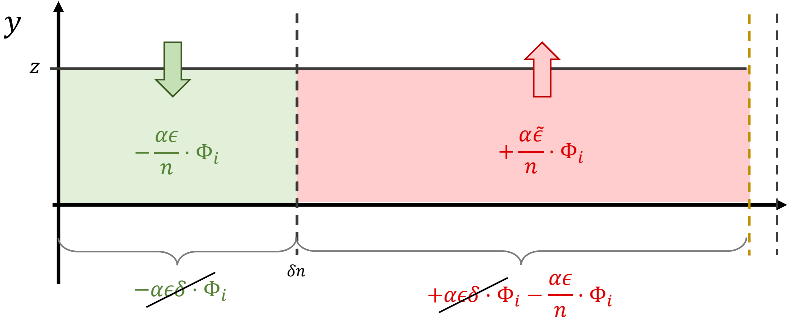
What could go wrong?

- There could be *too many overloaded bins*.
- Consider a step $t \geq 0$, where all but one bins have the same load:



What could go wrong?

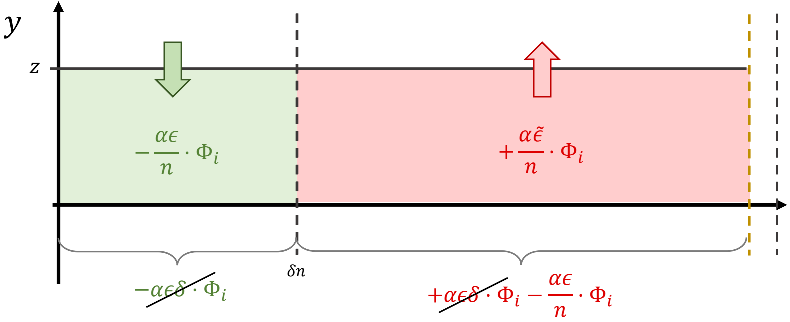
- There could be *too many overloaded bins.*
- Consider a step $t \geq 0$, where all but one bins have the same load:



- We could get only a *very small* decrease.

What could go wrong?

- There could be *too many overloaded bins*.
- Consider a step $t \geq 0$, where all but one bins have the same load:



- We could get only a *very small* decrease. $\rightsquigarrow$ Gives $\mathcal{O}(n \log n/\epsilon)$ bound on the gap.

The hyperbolic cosine potential

The hyperbolic cosine potential

- The *hyperbolic cosine potential* [PTW15, Spe77] is defined as

$$\Gamma^t := \Phi^t + \Psi^t := \sum_{i=1}^{n} e^{\alpha\cdot(x_i^t - t/n)} + e^{-\alpha\cdot(x_i^t - t/n)}$$

The hyperbolic cosine potential

- The *hyperbolic cosine potential* [PTW15, Spe77] is defined as

$$\Gamma^t := \Phi^t + \Psi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} + e^{-\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t} + e^{-\alpha \cdot y_i^t}.$$

The hyperbolic cosine potential

- The *hyperbolic cosine potential* [PTW15, Spe77] is defined as

$$\Gamma^t := \Phi^t + \Psi^t := \sum_{i=1}^{n} e^{\alpha \cdot (x_i^t - t/n)} + e^{-\alpha \cdot (x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha \cdot y_i^t} + e^{-\alpha \cdot y_i^t}.$$

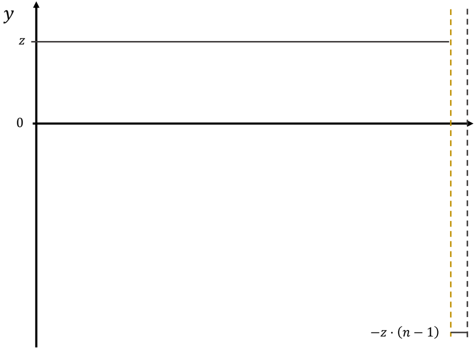
Question: Why does the *second term* help?

The hyperbolic cosine potential

- The *hyperbolic cosine potential* [PTW15, Spe77] is defined as

$$\Gamma^t := \Phi^t + \Psi^t := \sum_{i=1}^{n} e^{\alpha\cdot(x_i^t - t/n)} + e^{-\alpha\cdot(x_i^t - t/n)} = \sum_{i=1}^{n} e^{\alpha\cdot y_i^t} + e^{-\alpha\cdot y_i^t}.$$

Question: Why does the *second term* help?



The underload potential Ψ

The underload potential Ψ

- For any bin $i \in [n]$, the *underload potential* satisfies the following drift inequality,

The underload potential Ψ

- For any bin $i \in [n]$, the *underload potential* satisfies the following drift inequality,

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

The underload potential Ψ

- For any bin $i \in [n]$, the *underload potential* satisfies the following drift inequality,

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

- So, the dominant term is the *negative* of that in Φ.

The underload potential Ψ

- For any bin $i \in [n]$, the *underload potential* satisfies the following drift inequality,

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

- So, the dominant term is the *negative* of that in Φ.
- More specifically, if $p_i = \frac{1+\widetilde{\epsilon}}{n}$, then

$$\mathbf{E}\left[\,\Delta\Psi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Psi_i^t \cdot \left(-\frac{\alpha\widetilde{\epsilon}}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Good bin.}$$

The underload potential Ψ

- For any bin $i \in [n]$, the *underload potential* satisfies the following drift inequality,
$$\mathbf{E}\left[\, \Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(1 + \alpha \cdot \left(\frac{1}{n} - p_i\right) + \alpha^2 \cdot \left(p_i + (1 - p_i) \cdot \frac{1}{n^2}\right)\right).$$

- So, the dominant term is the *negative* of that in Φ.
- More specifically, if $p_i = \frac{1+\widetilde{\epsilon}}{n}$, then
$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\frac{\alpha\widetilde{\epsilon}}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Good bin.}$$

- Otherwise, if $p_i = \frac{1-\epsilon}{n}$
$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(+\frac{\alpha\epsilon}{n} + \mathcal{O}\left(\frac{\alpha^2}{n}\right)\right) \rightsquigarrow \text{Bad bin.}$$

Drift Theorem

Theorem ([PTW15, Section 2])

Consider any process with *non-decreasing* allocation vector p which is ϵ-biased for some $\epsilon \in (0,1)$ and some constant δ, in the setting with weights sampled from a distribution with finite MGF. Then, for $\Gamma := \Gamma(\alpha)$ with $\alpha := \Theta(\epsilon)$, for any step $t \geq 0$,

$$\mathbf{E}\left[\, \Delta\Gamma^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq -\Gamma^t \cdot \frac{\alpha\epsilon}{4n} + \mathrm{poly}(1/\epsilon),$$

and

$$\mathbf{E}\left[\, \Gamma^t \,\right] \leq n \cdot \mathrm{poly}(1/\epsilon).$$

Refined Drift Theorem

Theorem ([LS22, Corollary 3.2])

Consider any process and a *probability vector* p being ϵ-biased for some $\epsilon \in (0,1)$ and some constant δ. Further assume that it satisfies for some $K > 0$ and for any $t \geq 0$,

$$\mathbf{E}\left[\,\Phi^{t+1} \mid \mathfrak{F}^t\,\right] \leq \sum_{i=1}^{n} \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot \alpha + K \cdot \frac{\alpha^2}{n}\right),$$

and

$$\mathbf{E}\left[\,\Psi^{t+1} \mid \mathfrak{F}^t\,\right] \leq \sum_{i=1}^{n} \Psi_i^t \cdot \left(1 + \left(\frac{1}{n} - p_i\right) \cdot \alpha + K \cdot \frac{\alpha^2}{n}\right).$$

Then, there exists a constant $c := c(\delta) > 0$, such that for $\alpha \in \left(0, \min\left\{1, \frac{\epsilon\delta}{8K}\right\}\right)$

$$\mathbf{E}\left[\,\Gamma^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Gamma^t \cdot \left(1 - \frac{\alpha\epsilon\delta}{8n}\right) + c\alpha\epsilon,$$

and

$$\mathbf{E}\left[\,\Gamma^t\,\right] \leq \frac{8c}{\delta} \cdot n.$$

Refined Drift Theorem

Theorem ([LS22, Corollary 3.2])

Consider any process and a probability vector p being ϵ-biased for some $\epsilon \in (0,1)$ and some constant δ. Further assume that it satisfies for some $K > 0$ and some $R > 0$, for any $t \geq 0$,

$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq \sum_{i=1}^{n} \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot R \cdot \alpha + K \cdot R \cdot \frac{\alpha^2}{n}\right),$$

and

$$\mathbf{E}\left[\, \Psi^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq \sum_{i=1}^{n} \Psi_i^t \cdot \left(1 + \left(\frac{1}{n} - p_i\right) \cdot R \cdot \alpha + K \cdot R \cdot \frac{\alpha^2}{n}\right).$$

Then, there exists a constant $c := c(\delta) > 0$, such that for $\alpha \in \left(0, \min\left\{1, \frac{\epsilon\delta}{8K}\right\}\right)$

$$\mathbf{E}\left[\, \Gamma^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq \Gamma^t \cdot R \cdot \left(1 - \frac{\alpha\epsilon\delta}{8n}\right) + R \cdot c\alpha\epsilon,$$

and

$$\mathbf{E}\left[\, \Gamma^t \,\right] \leq \frac{8c}{\delta} \cdot n.$$

Recap

Recap

- Our goal is to show that:

$$\sum_{i=1}^{n} \Phi_i^t \cdot \left(\alpha \cdot \left(p_i - \frac{1}{n}\right) + K \cdot \frac{\alpha^2}{n}\right) + \Psi_i^t \cdot \left(\alpha \cdot \left(\frac{1}{n} - p_i\right) + K \cdot \frac{\alpha^2}{n}\right)$$

Recap

- Our goal is to show that:

$$\sum_{i=1}^{n} \Phi_i^t \cdot \left(\alpha \cdot \left(p_i - \frac{1}{n}\right) + K \cdot \frac{\alpha^2}{n}\right) + \Psi_i^t \cdot \left(\alpha \cdot \left(\frac{1}{n} - p_i\right) + K \cdot \frac{\alpha^2}{n}\right)$$

$$\leq -\frac{\alpha\epsilon}{4n} \cdot \Gamma^t + K \cdot \frac{\alpha^2}{n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon)$$

Recap

- Our goal is to show that:

$$\begin{aligned}\sum_{i=1}^n \Phi_i^t \cdot \left(\alpha \cdot \left(p_i - \frac{1}{n}\right) + K \cdot \frac{\alpha^2}{n}\right) + \Psi_i^t \cdot \left(\alpha \cdot \left(\frac{1}{n} - p_i\right) + K \cdot \frac{\alpha^2}{n}\right) \\ \leq -\frac{\alpha\epsilon}{4n} \cdot \Gamma^t + K \cdot \frac{\alpha^2}{n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon) \\ \leq -\frac{\alpha\epsilon}{8n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon),\end{aligned}$$

for any $\alpha \leq \frac{\epsilon}{8K}$. (**Key Observation 2**)

Recap

- Our goal is to show that:

$$\sum_{i=1}^{n} \Phi_i^t \cdot \left(\alpha \cdot \left(p_i - \frac{1}{n}\right) + K \cdot \frac{\alpha^2}{n}\right) + \Psi_i^t \cdot \left(\alpha \cdot \left(\frac{1}{n} - p_i\right) + K \cdot \frac{\alpha^2}{n}\right)$$
$$\leq -\frac{\alpha\epsilon}{4n} \cdot \Gamma^t + K \cdot \frac{\alpha^2}{n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon)$$
$$\leq -\frac{\alpha\epsilon}{8n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon),$$

for any $\alpha \leq \frac{\epsilon}{8K}$. (**Key Observation 2**)

- We have the following types of bins

Set	Load	Index	r_i	Dominant Contribution
Good overloaded $\mathcal{G}_+$	$y_i \geq 0$	$i \leq \delta n$	$\frac{1-\epsilon}{n}$	$-\Phi_i \cdot \frac{\alpha\epsilon}{n} + \Psi_i \cdot \frac{\alpha\epsilon}{n}$
Bad overloaded $\mathcal{B}_+$	$y_i \geq 0$	$i > \delta n$	$\frac{1+\epsilon}{n}$	$+\Phi_i \cdot \frac{\alpha\epsilon}{n} - \Psi_i \cdot \frac{\alpha\epsilon}{n}$
Good underloaded $\mathcal{G}_-$	$y_i < 0$	$i > \delta n$	$\frac{1+\epsilon}{n}$	$+\Phi_i \cdot \frac{\alpha\epsilon}{n} - \Psi_i \cdot \frac{\alpha\epsilon}{n}$
Bad overloaded $\mathcal{B}_-$	$y_i < 0$	$i \leq \delta n$	$\frac{1-\epsilon}{n}$	$-\Phi_i \cdot \frac{\alpha\epsilon}{n} + \Psi_i \cdot \frac{\alpha\epsilon}{n}$

Recap

- Our goal is to show that:

$$\sum_{i=1}^{n} \Phi_i^t \cdot \left(\alpha \cdot \left(p_i - \frac{1}{n}\right) + K \cdot \frac{\alpha^2}{n}\right) + \Psi_i^t \cdot \left(\alpha \cdot \left(\frac{1}{n} - p_i\right) + K \cdot \frac{\alpha^2}{n}\right)$$

$$\leq -\frac{\alpha\epsilon}{4n} \cdot \Gamma^t + K \cdot \frac{\alpha^2}{n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon)$$

$$\leq -\frac{\alpha\epsilon}{8n} \cdot \Gamma^t + \mathcal{O}(\alpha\epsilon),$$

for any $\alpha \leq \frac{\epsilon}{8K}$. (**Key Observation 2**)

- We have the following types of bins

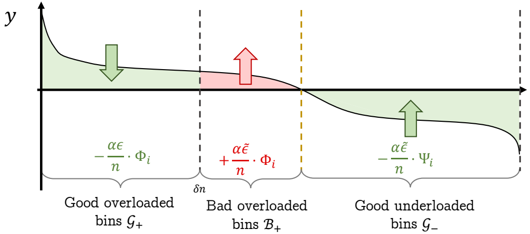
Set	Load	Index	r_i	Dominant Contribution
Good overloaded $\mathcal{G}_+$	$y_i \geq 0$	$i \leq \delta n$	$\frac{1-\epsilon}{\underset{\sim}{n}}$	$-\Phi_i \cdot \frac{\alpha\epsilon}{\underset{\sim}{n}} + \Psi_i \cdot \frac{\alpha\epsilon}{\underset{\sim}{n}}$
Bad overloaded $\mathcal{B}_+$	$y_i \geq 0$	$i > \delta n$	$\frac{1+\epsilon}{\underset{\sim}{n}}$	$+\Phi_i \cdot \frac{\alpha\epsilon}{\underset{\sim}{n}} - \Psi_i \cdot \frac{\alpha\epsilon}{\underset{\sim}{n}}$
Good underloaded $\mathcal{G}_-$	$y_i < 0$	$i > \delta n$	$\frac{1+\epsilon}{n}$	$+\Phi_i \cdot \frac{\alpha\epsilon}{n} - \Psi_i \cdot \frac{\alpha\epsilon}{n}$
Bad overloaded $\mathcal{B}_-$	$y_i < 0$	$i \leq \delta n$	$\frac{1-\epsilon}{n}$	$-\Phi_i \cdot \frac{\alpha\epsilon}{n} + \Psi_i \cdot \frac{\alpha\epsilon}{n}$

- **Key Observation 3:** For overloaded bins $\Psi_i^t \leq 1$ and for underloaded bins $\Phi_i^t \leq 1$,
$\rightsquigarrow$ their contribution is $\mathcal{O}(\alpha\epsilon)$.

The two general cases of bad bins

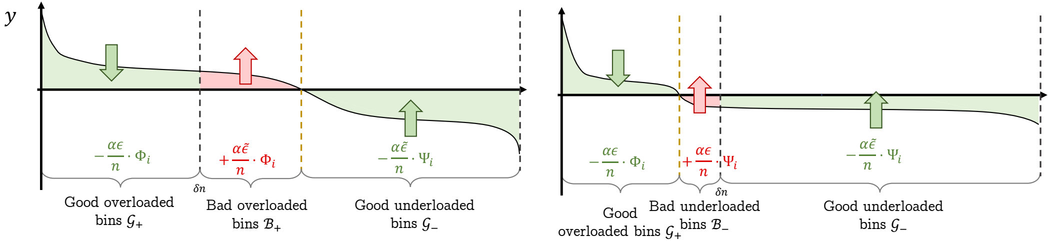
The two general cases of bad bins

- There can either be *overloaded bad bins* .



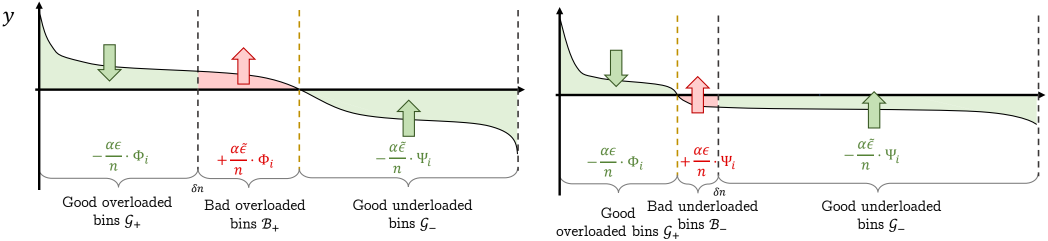
The two general cases of bad bins

- There can either be *overloaded bad bins* or *underloaded bad bins*.



The two general cases of bad bins

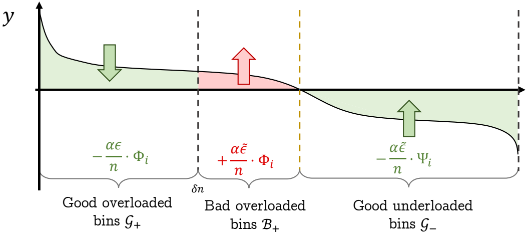
- There can either be *overloaded bad bins* or *underloaded bad bins*.

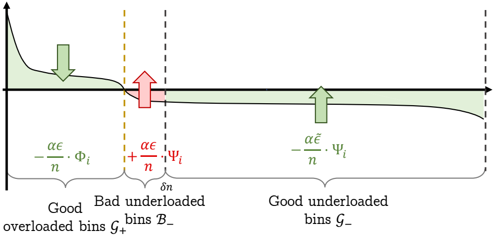


- **Key Observation 4:** The second case is symmetric to the first: $\delta' = 1 - \delta$, $\Phi' = \Psi$, $\Psi' = \Phi$ and $y' = -y$.

The two general cases of bad bins

- There can either be *overloaded bad bins* or *underloaded bad bins*.

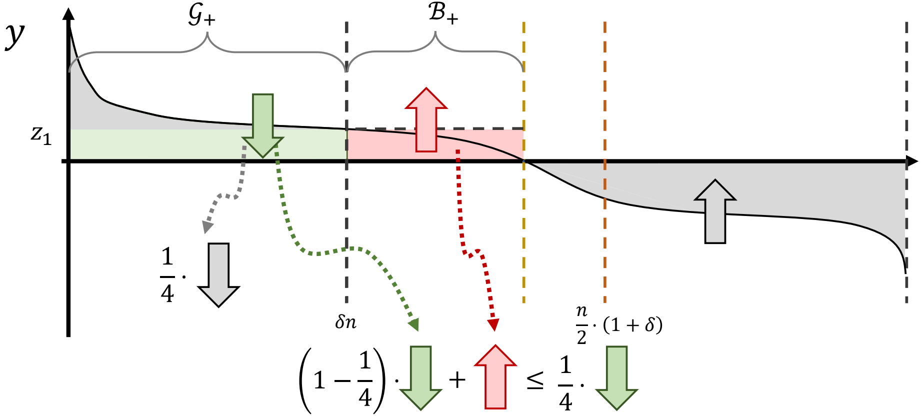




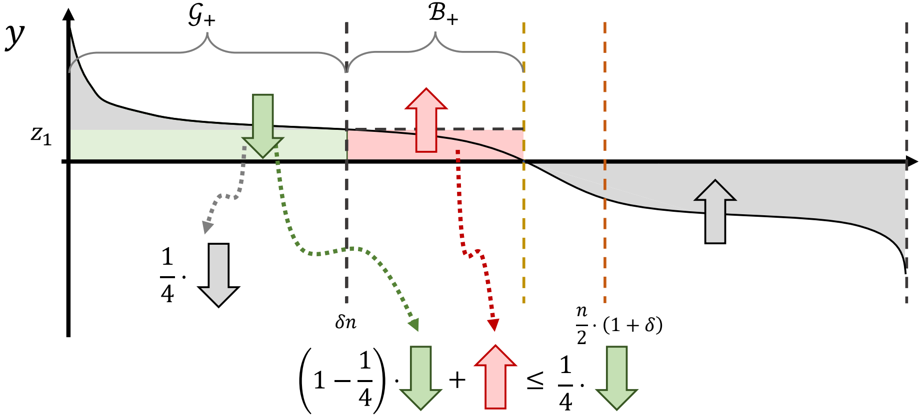
- **Key Observation 4:** The second case is symmetric to the first: $\delta' = 1 - \delta$, $\Phi' = \Psi$, $\Psi' = \Phi$ and $y' = -y$.
- So we only consider Case A.

Case A.1: Not too many overloaded bins

Case A.1: Not too many overloaded bins

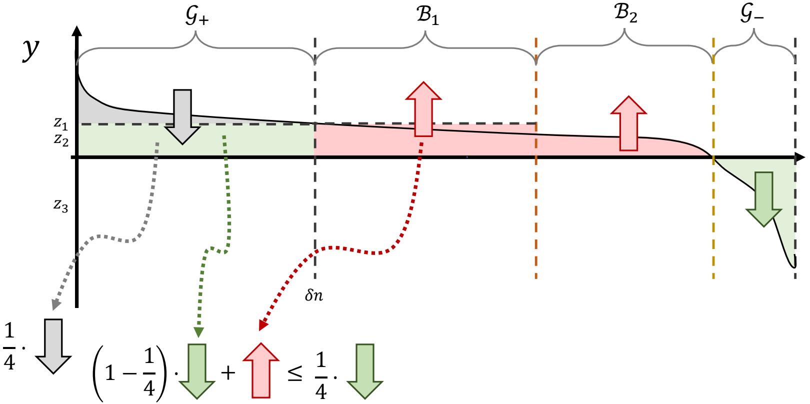


Case A.1: Not too many overloaded bins

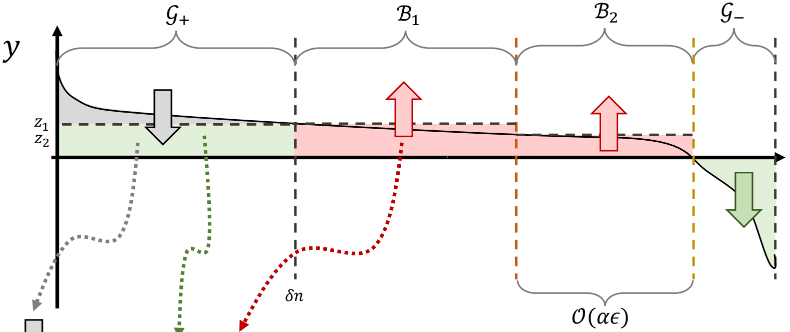


- As with the exponential potential, we counteract the bad bins with a fraction of the decrease of the overloaded good bins. *All* underloaded bins are good.

Case A.2: Too many overloaded bins



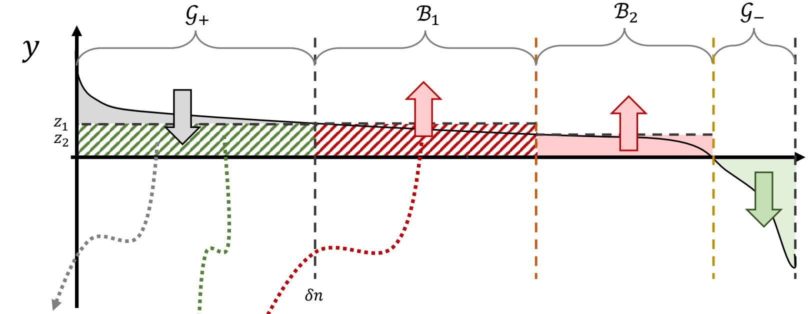
Case A.2: Too many overloaded bins

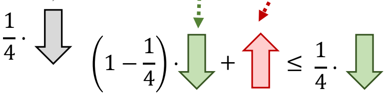


$\frac{1}{4} \cdot \Downarrow \quad \left(1 - \frac{1}{4}\right) \cdot \Downarrow + \Uparrow \leq \frac{1}{4} \cdot \Downarrow$

If $z_2 \leq \frac{1}{\alpha} \cdot \frac{1-\delta}{2\delta} \cdot \log 3$

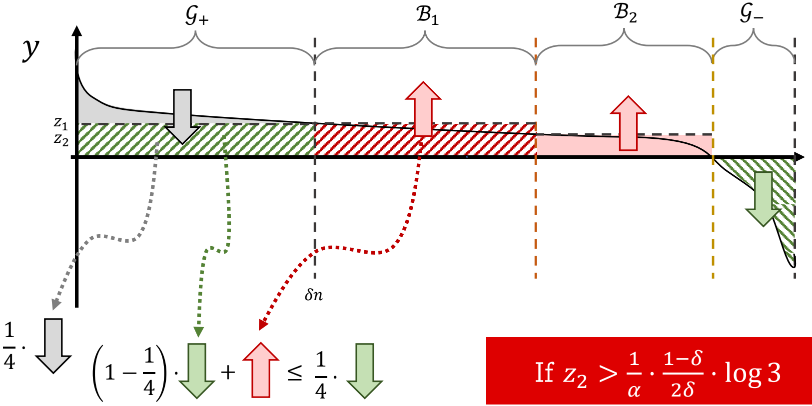
Case A.2: Too many overloaded bins





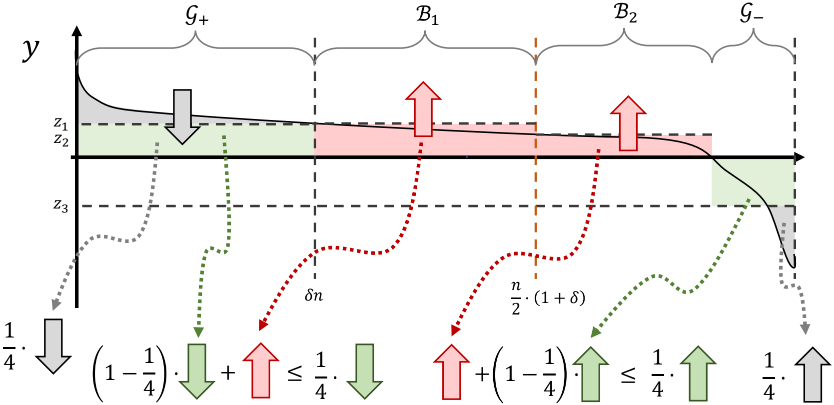
If $z_2 > \frac{1}{\alpha} \cdot \frac{1-\delta}{2\delta} \cdot \log 3$

Case A.2: Too many overloaded bins



If $z_2 > \frac{1}{\alpha} \cdot \frac{1-\delta}{2\delta} \cdot \log 3$

Case A.2: Too many overloaded bins



Recap: Proof

We used the following techniques:

Recap: Proof

We used the following techniques:

1. Consider only step probability vectors (**Key observation 1**).

Recap: Proof

We used the following techniques:

1. Consider only step probability vectors (**Key observation 1**).
2. Consider only the coefficients of α (**Key observation 2**).

Recap: Proof

We used the following techniques:

1. Consider only step probability vectors (**Key observation 1**).
2. Consider only the coefficients of α (**Key observation 2**).
3. Consider only Φ_i for overloaded bins (and Ψ_i otherwise) (**Key observation 3**).

Recap: Proof

We used the following techniques:

1. Consider only step probability vectors (**Key observation 1**).
2. Consider only the coefficients of α (**Key observation 2**).
3. Consider only Φ_i for overloaded bins (and Ψ_i otherwise) (**Key observation 3**).
4. Consider only Case A by symmetry (**Key observation 4**).

Recap: Proof

We used the following techniques:

1. Consider only step probability vectors (**Key observation 1**).
2. Consider only the coefficients of α (**Key observation 2**).
3. Consider only Φ_i for overloaded bins (and Ψ_i otherwise) (**Key observation 3**).
4. Consider only Case A by symmetry (**Key observation 4**).
5. Use the decrease of the underload potential to counteract the increase of bad bins.

The drift theorem

Theorem ([LS22, Corollary 3.2])

Consider any allocation process and a probability vector p being ϵ-biased for some $\epsilon \in (0,1)$ and some constant δ. Further assume that it satisfies for some $K > 0$ and some $R > 0$, for any $t \geq 0$,

$$\mathbf{E}\left[\, \Phi^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq \sum_{i=1}^{n} \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot R \cdot \alpha + K \cdot R \cdot \frac{\alpha^2}{n}\right),$$

and

$$\mathbf{E}\left[\, \Psi^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq \sum_{i=1}^{n} \Psi_i^t \cdot \left(1 + \left(\frac{1}{n} - p_i\right) \cdot R \cdot \alpha + K \cdot R \cdot \frac{\alpha^2}{n}\right).$$

Then, there exists a constant $c := c(\delta) > 0$, such that for $\alpha \in \left(0, \min\left\{1, \frac{\epsilon\delta}{8K}\right\}\right)$

$$\mathbf{E}\left[\, \Gamma^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] \leq \Gamma^t \cdot R \cdot \left(1 - \frac{\alpha\epsilon\delta}{8n}\right) + R \cdot c\alpha\epsilon,$$

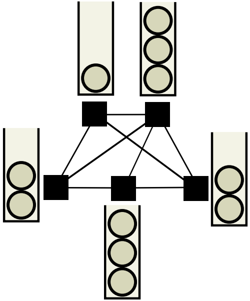
and

$$\mathbf{E}\left[\, \Gamma^t \,\right] \leq \frac{8c}{\delta} \cdot n.$$

Example 1: The Graphical Setting

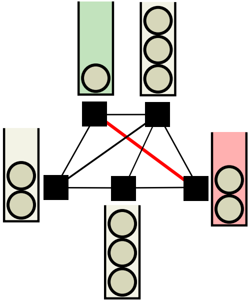
Example 1: The Graphical Setting

- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:



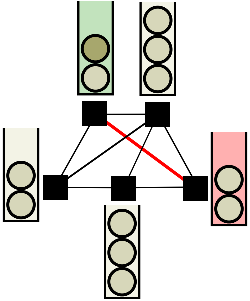
Example 1: The Graphical Setting

- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.



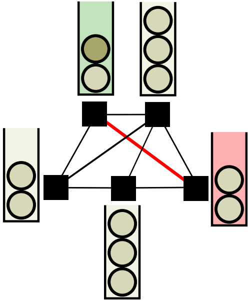
Example 1: The Graphical Setting

- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



Example 1: The Graphical Setting

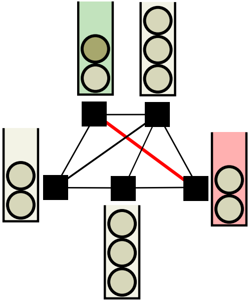
- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



- For any d-regular graph with *conductance* ϵ,

Example 1: The Graphical Setting

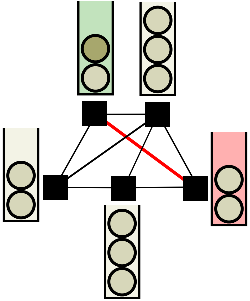
- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



- For any d-regular graph with *conductance* ϵ, p^t is *majorized* by an ϵ-biased probability vector.

Example 1: The Graphical Setting

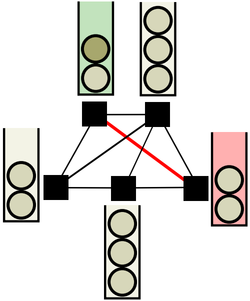
- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



- For any d-regular graph with *conductance* ϵ, p^t is *majorized* by an ϵ-biased probability vector. $\rightsquigarrow$ gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\epsilon}\right)$ [PTW15].

Example 1: The Graphical Setting

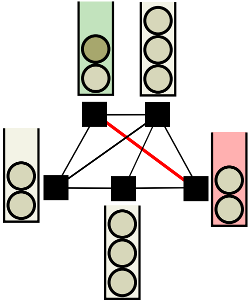
- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



- For any d-regular graph with *conductance* ϵ, p^t is *majorized* by an ϵ-biased probability vector. $\rightsquigarrow$ gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\epsilon}\right)$ [PTW15].
- Majorization does not apply for weights.

Example 1: The Graphical Setting

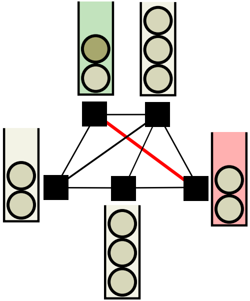
- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



- For any *d*-regular graph with *conductance* ϵ, p^t is *majorized* by an ϵ-biased probability vector. $\rightsquigarrow$ gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\epsilon}\right)$ [PTW15].
- Majorization does not apply for weights. But the refined drift theorem applies for the majorized vector.

Example 1: The Graphical Setting

- Given a graph $G = (V, E)$, where the vertices are bins. For each ball [KP06]:
  - Sample an edge u.a.r.
  - Allocate the ball to the least loaded of its *two* adjacent bins.



- For any d-regular graph with *conductance* ϵ, p^t is *majorized* by an ϵ-biased probability vector. $\rightsquigarrow$ gap is w.h.p. $\mathcal{O}\left(\frac{\log n}{\epsilon}\right)$ [PTW15].
- Majorization does not apply for weights. But the refined drift theorem applies for the majorized vector. $\rightsquigarrow$ Resolves [PTW15, Open problem 1]

Example 2: The TWINNING process

Example 2: The Twinning process

- The Twinning process at quantile δ, samples *one* bin i u.a.r. and:

Example 2: The Twinning process

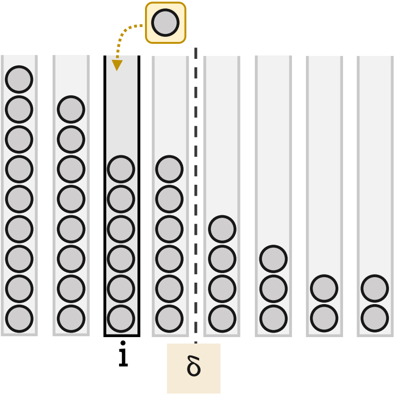
- The Twinning process at quantile δ, samples *one* bin i u.a.r. and:
  - If i is among the heaviest $n \cdot \delta$ bins, it allocates *one* ball there.

Example 2: The Twinning process

- The Twinning process at quantile δ, samples *one* bin i u.a.r. and:
  - If i is among the heaviest $n \cdot \delta$ bins, it allocates *one* ball there.
  - Otherwise, it allocates *two* balls.

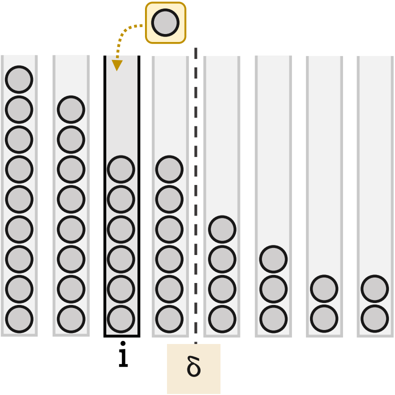
Example 2: The TWINNING process

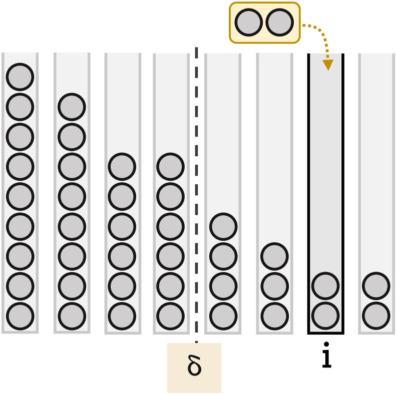
- The TWINNING process at quantile δ, samples *one* bin i u.a.r. and:
  - If i is among the heaviest $n \cdot \delta$ bins, it allocates *one* ball there.
  - Otherwise, it allocates *two* balls.



Example 2: The Twinning process

- The Twinning process at quantile δ, samples *one* bin i u.a.r. and:
  - If i is among the heaviest $n \cdot \delta$ bins, it allocates *one* ball there.
  - Otherwise, it allocates *two* balls.





Example 2: The TWINNING process

Example 2: The Twinning process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

Example 2: The Twinning process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right).$$

Example 2: The TWINNING process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right).$$

- For TWINNING, for any heavy bin $i \leq n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] = \underbrace{\left(1 - \frac{1}{n}\right) \cdot \frac{1}{n}}_{\text{Allocate one ball to bin } i} + \underbrace{\left(-\frac{1}{n}\right) \cdot \left(\delta - \frac{1}{n}\right)}_{\substack{\text{Allocate one ball} \\ \text{to any other heavy bin}}} + \underbrace{\left(-\frac{2}{n}\right) \cdot (1 - \delta)}_{\substack{\text{Allocate two balls} \\ \text{to any light bin}}}$$

Example 2: The Twinning process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right).$$

- For Twinning, for any heavy bin $i \leq n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] = \underbrace{\left(1 - \frac{1}{n}\right) \cdot \frac{1}{n}}_{\text{Allocate one ball to bin } i} + \underbrace{\left(-\frac{1}{n}\right) \cdot \left(\delta - \frac{1}{n}\right)}_{\substack{\text{Allocate one ball} \\ \text{to any other heavy bin}}} + \underbrace{\left(-\frac{2}{n}\right) \cdot (1 - \delta)}_{\substack{\text{Allocate two balls} \\ \text{to any light bin}}} = \frac{\delta}{n} - \frac{1}{n}.$$

Example 2: The Twinning process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right).$$

- For Twinning, for any heavy bin $i \leq n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] = \underbrace{\left(1 - \frac{1}{n}\right) \cdot \frac{1}{n}}_{\text{Allocate one ball to bin } i} + \underbrace{\left(-\frac{1}{n}\right) \cdot \left(\delta - \frac{1}{n}\right)}_{\substack{\text{Allocate one ball} \\ \text{to any other heavy bin}}} + \underbrace{\left(-\frac{2}{n}\right) \cdot (1 - \delta)}_{\substack{\text{Allocate two balls} \\ \text{to any light bin}}} = \frac{\delta}{n} - \frac{1}{n}.$$

- Similarly for a light bin $i > n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \,\middle|\, \mathfrak{F}^t \,\right] = \frac{\delta}{n}.$$

Example 2: The Twinning process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right).$$

- For Twinning, for any heavy bin $i \leq n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] = \underbrace{\left(1 - \frac{1}{n}\right) \cdot \frac{1}{n}}_{\text{Allocate one ball to bin } i} + \underbrace{\left(-\frac{1}{n}\right) \cdot \left(\delta - \frac{1}{n}\right)}_{\substack{\text{Allocate one ball} \\ \text{to any other heavy bin}}} + \underbrace{\left(-\frac{2}{n}\right) \cdot (1 - \delta)}_{\substack{\text{Allocate two balls} \\ \text{to any light bin}}} = \frac{\delta}{n} - \frac{1}{n}.$$

- Similarly for a light bin $i > n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] = \frac{\delta}{n}.$$

- So, we can apply the drift theorem with probability vector

$$p = \begin{cases} \frac{\delta}{n} & \text{for } i \leq n \cdot \delta, \\ \frac{1+\delta}{n} & \text{otherwise,} \end{cases}$$

Example 2: The TWINNING process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\, \Delta\Phi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right),$$

$$\mathbf{E}\left[\, \Delta\Psi_i^{t+1} \mid \mathfrak{F}^t \,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] + \alpha^2 \cdot \mathbf{E}\left[\, (\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t \,\right]\right).$$

- For TWINNING, for any heavy bin $i \leq n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] = \underbrace{\left(1 - \frac{1}{n}\right) \cdot \frac{1}{n}}_{\text{Allocate one ball to bin } i} + \underbrace{\left(-\frac{1}{n}\right) \cdot \left(\delta - \frac{1}{n}\right)}_{\substack{\text{Allocate one ball} \\ \text{to any other heavy bin}}} + \underbrace{\left(-\frac{2}{n}\right) \cdot (1 - \delta)}_{\substack{\text{Allocate two balls} \\ \text{to any light bin}}} = \frac{\delta}{n} - \frac{1}{n}.$$

- Similarly for a light bin $i > n \cdot \delta$:

$$\mathbf{E}\left[\, \Delta y_i^{t+1} \mid \mathfrak{F}^t \,\right] = \frac{\delta}{n}.$$

- So, we can apply the drift theorem with probability vector

$$p = \begin{cases} \frac{\delta}{n} & \text{for } i \leq n \cdot \delta, \\ \frac{1+\delta}{n} & \text{otherwise,} \end{cases}$$

Example 2: The TWINNING process

- For processes allocating more than one balls we have that:

$$\mathbf{E}\left[\,\Delta\Phi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Phi_i^t \cdot \left(\alpha \cdot \mathbf{E}\left[\,\Delta y_i^{t+1} \mid \mathfrak{F}^t\,\right] + \alpha^2 \cdot \mathbf{E}\left[\,(\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t\,\right]\right),$$

$$\mathbf{E}\left[\,\Delta\Psi_i^{t+1} \mid \mathfrak{F}^t\,\right] \leq \Psi_i^t \cdot \left(-\alpha \cdot \mathbf{E}\left[\,\Delta y_i^{t+1} \mid \mathfrak{F}^t\,\right] + \alpha^2 \cdot \mathbf{E}\left[\,(\Delta y_i^{t+1})^2 \mid \mathfrak{F}^t\,\right]\right).$$

- For TWINNING, for any heavy bin $i \leq n \cdot \delta$:

$$\mathbf{E}\left[\,\Delta y_i^{t+1} \mid \mathfrak{F}^t\,\right] = \underbrace{\left(1 - \frac{1}{n}\right) \cdot \frac{1}{n}}_{\text{Allocate one ball to bin } i} + \underbrace{\left(-\frac{1}{n}\right) \cdot \left(\delta - \frac{1}{n}\right)}_{\substack{\text{Allocate one ball} \\ \text{to any other heavy bin}}} + \underbrace{\left(-\frac{2}{n}\right) \cdot (1 - \delta)}_{\substack{\text{Allocate two balls} \\ \text{to any light bin}}} = \frac{\delta}{n} - \frac{1}{n}.$$

- Similarly for a light bin $i > n \cdot \delta$:

$$\mathbf{E}\left[\,\Delta y_i^{t+1} \mid \mathfrak{F}^t\,\right] = \frac{\delta}{n}.$$

- So, we can apply the drift theorem with probability vector

$$p = \begin{cases} \frac{\delta}{n} & \text{for } i \leq n \cdot \delta, \\ \frac{1+\delta}{n} & \text{otherwise,} \end{cases}$$

Example 3: Memory with resets

Example 3: Memory with resets

- Mitzenmacher, Prabhakar and Shah [MPS02] introduced the Memory process which is allowed to store a bin in a *cache*.

Example 3: Memory with resets

- Mitzenmacher, Prabhakar and Shah [MPS02] introduced the Memory process which is allowed to store a bin in a *cache*. In each step, it can either allocate to the cache or to a random bin.

Example 3: Memory with resets

- Mitzenmacher, Prabhakar and Shah [MPS02] introduced the Memory process which is allowed to store a bin in a *cache*. In each step, it can either allocate to the cache or to a random bin.

- In Memory with resets the cache is *emptied* every r steps.

Example 3: Memory with resets

- Mitzenmacher, Prabhakar and Shah [MPS02] introduced the Memory process which is allowed to store a bin in a *cache*. In each step, it can either allocate to the cache or to a random bin.

- In Memory with resets the cache is *emptied* every r steps.

- For $r = 2$, a similar analysis to that of the Twinning process, gives rise to the probability vector p of the Two-Choice process, i.e., $p_i = \frac{2i-1}{n^2}$.

Example 3: Memory with resets

- Mitzenmacher, Prabhakar and Shah [MPS02] introduced the Memory process which is allowed to store a bin in a *cache*. In each step, it can either allocate to the cache or to a random bin.

- In Memory with resets the cache is *emptied* every r steps.

- For $r = 2$, a similar analysis to that of the Twinning process, gives rise to the probability vector p of the Two-Choice process, i.e., $p_i = \frac{2i-1}{n^2}$.

- Again, applying the drift theorem gives w.h.p. an $\mathcal{O}(\log n)$ upper bound on the gap.

Example 4: The b-Batched setting

Example 4: The b-Batched setting

- Recall that in the b-Batched setting, a *batch* of b balls is allocated in *parallel*, using the allocation vector at the start of the batch.

Example 4: The b-Batched setting

- Recall that in the b-Batched setting, a *batch* of b balls is allocated in *parallel*, using the allocation vector at the start of the batch.
- Consider a process with an ϵ-biased allocation vector p which further satisfies $\max_{i \in [n]} p_i \leq \frac{C}{n}$.

Example 4: The b-Batched setting

- Recall that in the b-Batched setting, a *batch* of b balls is allocated in *parallel*, using the allocation vector at the start of the batch.
- Consider a process with an ϵ-biased allocation vector p which further satisfies $\max_{i \in [n]} p_i \leq \frac{C}{n}$.
- With a few *Taylor estimates*, we get

$$\mathbf{E}\left[\, \Phi_i^{t+b} \mid \Phi_i^t \,\right] \leq \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right),$$

Example 4: The b-Batched setting

- Recall that in the b-Batched setting, a *batch* of b balls is allocated in *parallel*, using the allocation vector at the start of the batch.
- Consider a process with an ϵ-biased allocation vector p which further satisfies $\max_{i\in[n]} p_i \leq \frac{C}{n}$.
- With a few *Taylor estimates*, we get

$$\mathbf{E}\left[\, \Phi_i^{t+b} \mid \Phi_i^t \,\right] \leq \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right),$$

and similarly,

$$\mathbf{E}\left[\, \Psi_i^{t+b} \mid \Psi_i^t \,\right] \leq \Psi_i^t \cdot \left(1 + \left(\frac{1}{n} - p_i\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right).$$

Example 4: The b-Batched setting

- Recall that in the b-Batched setting, a *batch* of b balls is allocated in *parallel*, using the allocation vector at the start of the batch.
- Consider a process with an ϵ-biased allocation vector p which further satisfies $\max_{i \in [n]} p_i \leq \frac{C}{n}$.
- With a few *Taylor estimates*, we get

$$\mathbf{E}\left[\, \Phi_i^{t+b} \mid \Phi_i^t \,\right] \leq \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right),$$

and similarly,

$$\mathbf{E}\left[\, \Psi_i^{t+b} \mid \Psi_i^t \,\right] \leq \Psi_i^t \cdot \left(1 + \left(\frac{1}{n} - p_i\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right).$$

- Therefore, by the *drift theorem* over b steps, we get w.h.p. an $\mathcal{O}(\frac{b}{n} \cdot \log n)$ gap.

Example 4: The b-Batched setting

- Recall that in the b-Batched setting, a *batch* of b balls is allocated in *parallel*, using the allocation vector at the start of the batch.
- Consider a process with an ϵ-biased allocation vector p which further satisfies $\max_{i \in [n]} p_i \leq \frac{C}{n}$.
- With a few *Taylor estimates*, we get

$$\mathbf{E}\left[\,\Phi_i^{t+b} \mid \Phi_i^t\,\right] \leq \Phi_i^t \cdot \left(1 + \left(p_i - \frac{1}{n}\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right),$$

and similarly,

$$\mathbf{E}\left[\,\Psi_i^{t+b} \mid \Psi_i^t\,\right] \leq \Psi_i^t \cdot \left(1 + \left(\frac{1}{n} - p_i\right) \cdot b \cdot \alpha + \frac{5C^2 b}{n} \cdot b \cdot \frac{\alpha^2}{n}\right).$$

- Therefore, by the *drift theorem* over b steps, we get w.h.p. an $\mathcal{O}(\frac{b}{n} \cdot \log n)$ gap.
- This is tight up to a $\log n$ factor for constant $C > 1$.

Conclusions

Summary

Summary

The drift theorem in [PTW15] can be used to:

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.
- Analyze graphical allocations via majorization (*without weights*).

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.
- Analyze graphical allocations via majorization (*without weights*).

The refined drift theorem can be used to:

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.
- Analyze graphical allocations via majorization (*without weights*).

The refined drift theorem can be used to:

- Analyze processes allocating to *more than one* bins in each step.

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.
- Analyze graphical allocations via majorization (*without weights*).

The refined drift theorem can be used to:

- Analyze processes allocating to *more than one* bins in each step.
- Provide *tighter bounds* on $\mathbf{E}[\Gamma^t]$ (and so tighter characterization of the load vector).

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.
- Analyze graphical allocations via majorization (*without weights*).

The refined drift theorem can be used to:

- Analyze processes allocating to *more than one* bins in each step.
- Provide *tighter bounds* on $\mathbf{E}\left[\,\Gamma^t\,\right]$ (and so tighter characterization of the load vector).
- Analyze a wider range of *settings* (including outdated information, noise, etc.).

Summary

The drift theorem in [PTW15] can be used to:

- Analyze processes with an ϵ-biased allocation vector *with weights*.
- Analyze graphical allocations via majorization (*without weights*).

The refined drift theorem can be used to:

- Analyze processes allocating to *more than one* bins in each step.
- Provide *tighter bounds* on $\mathbf{E}[\Gamma^t]$ (and so tighter characterization of the load vector).
- Analyze a wider range of *settings* (including outdated information, noise, etc.).
- It is *agnostic* of the balanced allocations setting.

Open problems

Open problems

Still many open problems, including:

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log \log n$ and $\log n$ for the MEMORY process with resets.

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log \log n$ and $\log n$ for the MEMORY process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log \log n$ and $\log n$ for the Memory process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).
- Analyse balls-into-bins *with deletions*.

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log\log n$ and $\log n$ for the Memory process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).
- Analyse balls-into-bins *with deletions*.
- Apply the drift theorem to *other dynamic processes*.

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log\log n$ and $\log n$ for the Memory process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).
- Analyse balls-into-bins *with deletions*.
- Apply the drift theorem to *other dynamic processes*.

Regarding techniques, there are also a few open questions:

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log \log n$ and $\log n$ for the MEMORY process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).
- Analyse balls-into-bins *with deletions*.
- Apply the drift theorem to *other dynamic processes*.

Regarding techniques, there are also a few open questions:

- Is there a simpler way to argue that ϵ-biased processes have $\Theta(n)$ bins above the average load?

Open problems

Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log \log n$ and $\log n$ for the Memory process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).
- Analyse balls-into-bins *with deletions*.
- Apply the drift theorem to *other dynamic processes*.

Regarding techniques, there are also a few open questions:

- Is there a simpler way to argue that ϵ-biased processes have $\Theta(n)$ bins above the average load?
- Is there a way to adapt the drift theorem to work for processes *with thresholds*?

Open problems

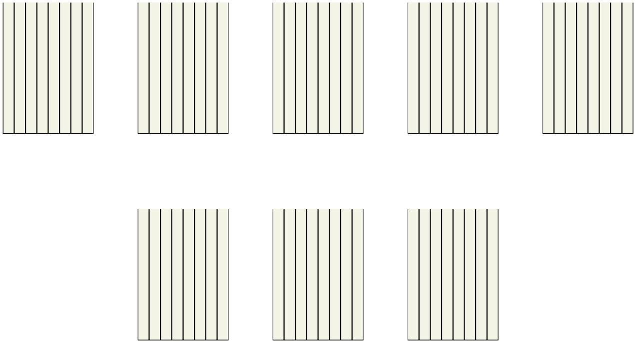
Still many open problems, including:

- Proving *sublogarithmic bounds* for the weighted graphical setting.
- Gaps between $\log \log n$ and $\log n$ for the MEMORY process with resets.
- Analyse various load balancing algorithms used *in practice* (cf. `envoy`, `nginx`).
- Analyse balls-into-bins *with deletions*.
- Apply the drift theorem to *other dynamic processes*.

Regarding techniques, there are also a few open questions:

- Is there a simpler way to argue that ϵ-biased processes have $\Theta(n)$ bins above the average load?
- Is there a way to adapt the drift theorem to work for processes *with thresholds*?
- Can we use a *single potential* to prove sublogarithmic bounds (e.g., the $\log_2 \log n + \Theta(1)$ bound for TWO-CHOICE)?

Questions?



More visualisations: `dimitrioslos.com/phdthesis`

Bibliography I

- D. Alistarh, J. Aspnes, and R. Gelashvili, *Space-optimal majority in population protocols*, 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'18), SIAM, 2018, pp. 2221–2239.
- 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, M. Mitzenmacher, and E. Upfal, *The ACM Paris Kanellakis Theory and Practice Award*, 2020, `https://www.acm.org/media-center/2021/may/technical-awards-2020`.
- Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, *Balanced allocations*, SIAM J. Comput. **29** (1999), no. 1, 180–200. MR 1710347
- D. Alistarh, R. Gelashvili, and J. Rybicki, *Fast graphical population protocols*, 25th International Conference on Principles of Distributed Systems (OPODIS'21), vol. 217, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 14:1–14:18.

Bibliography II

- D. Alistarh, J. Kopinsky, J. Li, and g. Nadiradze, *The power of choice in priority scheduling*, 36th Annual ACM-SIGOPT Principles of Distributed Computing (PODC'17), ACM, 2017, pp. 283–292.
- 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), Springer-Verlag, 2012, pp. 411–422.
- P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking, *Balanced allocations: the heavily loaded case*, SIAM J. Comput. **35** (2006), no. 6, 1350–1385. MR 2217150
- M. Dahlin, *Interpreting stale load information*, IEEE Trans. Parallel Distributed Syst.**11** (2000), no. 10, 1033–1047.
- P. Delgado, D. Didona, F. Dinu, and W. Zwaenepoel, *Job-aware scheduling in eagle: Divide and stick to your probes*, 7th ACM Symposium on Cloud Computing (SoCC'16), ACM, 2016, pp. 497–509.

Bibliography III

- P. Delgado, F. Dinu, A. M. Kermarrec, and W. Zwaenepoel, *Hawk: Hybrid datacenter scheduling*, 2015 USENIX Annual Technical Conference (USENIX'15), USENIX, 2015, pp. 499–510.
- A. Gupta, R. Krishnaswamy, A. Kumar, and S. Singla, *Online carpooling using expander decompositions*, 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'20), vol. 182, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, pp. 23:1–23:14.
- 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
- M. Khelghatdoust and V. Gramoli, *Peacock: Probe-based scheduling of jobs by rotating between elastic queues*, 24th International Conference on Parallel and Distributed Computing (Euro-Par'18), vol. 11014, Springer, 2018, pp. 178–191.
- 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

Bibliography IV

- K. Kenthapadi and R. Panigrahy, *Balanced allocation on graphs*, 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'06), ACM, 2006, pp. 434–443. MR 2368840
- 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. 389–399.
- Y. Lu, Q. Xie, G. Kliot, A. Geller, J. R. Larus, and A. G. Greenberg, *Join-idle-queue: A novel load balancing algorithm for dynamically scalable web services*, Perform. Evaluation **68** (2011), no. 11, 1056–1071.
- M. Mitzenmacher, *The power of two choices in randomized load balancing*, Ph.D. thesis, University of California at Berkeley, 1996.
- ———, *How useful is old information?*, IEEE Trans. Parallel Distributed Syst. **11** (2000), no. 1, 6–20.

Bibliography V

- M. Mitzenmacher, B. Prabhakar, and D. Shah, *Load balancing with memory*, The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings., IEEE, 2002, pp. 799–808.
- G. Nadiradze, *On achieving scalability through relaxation*, Ph.D. thesis, IST Austria, 2021.
- K. Ousterhout, P. Wendell, M. Zaharia, and I. Stoica, *Sparrow: distributed, low latency scheduling*, 24th ACM SIGOPS Symposium on Operating Systems Principles (SOSP'13), ACM, 2013, pp. 69–84.
- Y. Peres, K. Talwar, and U. Wieder, *Graphical balanced allocations and the* $(1+\beta)$*-choice process*, Random Structures & Algorithms **47** (2015), no. 4, 760–775. MR 3418914
- M. Raab and A. Steger, *"Balls into bins"—a simple and tight analysis*, 2nd International Workshop on Randomization and Computation (RANDOM'98), vol. 1518, Springer, 1998, pp. 159–170. MR 1729169

Bibliography VI

- J. Spencer, *Balancing games*, J. Combinatorial Theory Ser. B **23** (1977), no. 1, 68–74. MR 526057
- W. Whitt, *Deciding which queue to join: Some counterexamples*, Oper. Res. **34** (1986), no. 1, 55–62.