Adaptive Bin Packing with Overflow

Sebastian Perez-Salazar
Mohit Singh
Alejandro Toriello

Georgia Tech

Bin Packing Seminar 2021

Motivation

Motivation

Motivation

aws

Azure

Motivation

aws

Azure

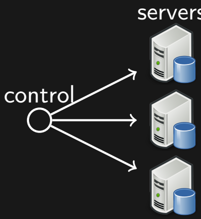
IBM Cloud

Motivation

aws

Azure

IBM Cloud

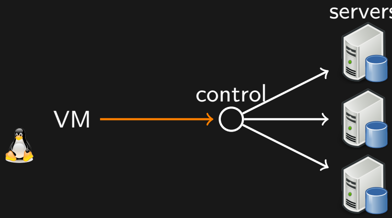


Motivation

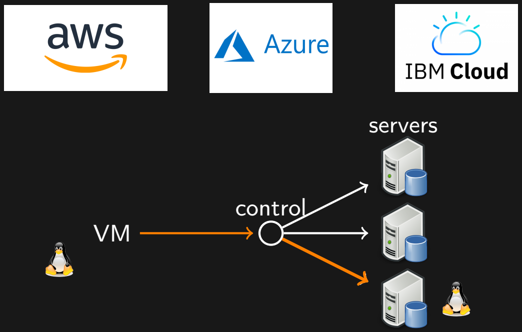
aws

Azure

IBM Cloud



Motivation



aws

Azure

IBM Cloud

servers

control

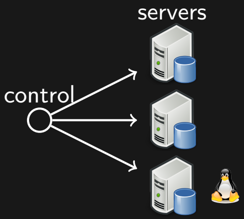
VM

Motivation

aws

Azure

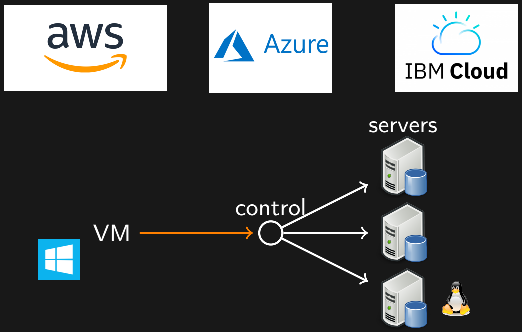
IBM Cloud



servers

control

Motivation

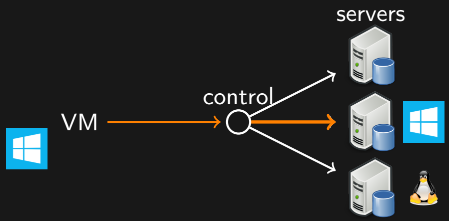


Motivation

aws

Azure

IBM Cloud



servers

VM

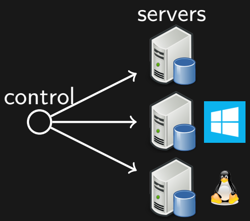
control

Motivation

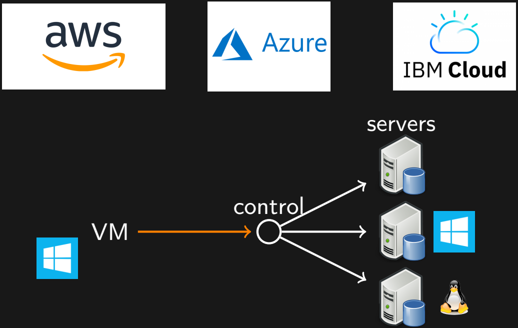
aws

Azure

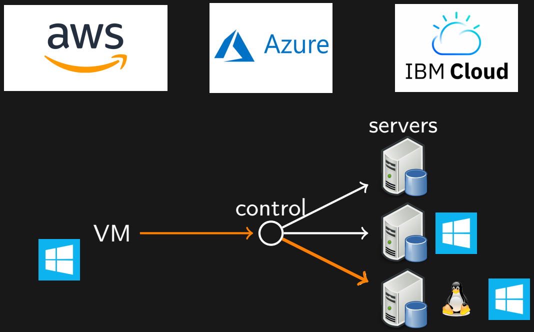
IBM Cloud



Motivation



Motivation

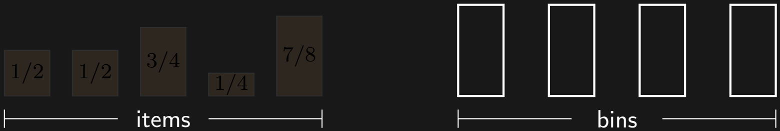


Online Bin Packing

Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

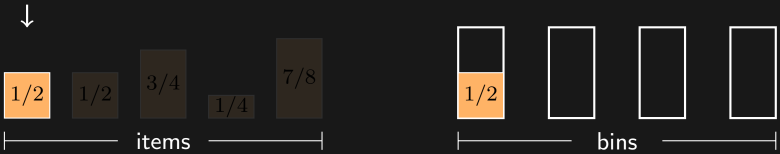
Online Bin Packing



Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

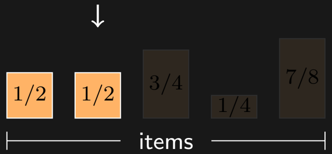
Online Bin Packing

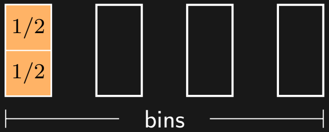


Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

Online Bin Packing

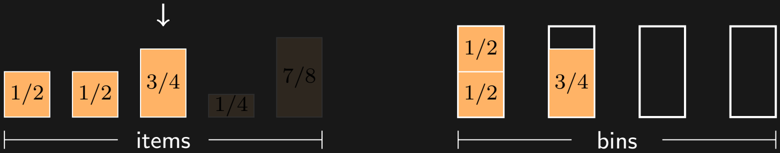




Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

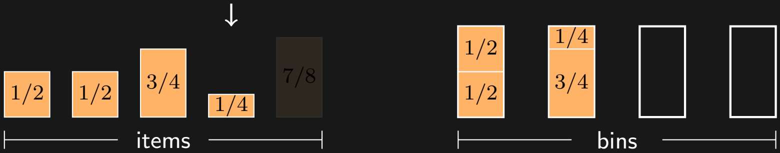
Online Bin Packing



Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

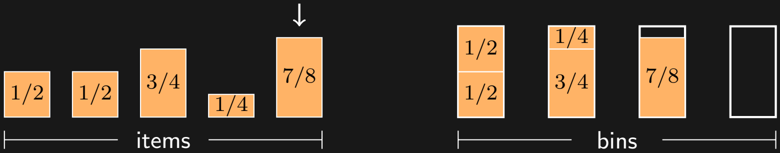
Online Bin Packing



Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

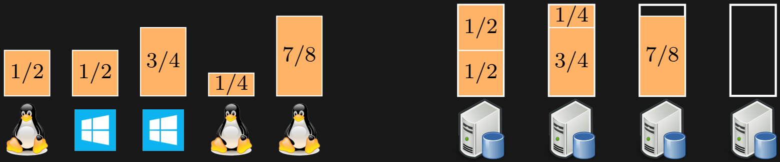
Online Bin Packing



Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

Online Bin Packing



Framework

- Assign items with sizes ≤ 1 into minimum amount of unit-capacity bins
- Examples
  - Assigning VM to servers [Gupta, Radovanovic]
  - Bandwidth allocation [Kleinberg, Rabani & Tardos]

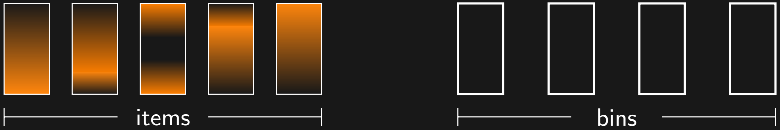
The stochastic viewpoint

- Uncertainty of items' sizes $\rightarrow$ probability distributions

- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

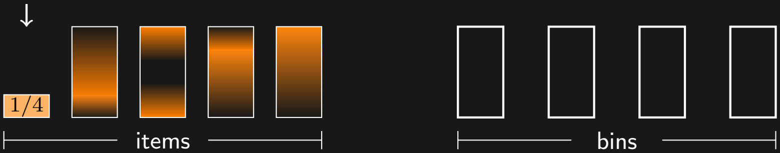
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

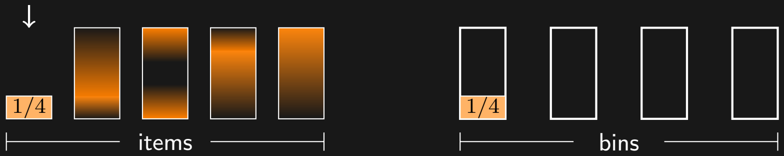
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

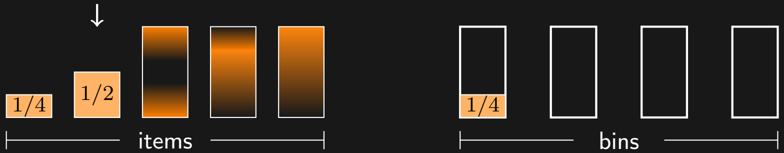
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

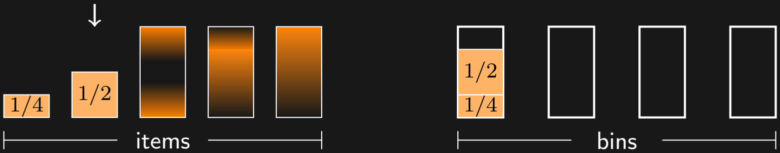
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

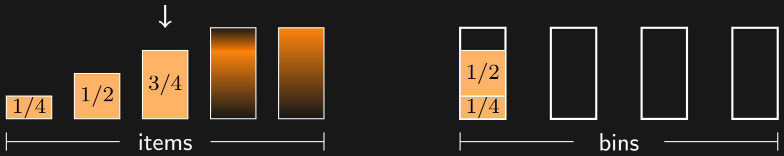
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

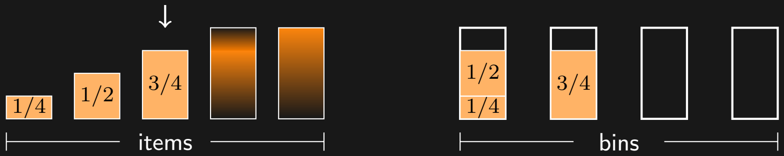
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

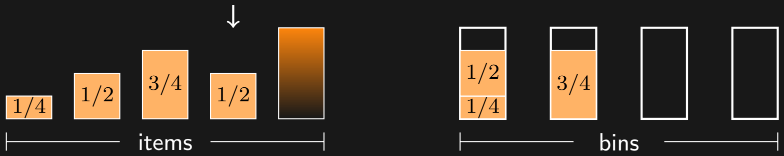
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

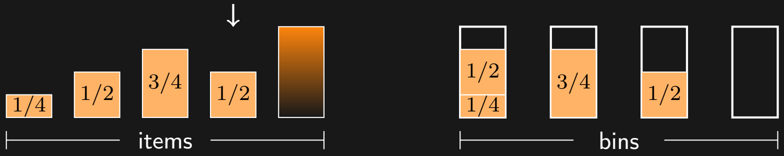
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

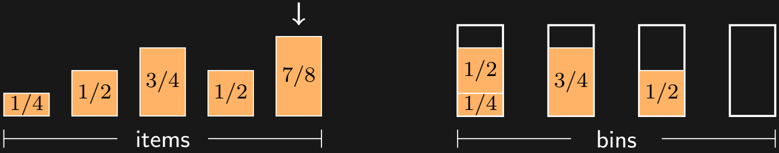
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

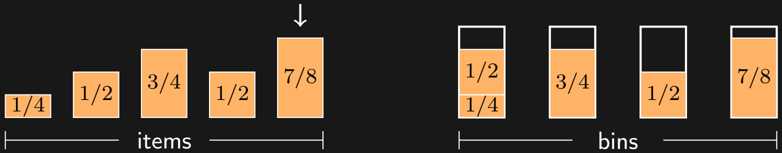
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

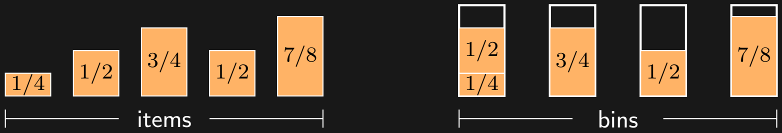
- Uncertainty of items' sizes → probability distributions



- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The stochastic viewpoint

- Uncertainty of items' sizes → probability distributions



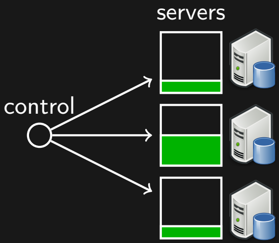
- Stochastic bin packing [Coffman et al '80]
- Sum of squares [Csirik et al '02]

The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic

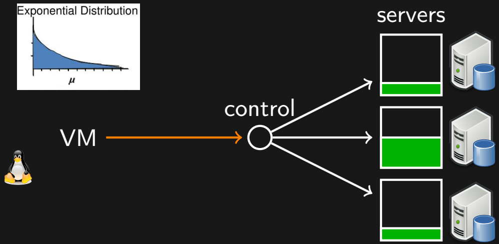
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



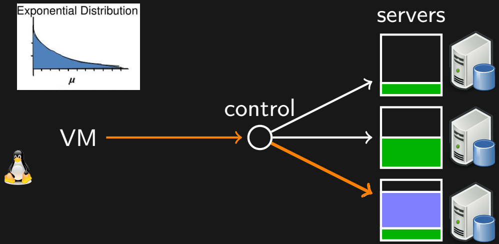
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



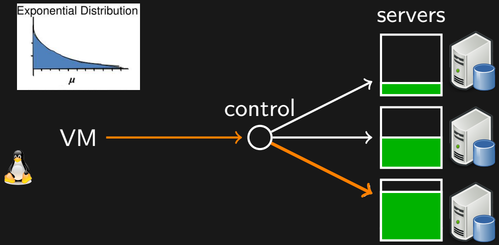
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



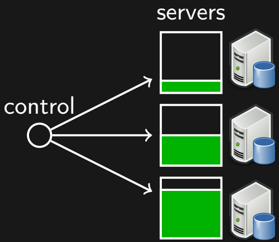
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



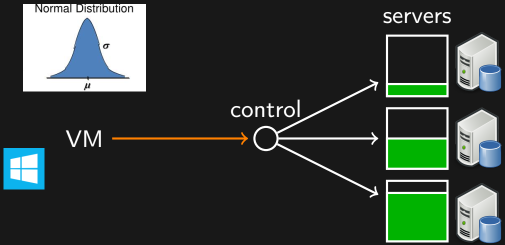
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



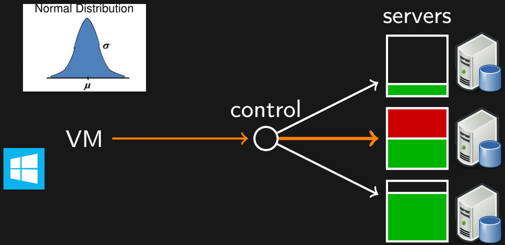
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



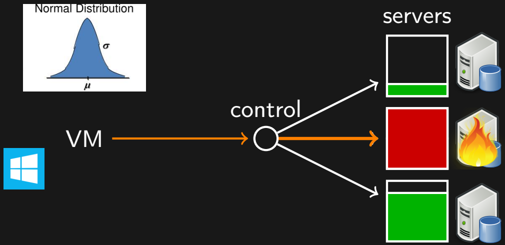
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



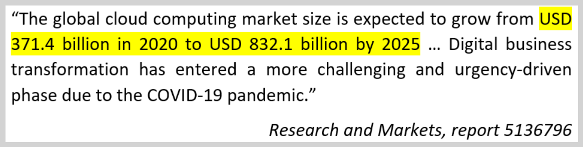
The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



The problem

- (Outcome) size observed before packing
- In many applications this is unrealistic



"The global cloud computing market size is expected to grow from USD 371.4 billion in 2020 to USD 832.1 billion by 2025 … Digital business transformation has entered a more challenging and urgency-driven phase due to the COVID-19 pandemic."

Research and Markets, report 5136796

Our approach

Our approach

- Our packing model that takes into account:

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$

Our approach

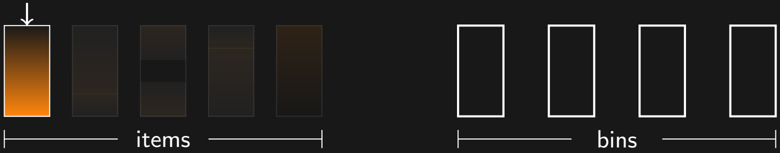
- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$

Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$

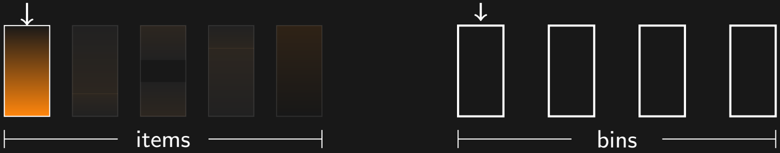
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



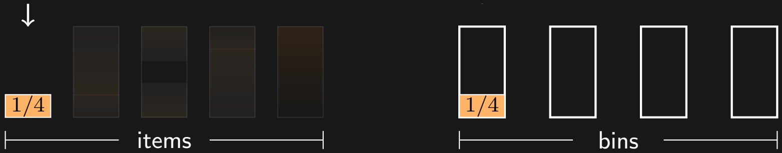
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



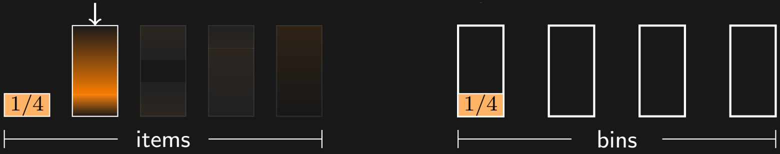
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



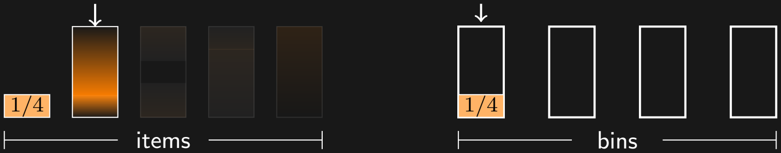
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



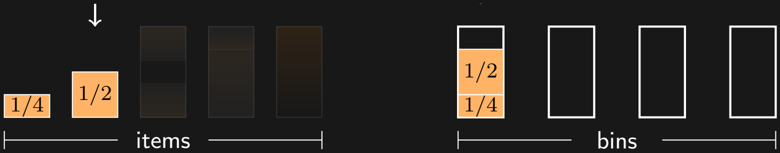
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



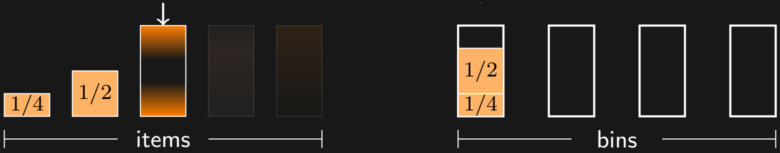
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



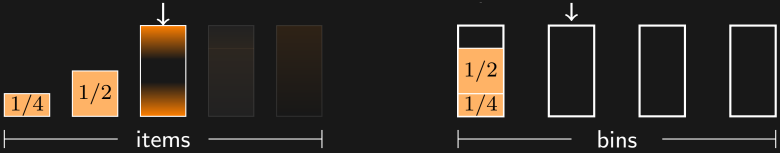
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



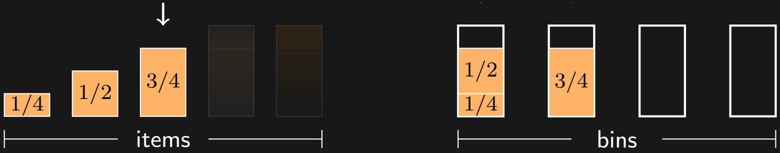
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



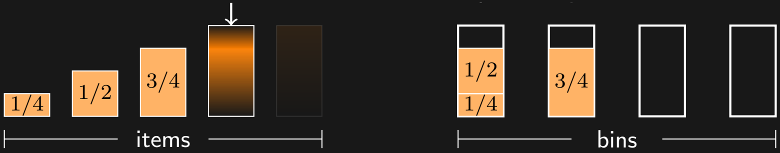
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



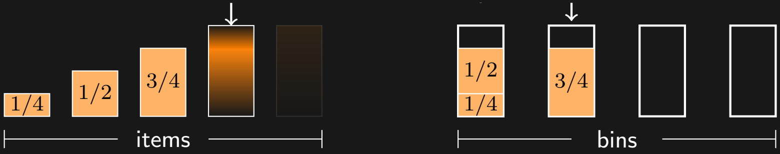
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



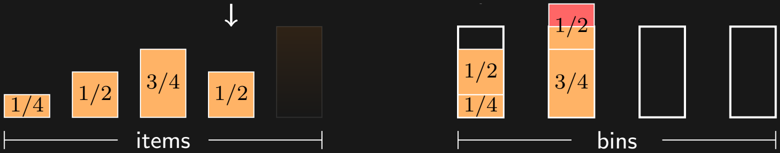
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



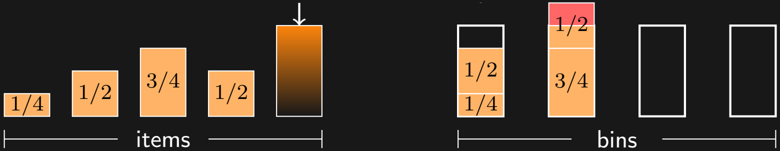
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



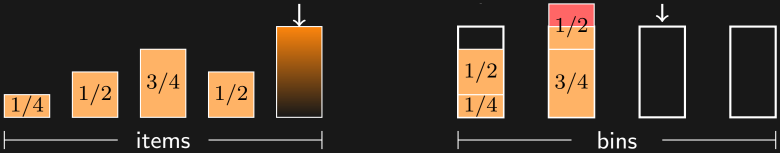
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



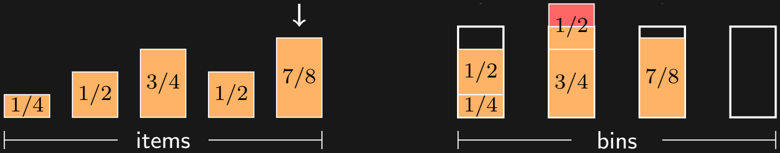
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



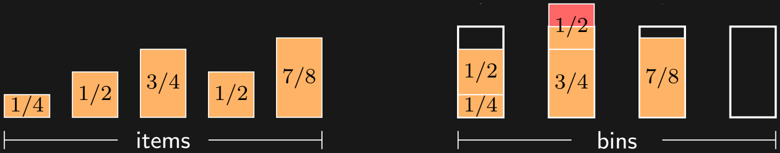
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



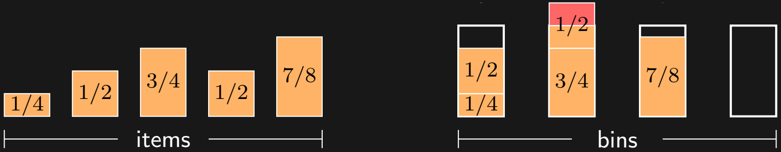
Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \ldots, X_n$
  - Items packed online in a fixed order $1, \ldots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



Our approach

- Our packing model that takes into account:
  - Uncertainty of items' sizes via independent r.v.'s $X_1, \dots, X_n$
  - Items packed online in a fixed order $1, \dots, n$
  - Sizes learned after the decision maker has packed it
  - Overflowing bin incurs a fixed penalty C and bin becomes unusable
    - Server overload, reduced quality of service, repacking, etc.
  - Goal: Minimize $\underbrace{\text{number of used bins} + \text{overall penalty}}_{\text{cost}}$



total cost $= 3 + C$

Extensible models

- Generalized extensible bin packing problem [Levin '19]
- Stochastic extensible bin packing [Sagnol, Schmidt & Tesch '18]
- Online bin packing with overload cost [Luo '21]

Adaptive models

- Adaptive knapsack [Derman, Lieberman & Ross '78][Dean, Goemans & Vondrák '08]
- GAP [Alaei et al '13]
- Bipartite matching [Mehta et al '14][Goyal & Udwani '19]
- Probing submodularity [Gupta, Nagarajan & Singla][Adamczyk & Sviridenko]

Some of our results

Some of our results

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq (3 + 2\sqrt{2})\mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

Some of our results

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq (3 + 2\sqrt{2})\mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

OPT is best sequential policy that packs $X_1, \ldots, X_n$ sequentially

Some of our results

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq (3 + 2\sqrt{2})\mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

OPT is best sequential policy that packs $X_1, \ldots, X_n$ sequentially

Result 2 (OFFline)

For any $X_1, \ldots, X_n$, there is a PTAS that computes a policy for bins with capacity $1 + \varepsilon$ and cost $\leq (1 + \varepsilon)\mathrm{cost}(OPT)$.

Some of our results

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq (3 + 2\sqrt{2})\mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

OPT is best sequential policy that packs $X_1, \ldots, X_n$ sequentially

Result 2 (OFFline)

For any $X_1, \ldots, X_n$, there is a PTAS that computes a policy for bins with capacity $1 + \varepsilon$ and cost $\leq (1 + \varepsilon)\mathrm{cost}(OPT)$.

Result 3 (OFFline)

Computing $\mathrm{cost}(OPT)$ is $\#\mathbf{P}$-hard.

Some of our results

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq 8 \cdot \mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

OPT is best sequential policy that packs $X_1, \ldots, X_n$ sequentially

Result 2 (OFFline)

For any $X_1, \ldots, X_n$, there is a PTAS that computes a policy for bins with capacity $1 + \varepsilon$ and cost $\leq (1 + \varepsilon)\mathrm{cost}(OPT)$.

Result 3 (OFFline)

Computing $\mathrm{cost}(OPT)$ is $\#\mathbf{P}$-hard.

Why OPT is a sequential policy?

Why OPT is a sequential policy?

- Cannot compare against optimal offline that knows all outcomes:

$$X_i = \begin{cases} 1/n & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

Why OPT is a sequential policy?

- Cannot compare against optimal offline that knows all outcomes:

$$X_i = \begin{cases} 1/n & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal offline cost is $\leq n/C + 1$

Why OPT is a sequential policy?

- Cannot compare against optimal offline that knows all outcomes:

$$X_i = \begin{cases} 1/n & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal offline cost is $\leq n/C + 1$
  - Any sequential policy incurs in cost $\geq n$.

Why OPT is a sequential policy?

- Cannot compare against optimal offline that knows all outcomes:

$$X_i = \begin{cases} 1/n & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal offline cost is $\leq n/C + 1$
  - Any sequential policy incurs in cost $\geq n$.

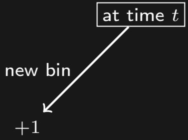
$\boxed{\text{at time } t}$

Why OPT is a sequential policy?

- Cannot compare against optimal offline that knows all outcomes:

$$X_i = \begin{cases} 1/n & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal offline cost is $\leq n/C + 1$
  - Any sequential policy incurs in cost $\geq n$.

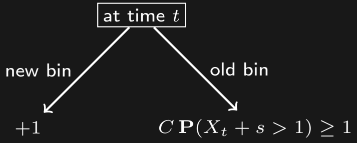


Why OPT is a sequential policy?

- Cannot compare against optimal offline that knows all outcomes:

$$X_i = \begin{cases} 1/n & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal offline cost is $\leq n/C + 1$
  - Any sequential policy incurs in cost $\geq n$.



at time t

new bin

old bin

$+1$

$C\,\mathbf{P}(X_t + s > 1) \geq 1$

For the rest of the talk

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq 8 \cdot \mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

For the rest of the talk

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq 8 \cdot \mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

Idea:

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$

For the rest of the talk

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq 8 \cdot \mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

Idea:

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- Phase 1:
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}] =$ sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$

For the rest of the talk

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq 8 \cdot \mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

Idea:

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- Phase 1:
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}] =$ sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$
- Phase 2:
  - Only bounded-risk policies are interesting: $\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 4 \cdot \mathrm{cost}(\mathcal{P}_{\textsf{old}})$

For the rest of the talk

Result 1 (ONline)

There is an algorithm ALG such that $\mathrm{cost}(ALG) \leq 8 \cdot \mathrm{cost}(OPT)$ when $X_1, \ldots, X_n$ are i.i.d.

Idea:

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- Phase 1:
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}] =$ sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$
- Phase 2:
  - Only bounded-risk policies are interesting: $\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 4 \cdot \mathrm{cost}(\mathcal{P}_{\textsf{old}})$
- Phase 3:
  - $\mathbf{E}[N_{\mathrm{ALG}}] = \min\limits_{\mathcal{P}\ \text{bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$

Greedy policies

behave poorly

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

Greedy policies

behave poorly

- Cost of opening a new bin ($+1$) vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_m	0	0		

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_m	0	0		

  - Cost of greedy $\geq n/2$

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_m	0	0		

  - Cost of greedy $\geq n/2$

Bin breaks:

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_m	0	0		

  - Cost of greedy $\geq n/2$

Bin breaks: $\underbrace{0\,0\,0\cdots0\,1}_{\approx\mathrm{Geom}(\frac{1}{C})}$

Greedy policies

behave poorly

- Cost of opening a new bin ($+1$) vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_m	0	0		

  - Cost of greedy $\geq n/2$

Bin breaks: $\underbrace{0\,0\,0\cdots0\,1}_{\approx\text{Geom}(\frac{1}{C})}\underbrace{0\,0\cdots0\,1}_{\approx\text{Geom}(\frac{1}{C})}$

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  - Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
⋮	⋮	⋮	⋮	⋮
B_m	0	0		

  - Cost of greedy $\geq n/2$

Bin breaks: $\underbrace{0\,0\,0\cdots0\,1}_{\approx\mathrm{Geom}(\frac{1}{C})}\underbrace{0\,0\cdots0\,1}_{\approx\mathrm{Geom}(\frac{1}{C})} \Longrightarrow \approx 2C$ items per bin

Greedy policies

behave poorly

- Cost of opening a new bin $(+1)$ vs expected cost using an available bin
- Consider the i.i.d. sequence

$$X_i = \begin{cases} 0 & \text{w.p. } 1 - 1/C \\ 1 & \text{w.p. } 1/C \end{cases}$$

  – Optimal policy incurs in a cost $\leq n/C + 1$

Bins	1st item	2nd item	3rd item	4th item
B_1	0	0	0	1
B_2	0	0	1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
B_m	0	0		

  – Cost of greedy $\geq n/2$

Bin breaks: $$\underbrace{0\,0\,0 \cdots 0\,1}_{\approx \text{Geom}(\frac{1}{C})}\underbrace{0\,0 \cdots 0\,1}_{\approx \text{Geom}(\frac{1}{C})} \implies \approx 2C \text{ items per bin}$$

$$\text{cost} \geq C\left(\frac{n}{2C}\right) = n/2$$

The risk

- Control number of times bins is pushed to limit

The risk

- Control number of times bins is pushed to limit

$$\mathrm{risk}(B) = \mathbf{P}(X_{i_1} \text{ overflows } B) + \mathbf{P}(X_{i_2} \text{ overflows } B) + \cdots$$

The risk

- Control number of times bins is pushed to limit

$$\mathrm{risk}(B) = \mathbf{P}(X_{i_1} \text{ overflows } B) + \mathbf{P}(X_{i_2} \text{ overflows } B) + \cdots$$

  - Greedy tries ≈ 1 times to break bin

The risk

- Control number of times bins is pushed to limit

$$\text{risk}(B) = \mathbf{P}(X_{i_1} \text{ overflows } B) + \mathbf{P}(X_{i_2} \text{ overflows } B) + \cdots$$

  - Greedy tries ≈ 1 times to break bin

- $\mathbf{E}[O_{\mathcal{P}}] = \sum_j \mathbf{E}[\text{risk}(B_j)]$
  - $\text{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C\,\mathbf{E}[O_{\mathcal{P}}]$

The Budgeted Greedy Algorithm

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
    - Item i to bin j pack if $\text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
    - $\text{risk}(B_j) \leftarrow \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

risk	0	0	0	
usage	0	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

$\mathrm{risk}(B_1) + \mathbf{P}(X_1 > 1) \leq \frac{1}{C}$?

risk	0	0	0	
usage	0	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
    - Item i to bin j pack if $\quad \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \le 1/C$
    - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

$$\mathrm{risk}(B_1) + \underset{=0}{\mathbf{P}(X_1 > 1)} \le \frac{1}{C}\text{?}$$

risk	0	0	0	
usage	0	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\text{risk}(B_j) \leftarrow \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \text{Bern}\left(\frac{1}{C}\right)$

$\text{risk}(B_1) + \mathbf{P}(X_1 > 1) \leq \frac{1}{C}$?

$= 0$

risk	0	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\quad \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

risk	0	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

$\mathrm{risk}(B_1) + \mathbf{P}(X_2 + 1 > 1) \leq \frac{1}{C}$?

risk	0	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a "risk" budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\text{risk}(B_j) \leftarrow \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \text{Bern}\left(\frac{1}{C}\right)$

$$\text{risk}(B_1) + \mathbf{P}(X_2 + 1 > 1) \leq \tfrac{1}{C}\text{?}$$
$$= \tfrac{1}{C}$$

risk	0	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\quad \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\text{risk}(B_j) \leftarrow \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \text{Bern}\left(\frac{1}{C}\right)$

$$\text{risk}(B_1) + \mathbf{P}(X_2 + 1 > 1) \leq \tfrac{1}{C}\text{?}$$

$$= \tfrac{1}{C}$$

risk	$\frac{1}{C}$	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\quad \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

risk	$\frac{1}{C}$	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a "risk" budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

$\mathrm{risk}(B_1) + \mathbf{P}(X_3 + 1 > 1) \leq \frac{1}{C}$?

risk	$\frac{1}{C}$	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a "risk" budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\text{risk}(B_j) \leftarrow \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \text{Bern}\left(\frac{1}{C}\right)$

$$\text{risk}(B_1) + \mathbf{P}(X_3 + 1 > 1) \leq \tfrac{1}{C}\,?$$

$$= \tfrac{2}{C}$$

risk	$\frac{1}{C}$	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a "risk" budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

$$\mathrm{risk}(B_2) + \mathbf{P}(X_3 > 1) \leq \tfrac{1}{C}?$$
$$= 0$$

risk	$\frac{1}{C}$	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a “risk” budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
  - Item i to bin j pack if $\mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
  - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Ex: $X_1, X_2, X_3, \ldots \sim \mathrm{Bern}\left(\frac{1}{C}\right)$

Can show $\mathrm{cost} \leq 2\frac{n}{C} + 1$

risk	$\frac{1}{C}$	0	0	
usage	1	0	0	
bins	B_1	B_2	B_3	$\cdots$

The Budgeted Greedy Algorithm

- Put a "risk" budget of $1/C$ for each bin
- Initialize risk at 0 for each bin
    - Item i to bin j pack if $\quad \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
    - $\mathrm{risk}(B_j) \leftarrow \mathrm{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Proposition

For bins in Budgeted Greedy, $\mathrm{risk}(B_j) \leq \frac{1}{C}$. Therefore,

$$\begin{aligned} \mathrm{cost}(\mathrm{ALG}) &= \mathbf{E}[N_{\mathrm{ALG}}] + C \cdot \mathbf{E}[O_{\mathrm{ALG}}] \\ &= \mathbf{E}[N_{\mathrm{ALG}}] + C \cdot \sum_j \mathbf{E}[\mathrm{risk}(B_j)] \\ &\leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]. \end{aligned}$$

Where we are ...

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- ~~Phase 1~~: ✓
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}]$ = sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$
- Phase 2:
  - Only bounded-risk policies are interesting: $\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 4 \cdot \mathrm{cost}(\mathcal{P}_{\textsf{old}})$
- Phase 3:
  - $\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P}\ \textsf{bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$

Only bounded risk policies are interesting

Only bounded risk policies are interesting

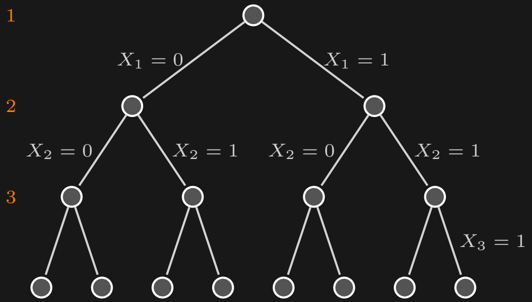
Theorem

For any policy $\mathcal{P}$ for packing $X_1, \ldots, X_n$, there is a policy $\mathcal{P}'$ with risk $\leq 1/C$ such that $\mathrm{cost}(\mathcal{P}') \leq 4\,\mathrm{cost}(\mathcal{P})$.

Only bounded risk policies are interesting

Theorem

For any policy $\mathcal{P}$ for packing $X_1, \ldots, X_n$, there is a policy $\mathcal{P}'$ with risk $\leq 1/C$ such that $\mathrm{cost}(\mathcal{P}') \leq 4\,\mathrm{cost}(\mathcal{P})$.

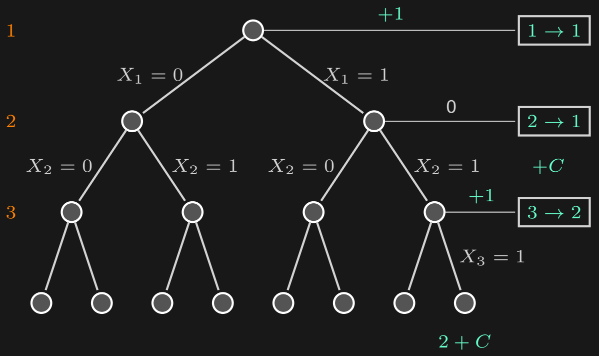


$X_1, X_2, X_3 \sim \mathrm{Bern}(1/C)$

Only bounded risk policies are interesting

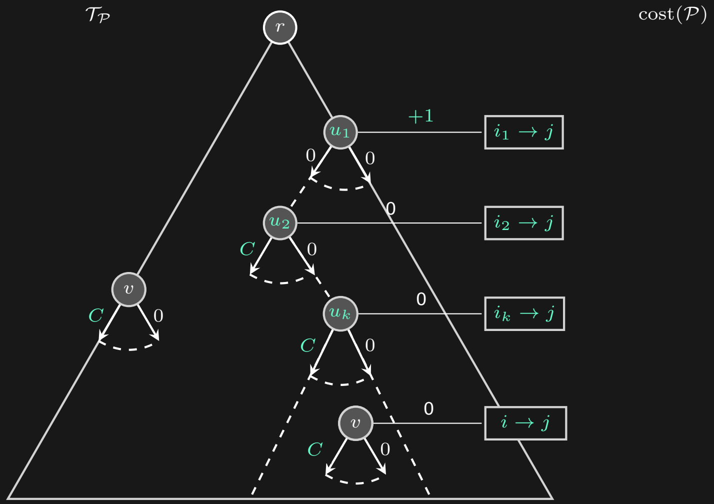
Theorem

For any policy $\mathcal{P}$ for packing $X_1, \ldots, X_n$, there is a policy $\mathcal{P}'$ with risk $\leq 1/C$ such that $\text{cost}(\mathcal{P}') \leq 4\,\text{cost}(\mathcal{P})$.



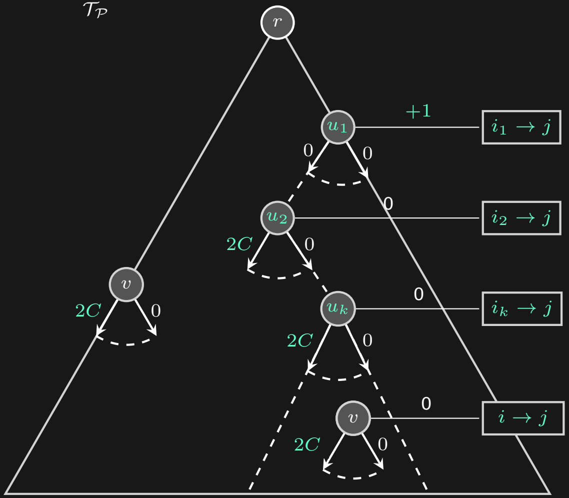
$X_1, X_2, X_3 \sim \text{Bern}(1/C)$

Proof

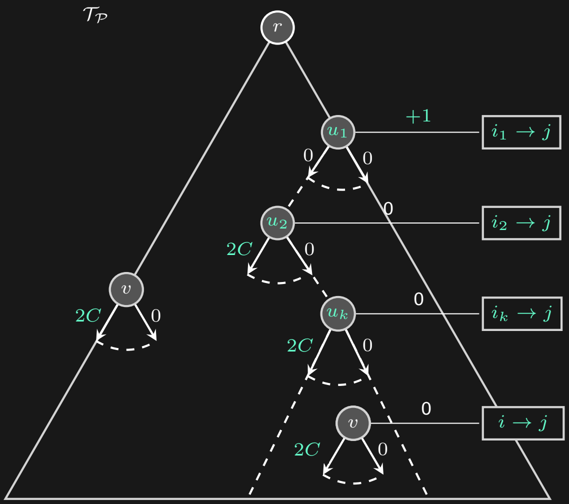

$\mathcal{T}_{\mathcal{P}}$
r
$\text{cost}(\mathcal{P})$
u_1
$+1$
$i_1 \to j$
0
0
0
u_2
$i_2 \to j$
C
0
v
C
0
u_k
0
$i_k \to j$
C
0
v
0
$i \to j$
C
0

Proof

$\mathrm{cost}'(\mathcal{P}) \leq 2\,\mathrm{cost}(\mathcal{P})$



Proof



$\mathrm{cost}'(\mathcal{P}) \leq 2\,\mathrm{cost}(\mathcal{P})$

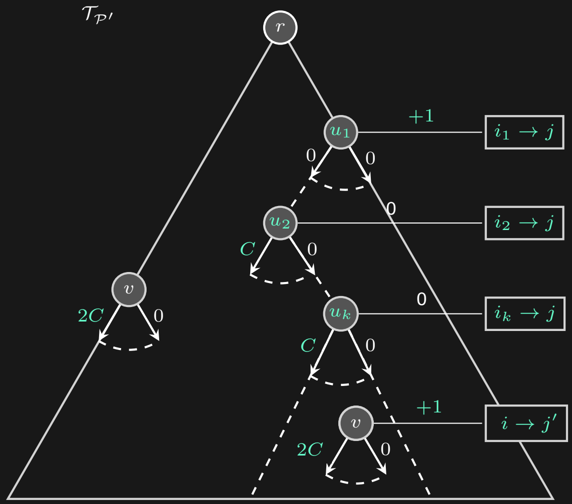
$\mathrm{risk}_j = \mathbf{P}(X_{i_1} > 1) \leq \frac{1}{C}$

$\mathrm{risk}_j \leq \frac{1}{C}$

$\mathrm{risk}_j > \frac{1}{C}$

$\mathbf{P}(X_{i_1}$ overflows $j) +$
$\mathbf{P}(X_{i_2}$ overflows $j) + \cdots$

Proof



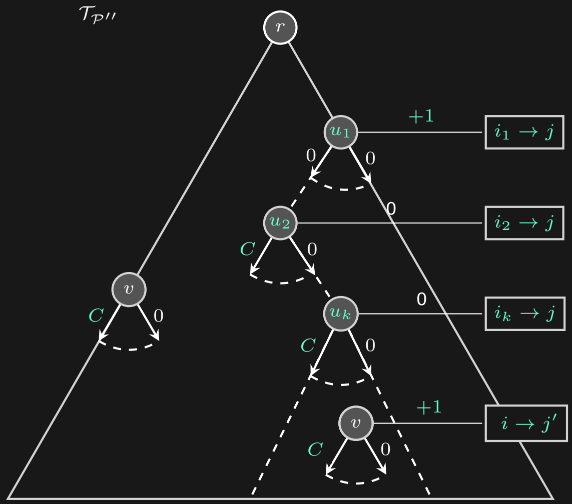
$\mathrm{cost}''(\mathcal{P}') \leq 2\,\mathrm{cost}(\mathcal{P})$

$\mathrm{risk}_j = \mathbf{P}(X_{i_1} > 1) \leq \frac{1}{C}$

$\mathrm{risk}_j \leq \frac{1}{C}$

$\mathrm{risk}_j > \frac{1}{C}$

Proof



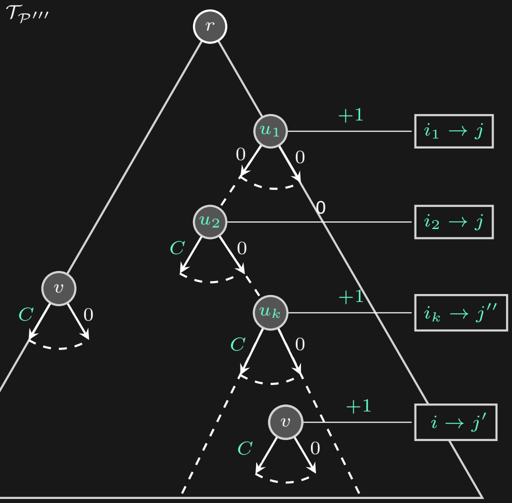
$\mathrm{cost}(\mathcal{P}'') \leq 2\,\mathrm{cost}(\mathcal{P})$

$\mathrm{risk}_j = \mathbf{P}(X_{i_1} > 1) \leq \frac{1}{C}$

$\mathrm{risk}_j \leq \frac{1}{C}$

$\mathrm{risk}_j > \frac{1}{C}$

Proof



$\mathrm{cost}(\mathcal{P}''') \leq 4\,\mathrm{cost}(\mathcal{P})$

$\mathrm{risk}_j = \mathbf{P}(X_{i_1} > 1) \leq \frac{1}{C}$

$\mathrm{risk}_j \leq \frac{1}{C}$

Where we are ...

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- ~~Phase 1~~: ✓
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}] =$ sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$
- ~~Phase 2~~: ✓
  - Only bounded-risk policies are interesting: $\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 4 \cdot \mathrm{cost}(\mathcal{P}_{\textsf{old}})$
- Phase 3:
  - $\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\substack{\mathcal{P}\ \text{bounded} \\ \text{risk}}} \mathbf{E}[N_{\mathcal{P}}]$

Number of Bins of Budgeted Greedy

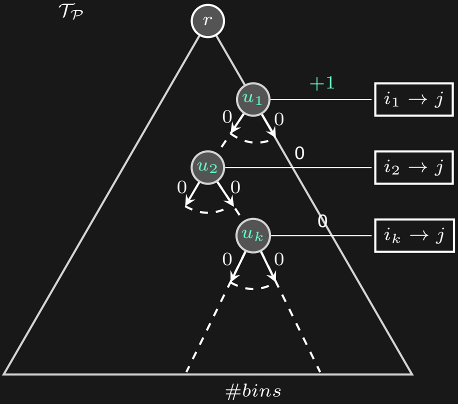
Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P}\ \substack{\textsf{bounded}\\\textsf{risk}}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

Number of Bins of Budgeted Greedy

Theorem

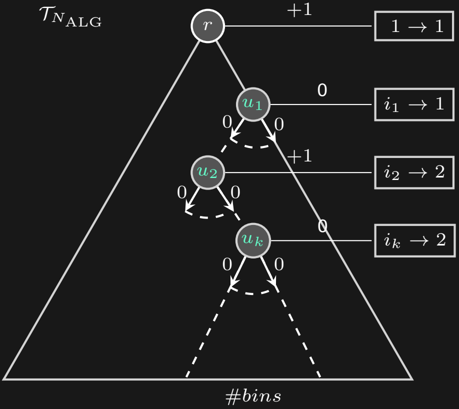
$\mathbf{E}[N_{\text{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.



Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\text{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

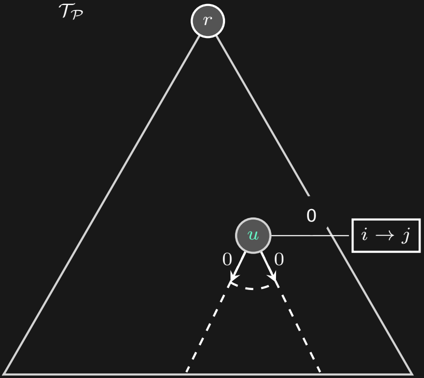


- i to bin j pack if $\text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j) \leq 1/C$
- $\text{risk}(B_j) \leftarrow \text{risk}(B_j) + \mathbf{P}(X_i \text{ overflow } j)$

Number of Bins of Budgeted Greedy

Theorem

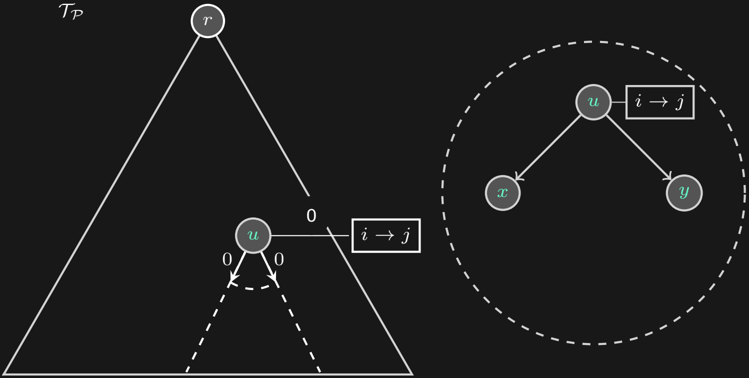
$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P}\ \text{bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.



Number of Bins of Budgeted Greedy

Theorem

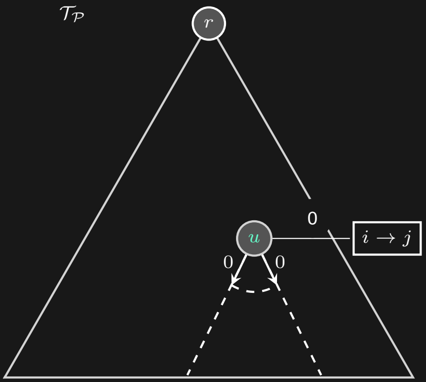
$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

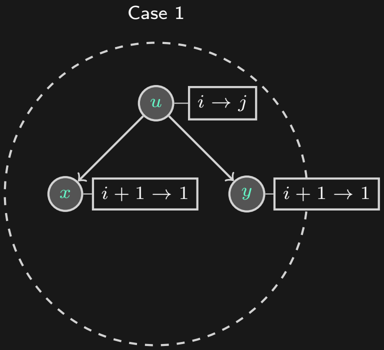


Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

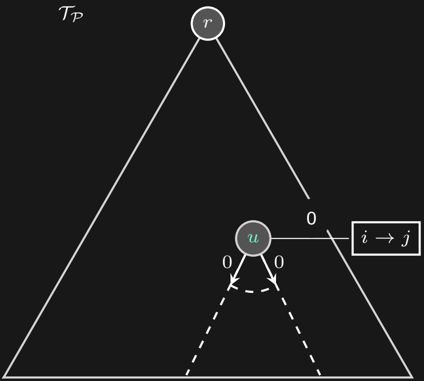


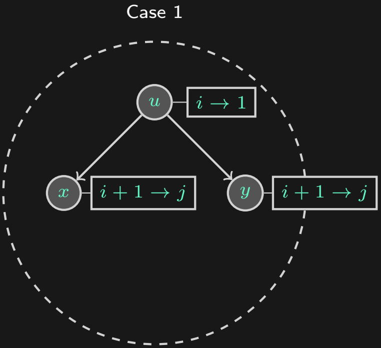


Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

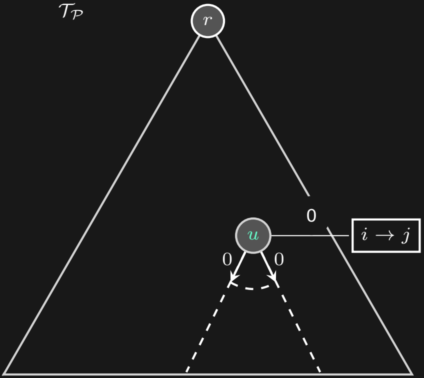




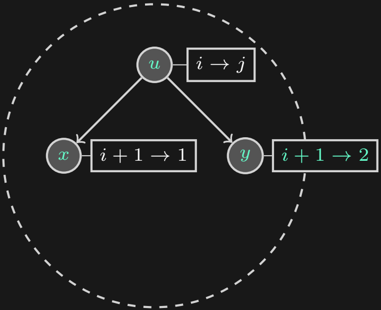
Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P}\ \text{bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.



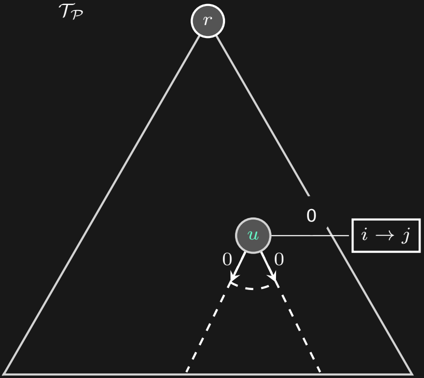
Case 2

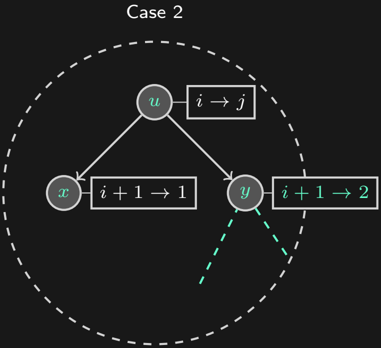


Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

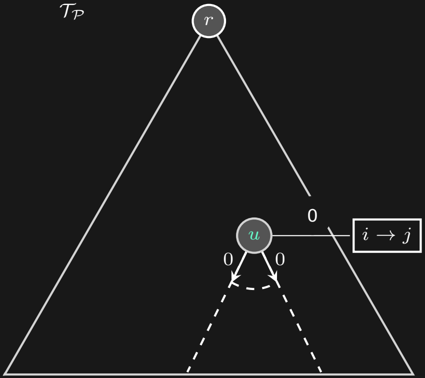


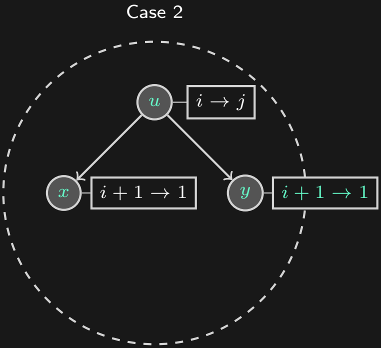


Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

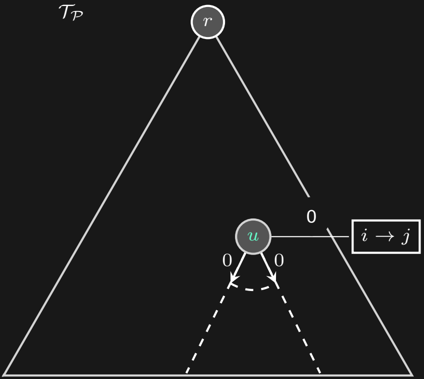


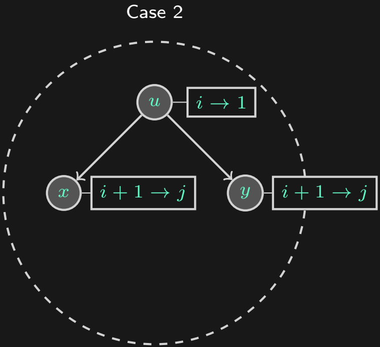


Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P}\ \text{bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.

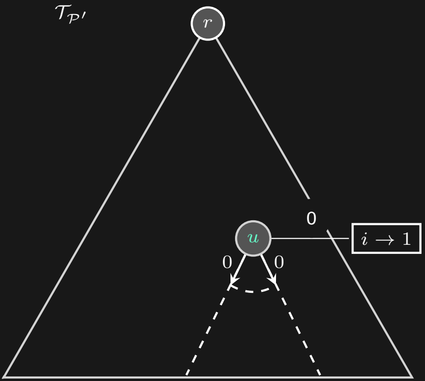




Number of Bins of Budgeted Greedy

Theorem

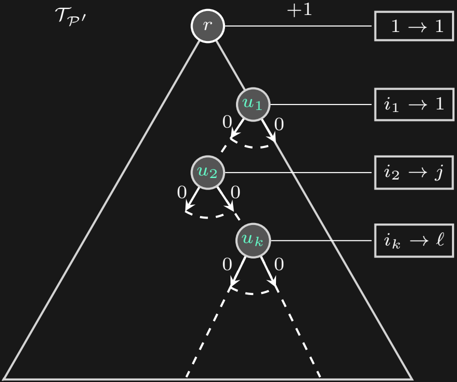
$\mathbf{E}[N_{\text{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.



Number of Bins of Budgeted Greedy

Theorem

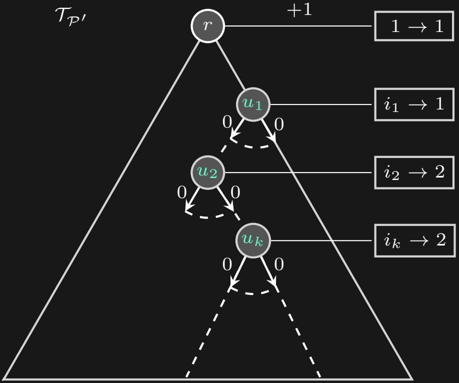
$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P} \text{ bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.



Number of Bins of Budgeted Greedy

Theorem

$\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\mathcal{P}\ \text{bounded risk}} \mathbf{E}[N_{\mathcal{P}}]$ if $X_1, \ldots, X_n$ are i.i.d.



Then ...

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- ~~Phase 1~~: ✓
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}]$ = sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$
- ~~Phase 2~~: ✓
  - Only bounded-risk policies are interesting: $\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 4 \cdot \mathrm{cost}(\mathcal{P}_{\textsf{old}})$
- ~~Phase 3~~: ✓
  - $\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\substack{\mathcal{P}\ \textsf{bounded} \\ \textsf{risk}}} \mathbf{E}[N_{\mathcal{P}}]$

Then ...

- $\mathrm{cost}(\mathcal{P}) = \mathbf{E}[N_{\mathcal{P}}] + C \cdot \mathbf{E}[O_{\mathcal{P}}]$
- ~~Phase 1~~: ✓
  - Notion of risk of bin → $\mathbf{E}[O_{\mathcal{P}}]$ = sum of risks of bins
  - $\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}]$
- ~~Phase 2~~: ✓
  - Only bounded-risk policies are interesting: $\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 4 \cdot \mathrm{cost}(\mathcal{P}_{\textsf{old}})$
- ~~Phase 3~~: ✓
  - $\mathbf{E}[N_{\mathrm{ALG}}] = \min_{\substack{\mathcal{P}\ \textsf{bounded} \\ \textsf{risk}}} \mathbf{E}[N_{\mathcal{P}}]$

$$\mathrm{cost}(\mathrm{ALG}) \leq 2\,\mathbf{E}[N_{\mathrm{ALG}}] \leq 2\,\mathbf{E}[N_{\mathcal{P}_{\textsf{new}}}] \leq 2\,\mathrm{cost}(\mathcal{P}_{\textsf{new}}) \leq 8\,\mathrm{cost}(\mathcal{P}_{\textsf{old}})$$

□

Some comments

- Careful analysis gives us the factor $3 + 2\sqrt{2} \approx 5.828$
- Budgeted Greedy also exhibits $\mathcal{O}(\log C)$ factor against *arbitrary* exponential distributions
- There are non-identical distributions in which Budgeted Greedy fails:
  - $\mathrm{cost}(\mathrm{ALG}) = n/2$ but $\mathrm{cost}(\mathrm{OPT}) \leq n/C$
- Still open for arbitrary distributions

Summary

- New packing model for items with random sizes
  - Outcomes are observed right after packing item
  - Overflowing bins incur in penalty

Model	Positive results(?)	Hardness(?)
Online	$3+2\sqrt{2}$ factor for i.i.d. input $\mathcal{O}(\log C)$ factor for exponential $\mathcal{O}(1)$ factor for exponential distributions if rates are larger than $2\log C$	$\Omega(\sqrt{\log C})$ factor for exp.
Offline	PTAS with extra capacity	$\#\mathbf{P}$-hard

- Preprint: `https://arxiv.org/abs/2007.11532v2`

- My site: `https://sites.google.com/view/sebastianps`