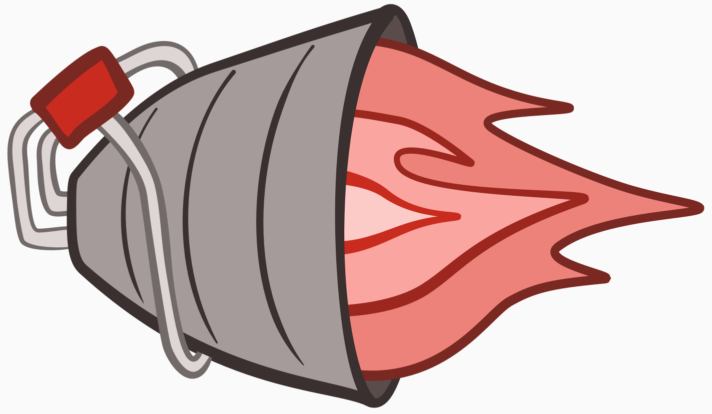
Strongly Tail-Optimal Scheduling *in the Light-Tailed M/G/1*

Ziv Scully Cornell ORIE

Joint work with

George Yu Cornell ORIE
Amit Harlev Cornell CAM



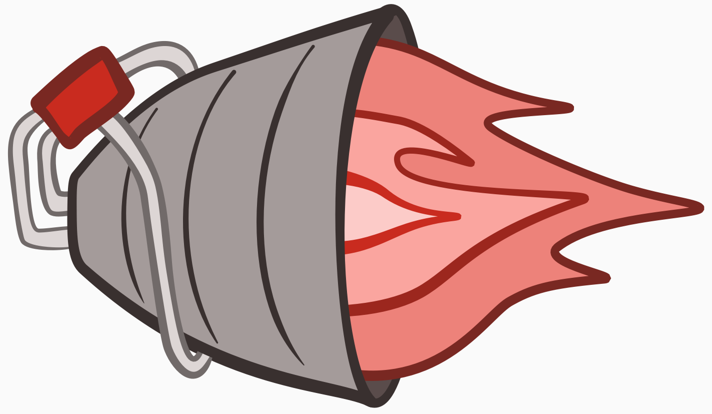
Strongly Tail-Optimal Scheduling *in the Light-Tailed M/G/1*

Ziv Scully Cornell ORIE

Joint work with

George Yu Cornell ORIE
Amit Harlev Cornell CAM

Best paper award at SIGMETRICS 2024



Strongly Tail-Optimal Scheduling *in the Light-Tailed M/G/1*

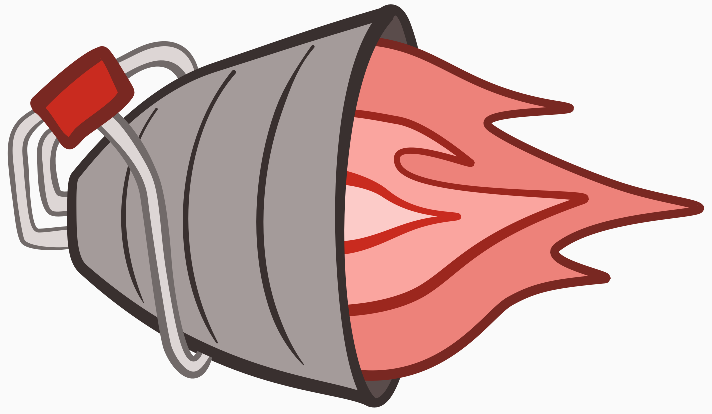
Ziv Scully Cornell ORIE

Joint work with

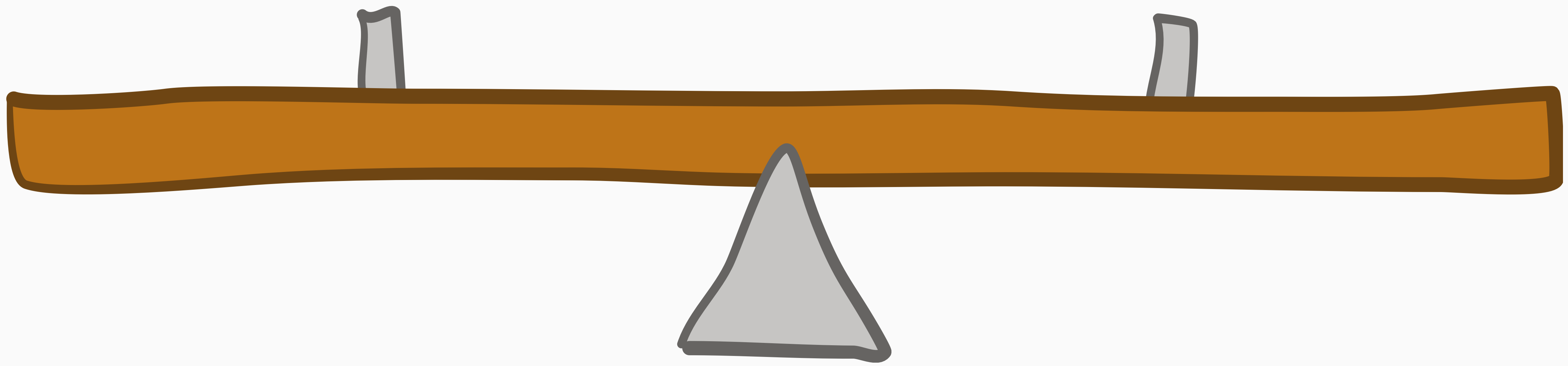
George Yu Cornell ORIE
Amit Harlev Cornell CAM

Best paper award at SIGMETRICS 2024

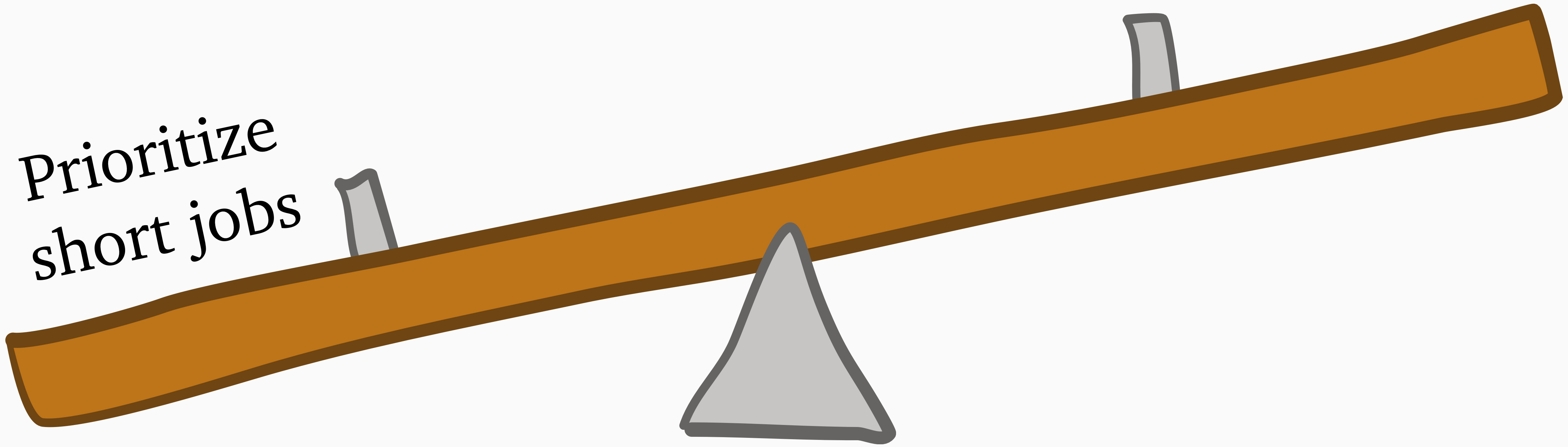
Preprint coming soon…



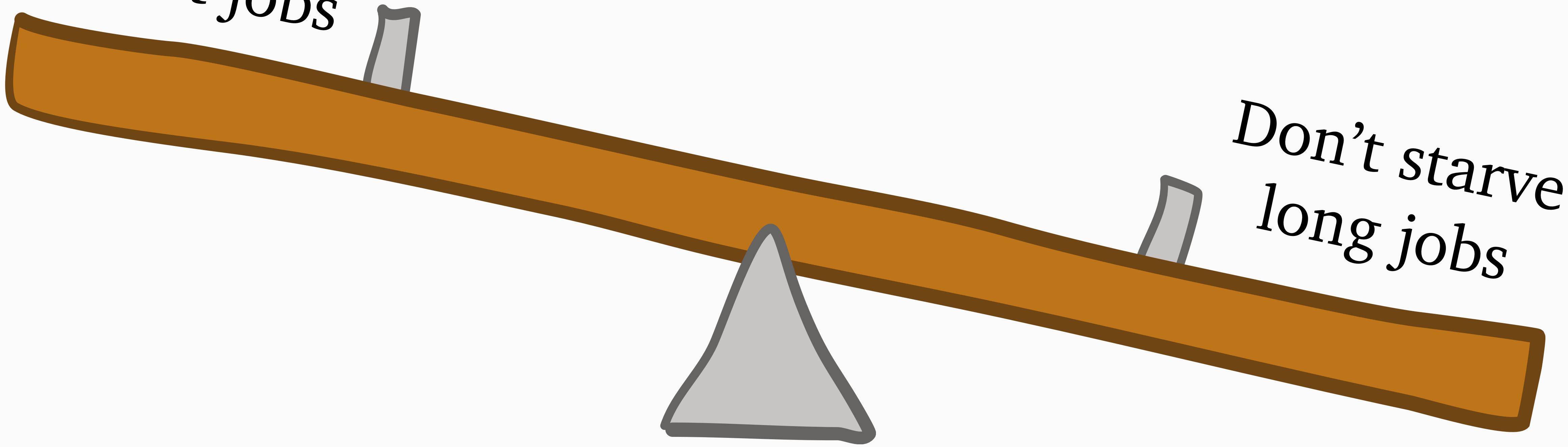
Tradeoff: priority vs. starvation



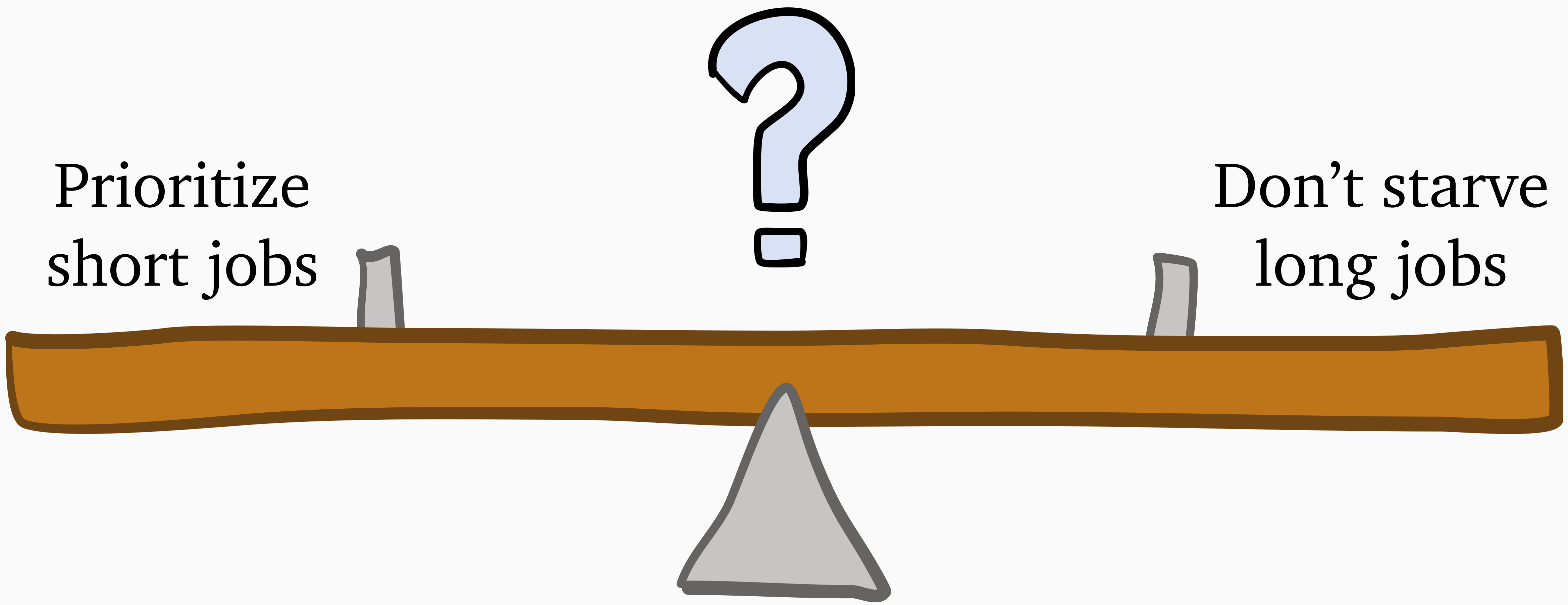
Tradeoff: priority vs. starvation



Tradeoff: priority vs. starvation

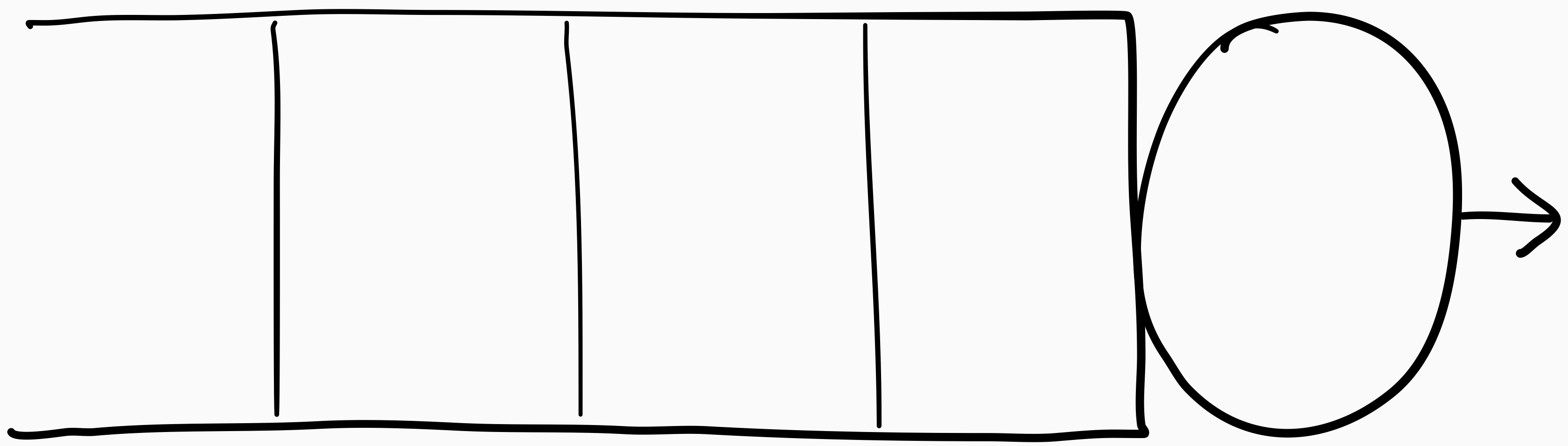


Tradeoff: priority vs. starvation

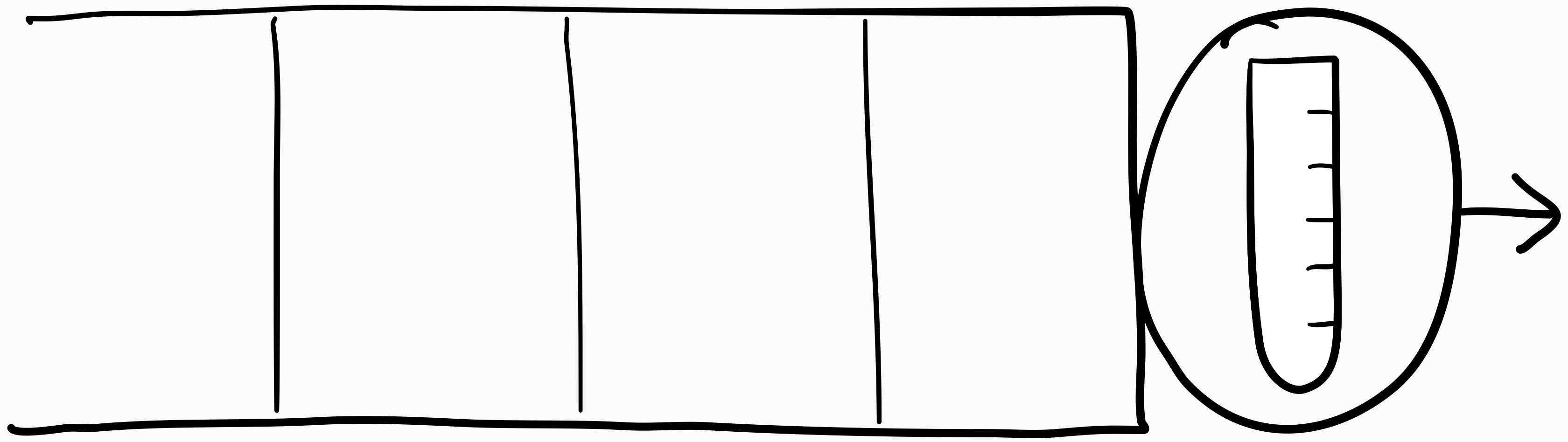


How should we schedule jobs to minimize delay?

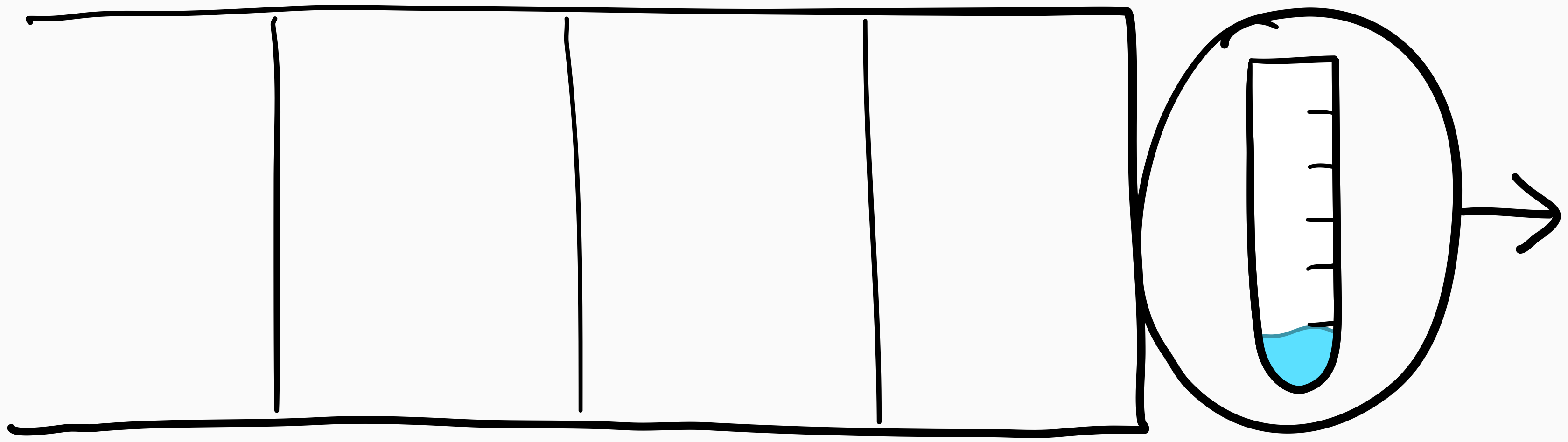
How should we schedule jobs to minimize delay?



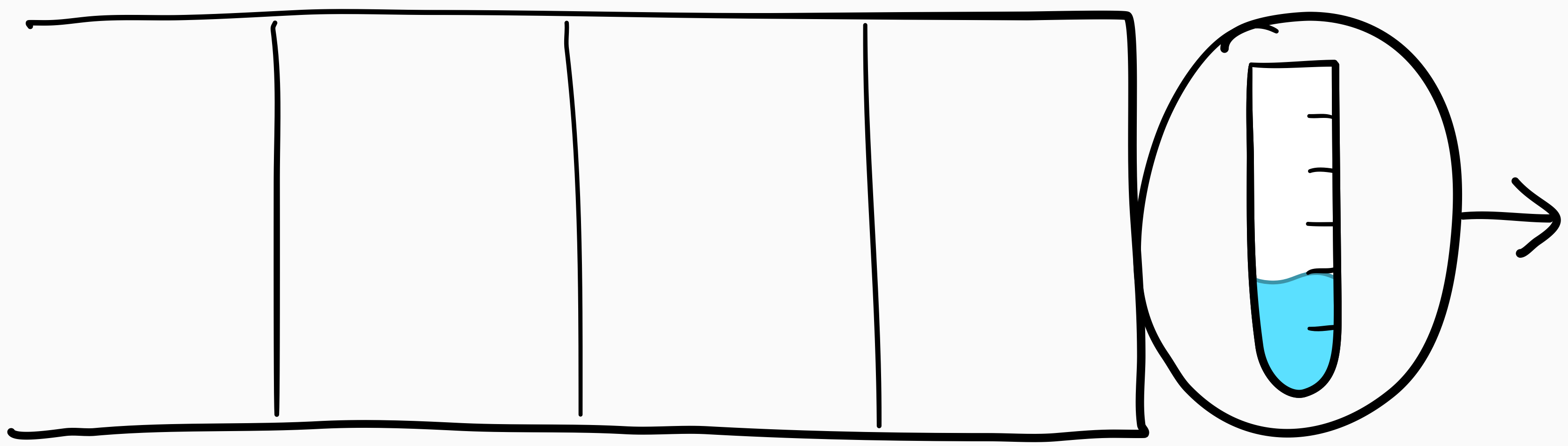
How should we schedule jobs to minimize delay?



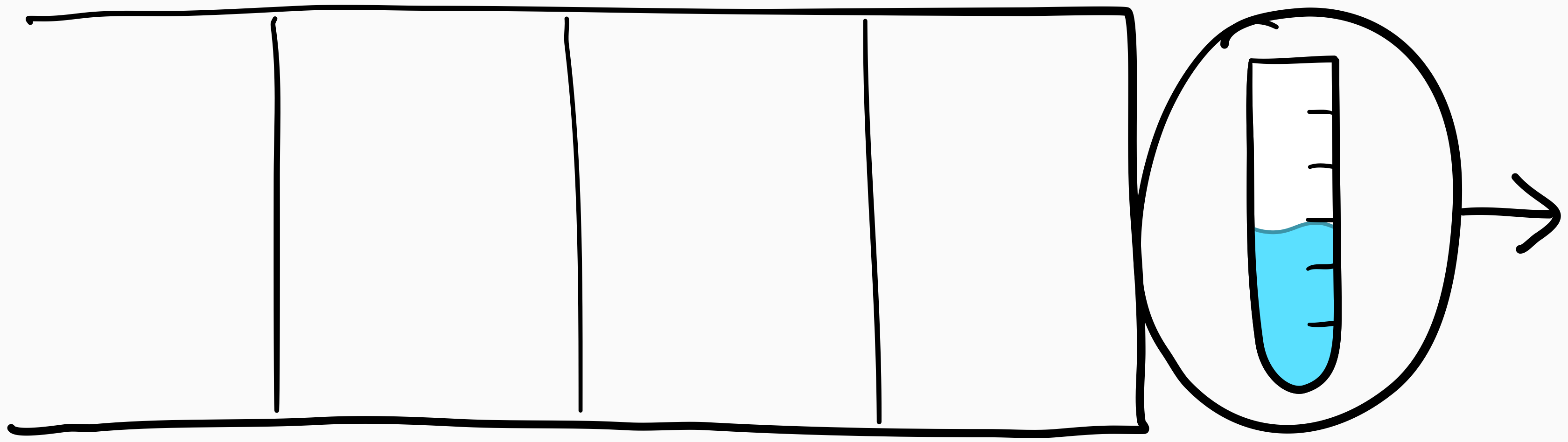
How should we schedule jobs to minimize delay?



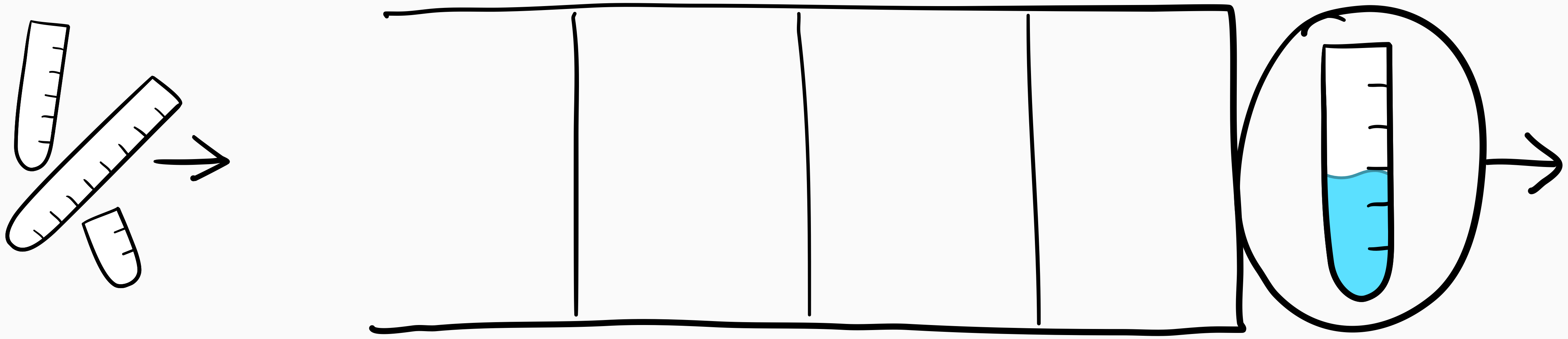
How should we schedule jobs to minimize delay?



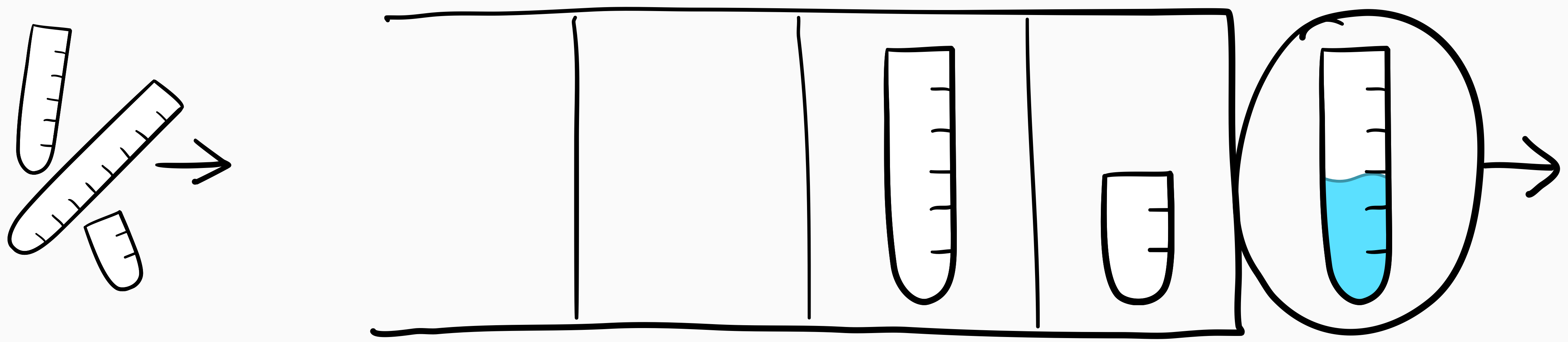
How should we schedule jobs to minimize delay?



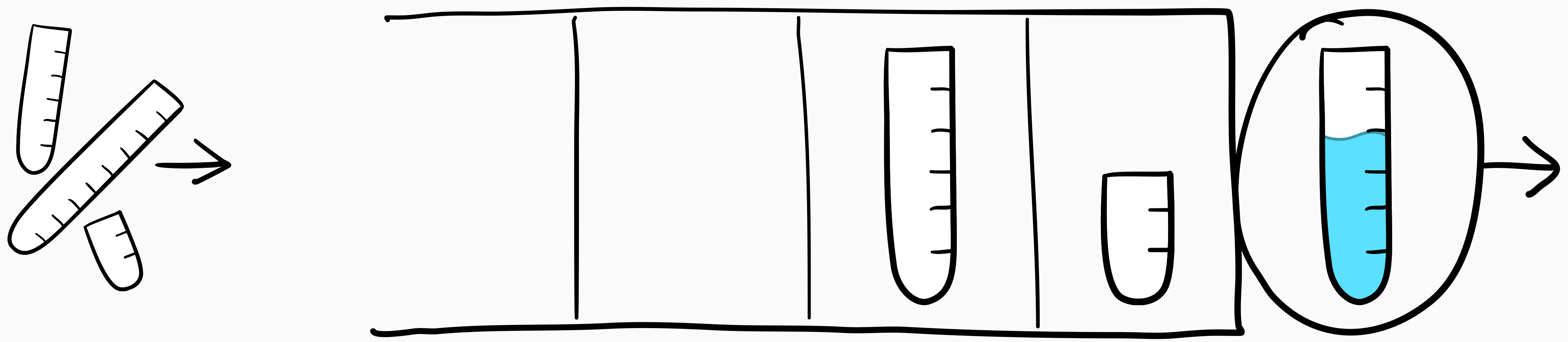
How should we schedule jobs to minimize delay?



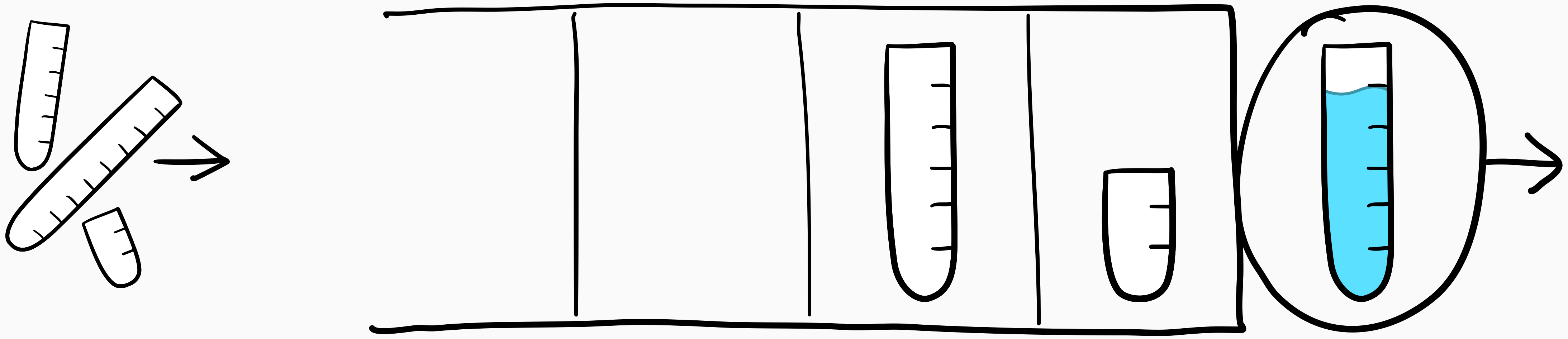
How should we schedule jobs to minimize delay?



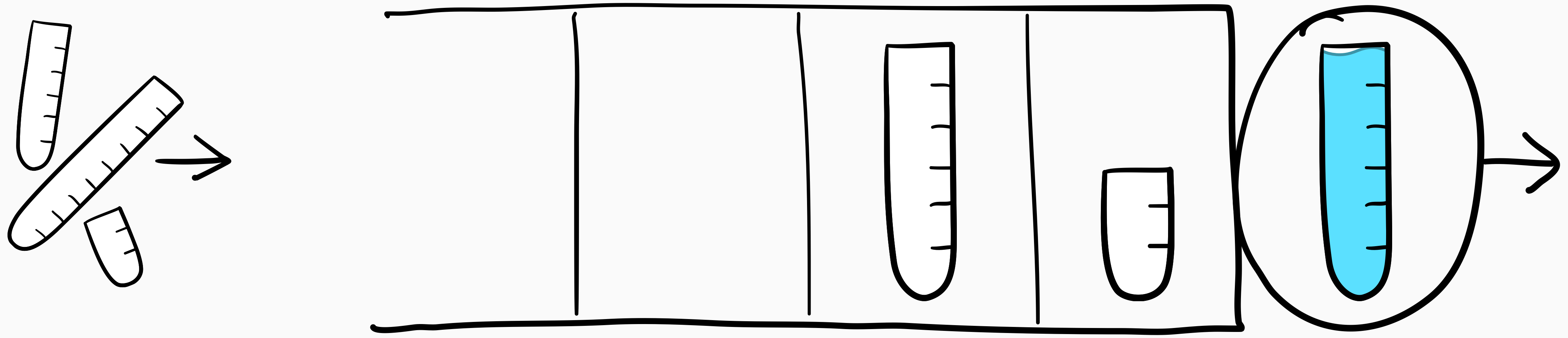
How should we schedule jobs to minimize delay?



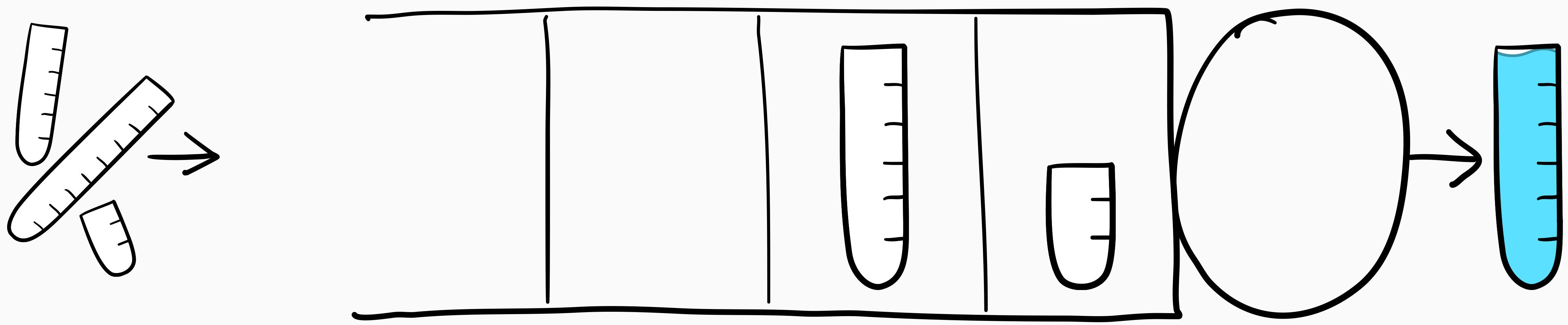
How should we schedule jobs to minimize delay?



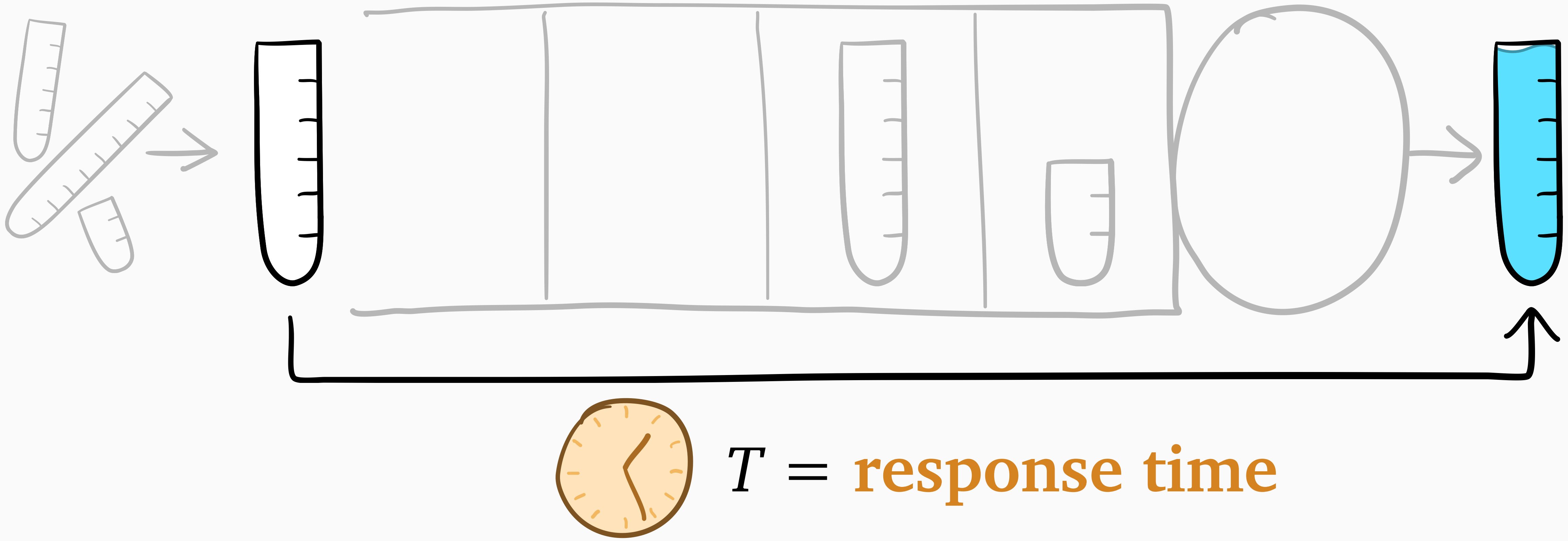
How should we schedule jobs to minimize delay?



How should we schedule jobs to minimize delay?

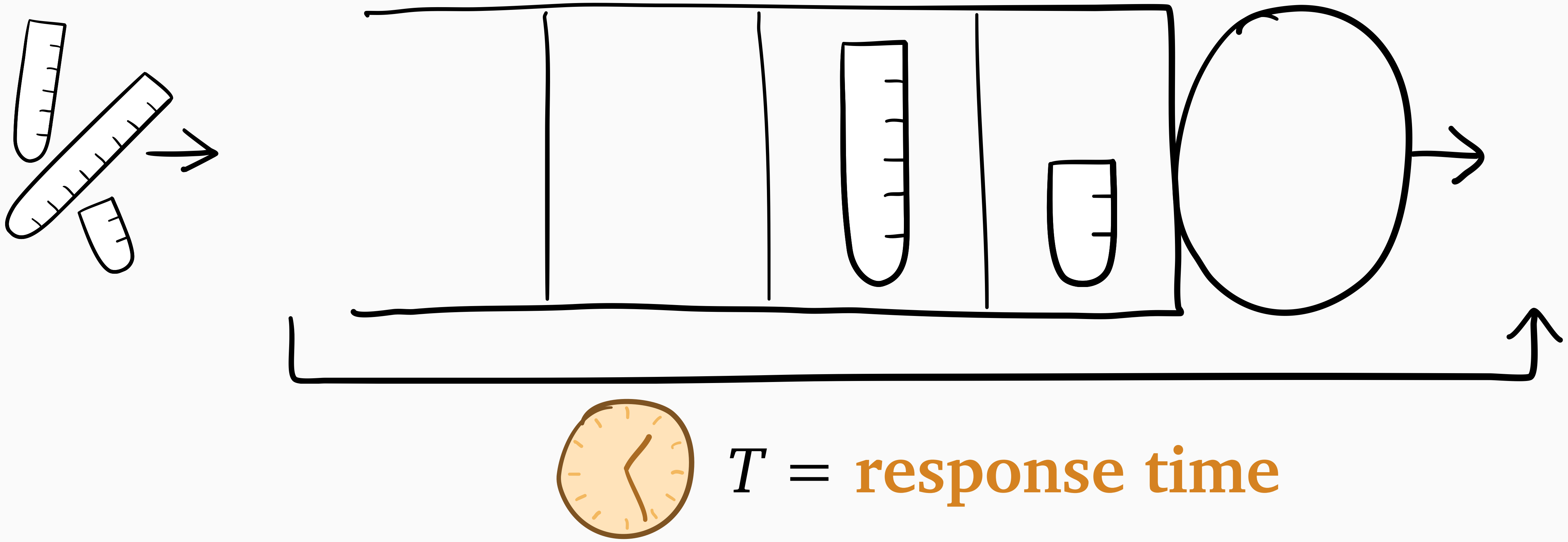


How should we schedule jobs to minimize delay?



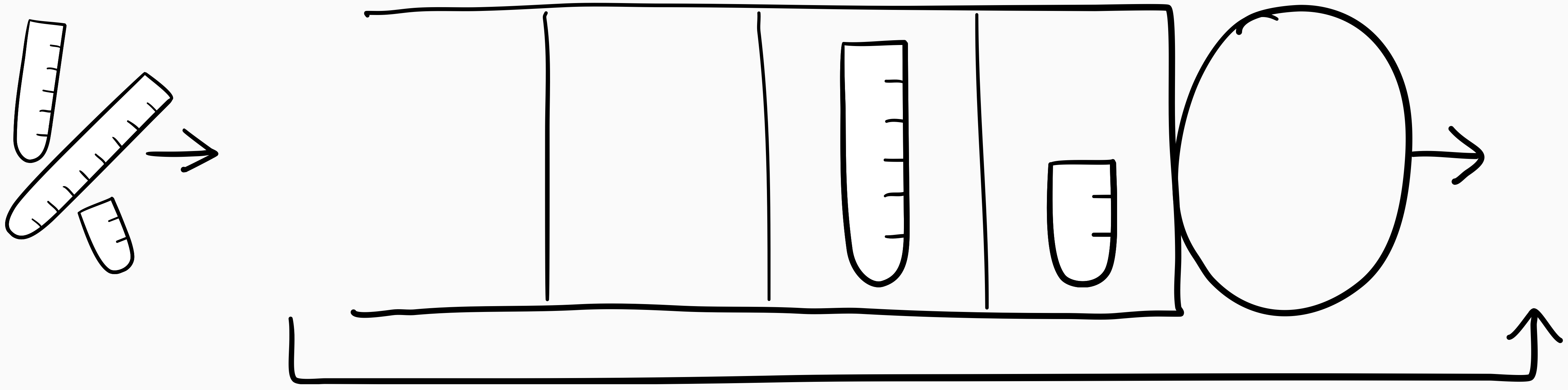
T = **response time**

How should we schedule jobs to minimize delay?



T = **response time**

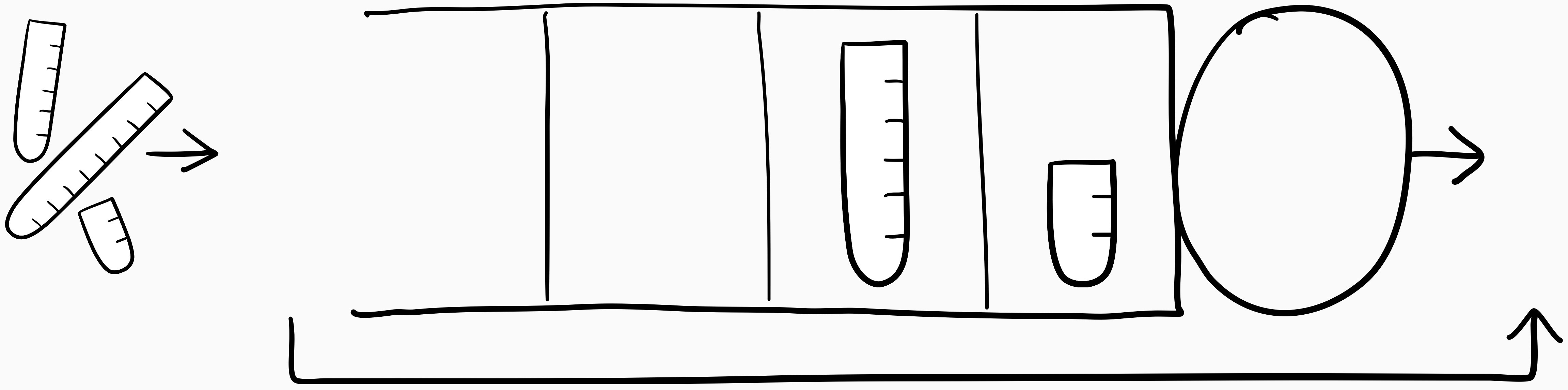
How should we schedule jobs to minimize delay?



T = **response time**

Minimize $\mathbf{E}[T]$?

How should we schedule jobs to minimize delay?

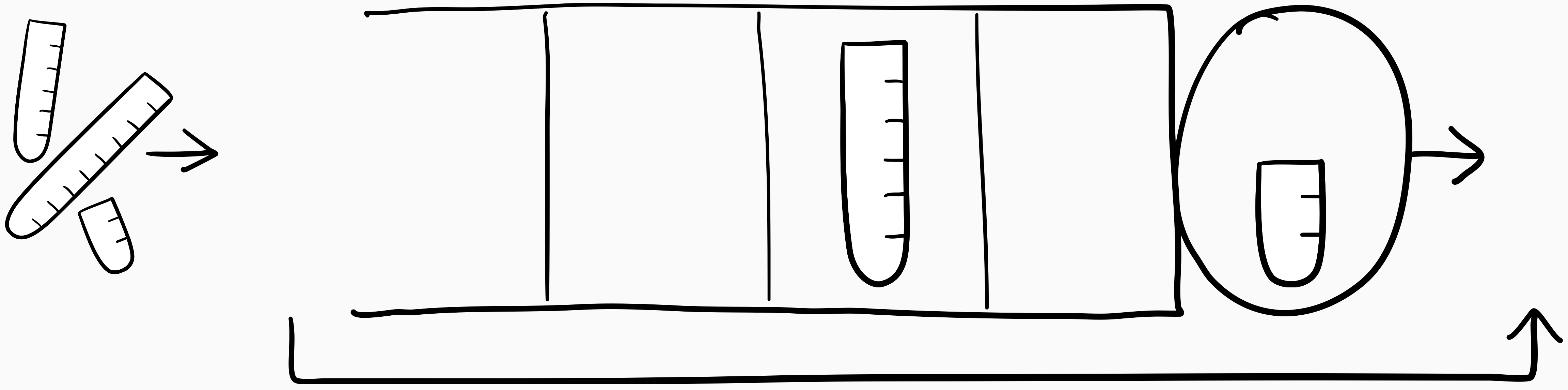


T = **response time**

Minimize $\mathbf{E}[T]$?

Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

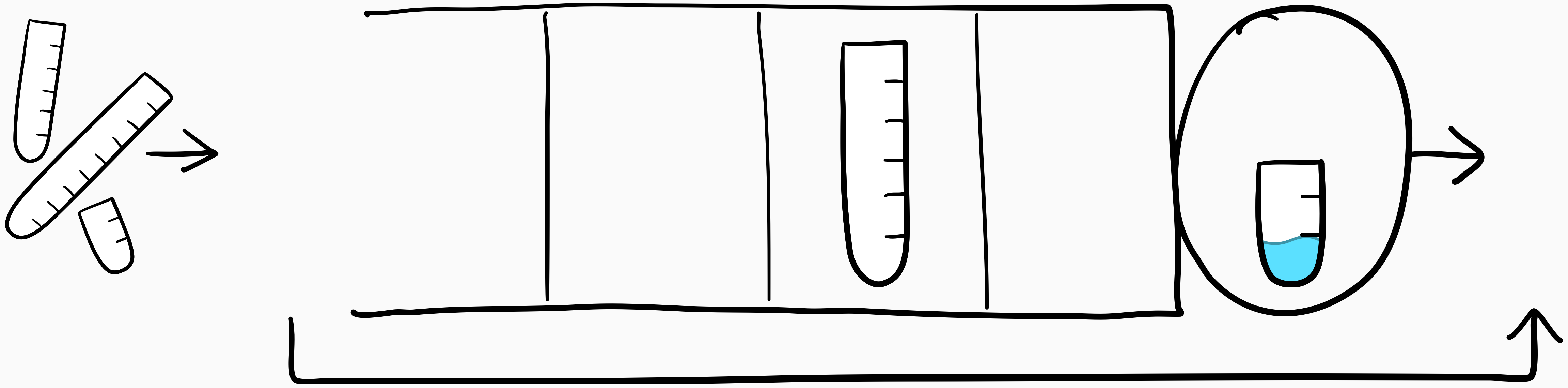


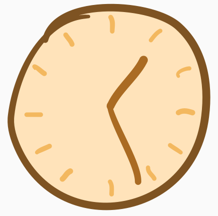
T = **response time**

Minimize $\mathbf{E}[T]$?

Serve short jobs before long jobs

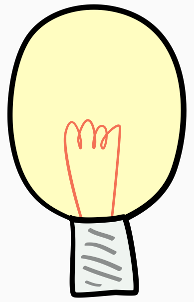
How should we schedule jobs to minimize delay?





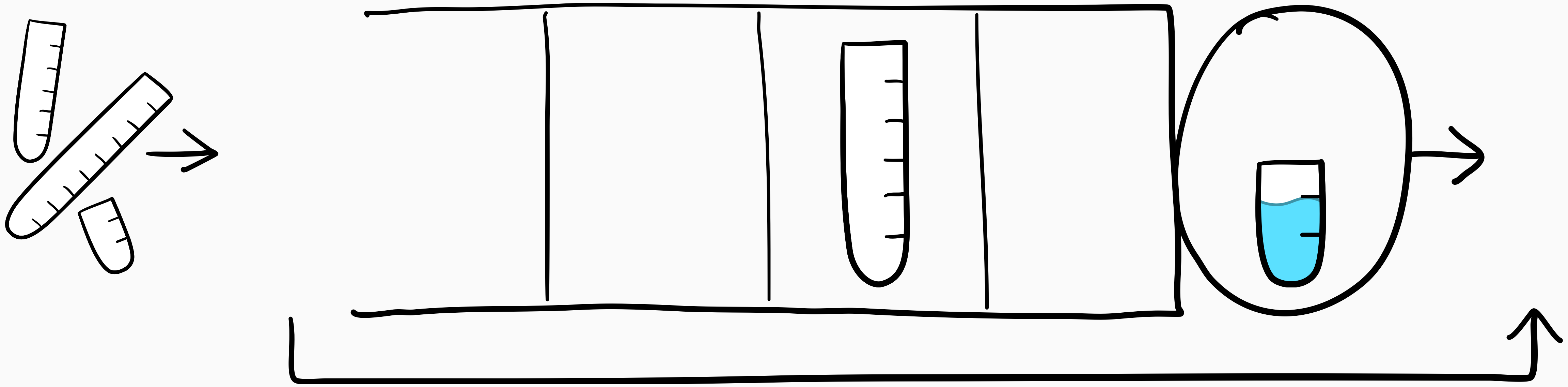
T = **response time**

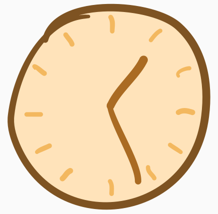
Minimize $\mathbf{E}[T]$?



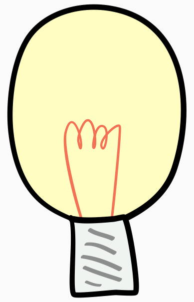
Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

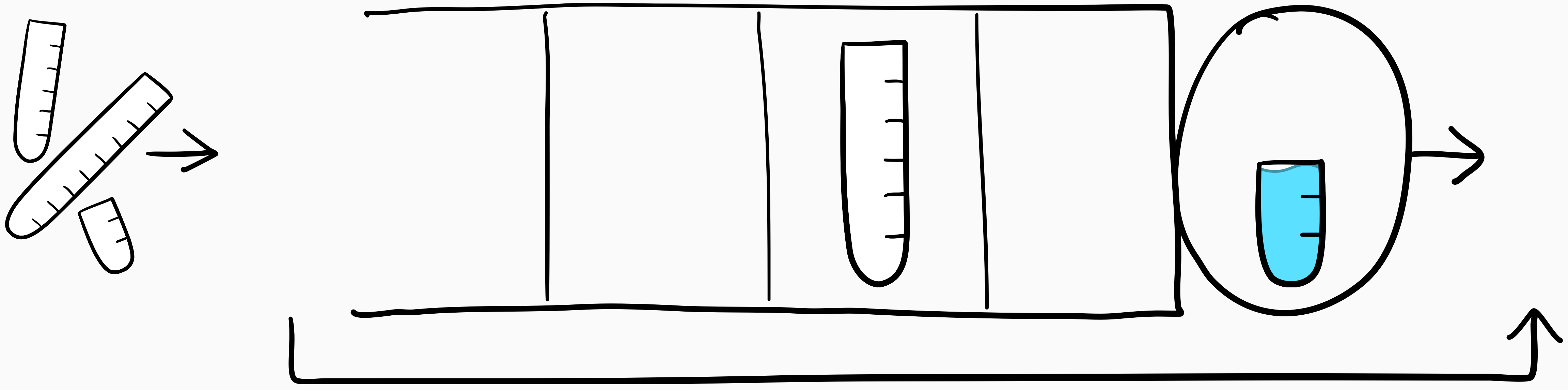


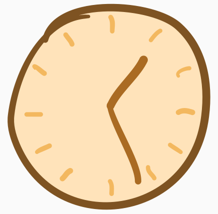
 T = **response time**

 Minimize $\mathbf{E}[T]$?

 Serve short jobs
before long jobs

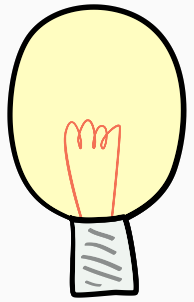
How should we schedule jobs to minimize delay?





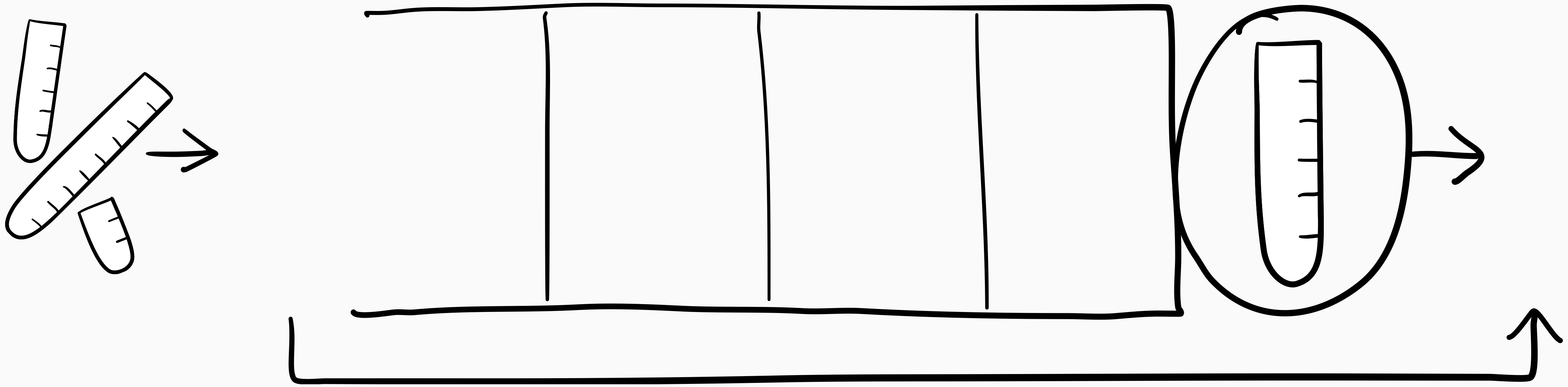
T = **response time**

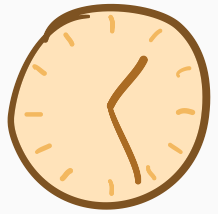
Minimize $\mathbf{E}[T]$?



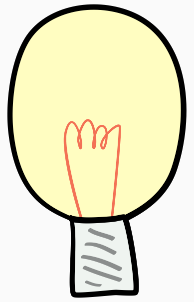
Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?

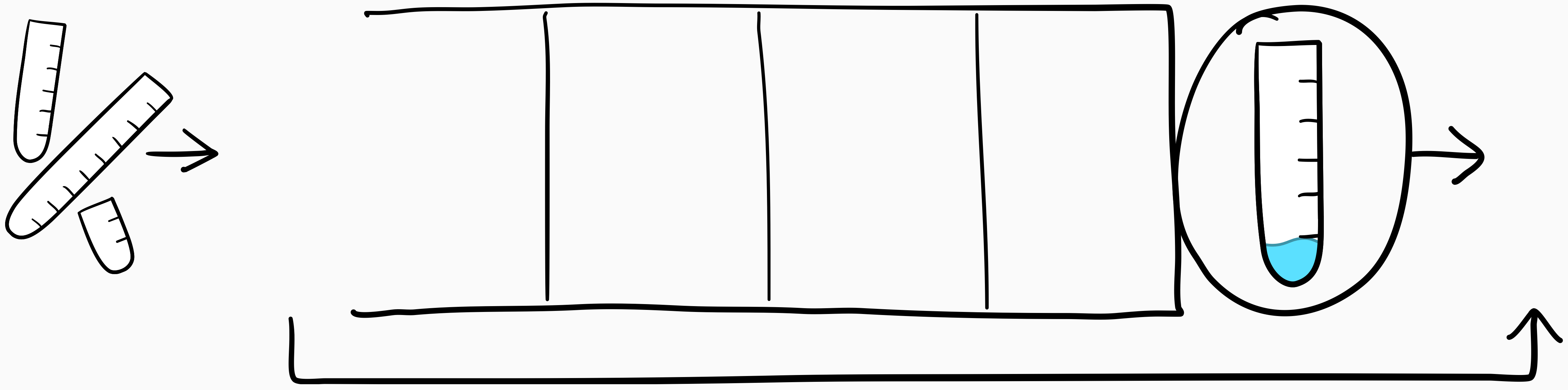


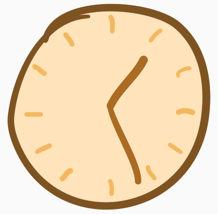
 T = **response time**

 Minimize $\mathbf{E}[T]$?

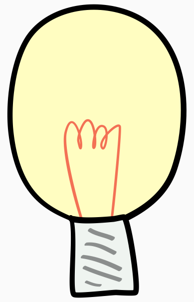
 Serve short jobs before long jobs

How should we schedule jobs to minimize delay?

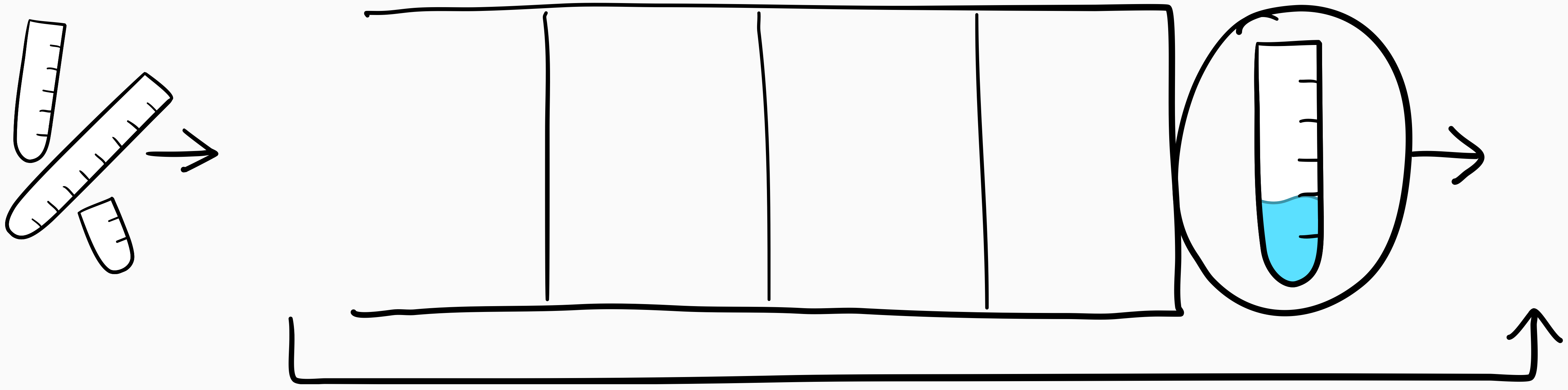


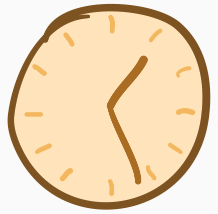
 T = **response time**

 Minimize $\mathbf{E}[T]$?

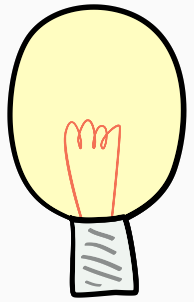
 Serve short jobs before long jobs

How should we schedule jobs to minimize delay?

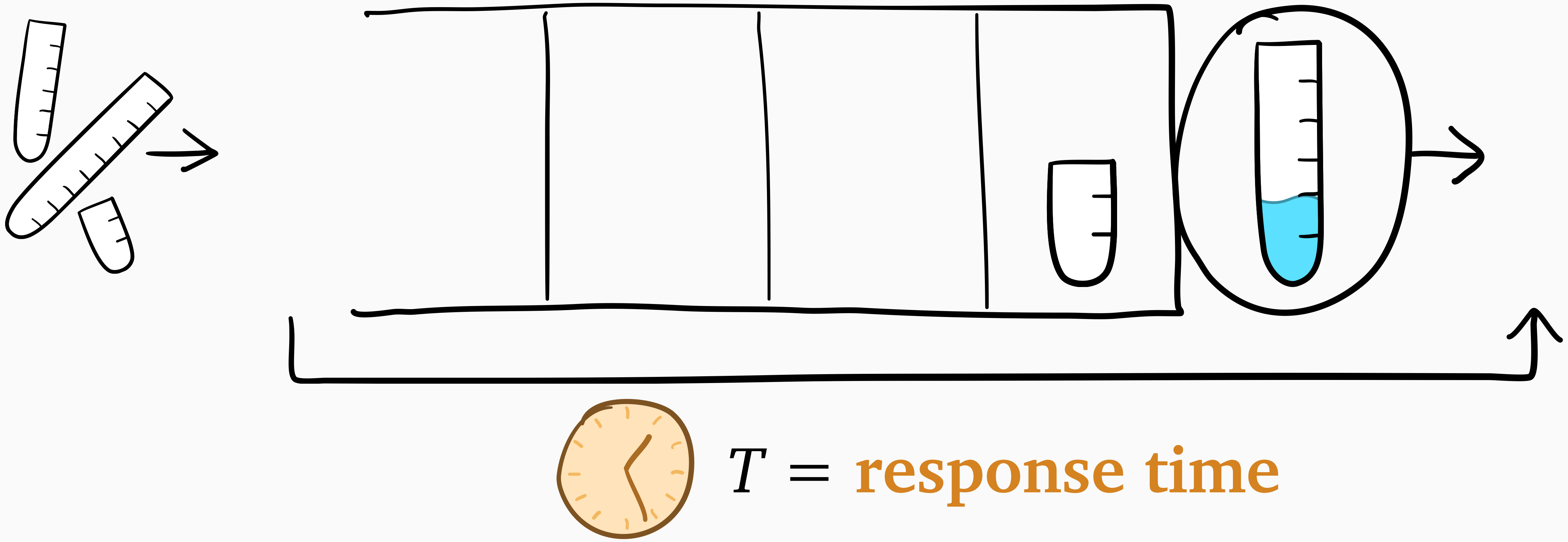


 T = **response time**

 Minimize $\mathbf{E}[T]$?

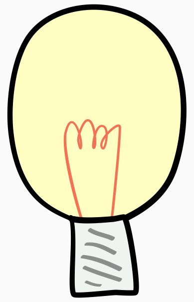
 Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?



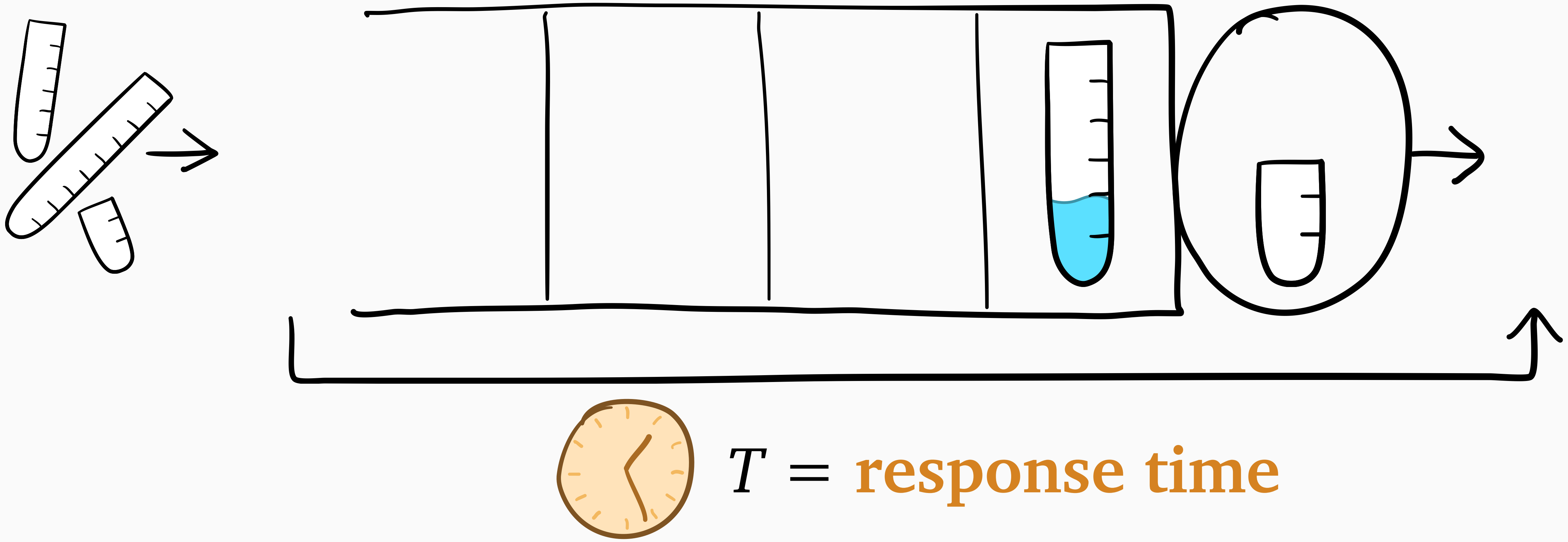
T = **response time**

Minimize $\mathbf{E}[T]$?



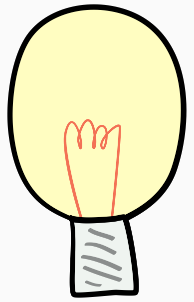
Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?



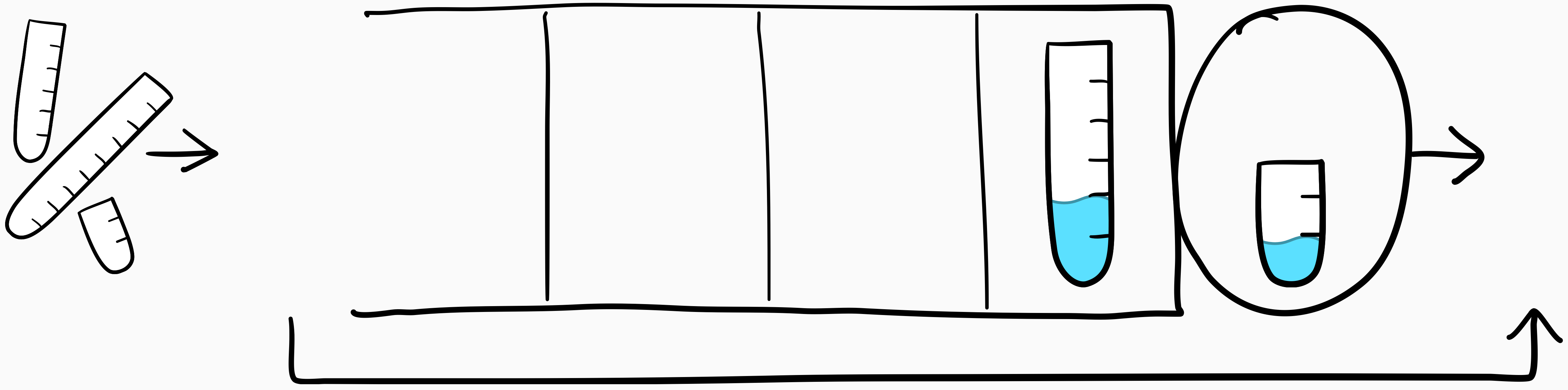
T = **response time**

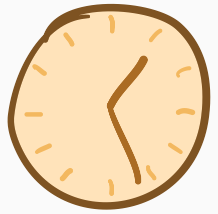
Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

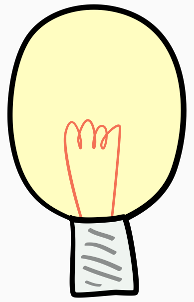
How should we schedule jobs to minimize delay?





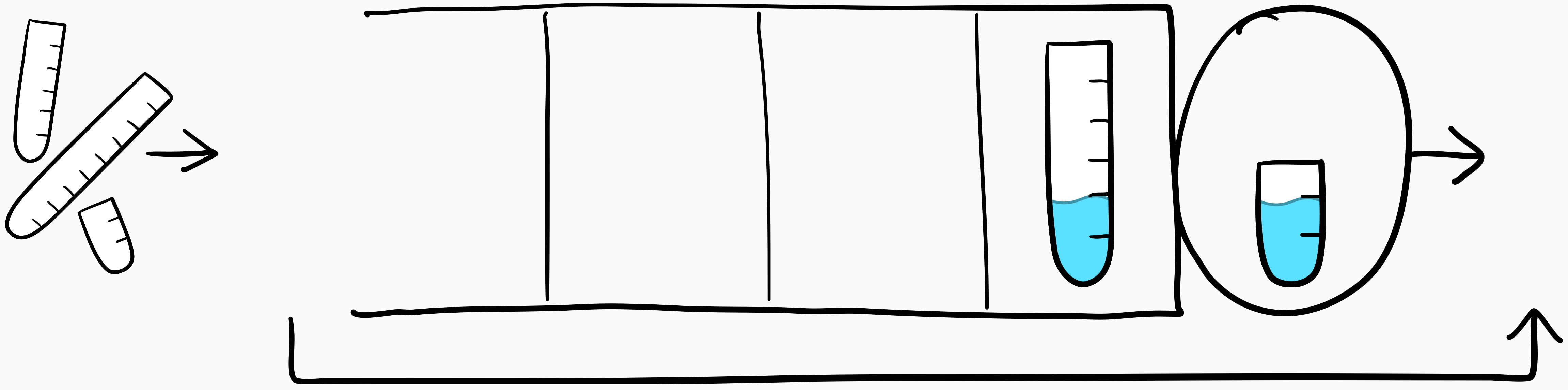
T = **response time**

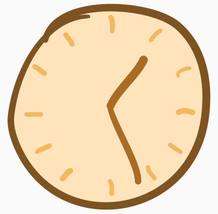
Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

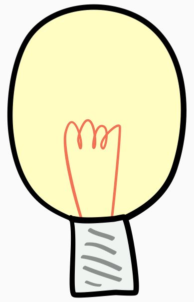
How should we schedule jobs to minimize delay?





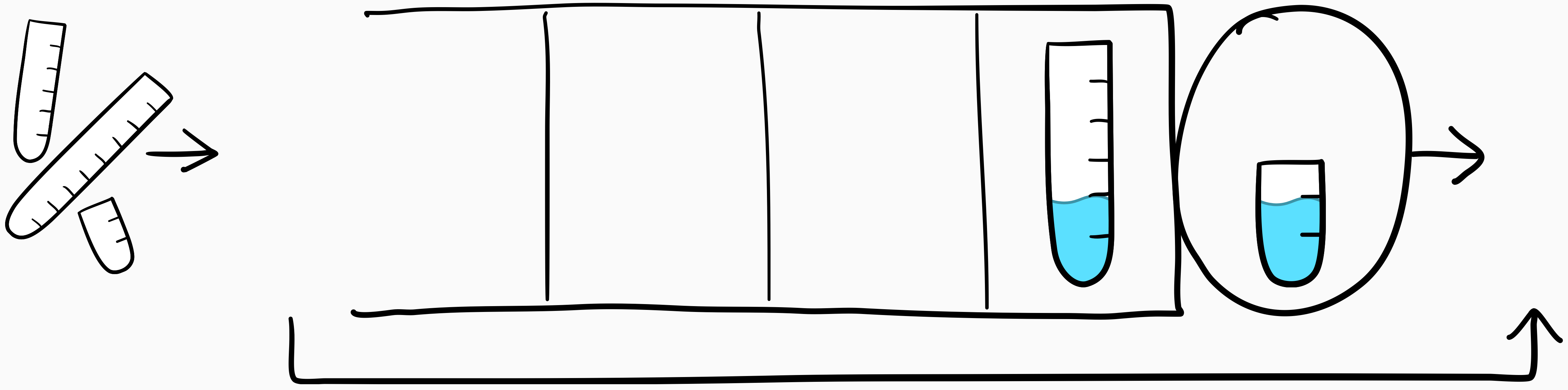
T = **response time**

Minimize $\mathbf{E}[T]$?



Serve short jobs
before long jobs

How should we schedule jobs to minimize delay?



T = **response time**

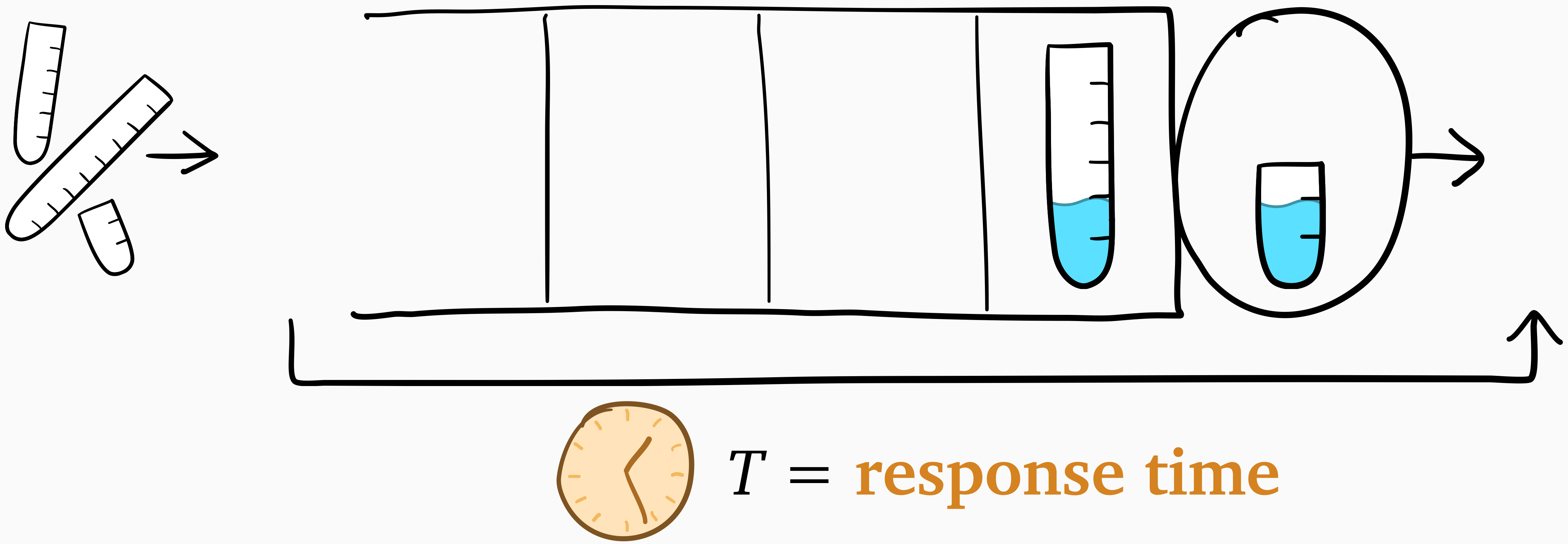
Minimize $\mathbf{E}[T]$?

Serve short jobs before long jobs

SRPT: minimizes $\mathbf{E}[T]$

shortest remaining processing time

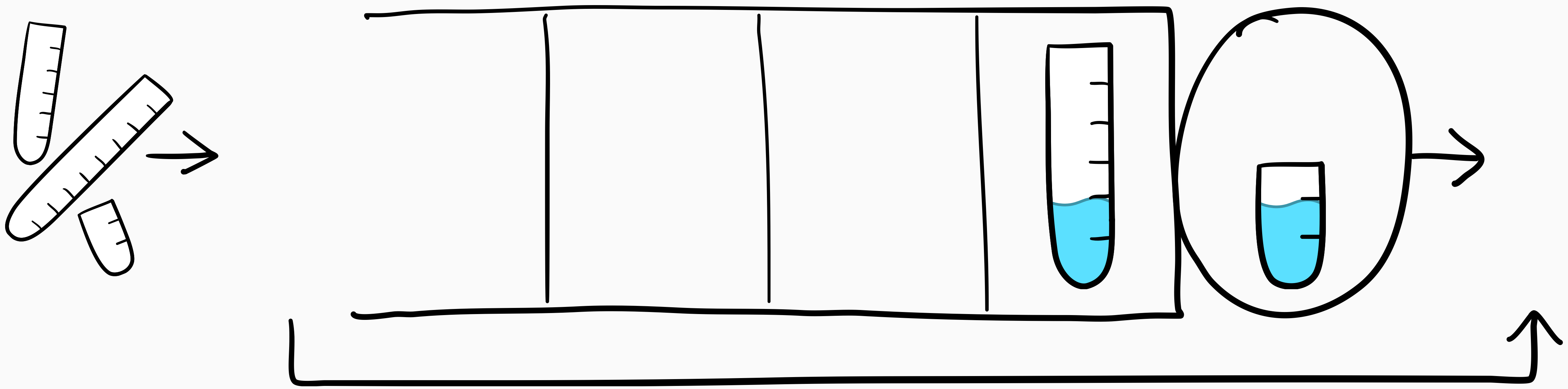
Beyond the mean: tail metrics

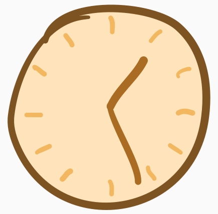


T = **response time**

Beyond the mean: tail metrics

Minimize
- $\mathbf{P}[T > t]$?
- $\mathbf{E}[(T - t)^+]$?
- quantiles of T?



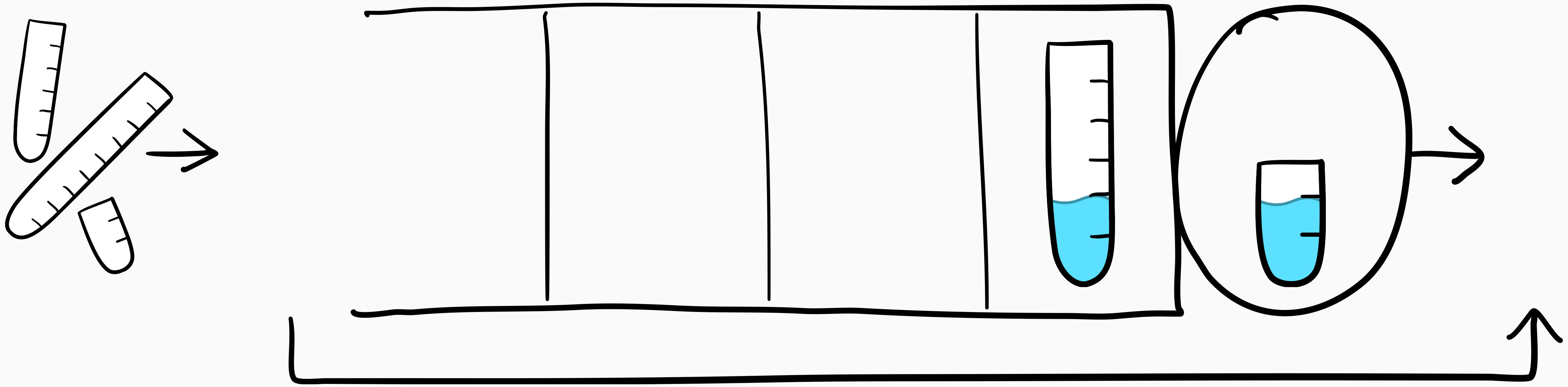


T = **response time**

Beyond the mean: tail metrics

Minimize
- $\mathbf{P}[T > t]$?
- $\mathbf{E}[(T - t)^+]$?
- quantiles of T?

Practice: important



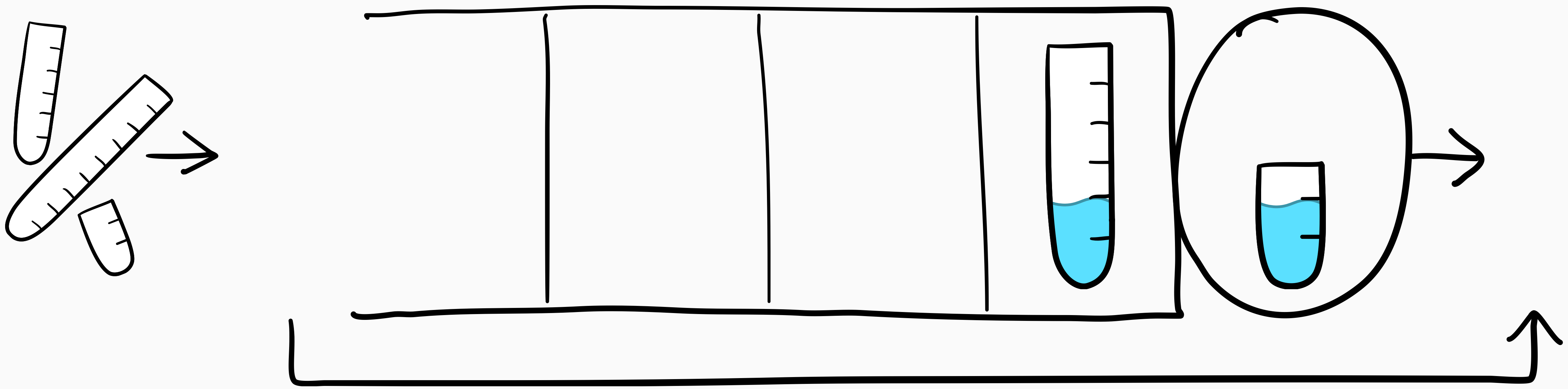
T = **response time**

Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

Practice: important

Theory: very hard



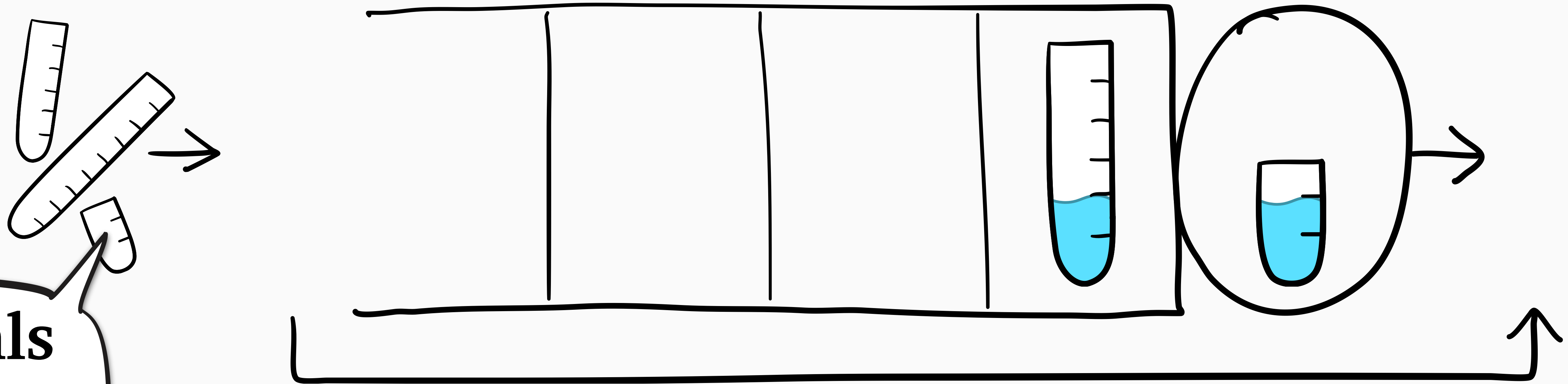
T = **response time**

Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^{+}]? \\ \text{quantiles of } T? \end{cases}$

Practice: important

Theory: very hard



M/G arrivals

- arrival rate λ
- job size dist. S

T = **response time**

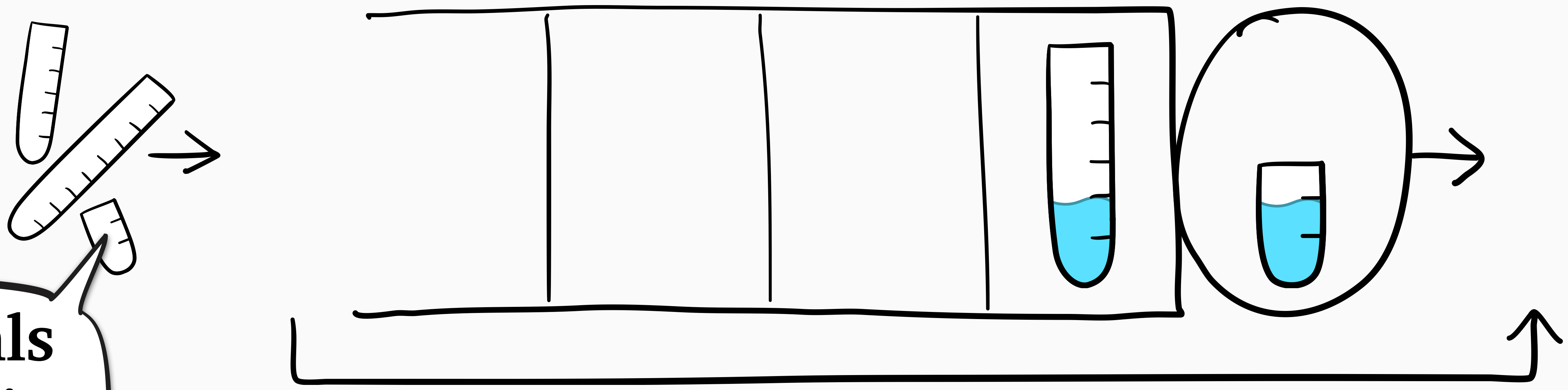
Beyond the mean: tail metrics

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

Practice: important

Theory: very hard

Tractable:
study $t \to \infty$
asymptotics



M/G arrivals
- arrival rate λ
- job size dist. S

T = **response time**

Beyond the mean: tail metrics

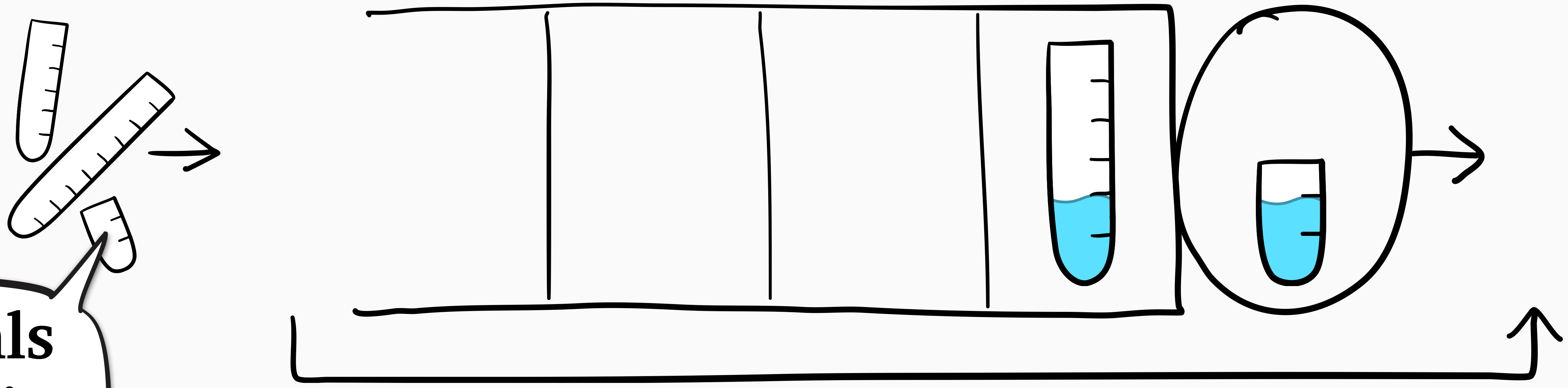
no single t value is most important

Minimize $\begin{cases} \mathbf{P}[T > t]? \\ \mathbf{E}[(T - t)^+]? \\ \text{quantiles of } T? \end{cases}$

Practice: important

Theory: very hard

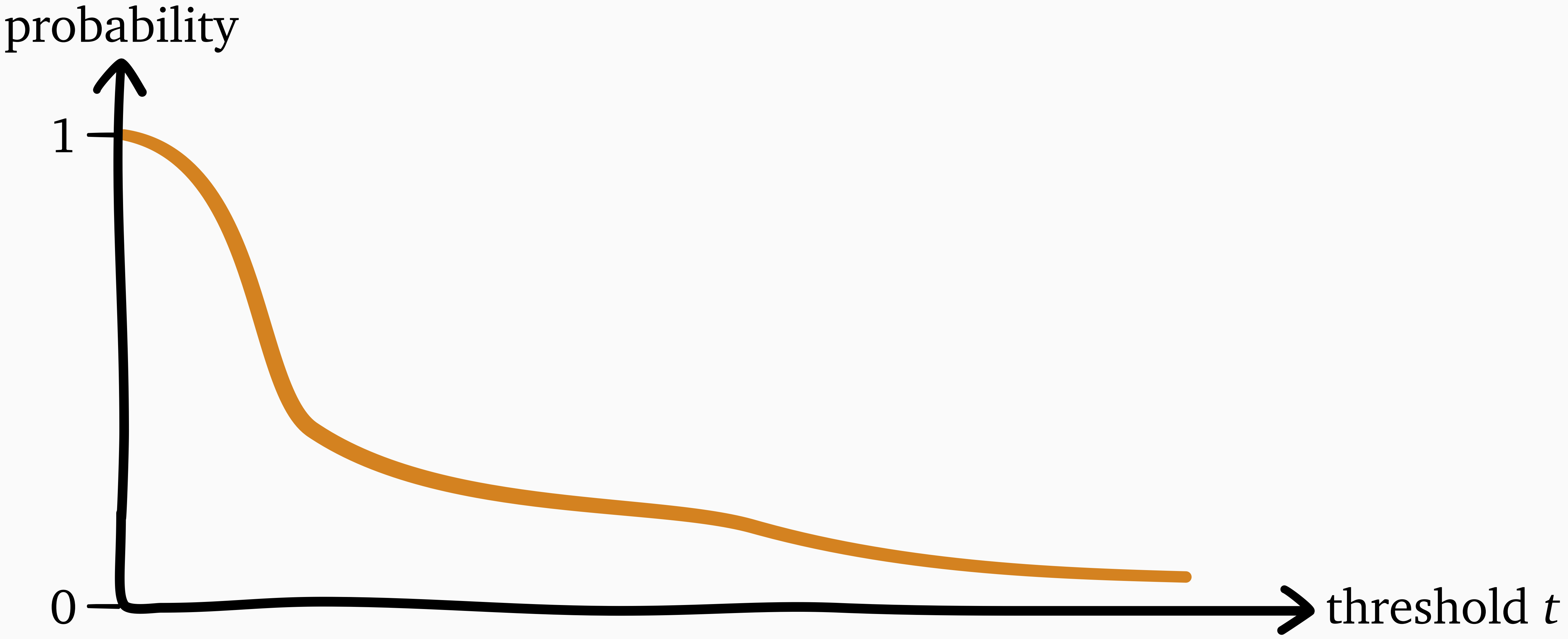
Tractable:
study $t \to \infty$
asymptotics



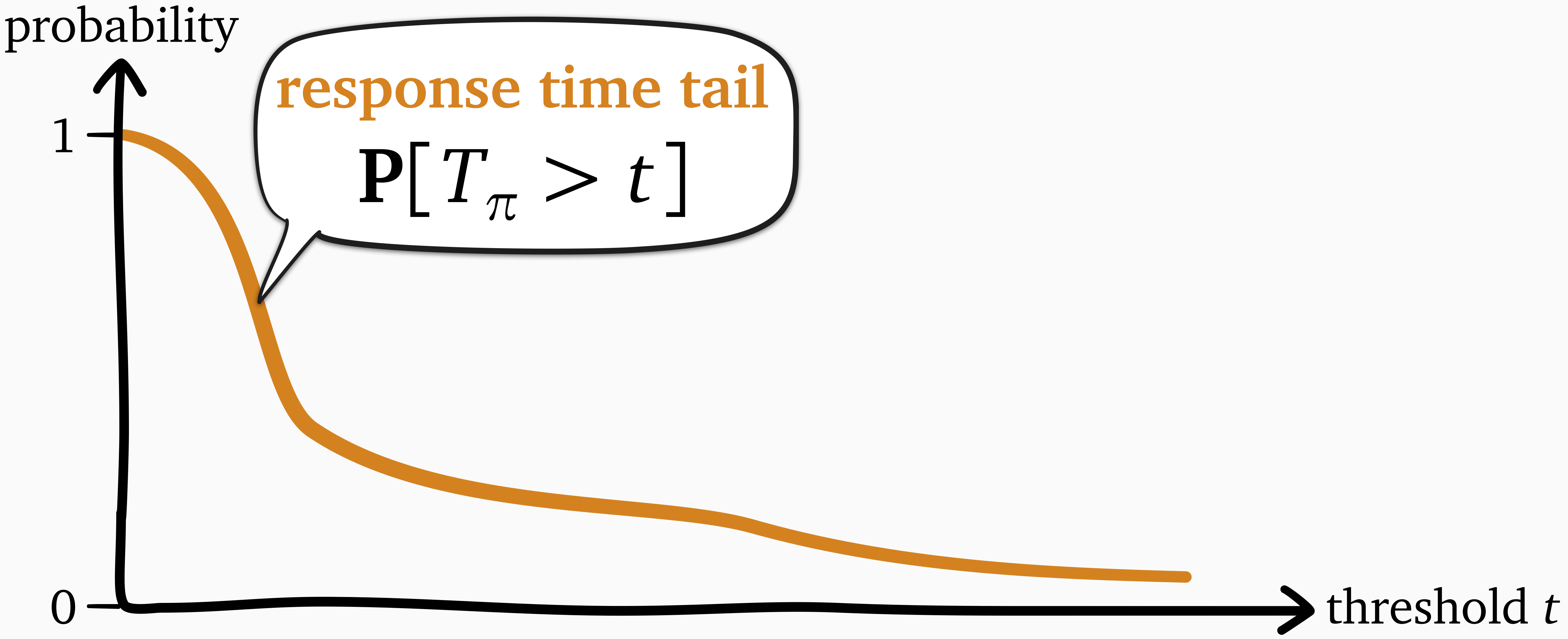
M/G arrivals
- arrival rate λ
- job size dist. S

T = **response time**

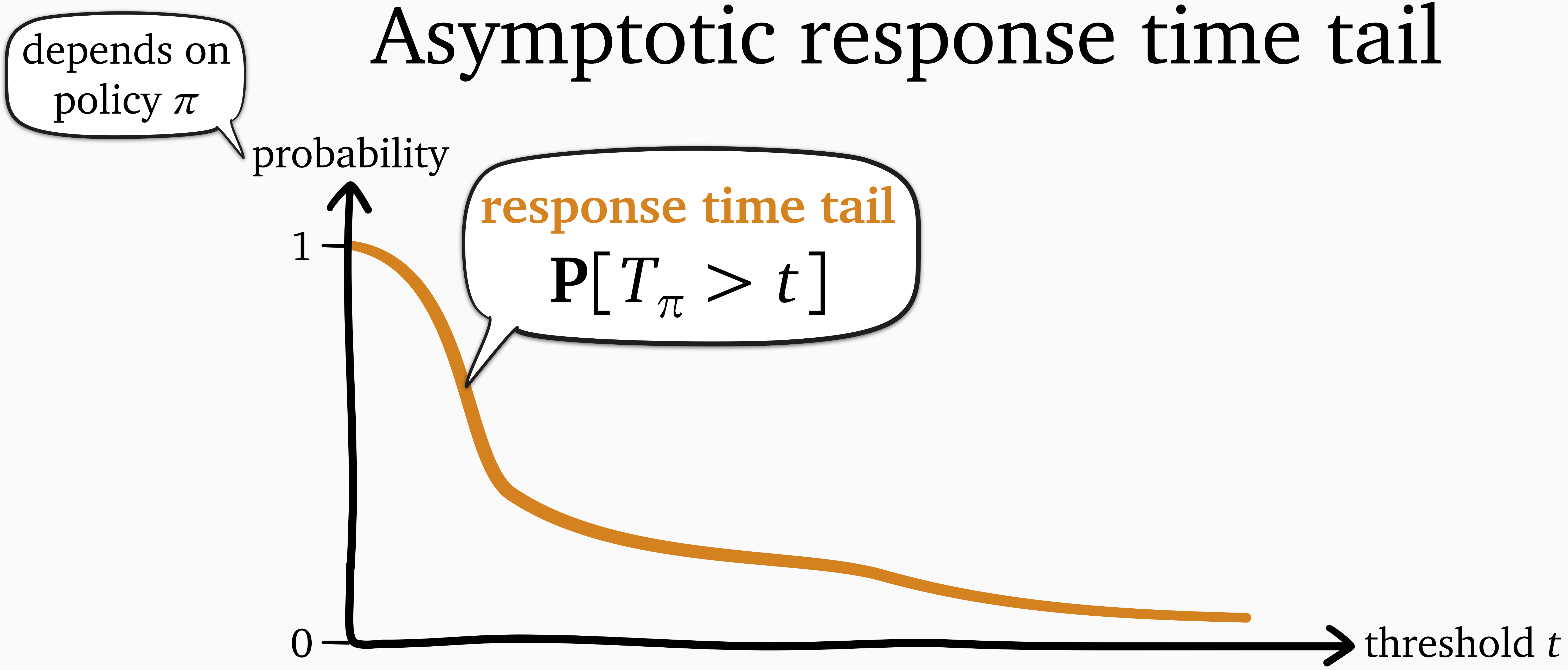
Asymptotic response time tail



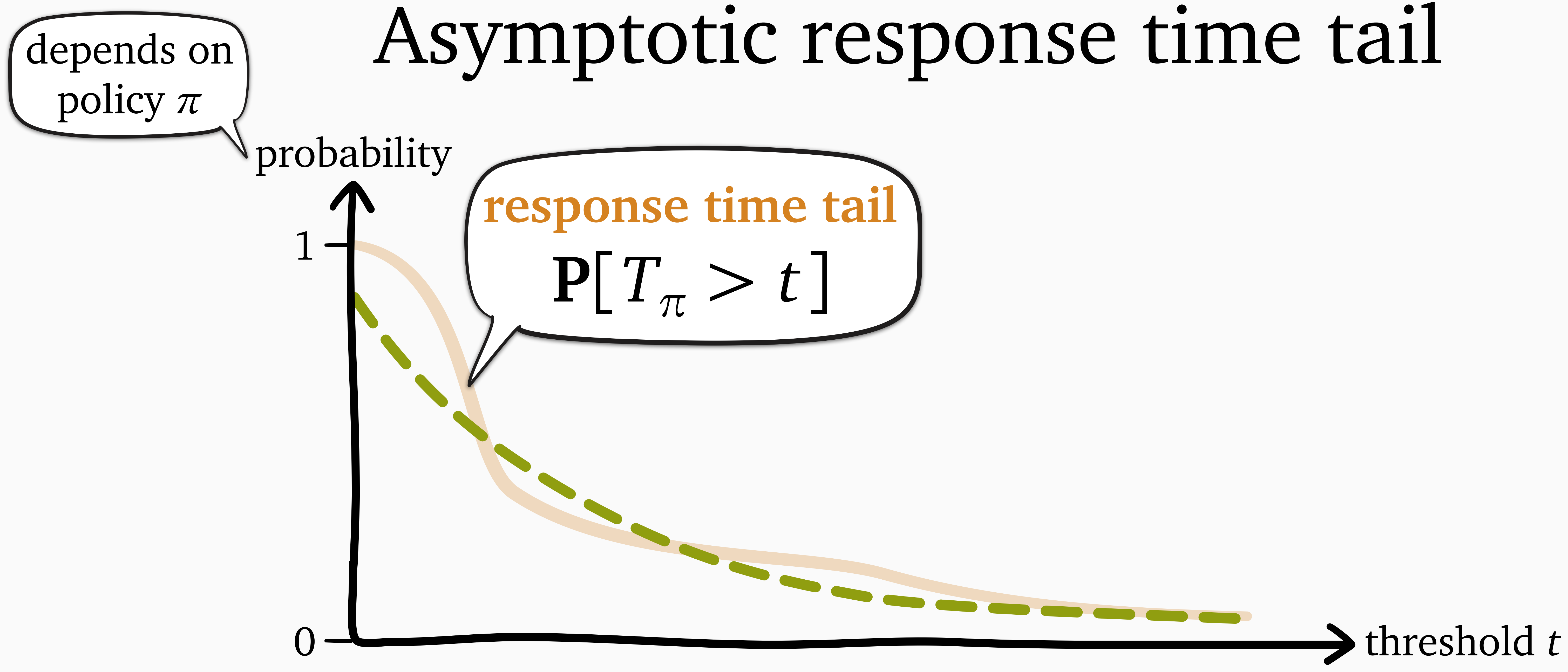
Asymptotic response time tail



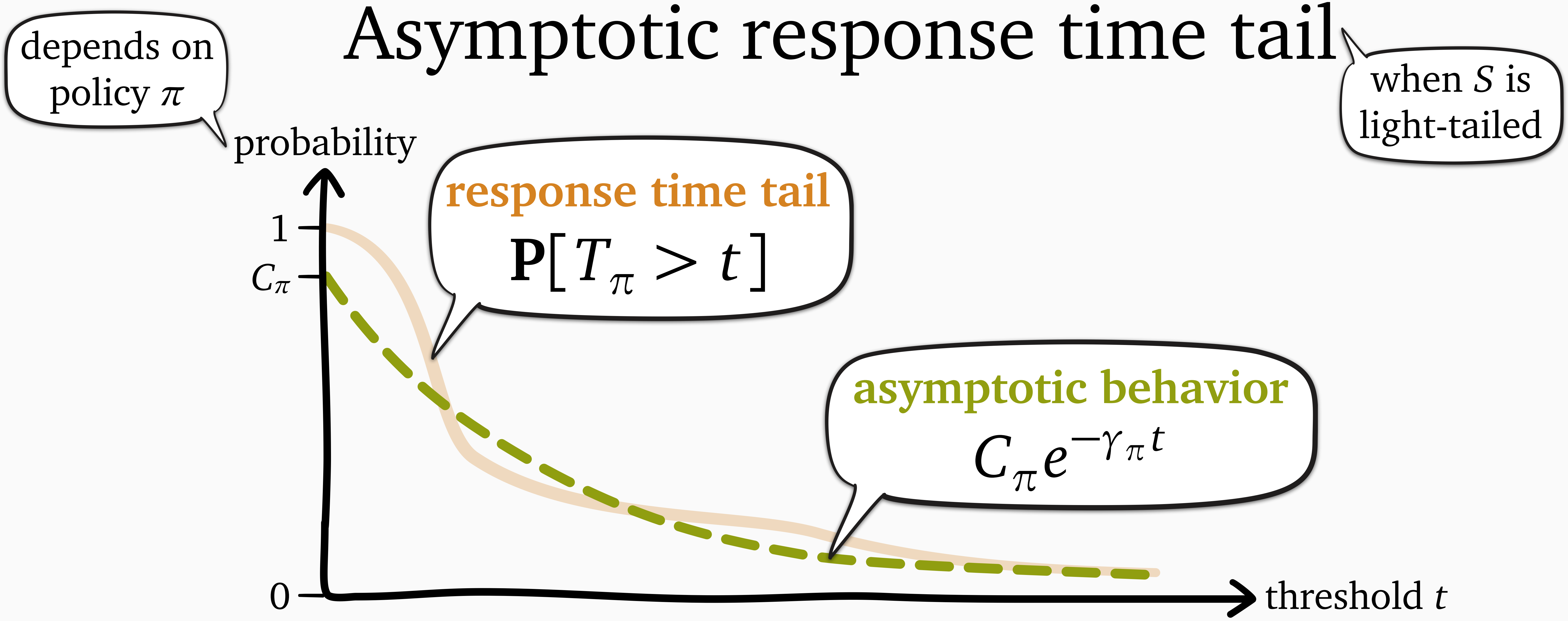
Asymptotic response time tail



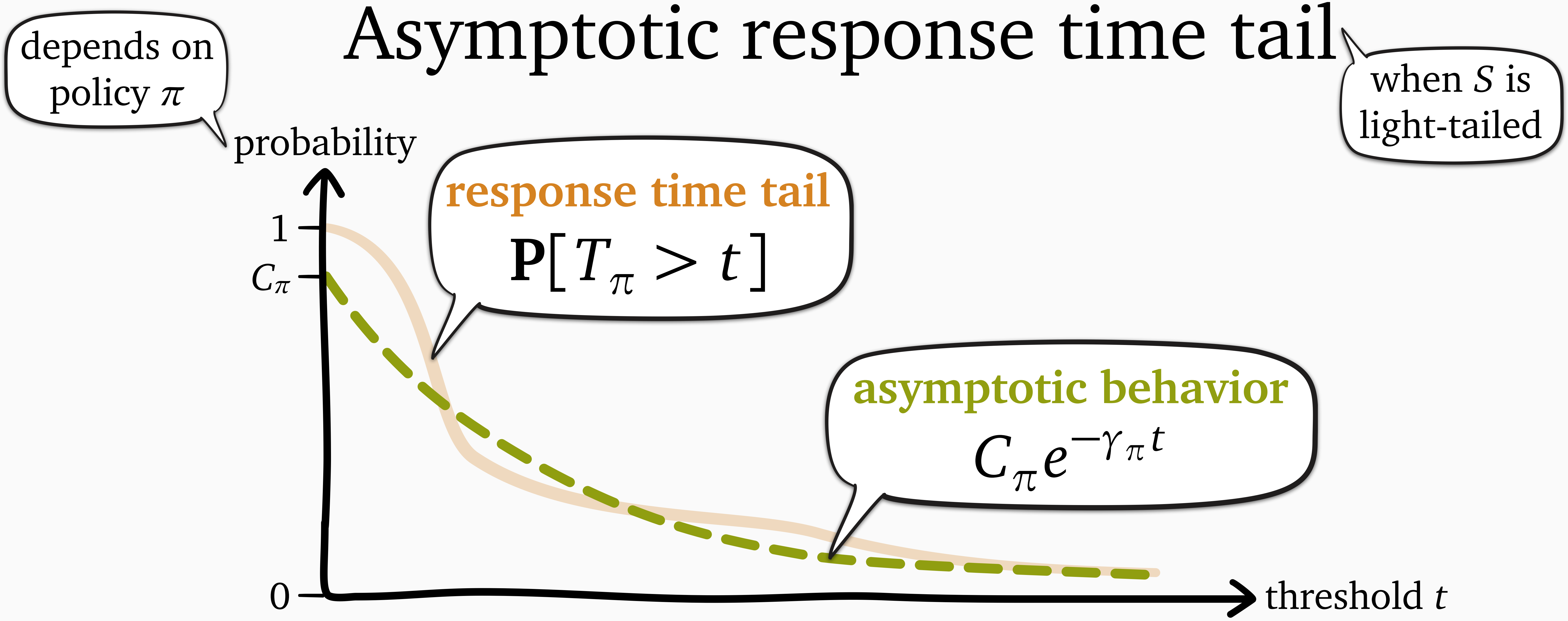
Asymptotic response time tail



Asymptotic response time tail



Asymptotic response time tail



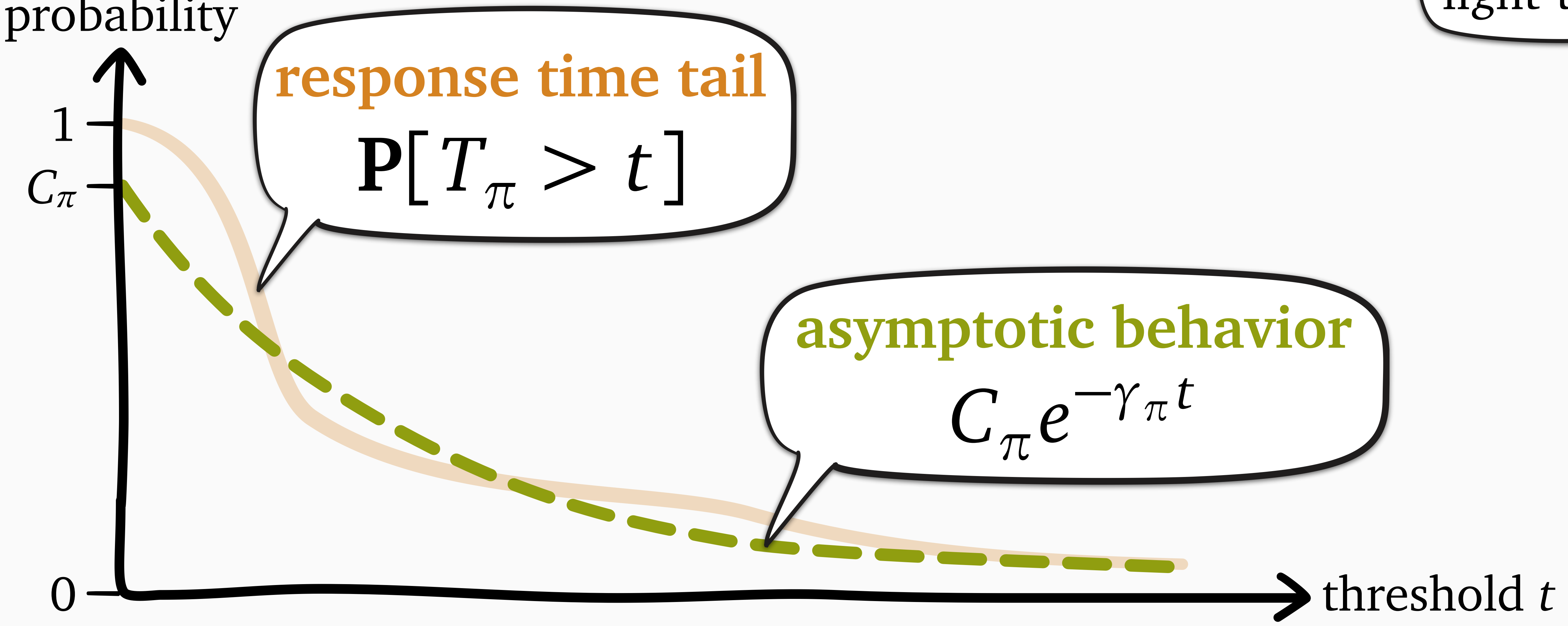
γ_π = *decay rate* of π

C_π = *tail constant* of π

Asymptotic response time tail

depends on policy π

when S is light-tailed

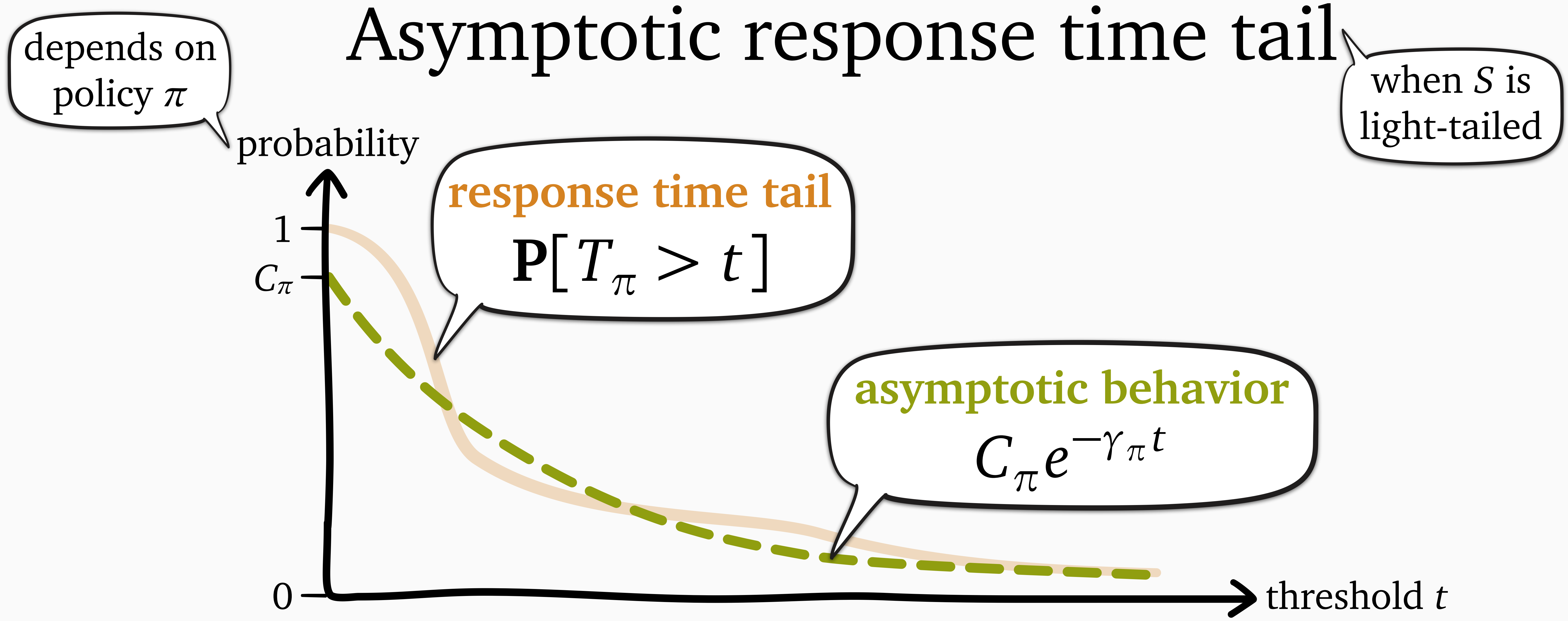


Weak optimality:
optimal γ_π

$\gamma_\pi =$ *decay rate* of π

$C_\pi =$ *tail constant* of π

Asymptotic response time tail



γ_π = *decay rate* of π

C_π = *tail constant* of π

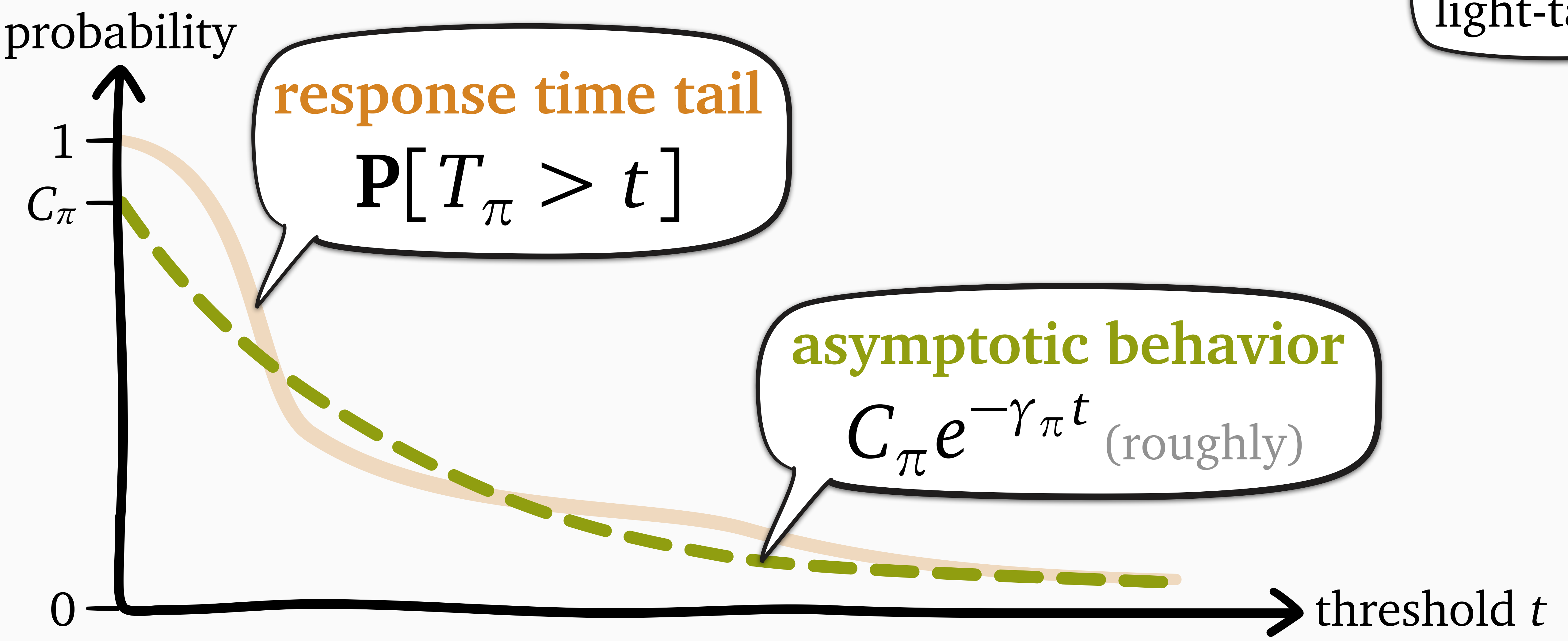
Weak optimality: optimal γ_π

Strong optimality: optimal γ_π and C_π

Asymptotic response time tail

depends on policy π

when S is light-tailed



γ_π = *decay rate* of π

C_π = *tail constant* of π

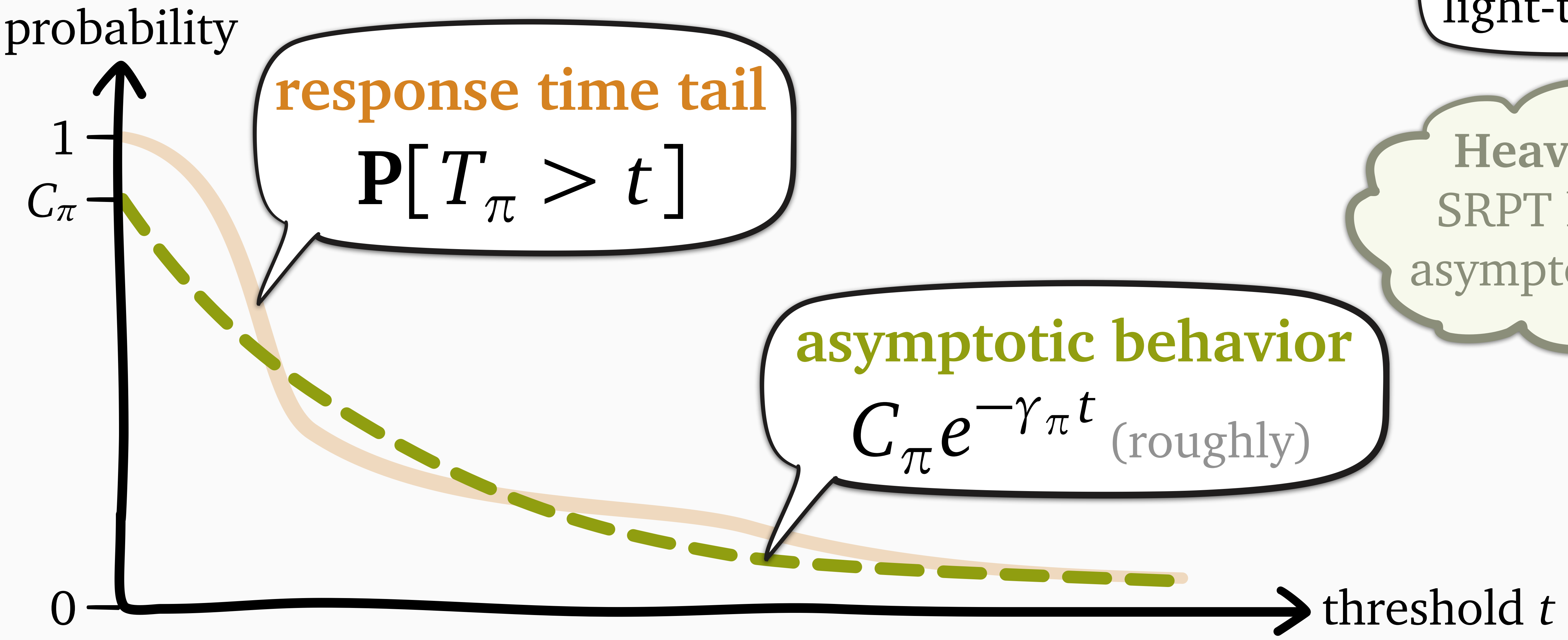
Weak optimality: optimal γ_π

Strong optimality: optimal γ_π and C_π

Asymptotic response time tail

depends on policy π

when S is light-tailed



Heavy-tailed S: SRPT has optimal asymptotic behavior

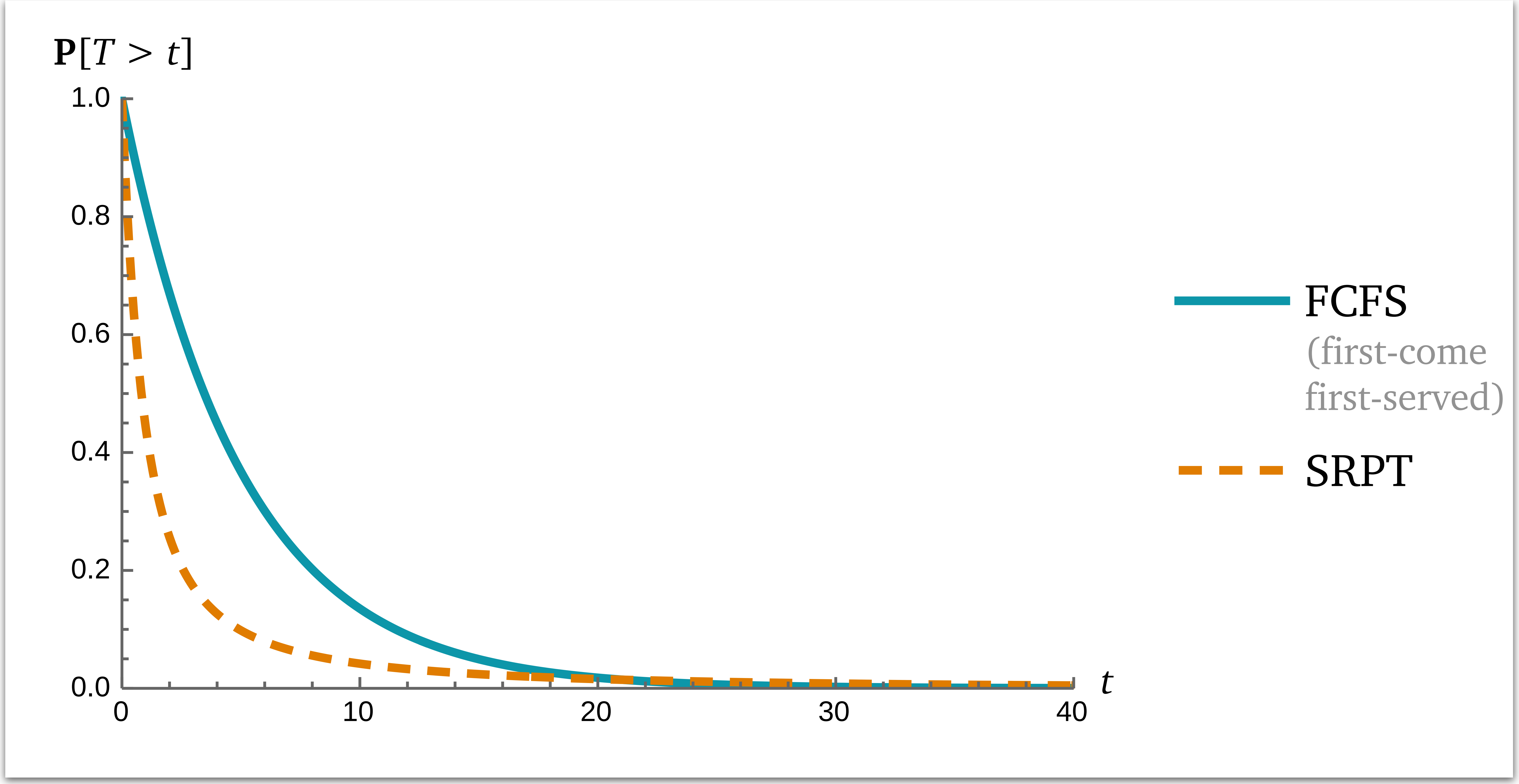
γ_π = *decay rate* of π

C_π = *tail constant* of π

Weak optimality: optimal γ_π

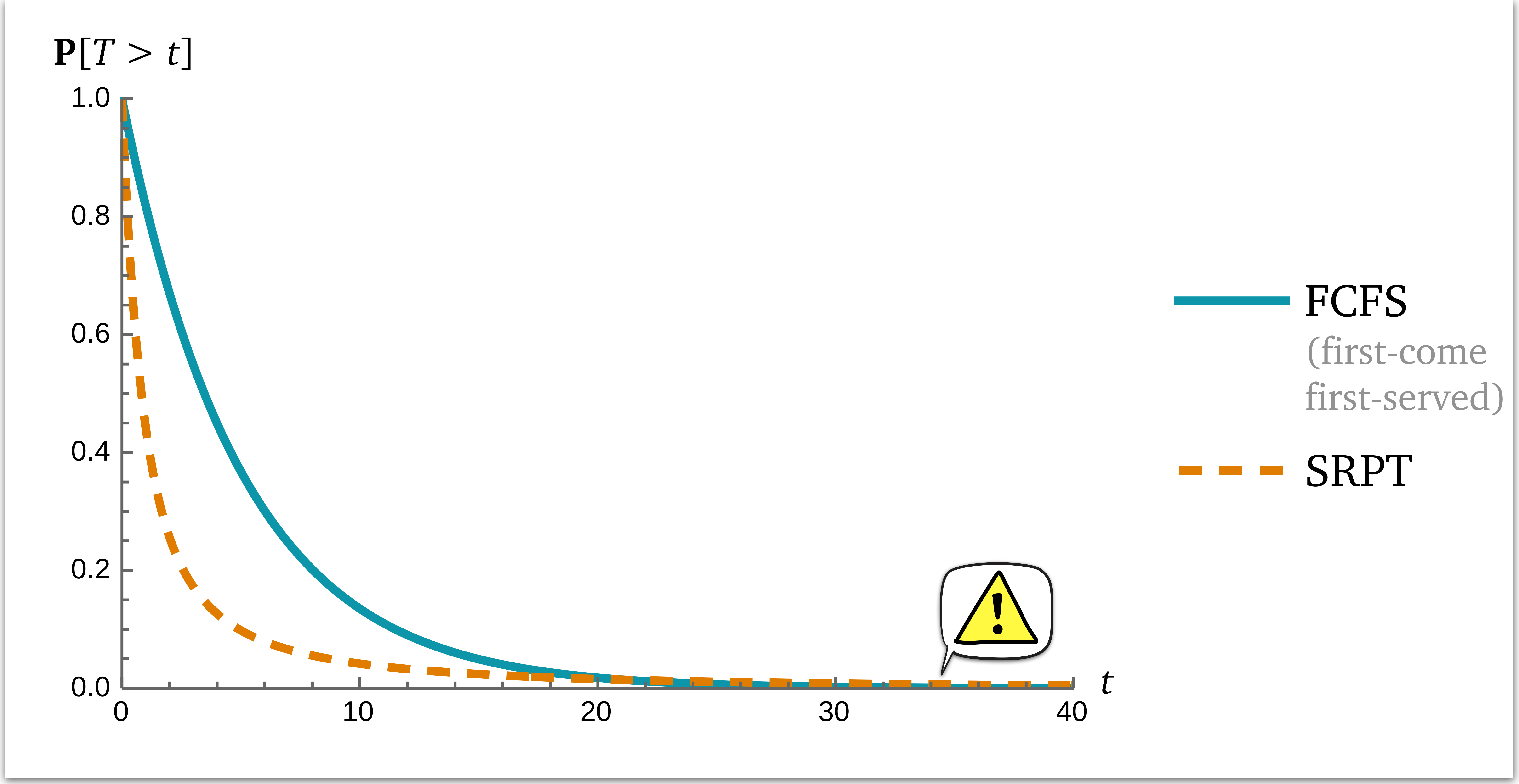
Strong optimality: optimal γ_π and C_π

FCFS vs. SRPT

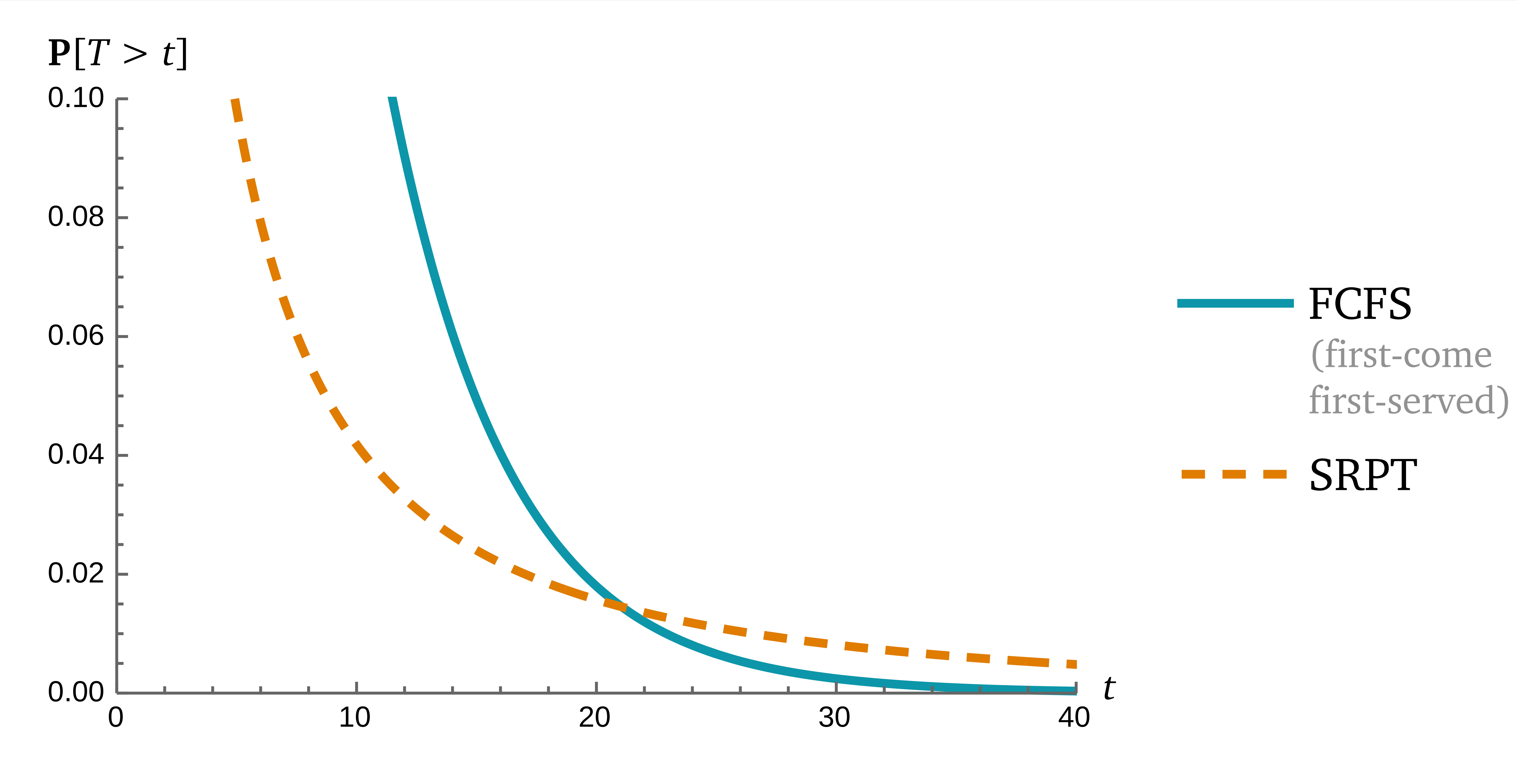


P[T > t]
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
t
FCFS
(first-come
first-served)
SRPT

FCFS vs. SRPT

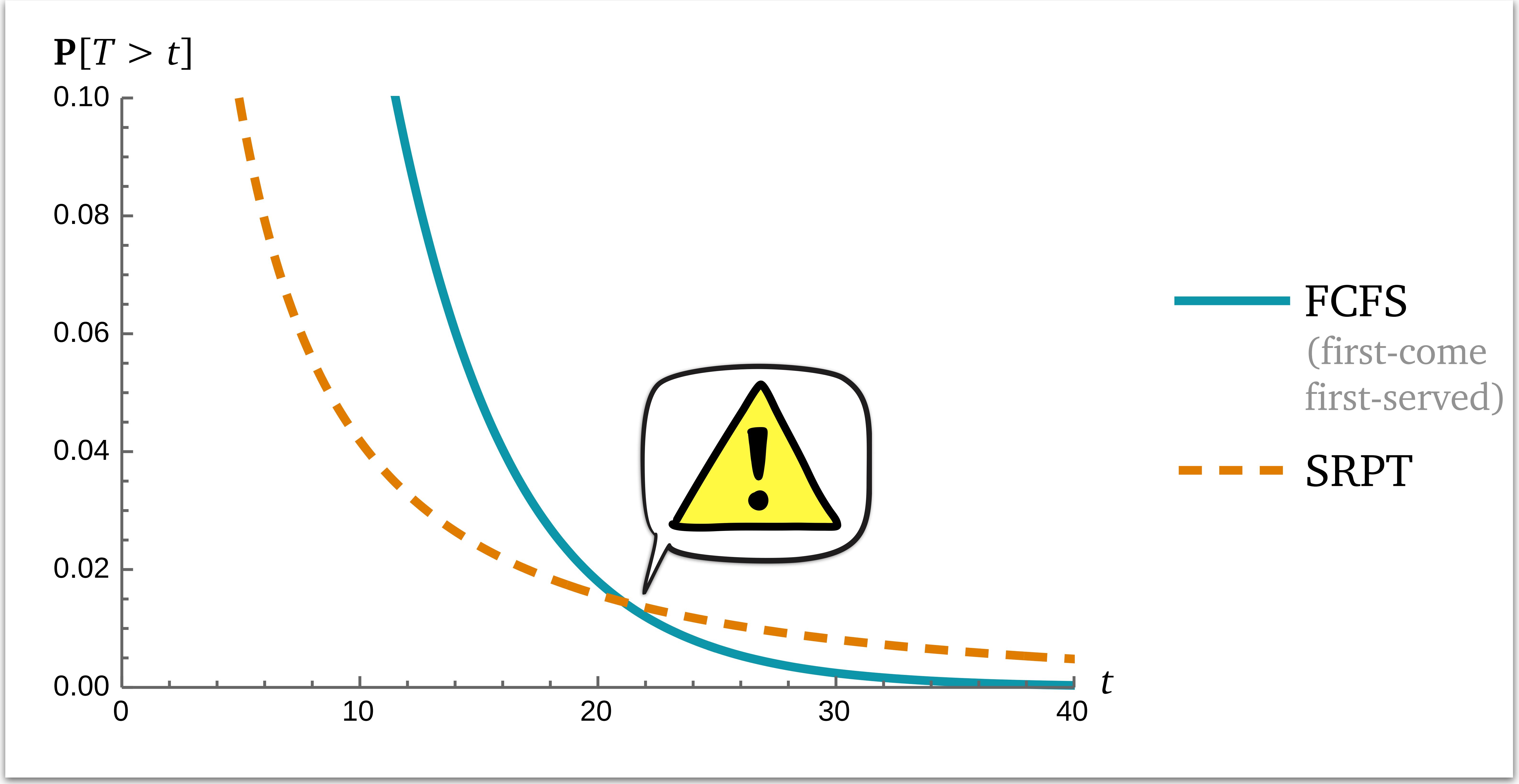

P[T > t]
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
t
FCFS
(first-come first-served)
SRPT

FCFS vs. SRPT



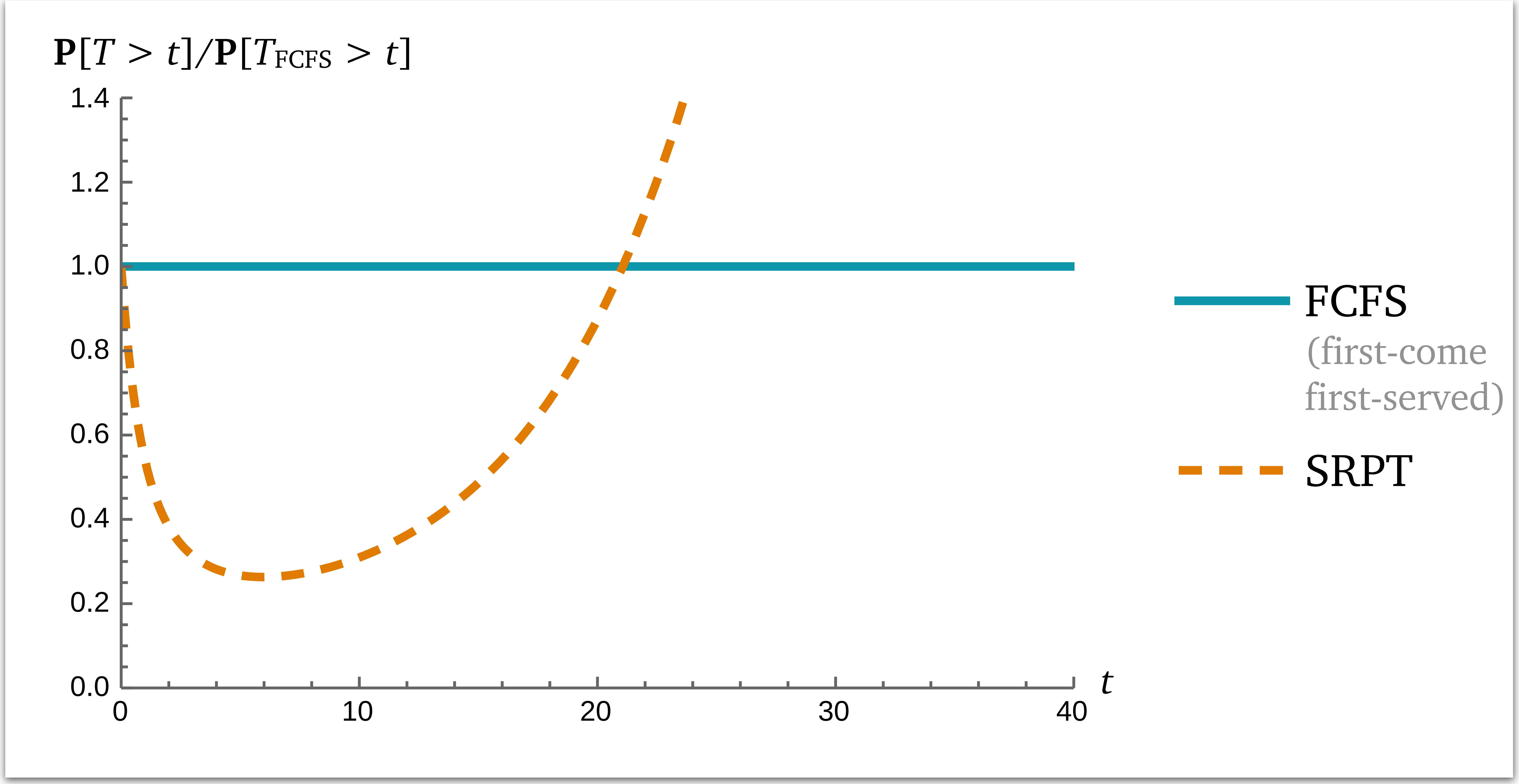
$\mathbf{P}[T > t]$
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
t
FCFS
(first-come first-served)
SRPT

FCFS vs. SRPT

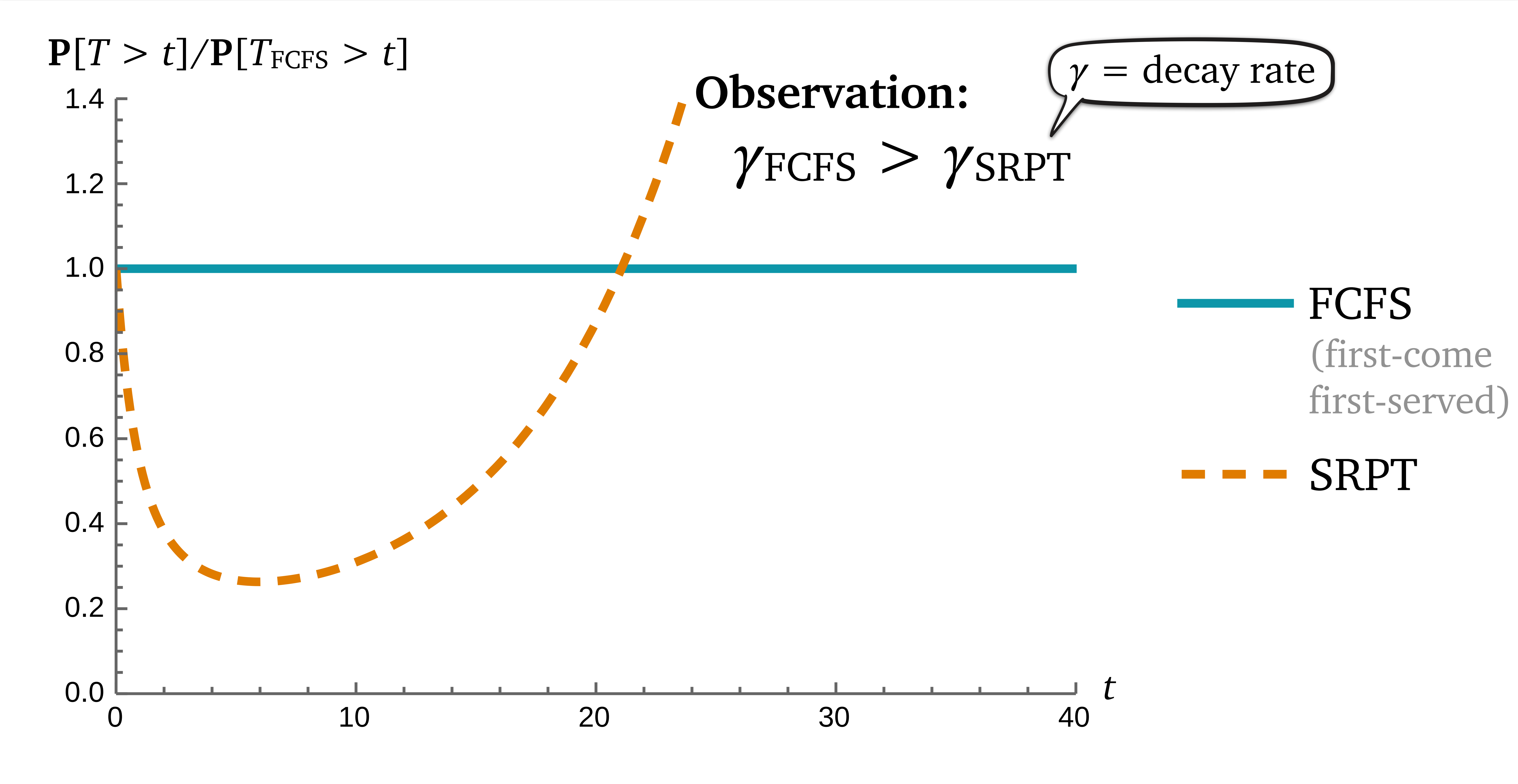


$\mathbf{P}[T > t]$
0.10
0.08
0.06
0.04
0.02
0.00
0
10
20
30
40
t
FCFS
(first-come
first-served)
SRPT

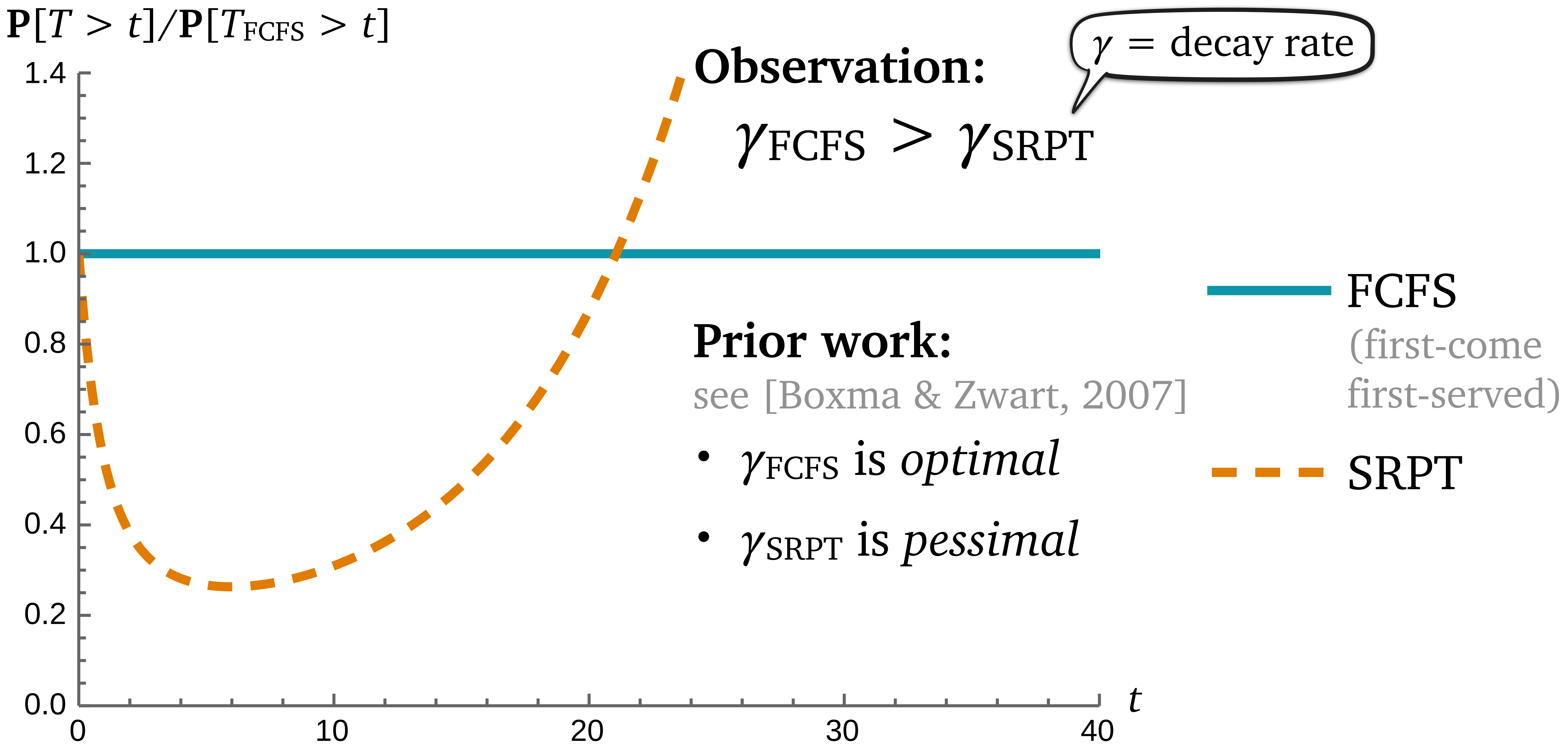
FCFS vs. SRPT


$\mathbf{P}[T > t]/\mathbf{P}[T_{\text{FCFS}} > t]$
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
t
FCFS
(first-come first-served)
SRPT

FCFS vs. SRPT

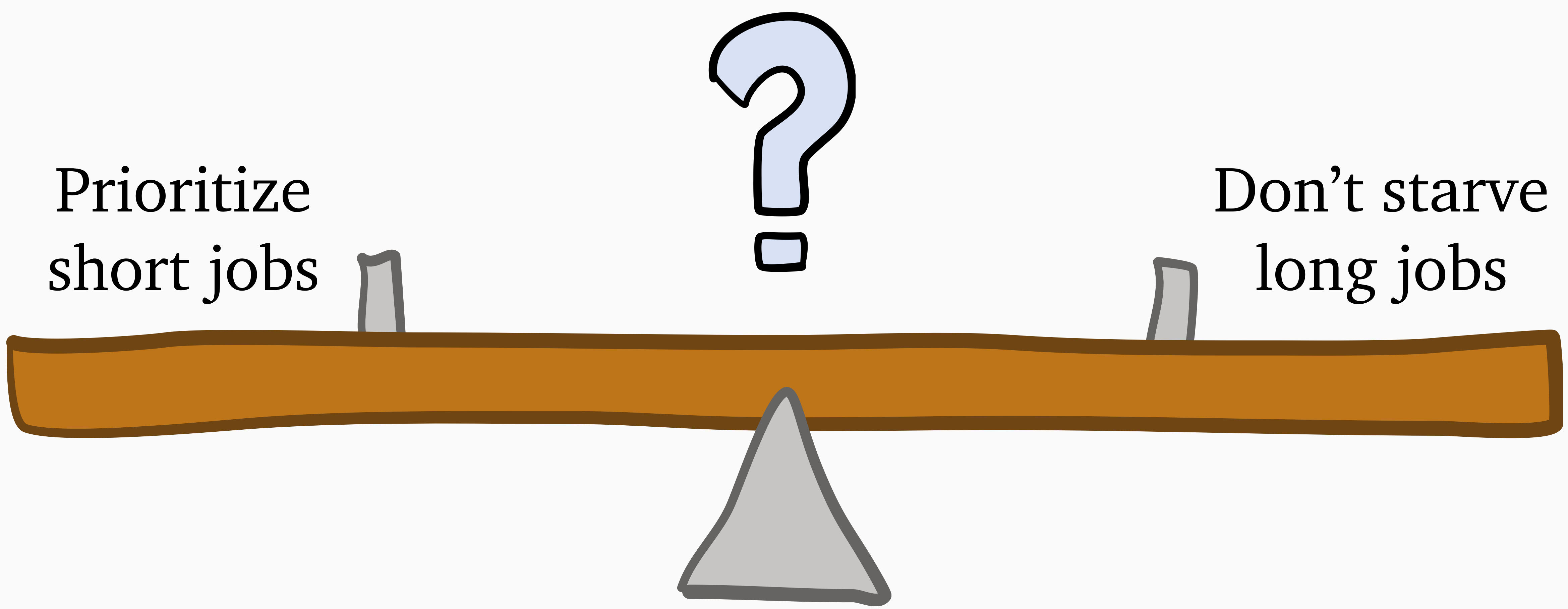

$\mathbf{P}[T > t]/\mathbf{P}[T_{\text{FCFS}} > t]$
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
t
Observation:
$\gamma_{\text{FCFS}} > \gamma_{\text{SRPT}}$
γ = decay rate
FCFS
(first-come first-served)
SRPT

FCFS vs. SRPT

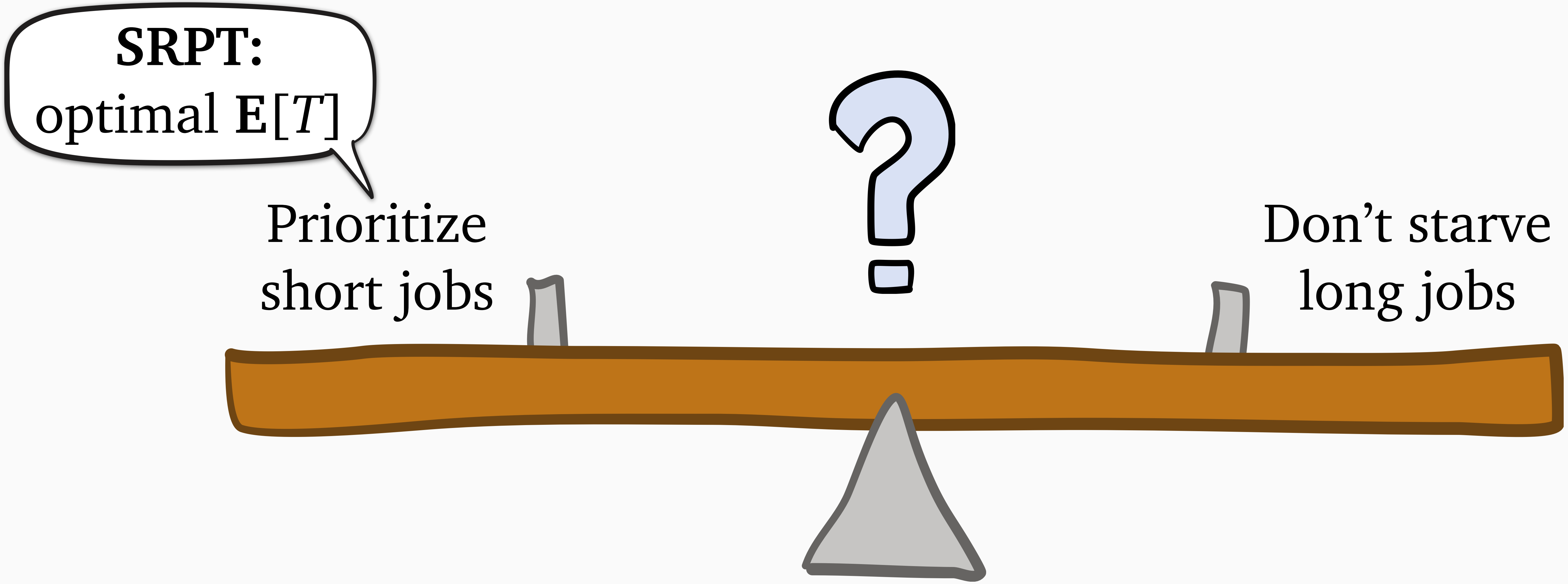


P[T > t]/P[T_FCFS > t]
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
t
Observation:
γ_FCFS > γ_SRPT
γ = decay rate
Prior work:
see [Boxma & Zwart, 2007]
• γ_FCFS is optimal
• γ_SRPT is pessimal
FCFS
(first-come first-served)
SRPT

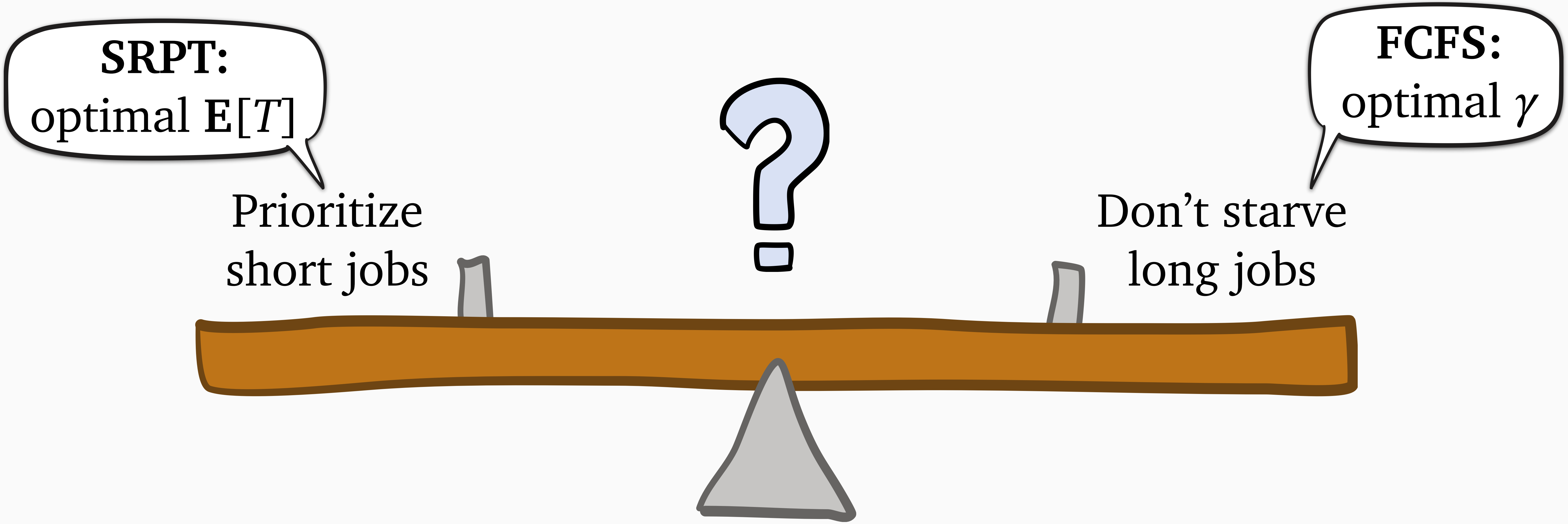
Tradeoff: priority vs. starvation



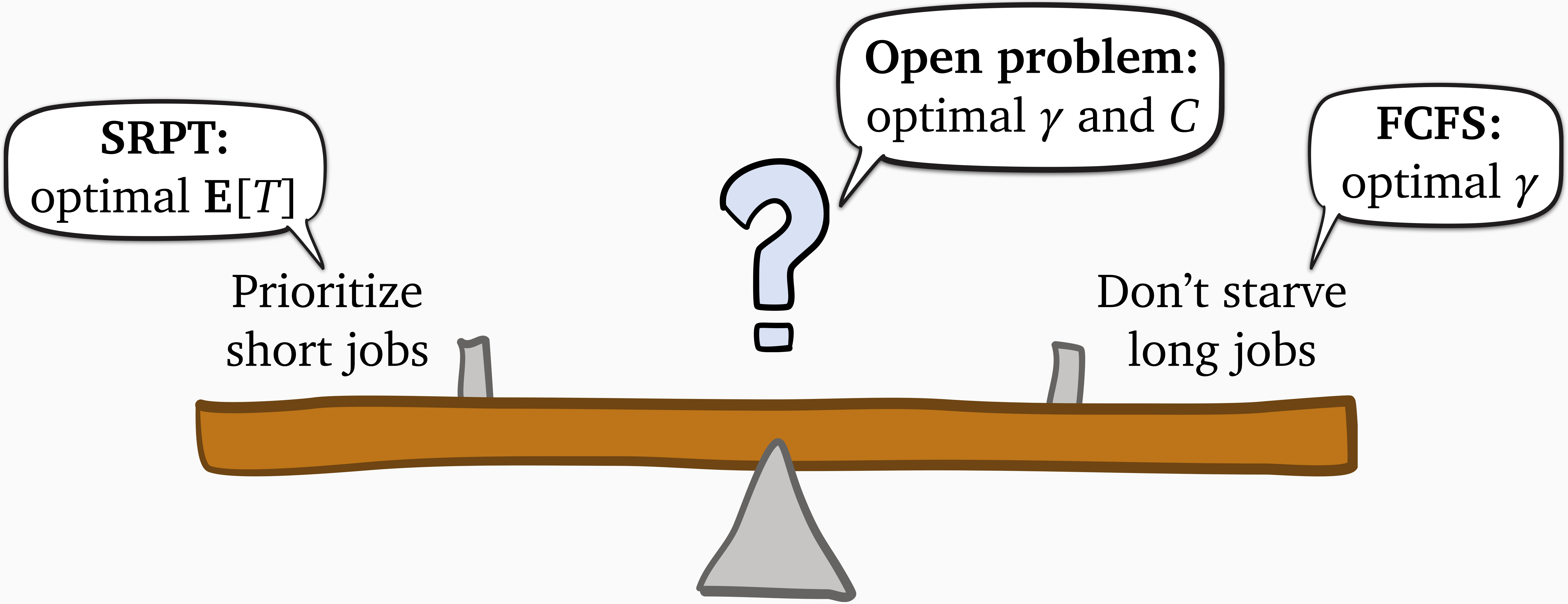
Tradeoff: priority vs. starvation



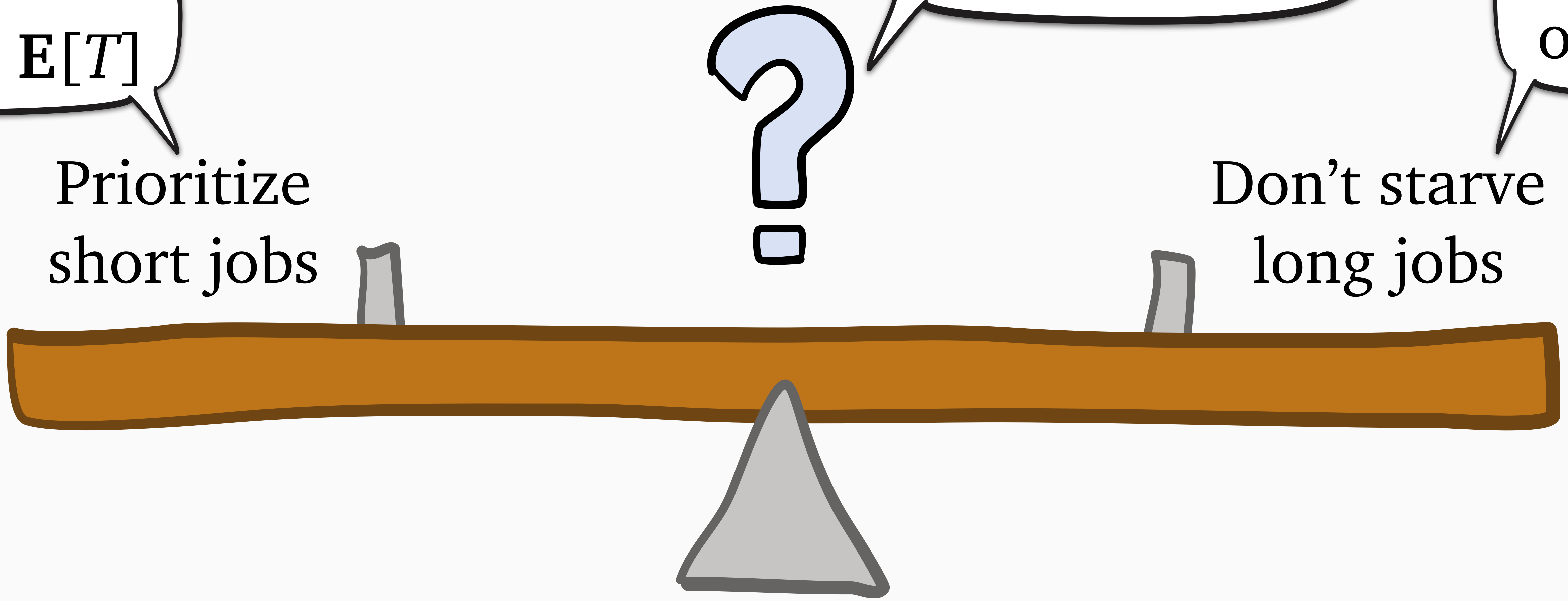
Tradeoff: priority vs. starvation



Tradeoff: priority vs. starvation



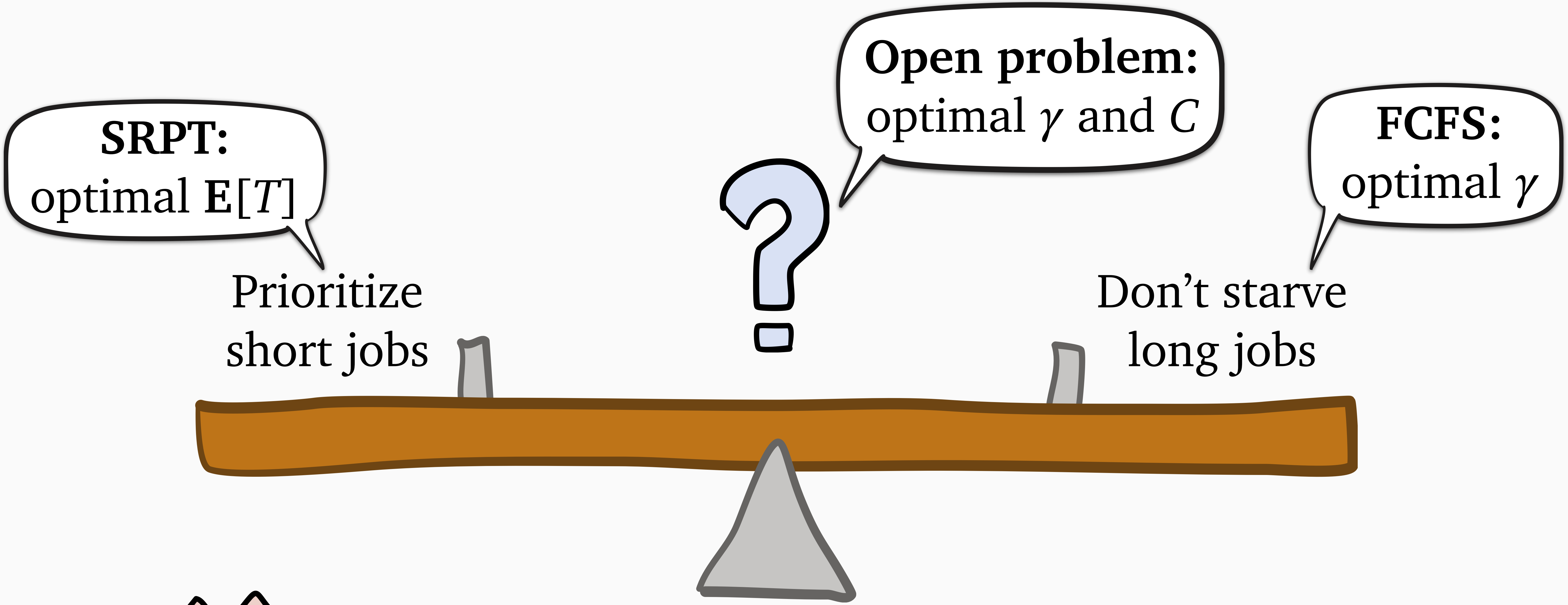
Tradeoff: priority vs. starvation



Conjecture: FCFS optimizes C, too

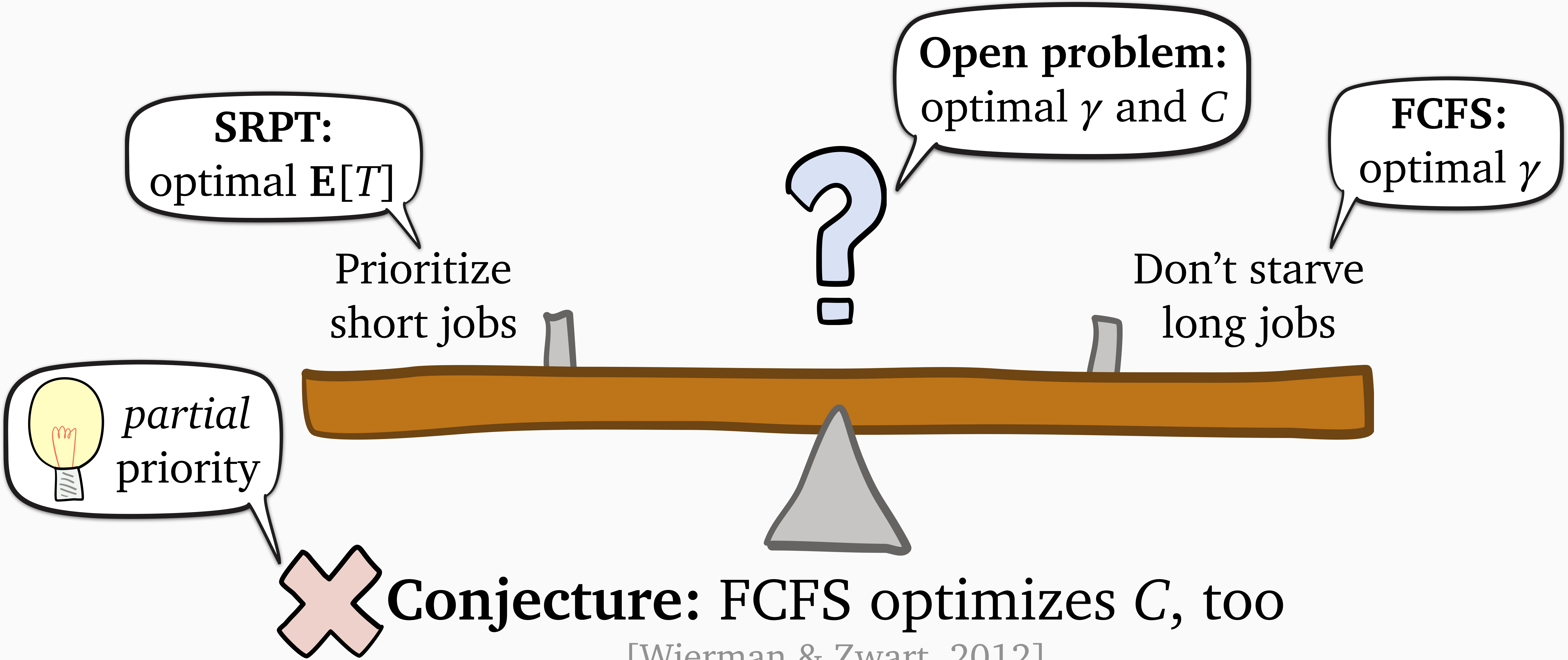
[Wierman & Zwart, 2012]

Tradeoff: priority vs. starvation



Conjecture: FCFS optimizes C, too
[Wierman & Zwart, 2012]

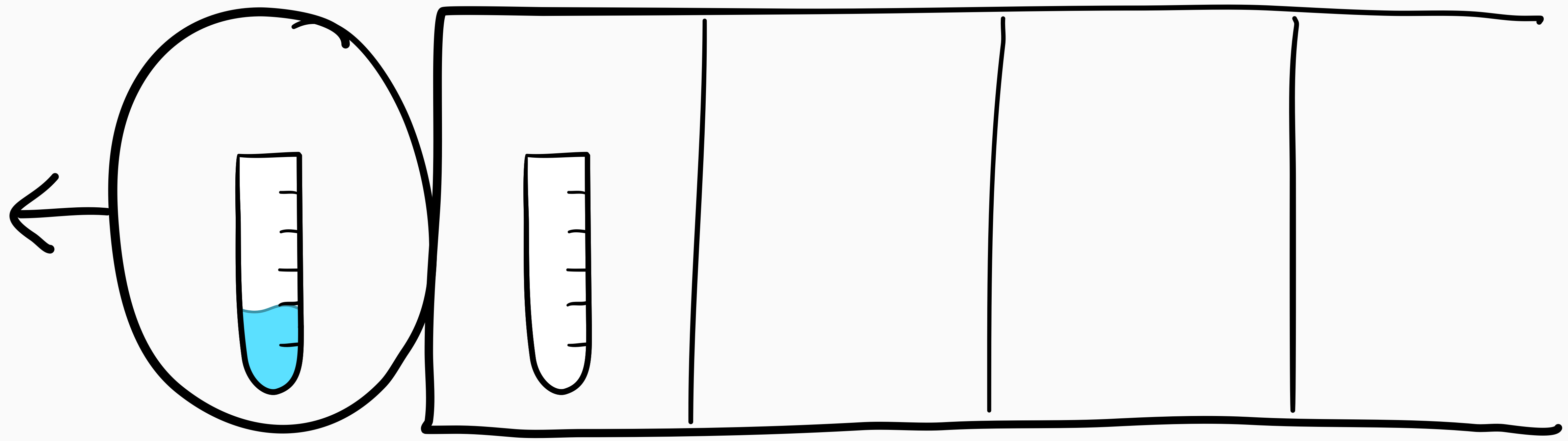
Tradeoff: priority vs. starvation



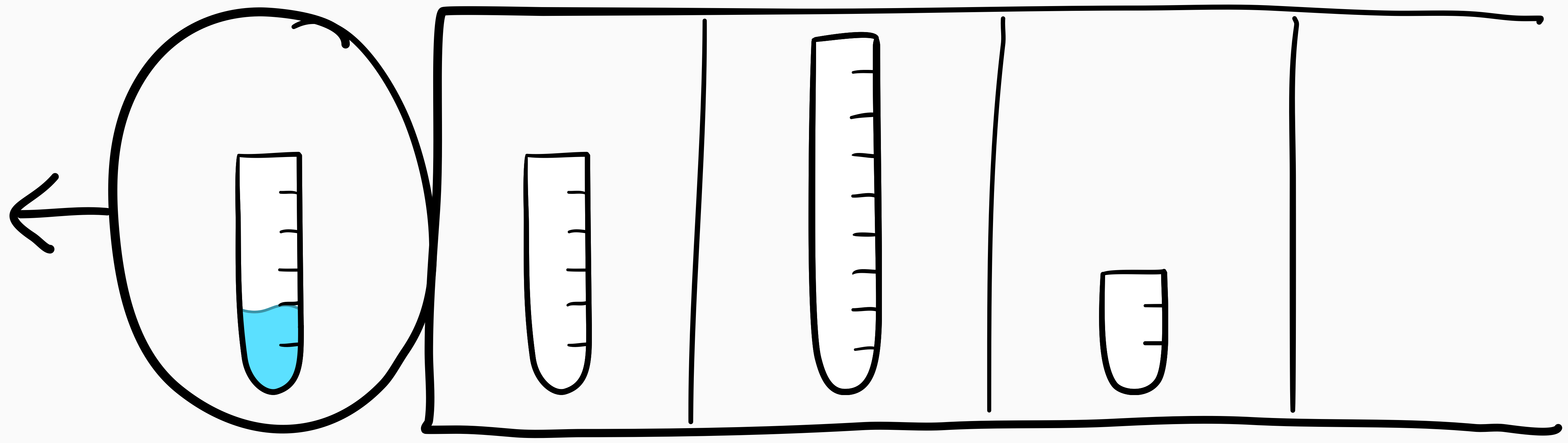
Conjecture: FCFS optimizes C, too

[Wierman & Zwart, 2012]

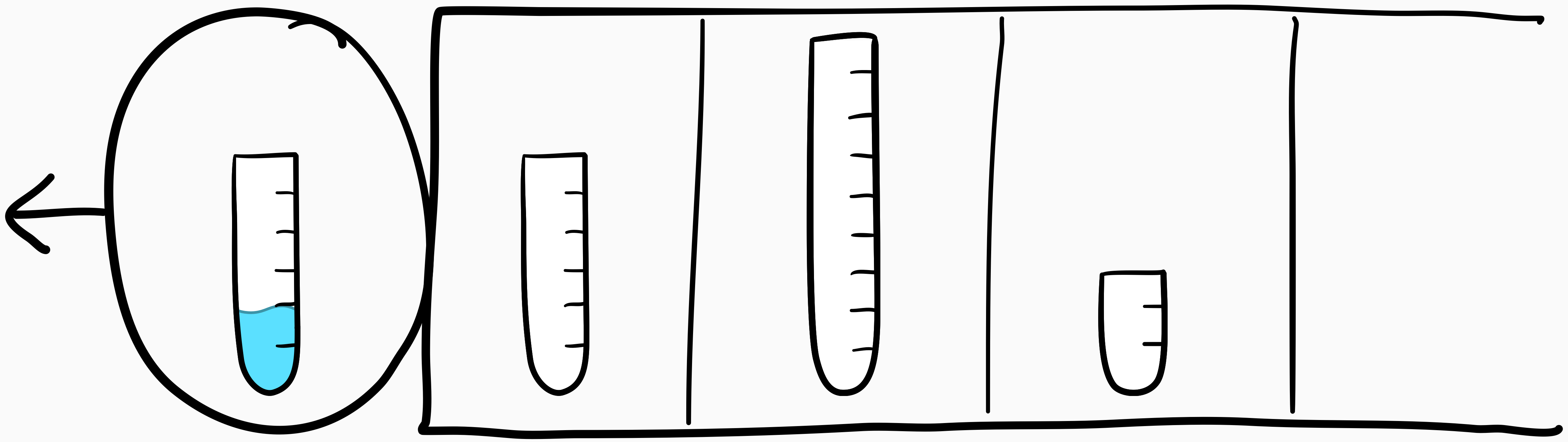
Can we beat FCFS?

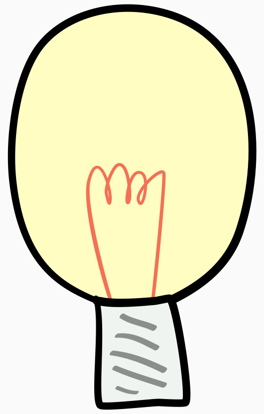


Can we beat FCFS?



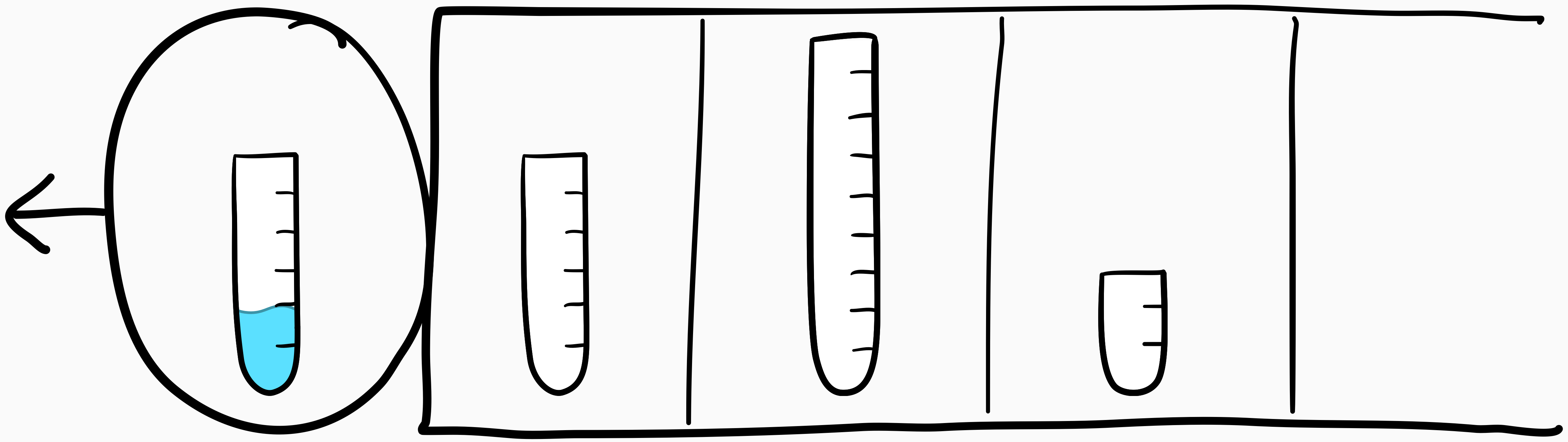
Can we beat FCFS?

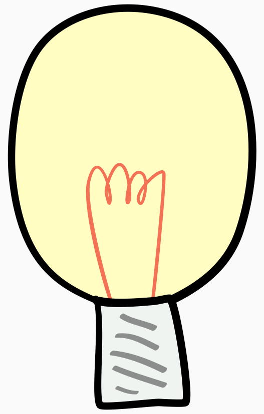




Nudge [Grosof et al., 2021]

Can we beat FCFS?

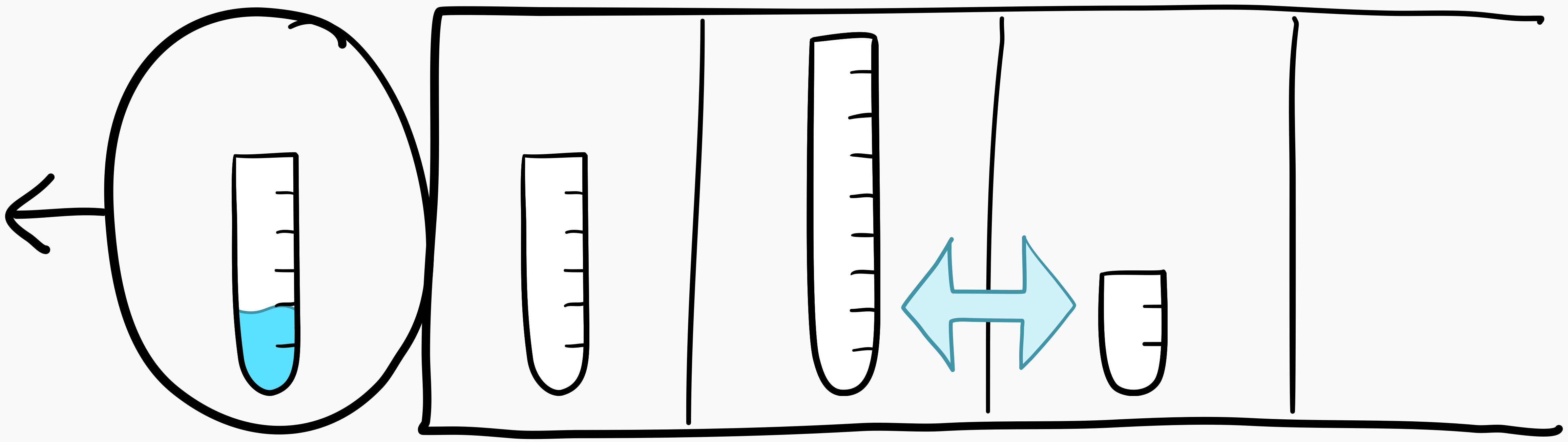


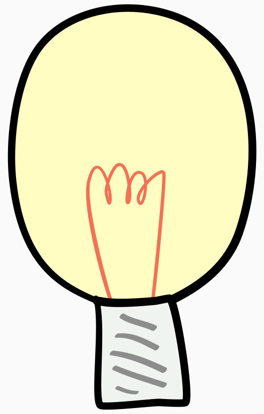


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

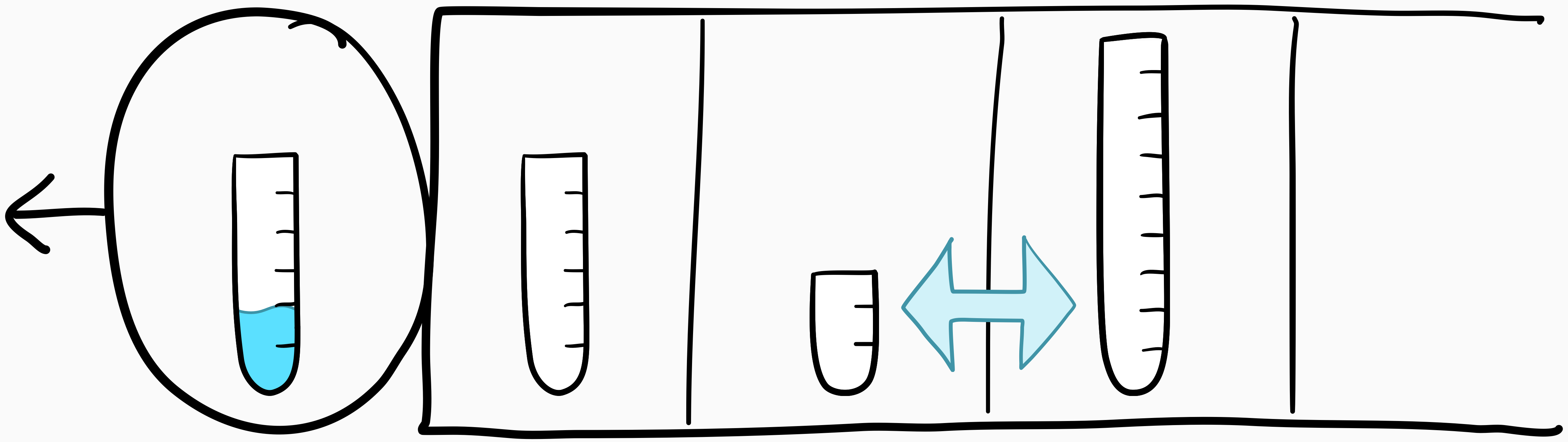


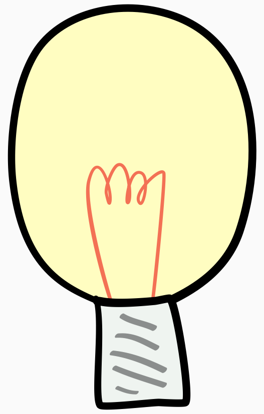


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

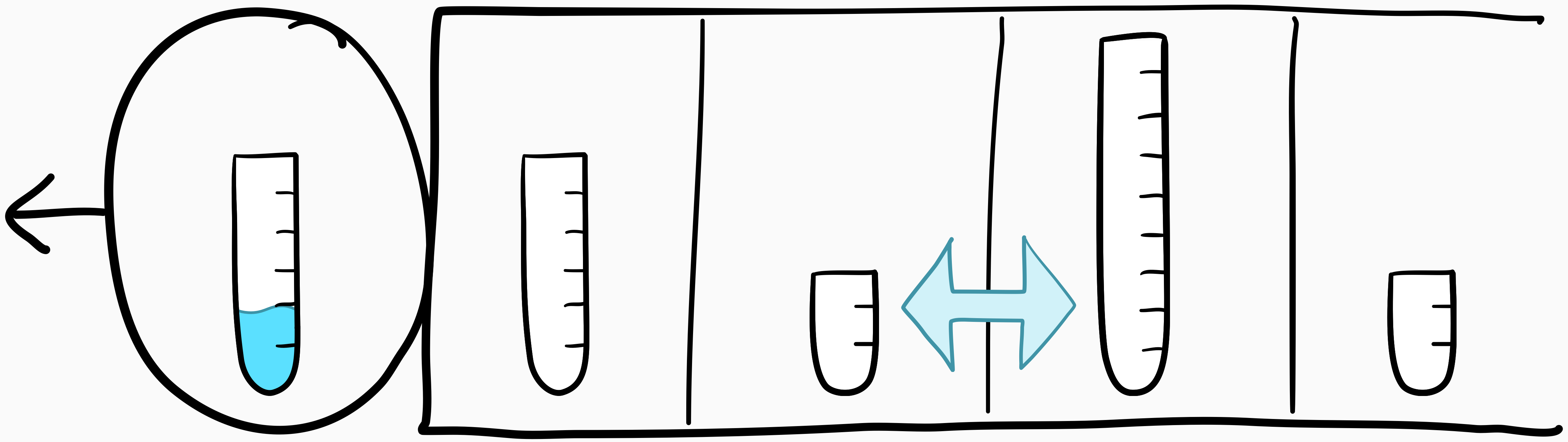


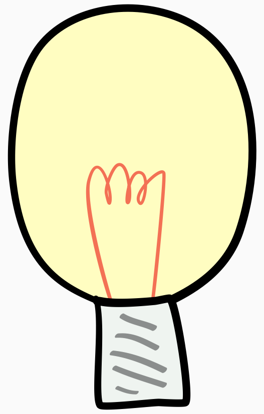


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

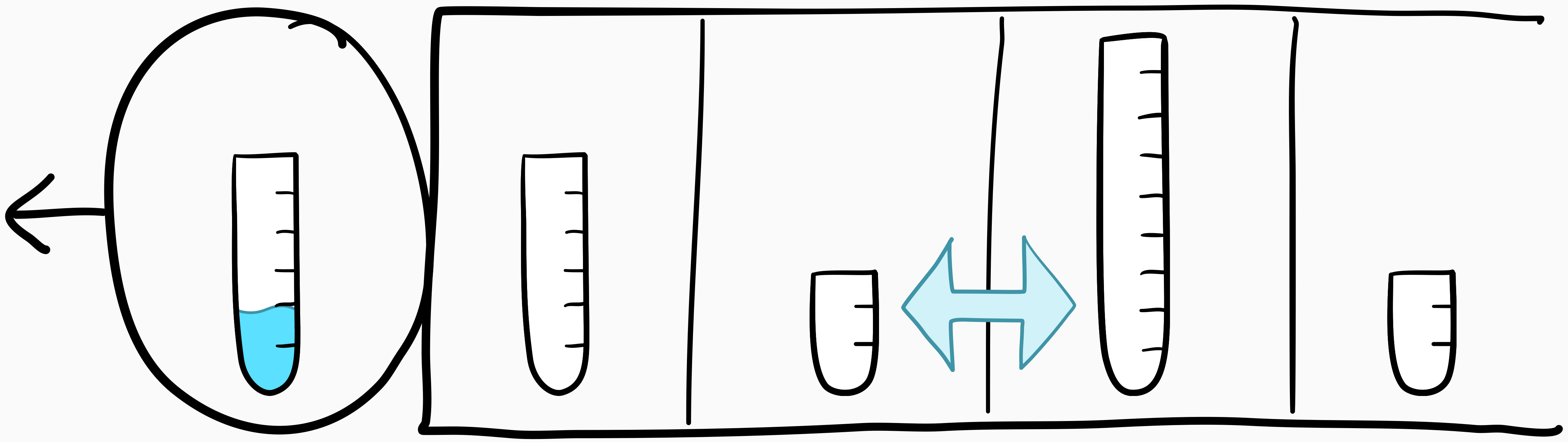


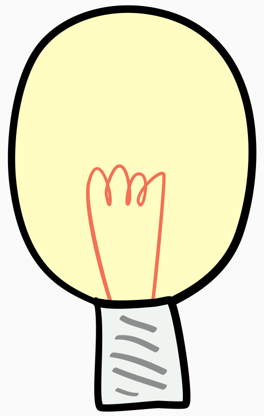


Nudge [Grosof et al., 2021]

- small job can pass one large job

Can we beat FCFS?

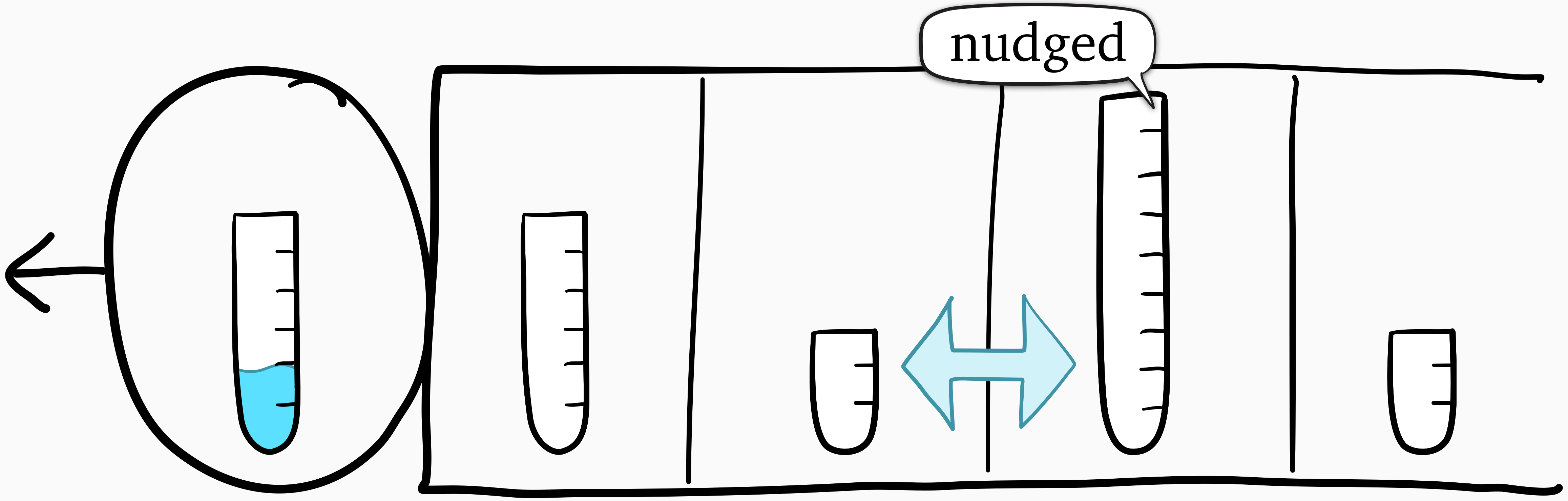


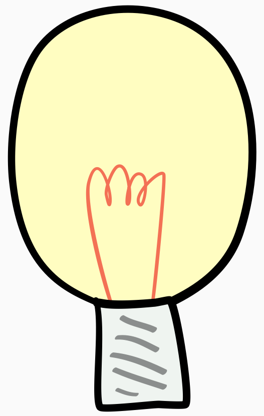


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?

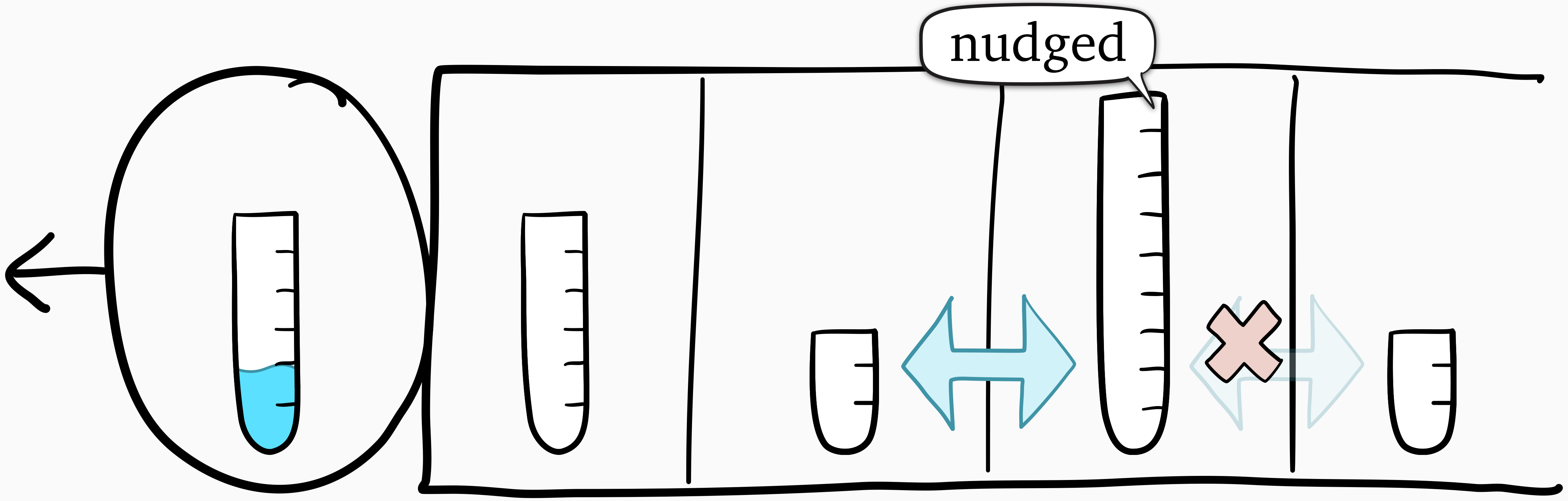


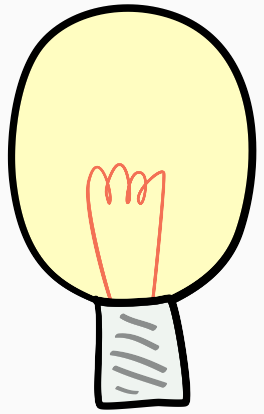


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?

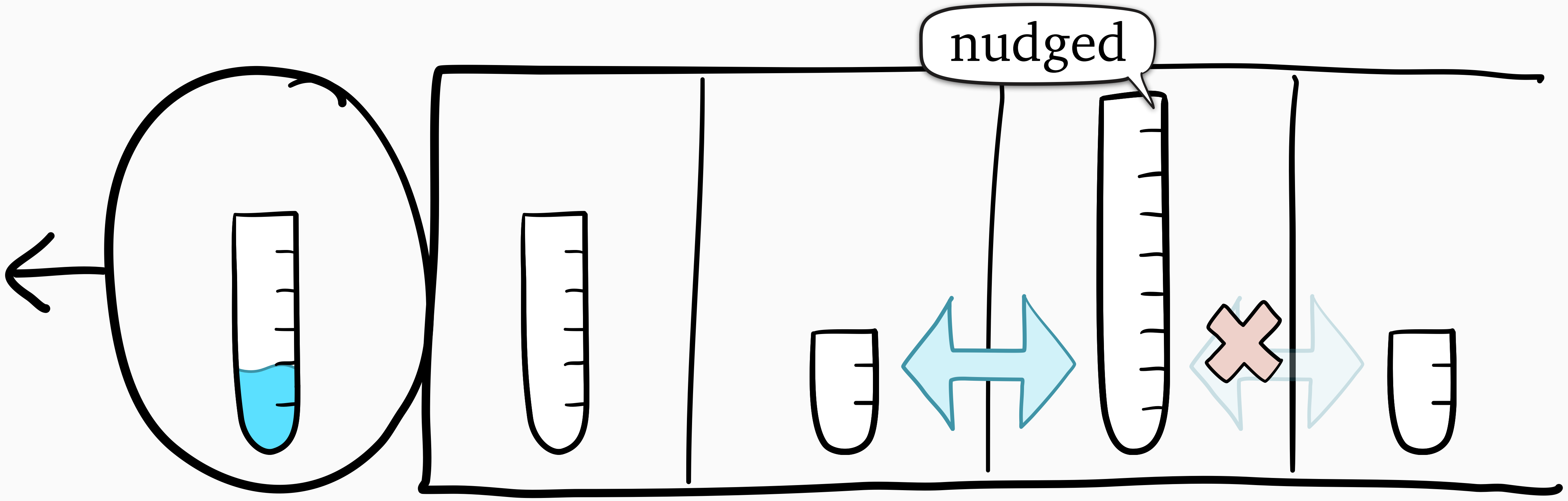


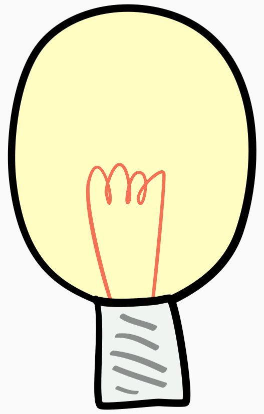


Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

Can we beat FCFS?





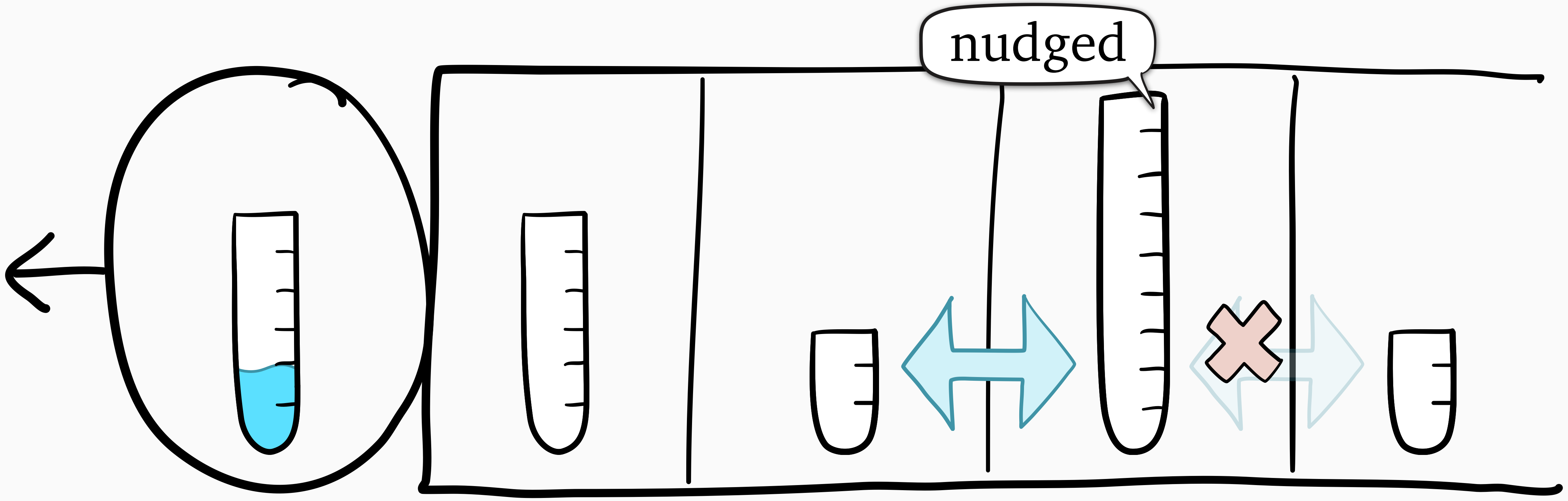
Nudge [Grosof et al., 2021]

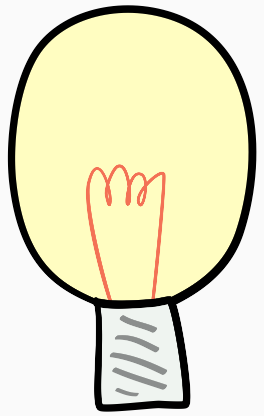
- small job can pass one large job
- large job can't be passed twice

Theorem:

$C_{\text{Nudge}} < C_{\text{FCFS}}$

Can we beat FCFS?





Nudge [Grosof et al., 2021]

- small job can pass one large job
- large job can't be passed twice

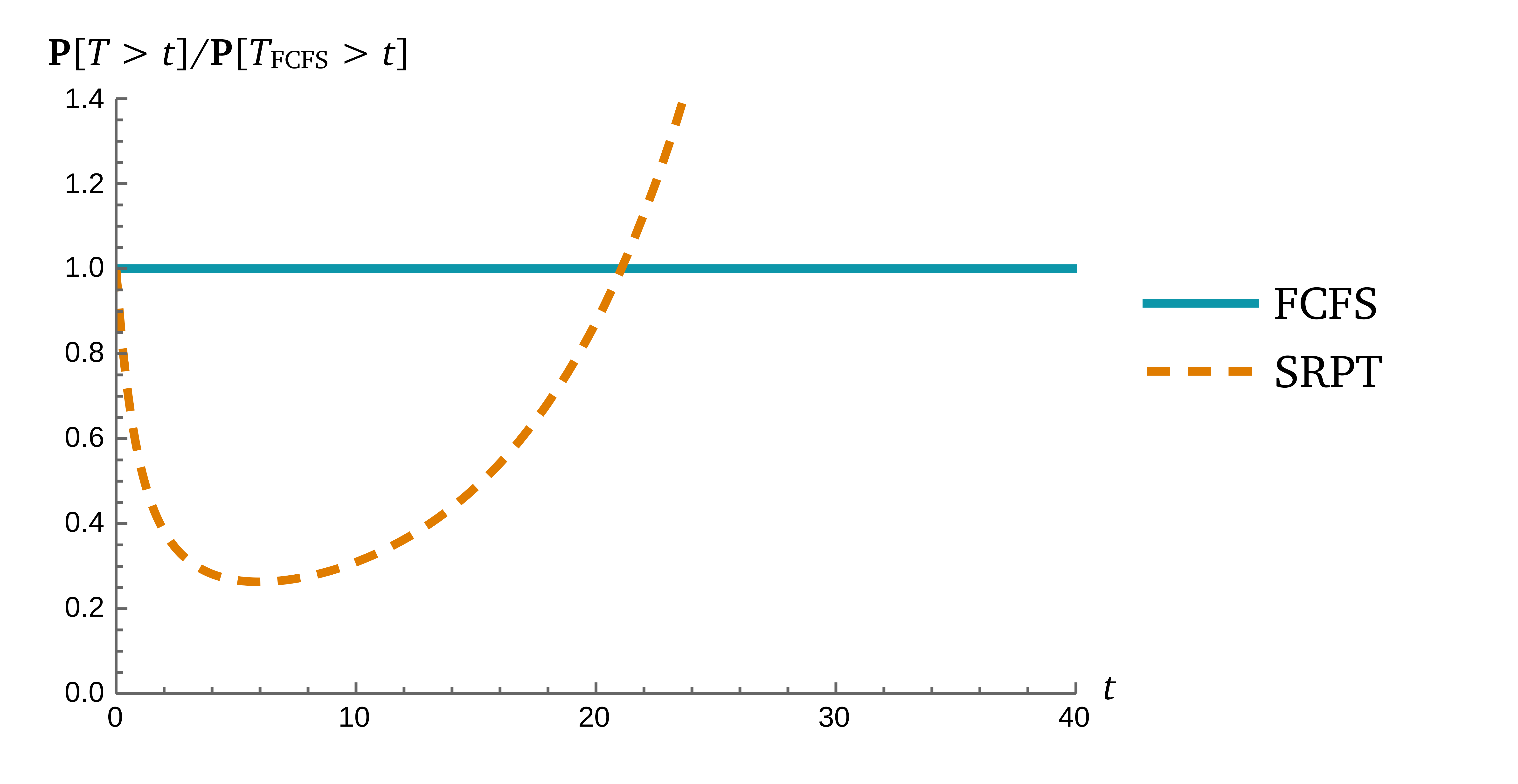
Theorem:

$C_{\text{Nudge}} < C_{\text{FCFS}}$

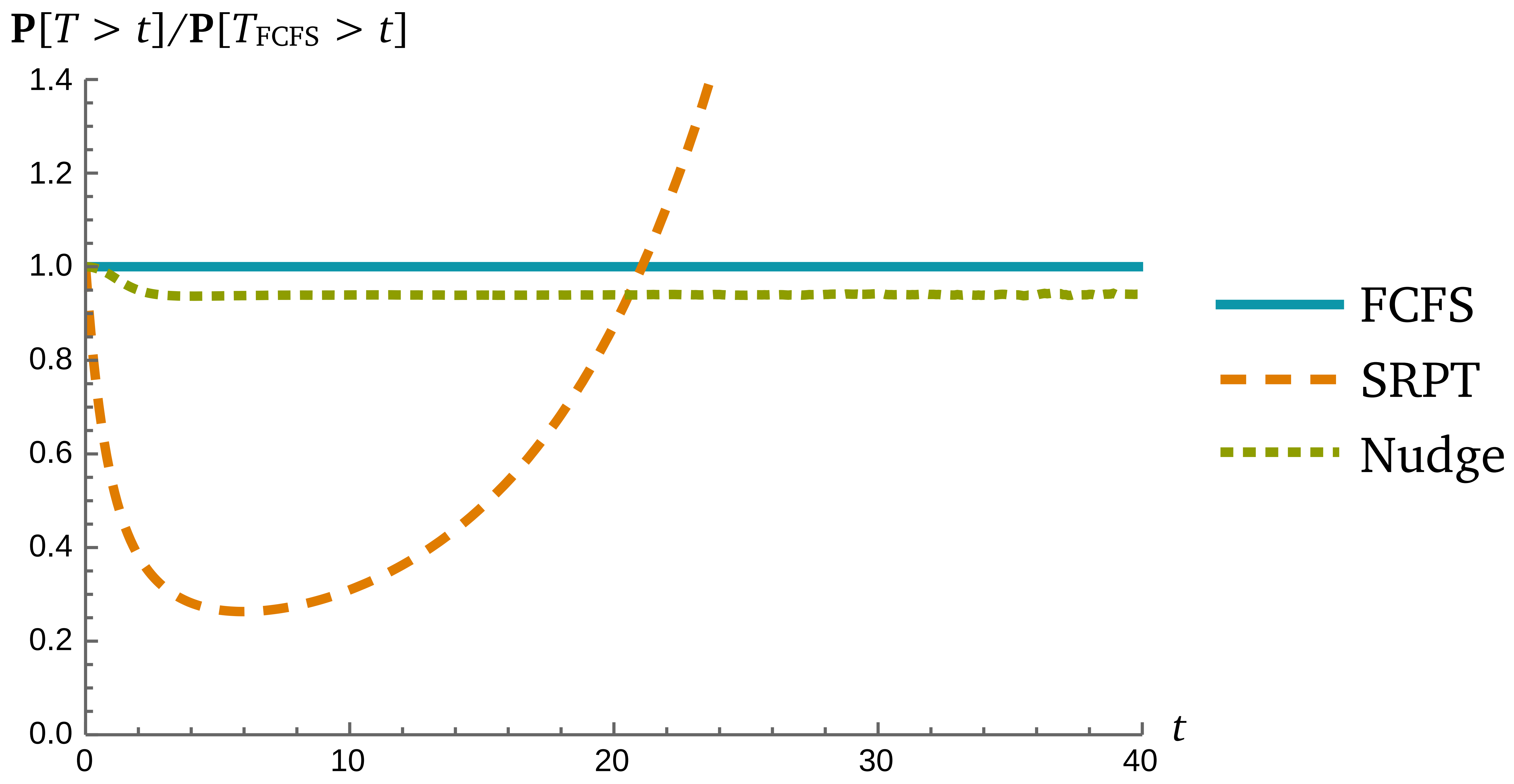
More complex variants get even lower C

[Van Houdt, 2022; Charlet & Van Houdt, 2024]

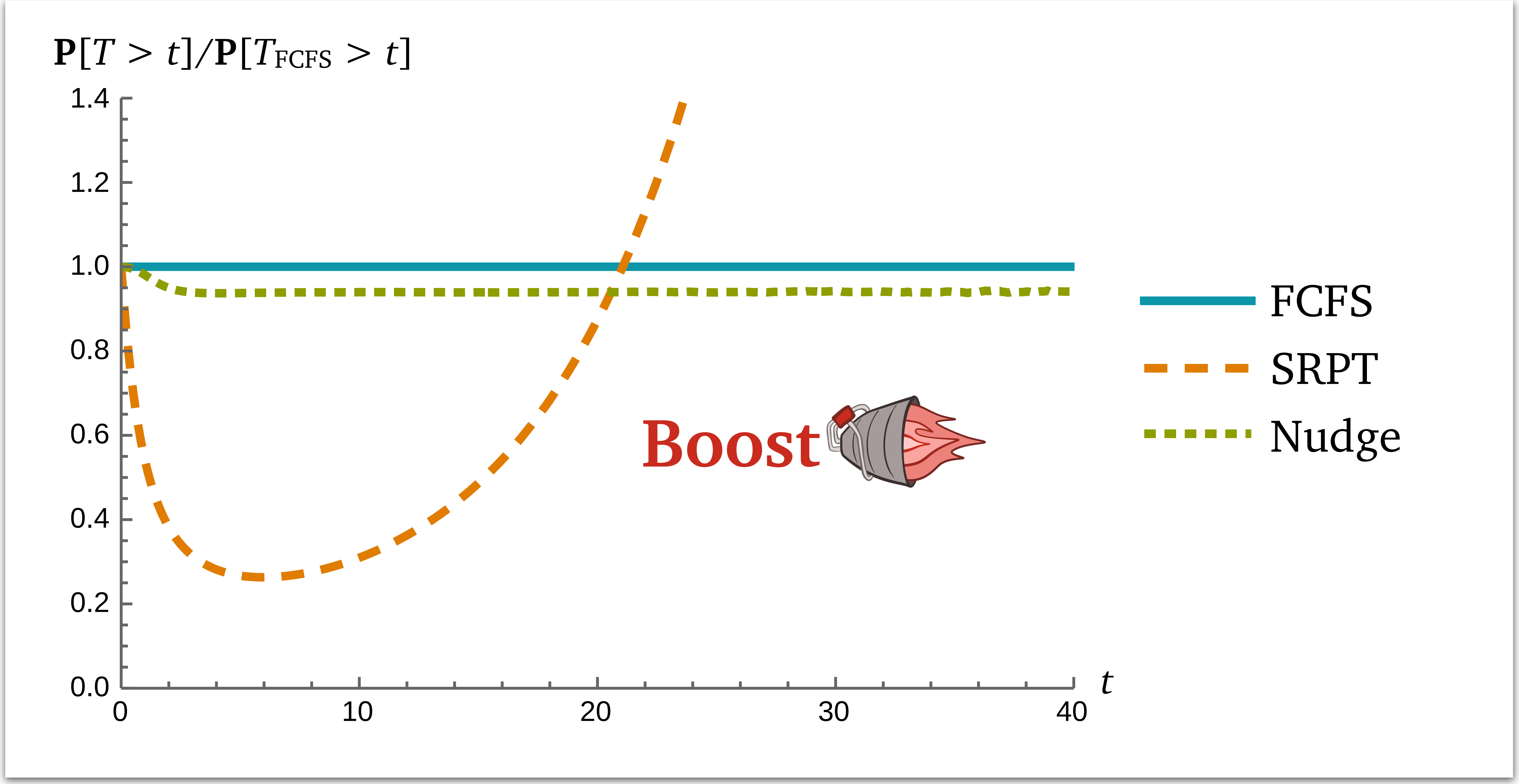
Optimizing the tail constant C



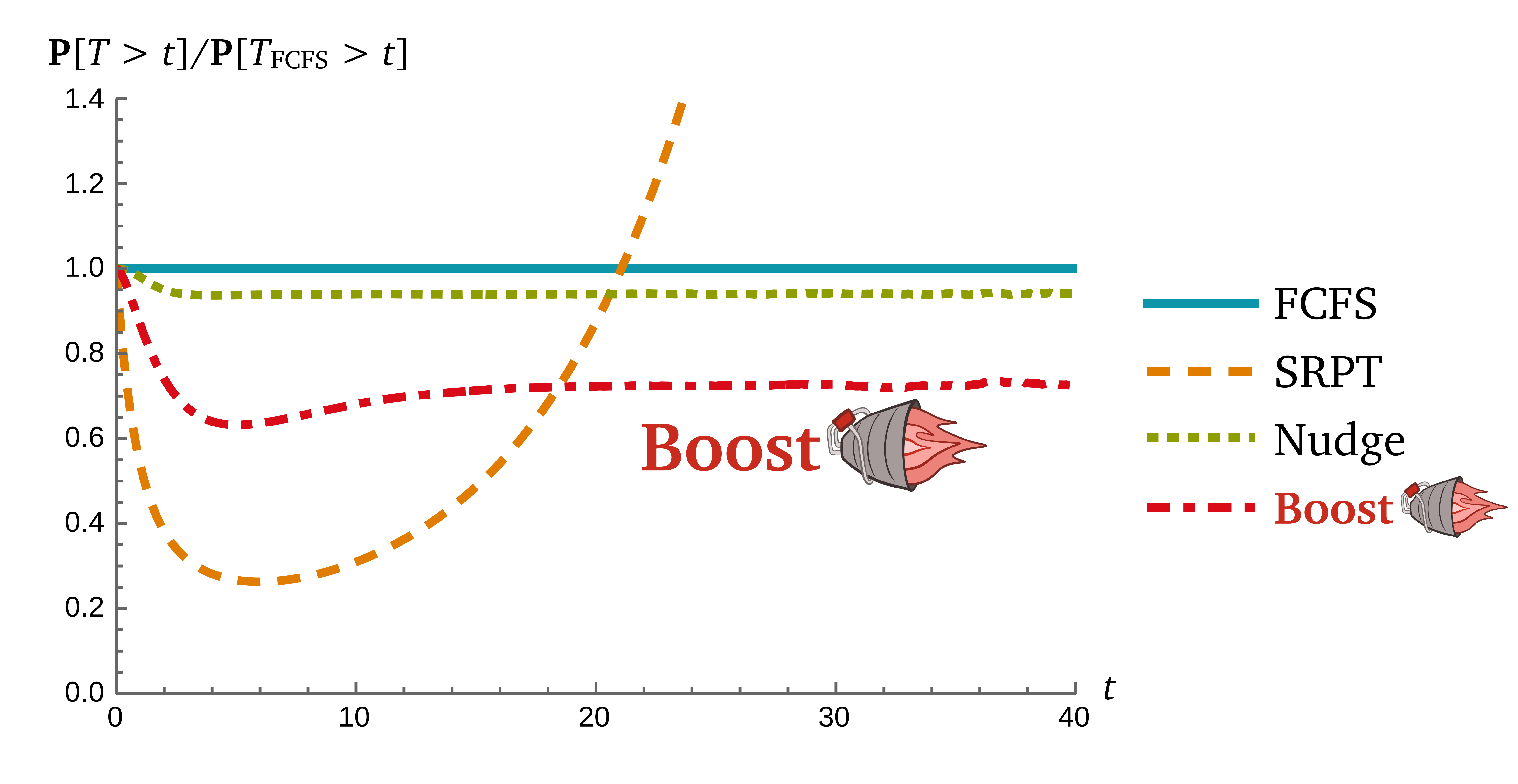
Optimizing the tail constant C



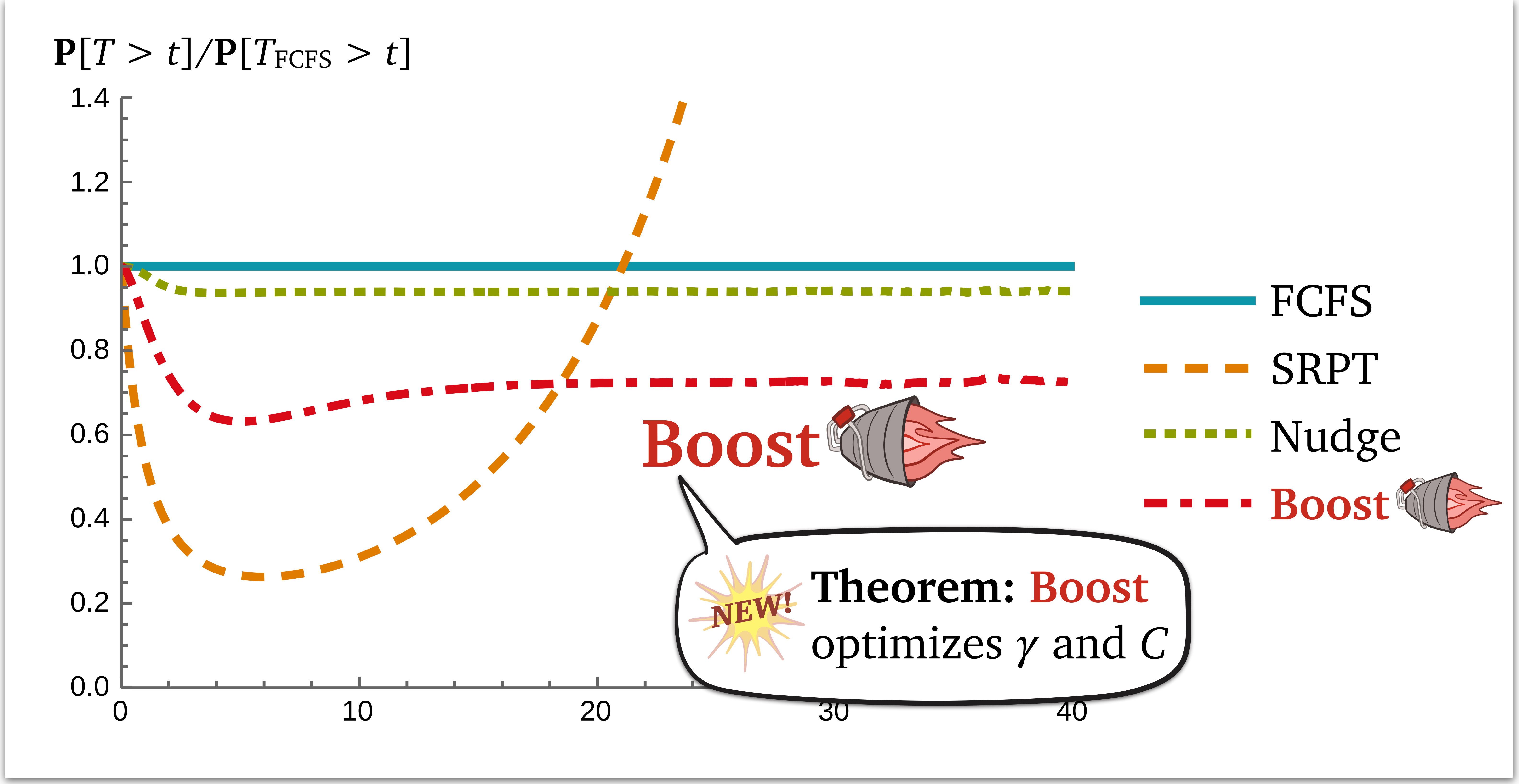
Optimizing the tail constant C



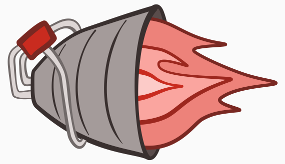
Optimizing the tail constant C

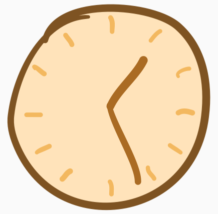


Optimizing the tail constant C



Our contributions:

 Design the **Boost** scheduling policy

 Analyze **Boost**'s performance

 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

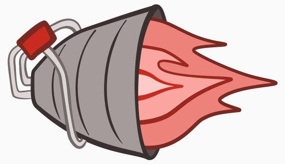
Our contributions:

actually a *family* of many policies

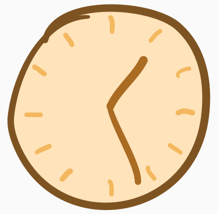
- Design the **Boost** scheduling policy
- Analyze **Boost**'s performance
- Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Our contributions:

actually a *family* of many policies

 Design the **Boost** scheduling policy

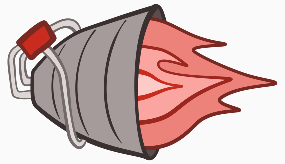
all instances

 Analyze **Boost**’s performance

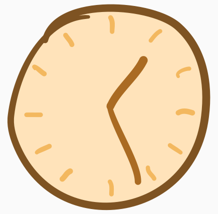
 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Our contributions:

actually a *family* of many policies

 Design the **Boost** scheduling policy

all instances

 Analyze **Boost**'s performance

specific instance called **γ-Boost**

 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Our contributions:

Design the **Boost** scheduling policy

actually a *family* of many policies

Analyze **Boost**'s performance

all instances

Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

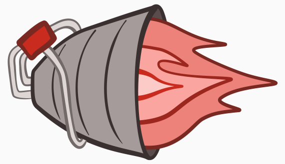
specific instance called **γ-Boost**

Known job sizes

Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

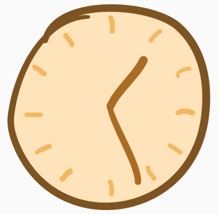
Our contributions:

actually a *family* of many policies



Design the **Boost** scheduling policy

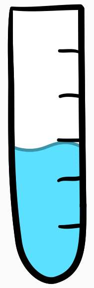
all instances



Analyze **Boost**'s performance

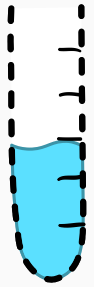
specific instance called ***γ*-Boost**

Prove **Boost** is *strongly tail-optimal* for light-tailed sizes



Known job sizes

Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

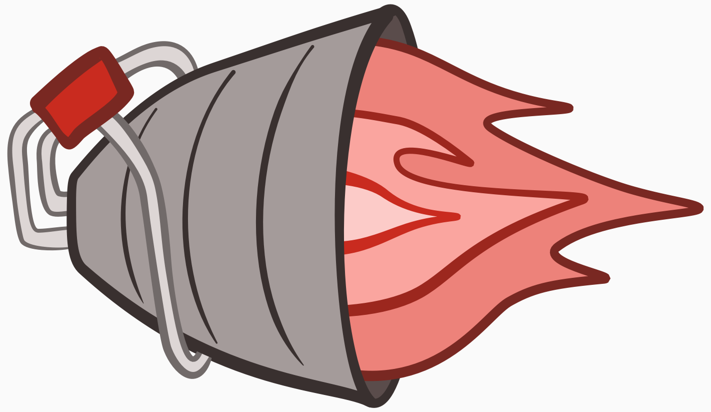


Unknown job sizes

Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.

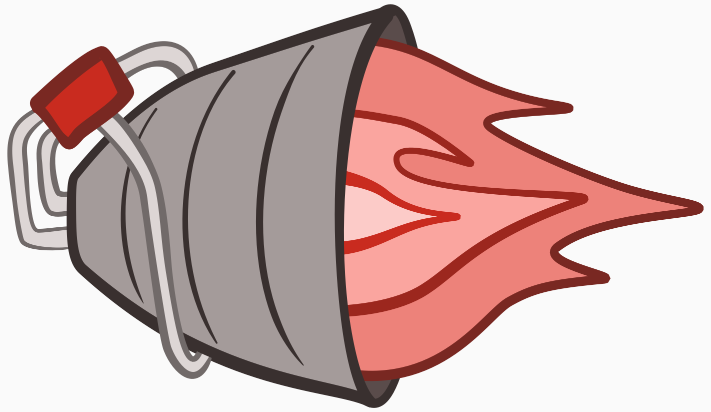
Boost

Boost



How does the **Boost** policy family work?

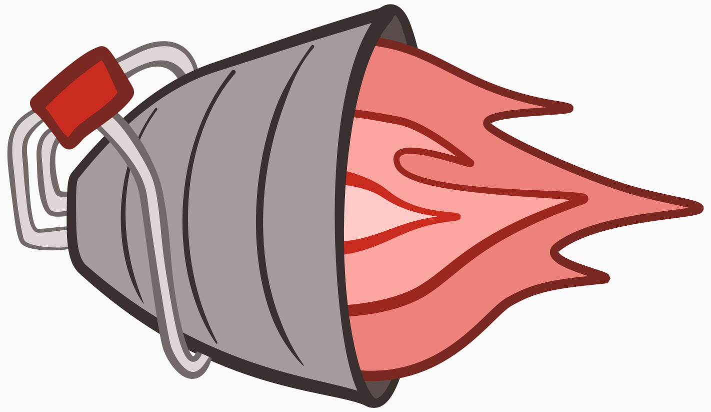
Boost



How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost

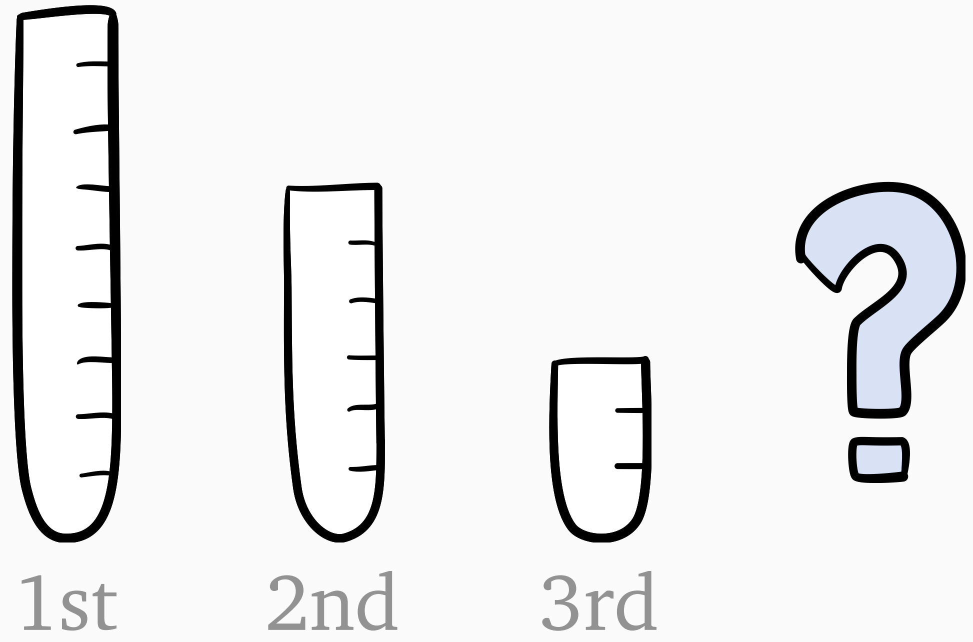
Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

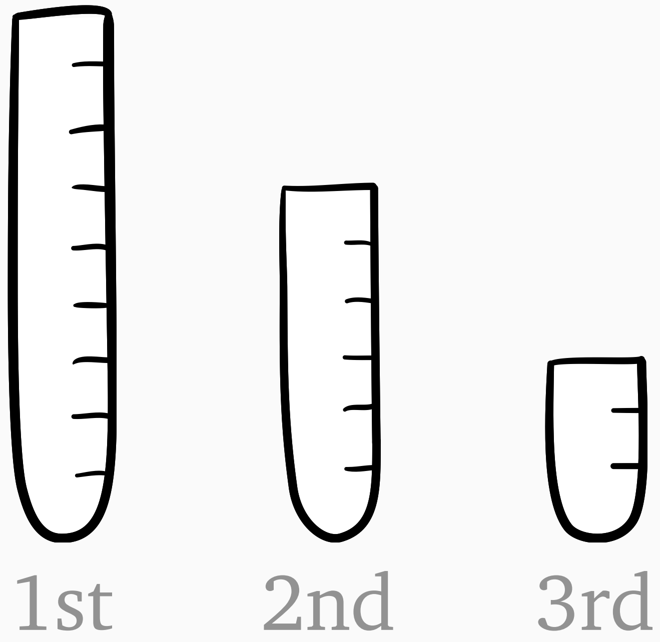
Can we beat Nudge?

Can we beat Nudge?



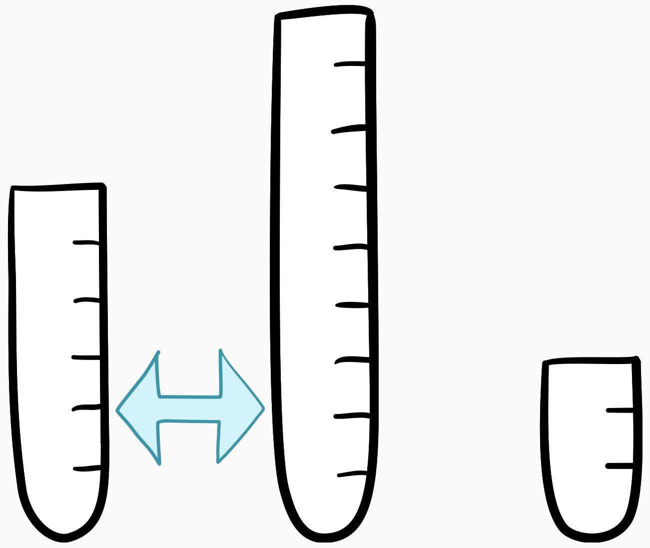
How to handle range of sizes?

Can we beat Nudge?

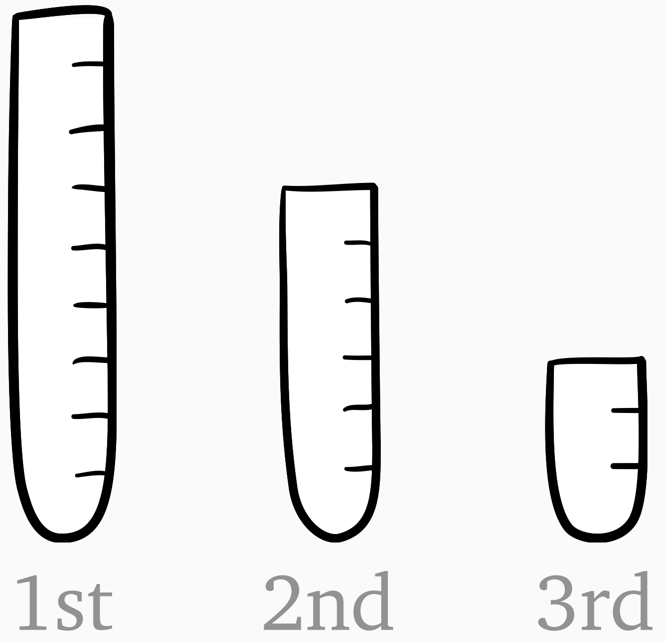


How to handle range of sizes?

medium = small?

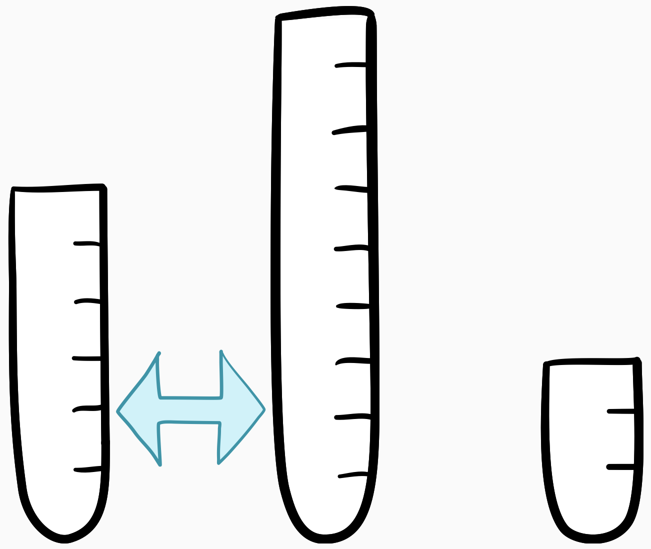


Can we beat Nudge?

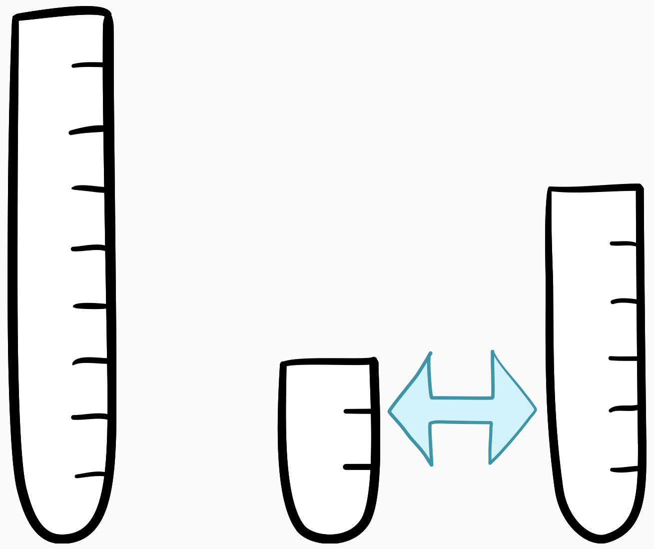


How to handle range of sizes?

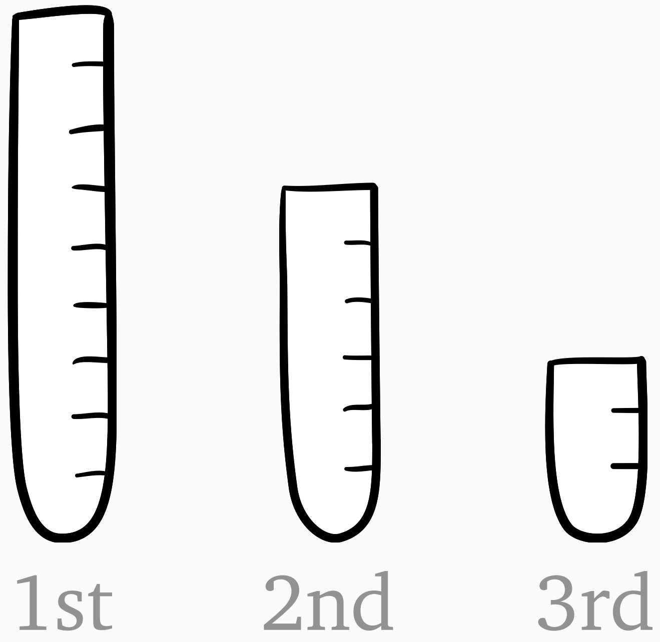
medium = small?



medium = large?

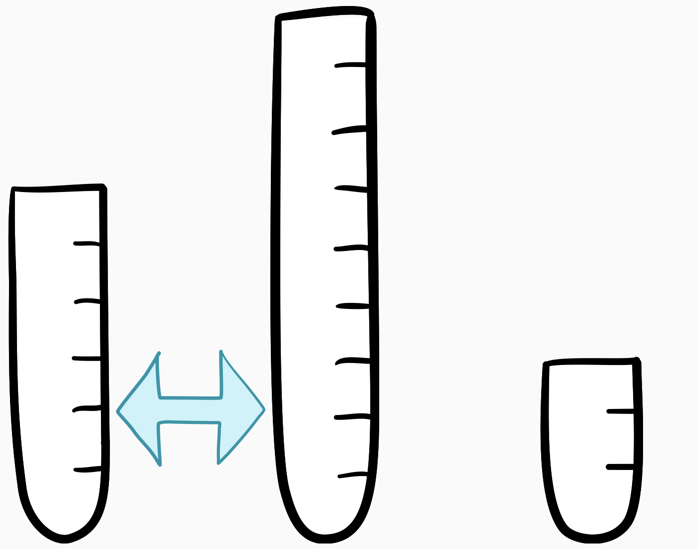


Can we beat Nudge?

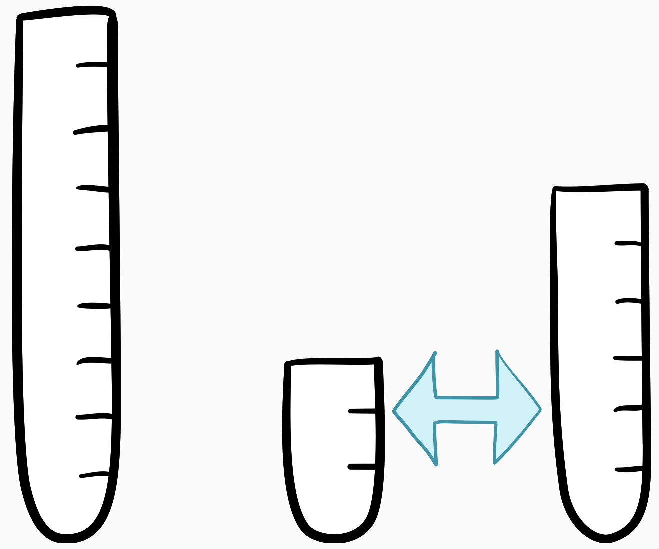


How to handle range of sizes?

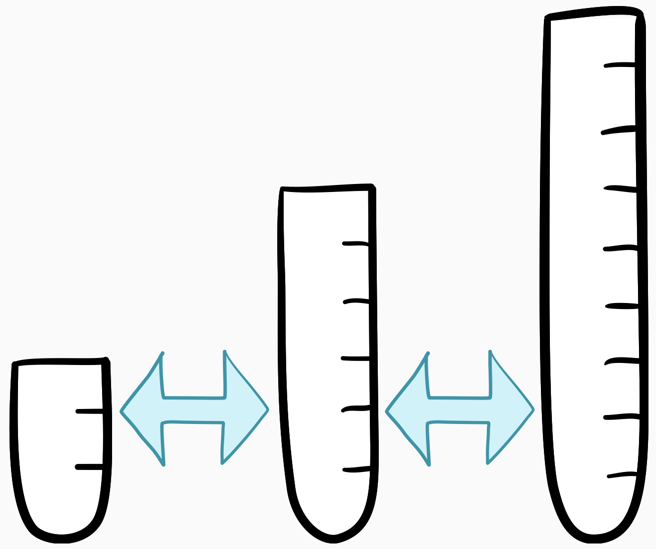
medium = small?



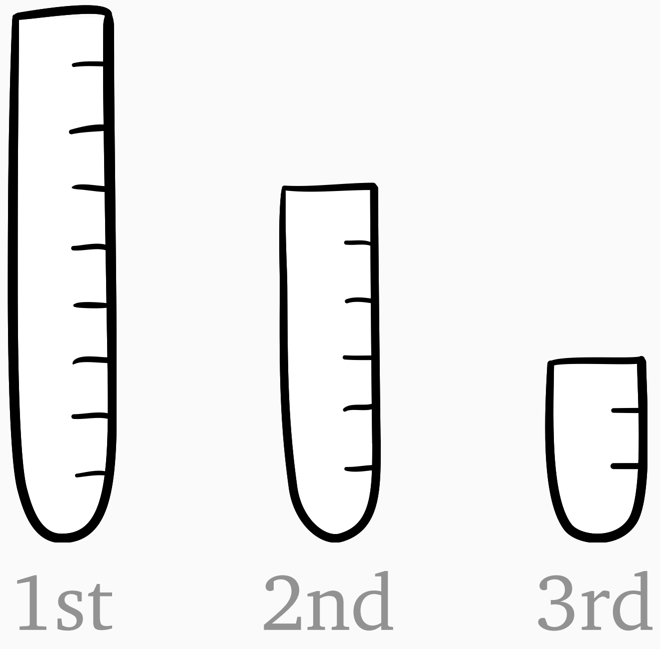
medium = large?



something else?



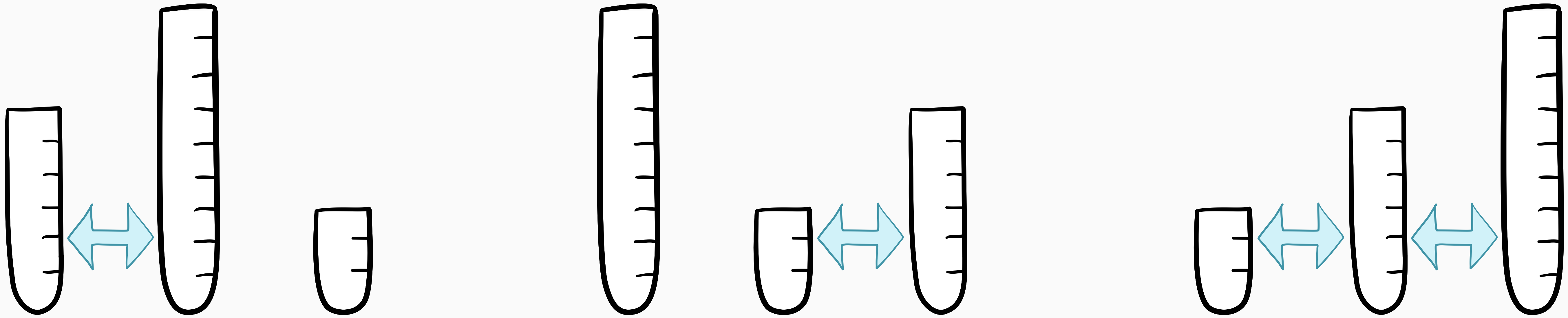
Can we beat Nudge?



How to handle range of sizes?

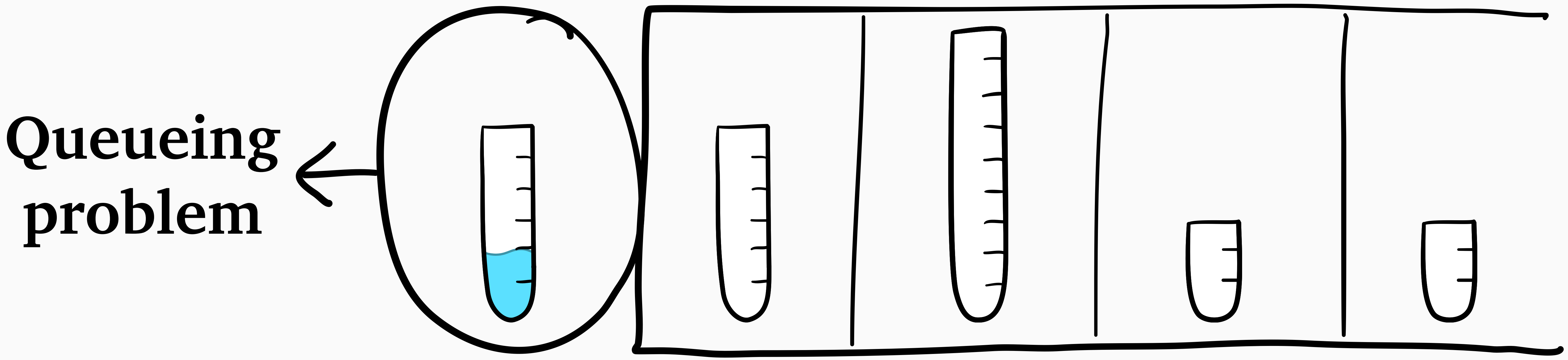
What info could help us decide?

medium = small? medium = large? something else?

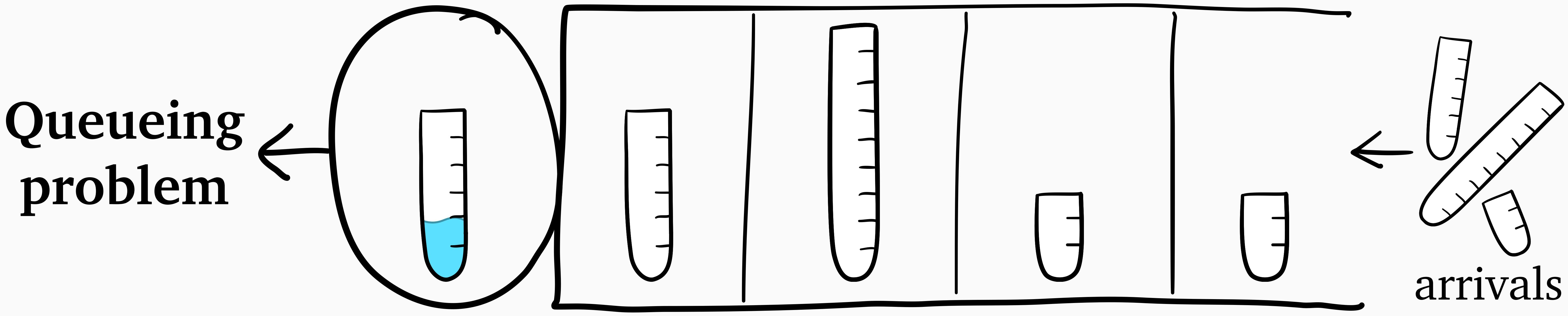


Where do optimal policies come from?

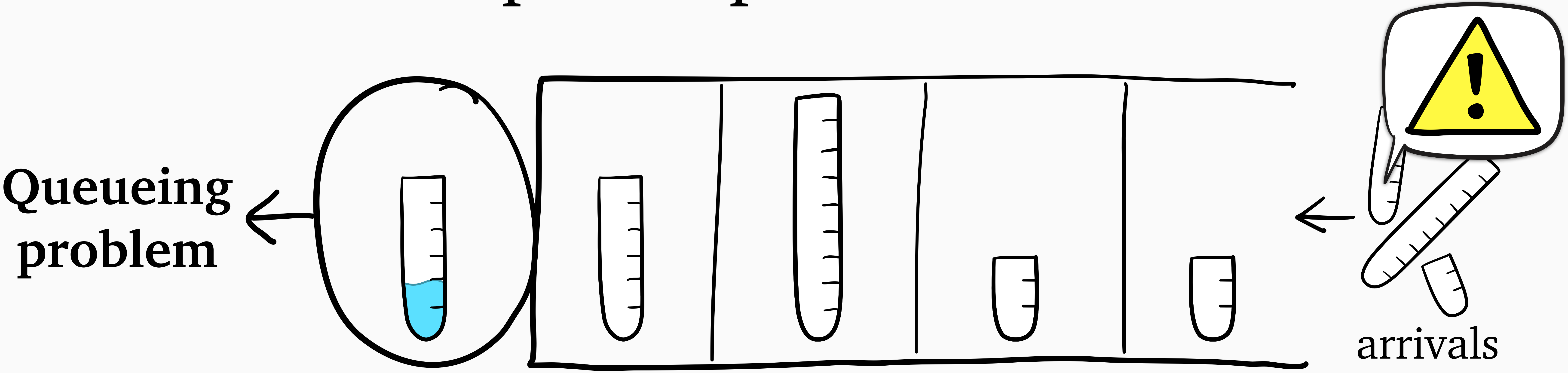
Where do optimal policies come from?



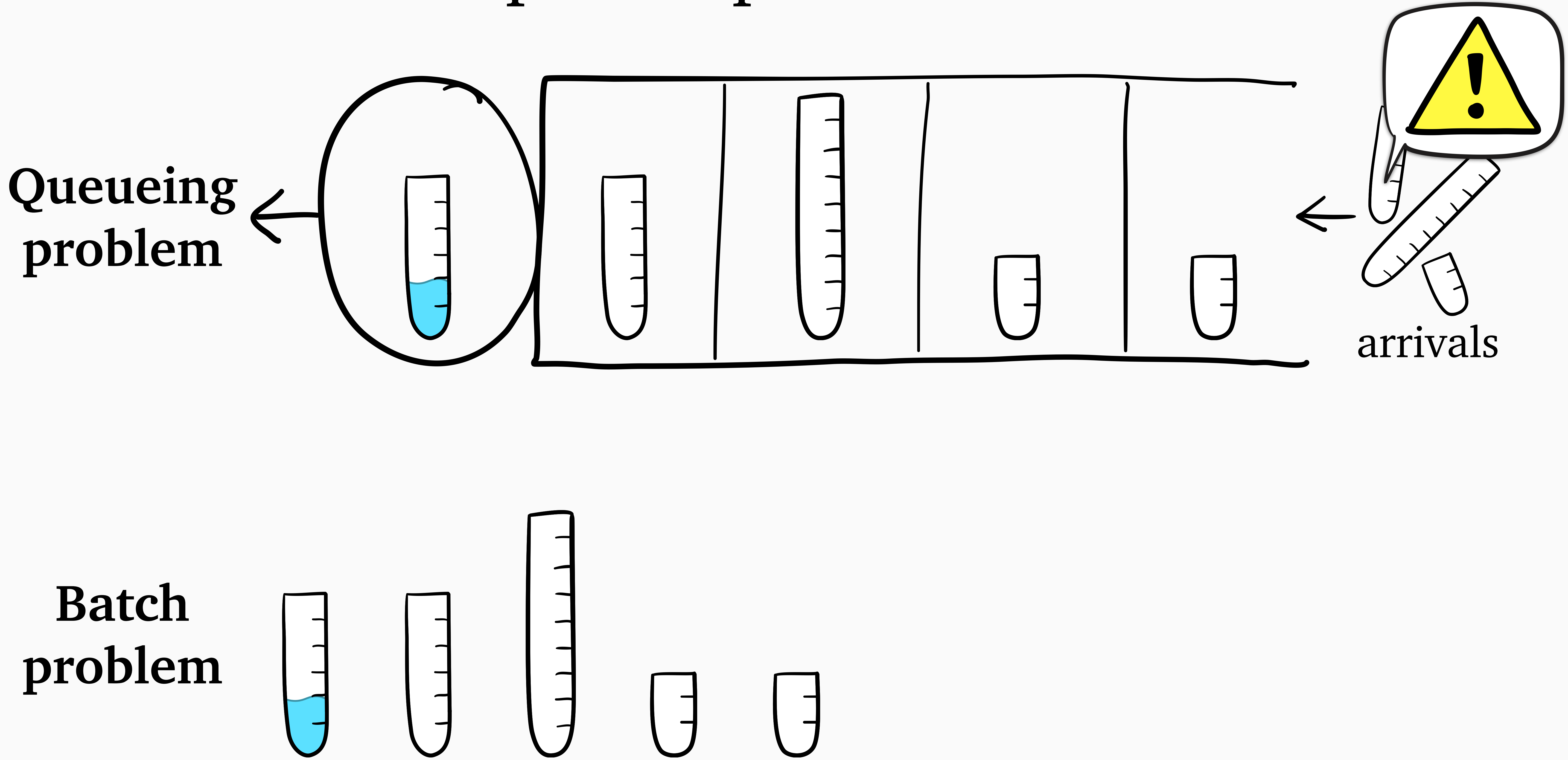
Where do optimal policies come from?



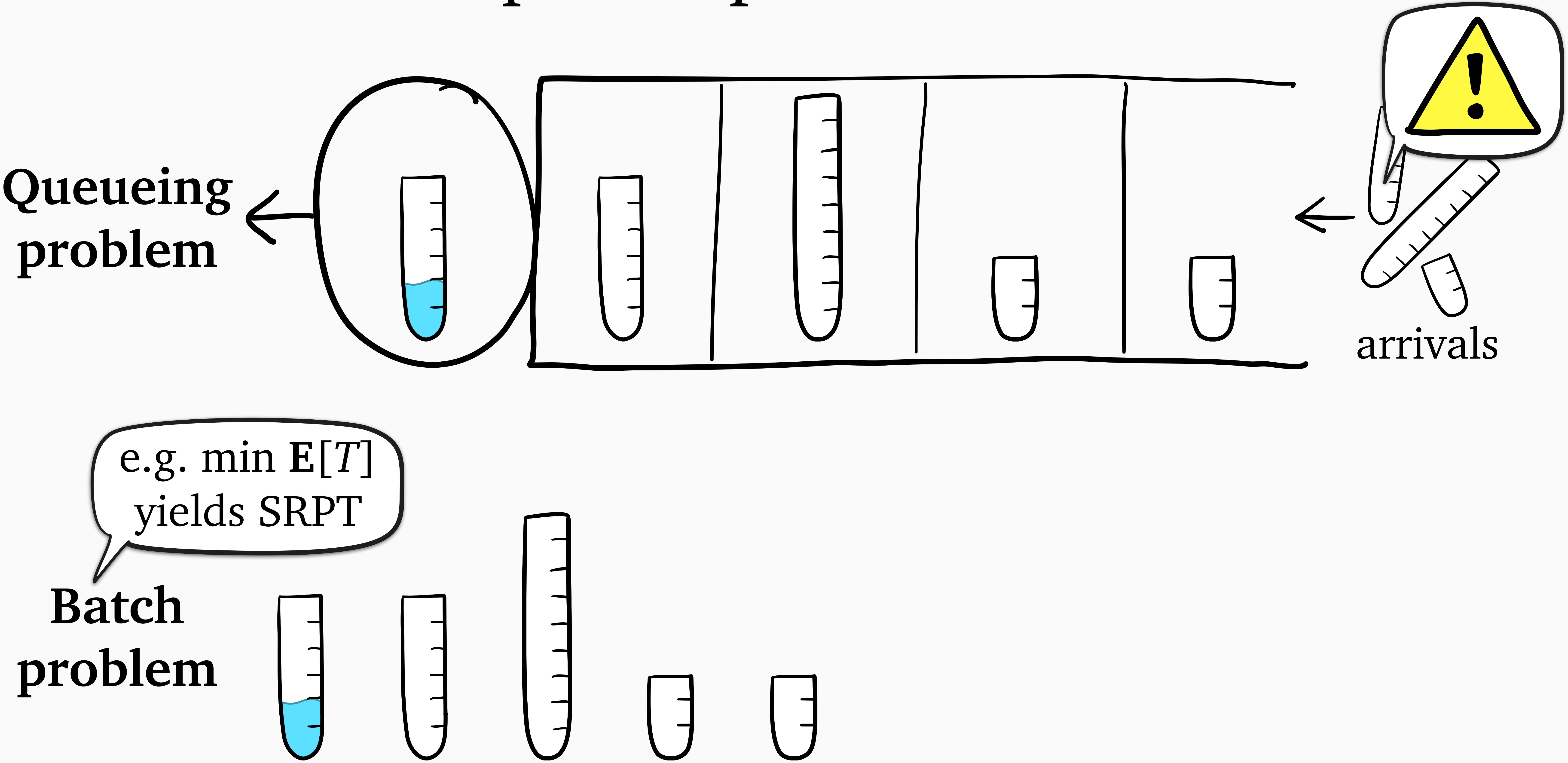
Where do optimal policies come from?



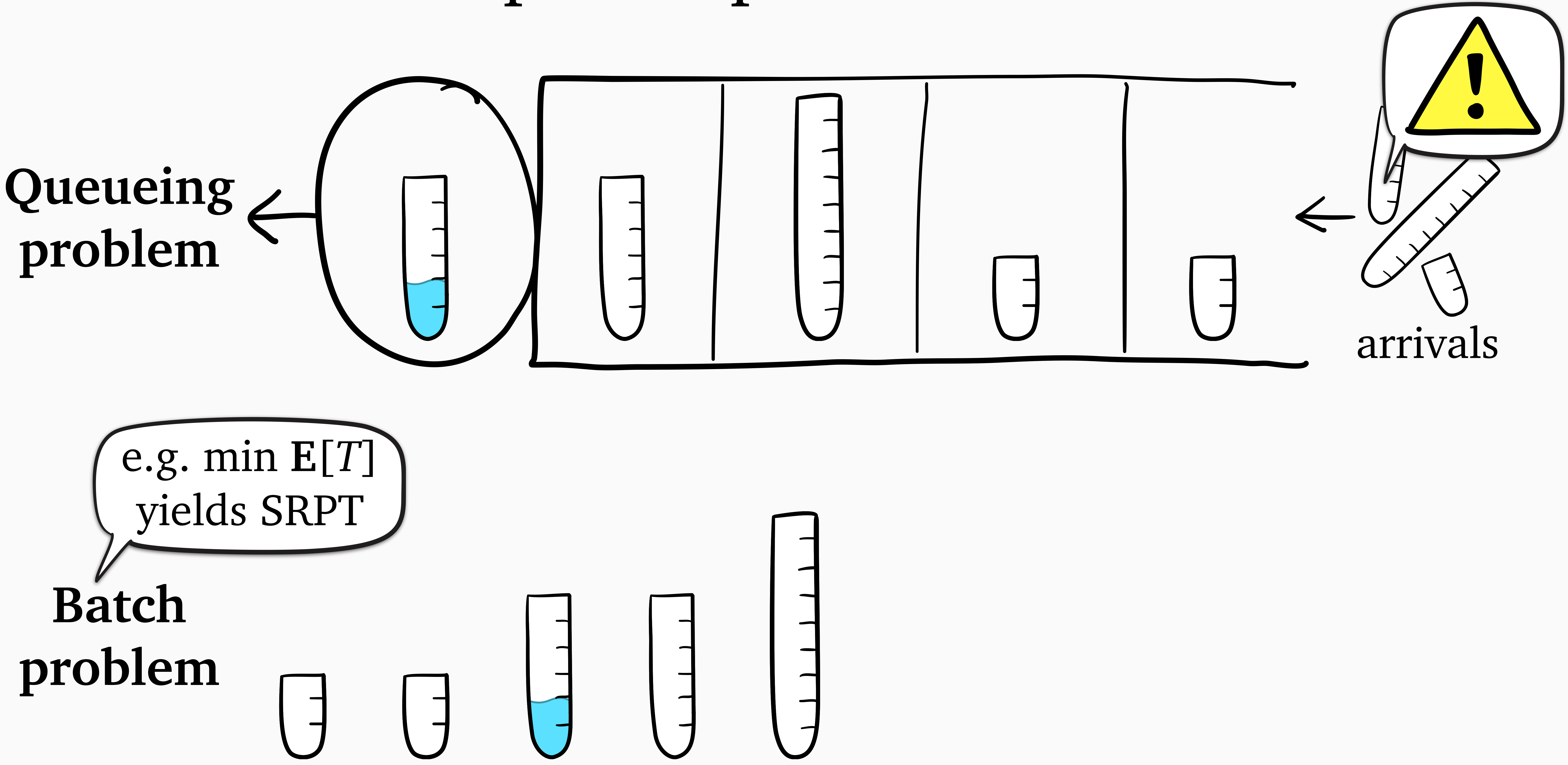
Where do optimal policies come from?



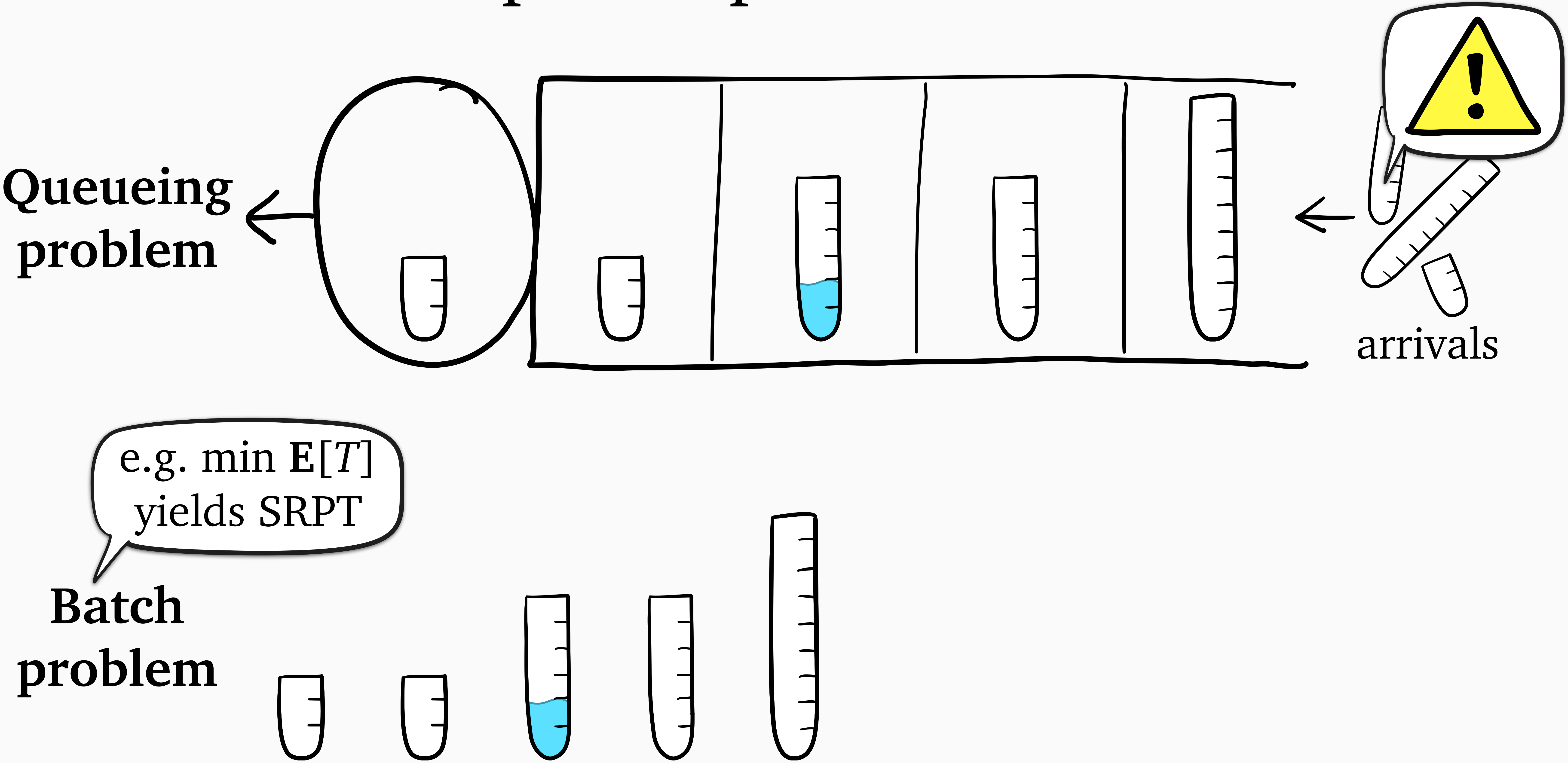
Where do optimal policies come from?



Where do optimal policies come from?

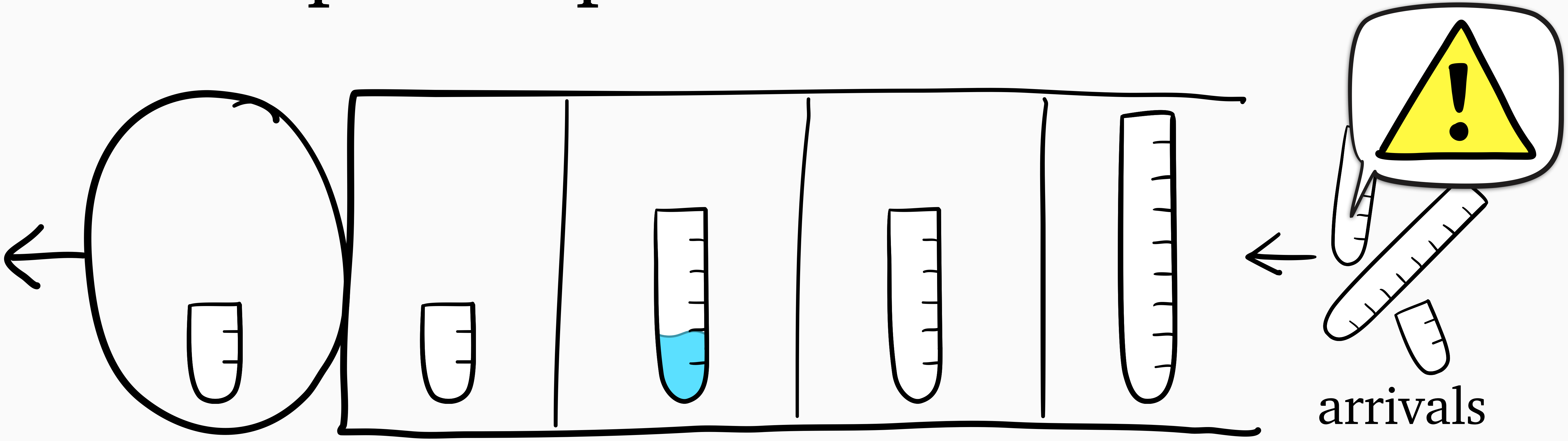


Where do optimal policies come from?



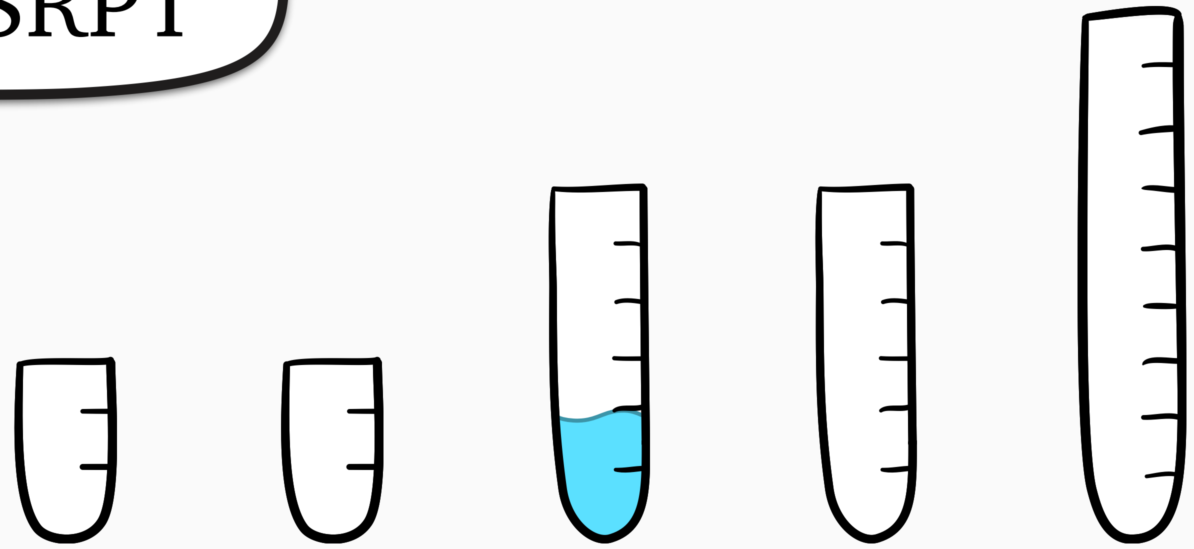
Where do optimal policies come from?

Queueing problem



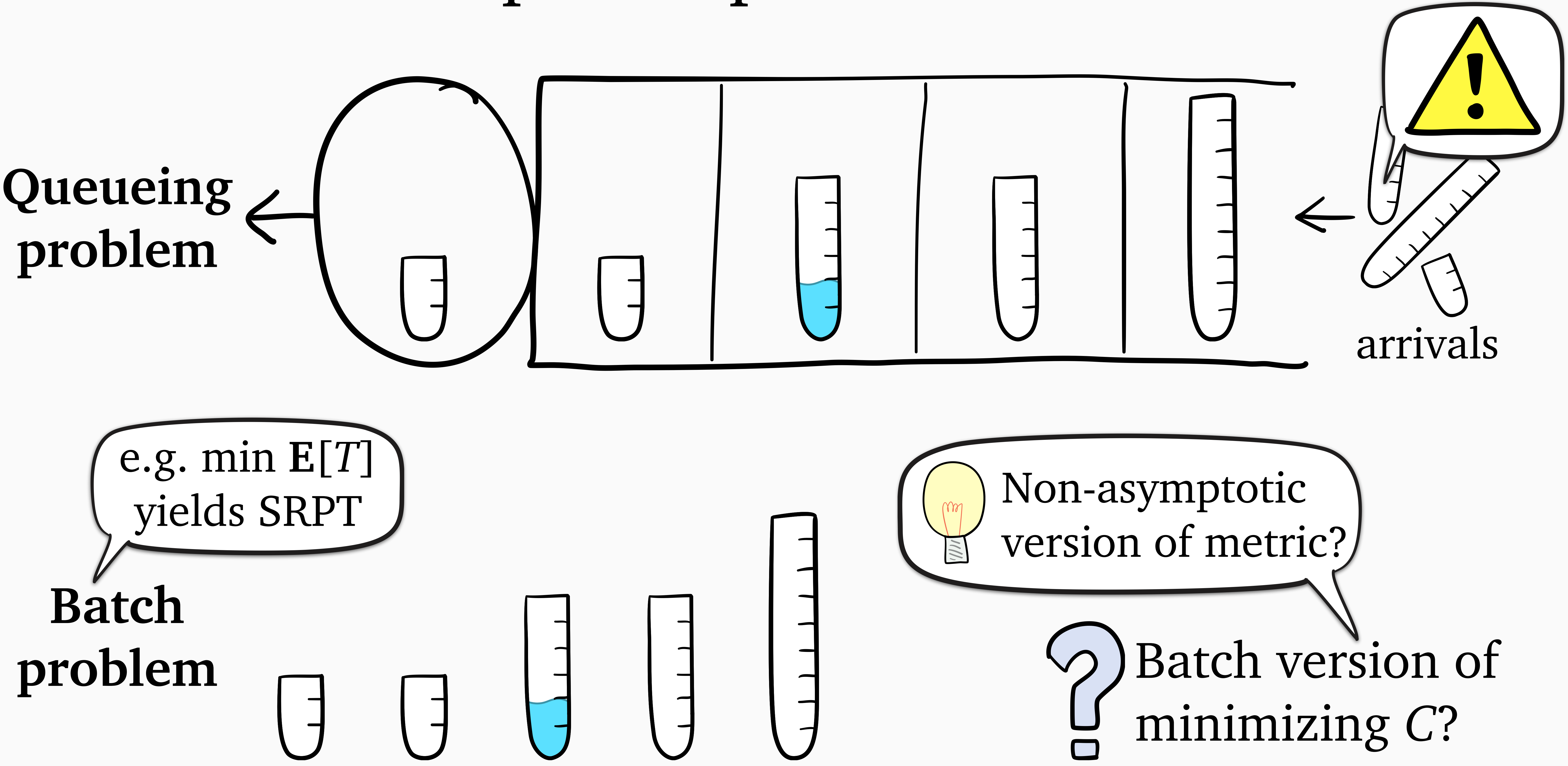
e.g. min $\mathbf{E}[T]$
yields SRPT

Batch problem



Batch version of minimizing C?

Where do optimal policies come from?



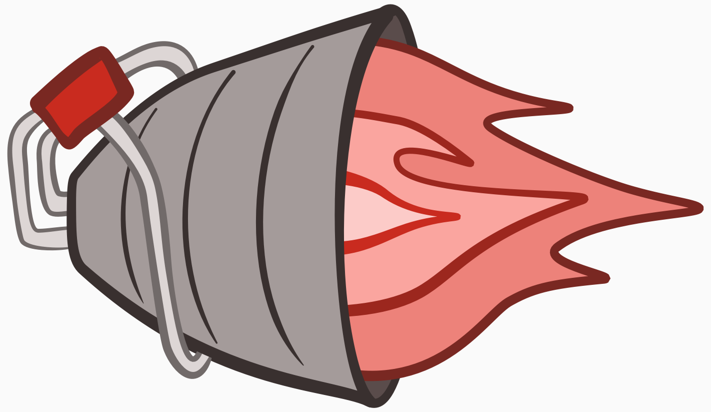
Boost

Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

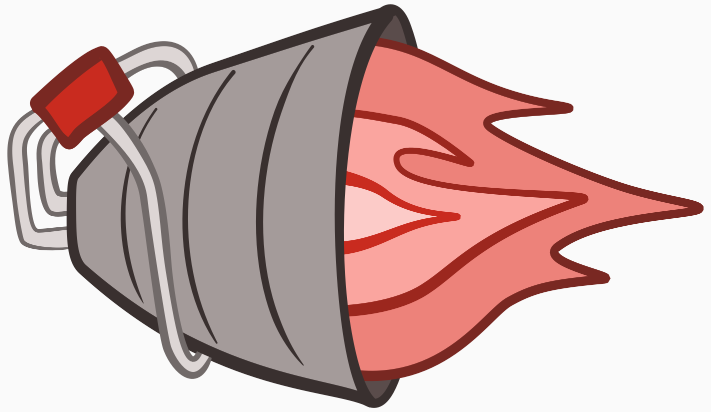
How to handle range of sizes?

Batch version of minimizing C?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

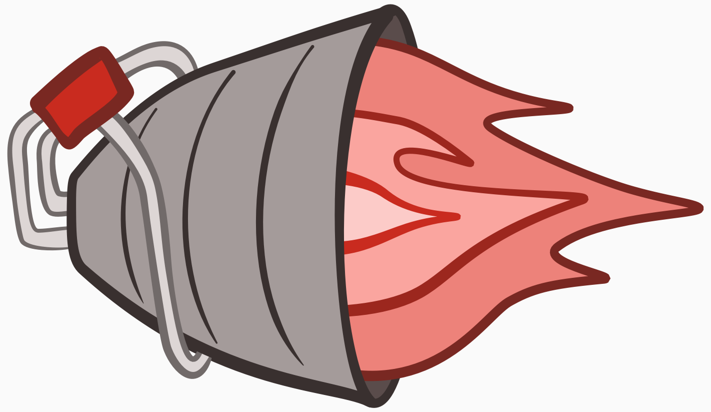
How to handle range of sizes?

Batch version of minimizing C?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Boost



Why is achieving strong tail optimality hard?

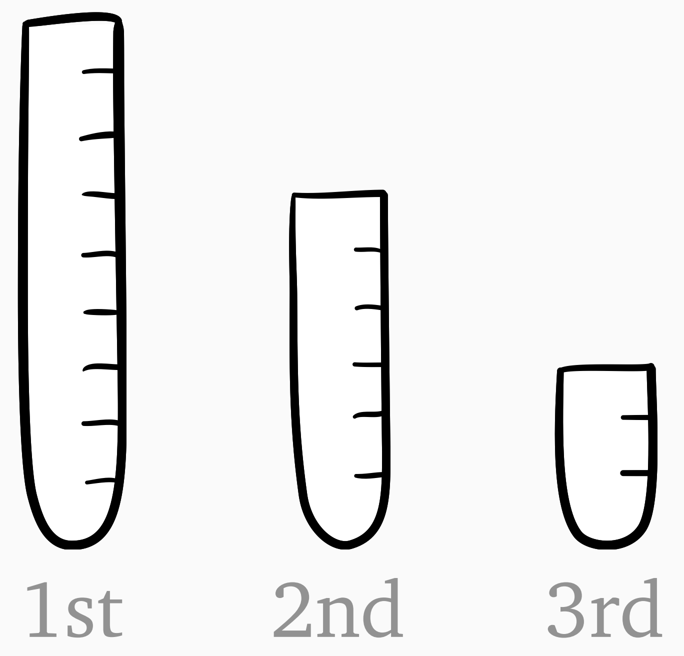
How to handle range of sizes?

Batch version of minimizing C?

How does the **Boost** policy family work?

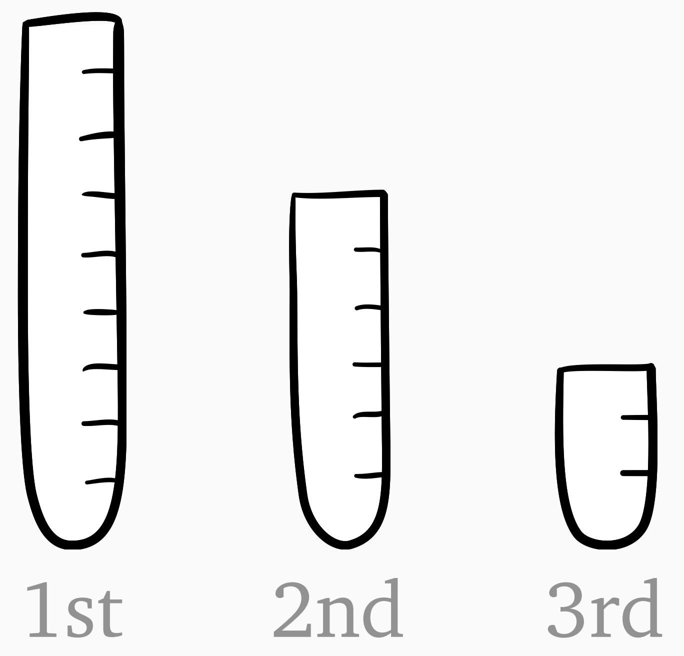
How do we achieve strong tail optimality?

Key information:



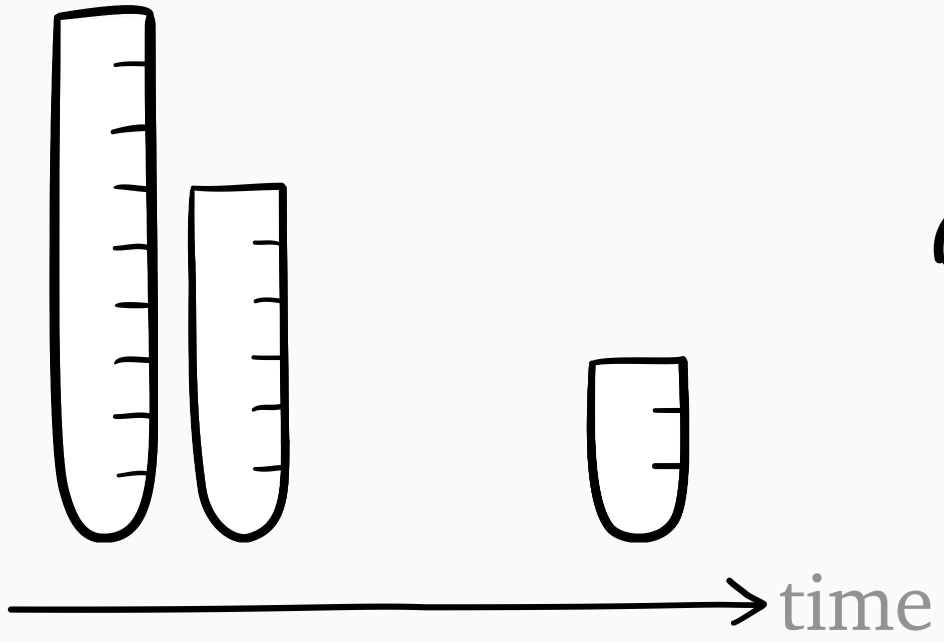
How to handle range of sizes?

Key information: *arrival times*



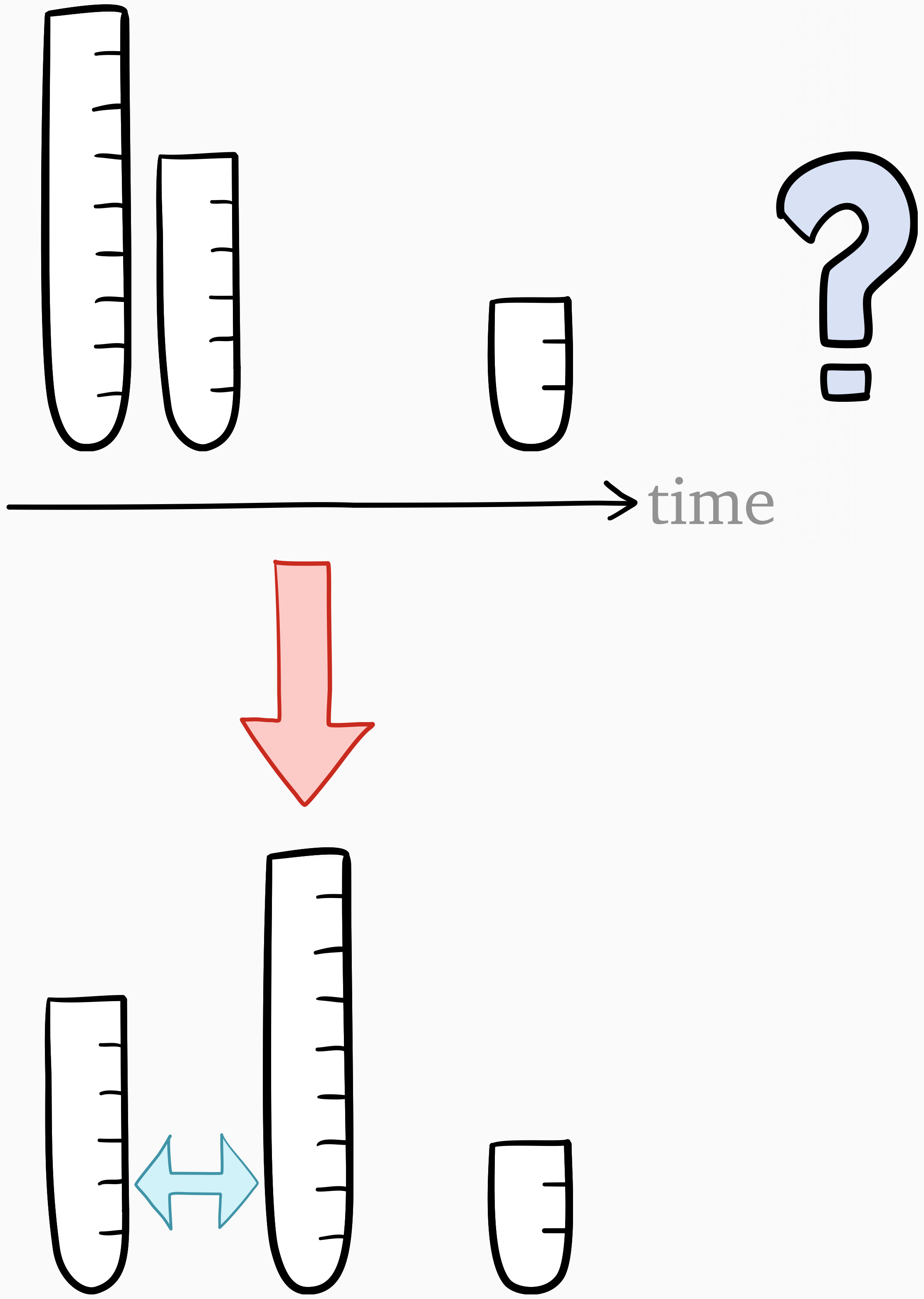
How to handle range of sizes?

Key information: *arrival times*



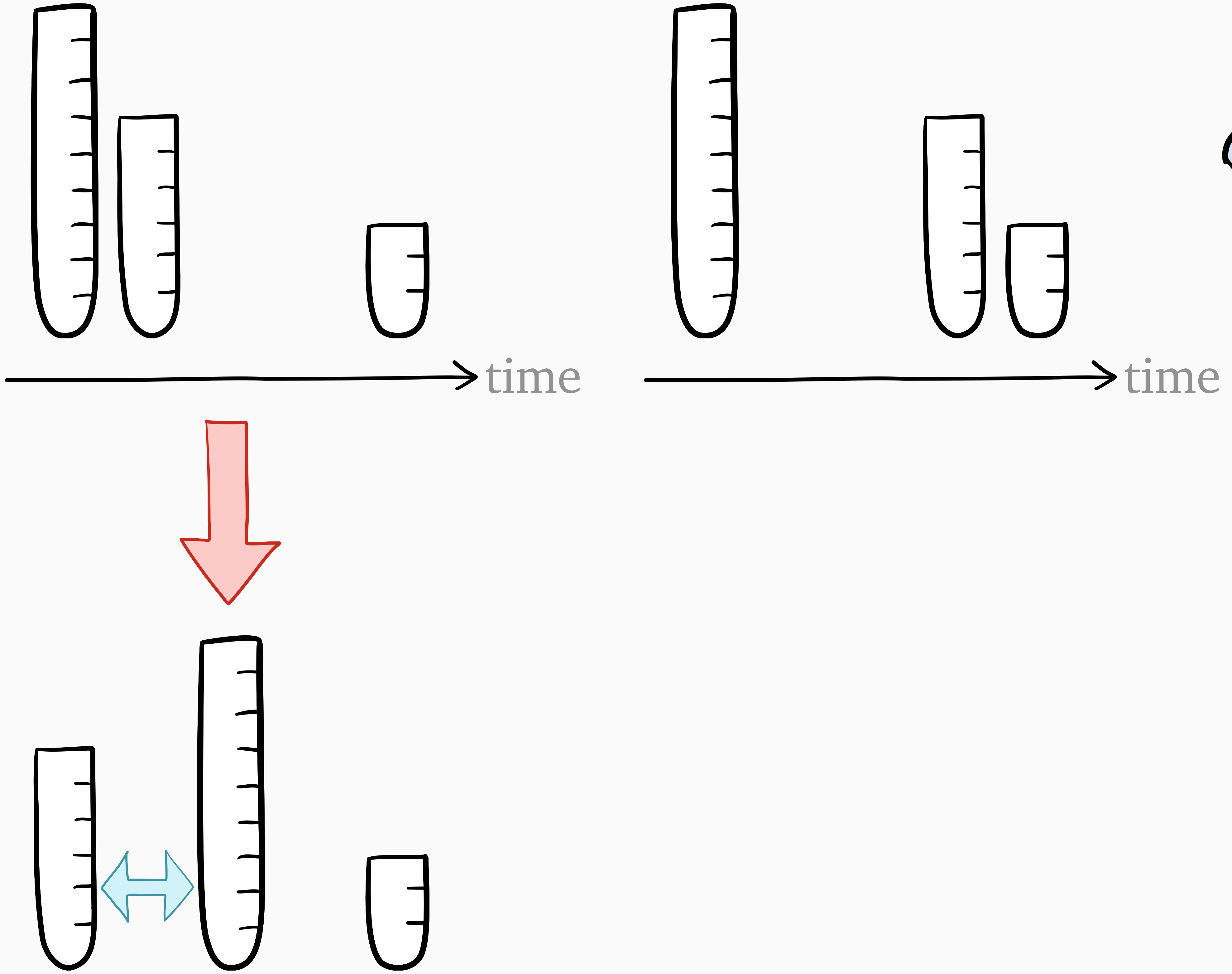
How to handle range of sizes?

Key information: *arrival times*



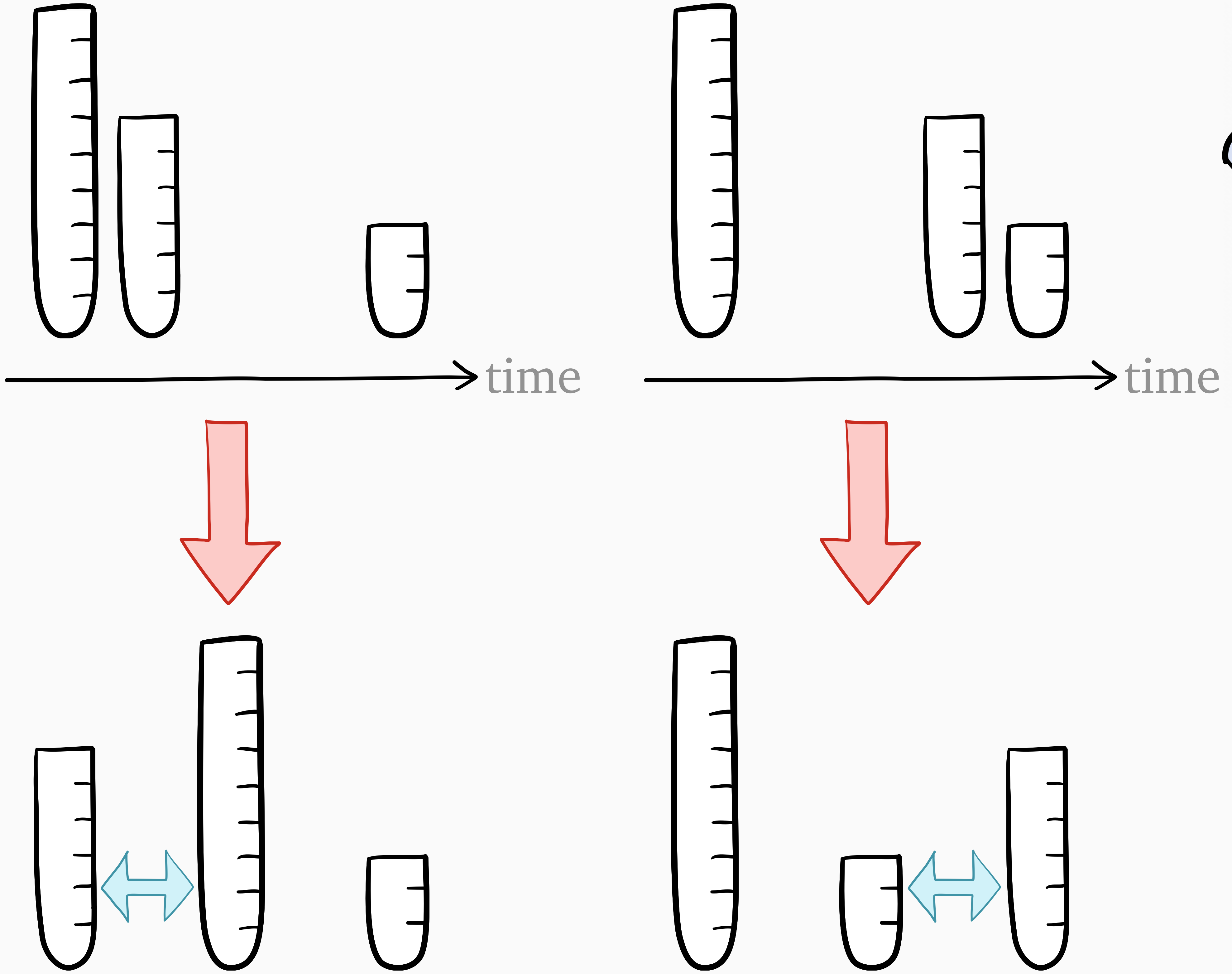
How to handle
range of sizes?

Key information: *arrival times*



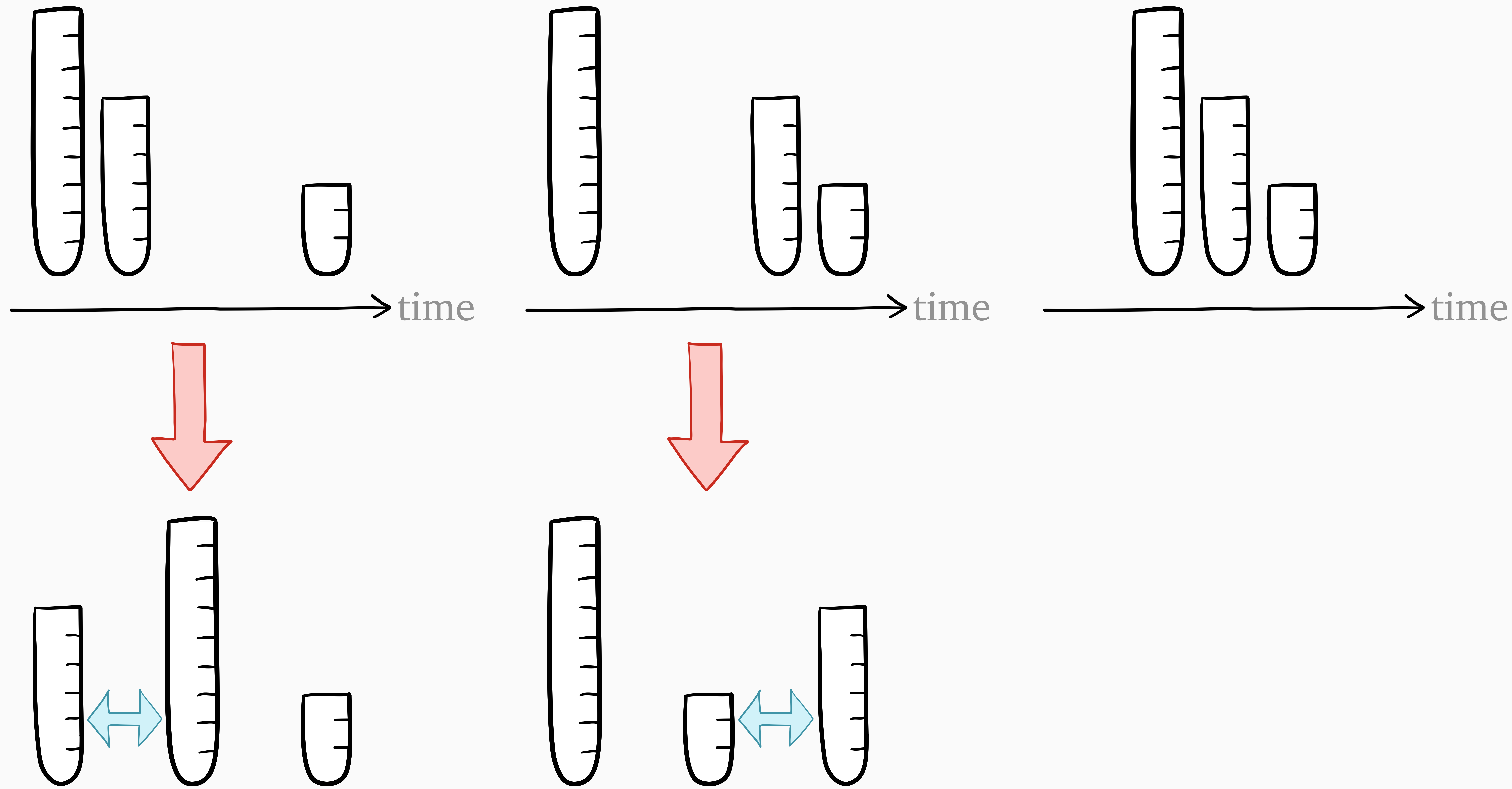
How to handle range of sizes?

Key information: *arrival times*

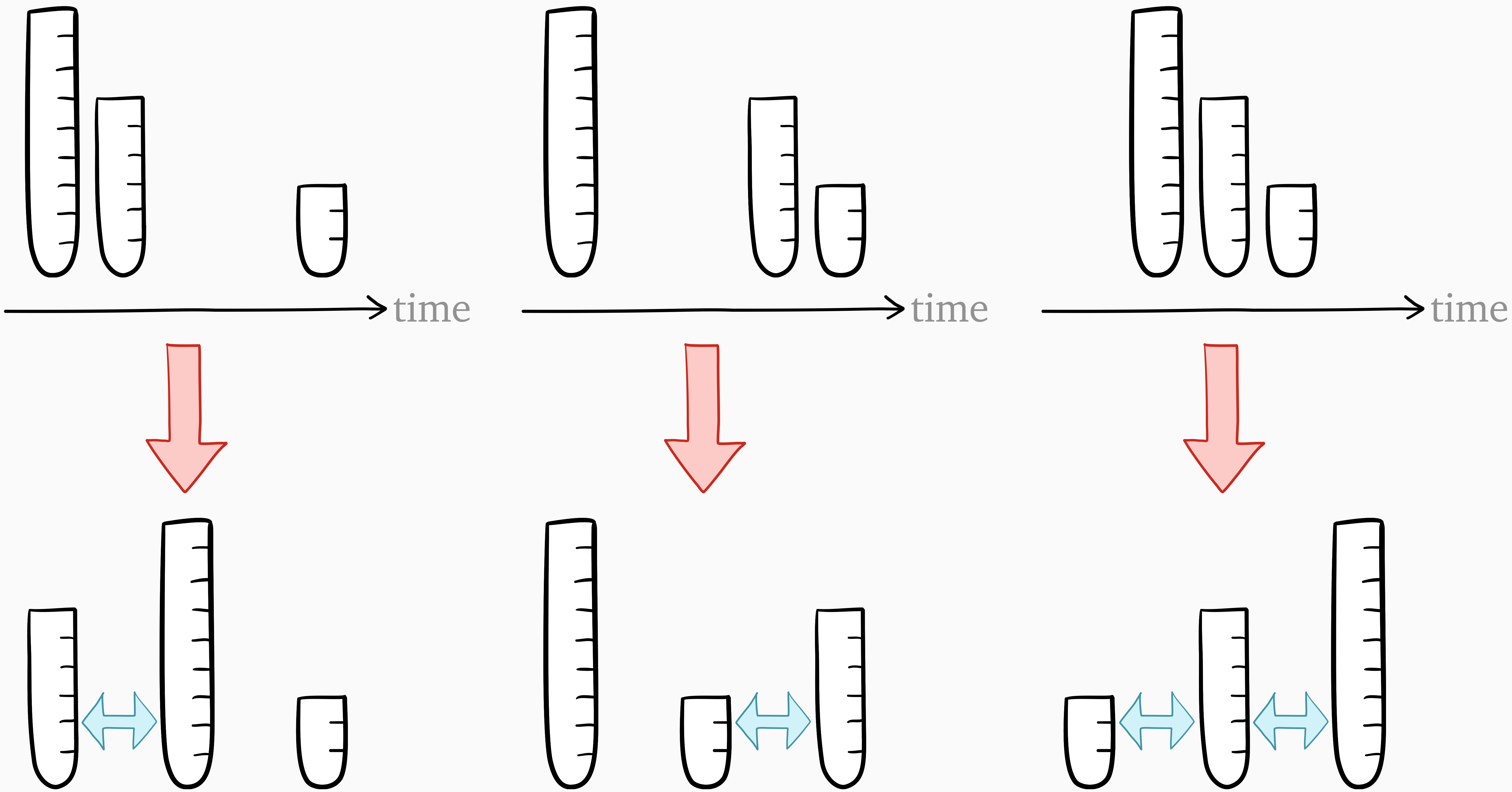


How to handle range of sizes?

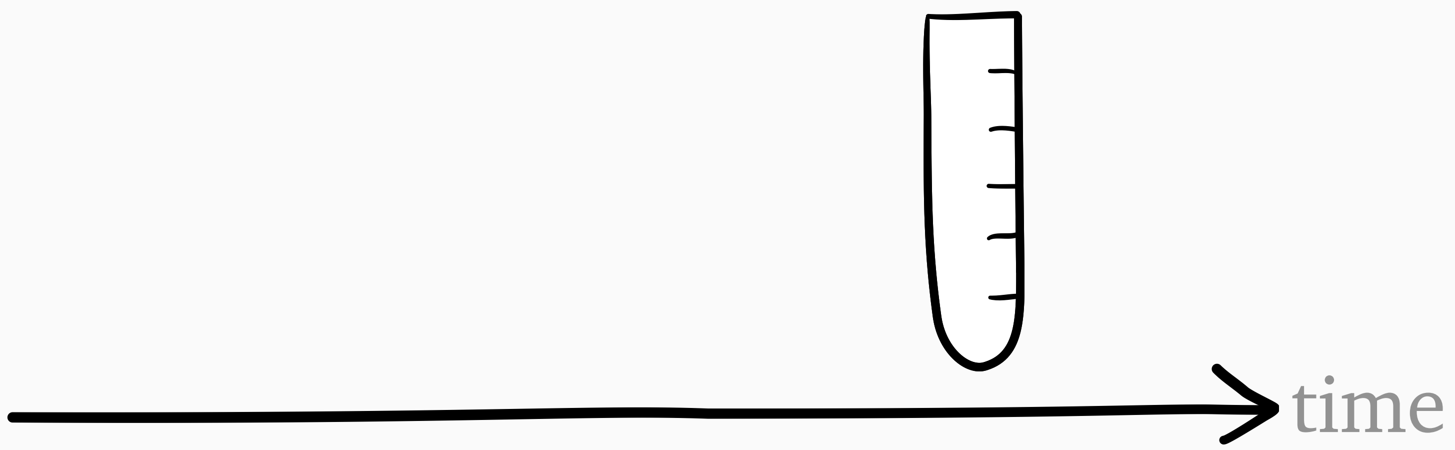
Key information: *arrival times*



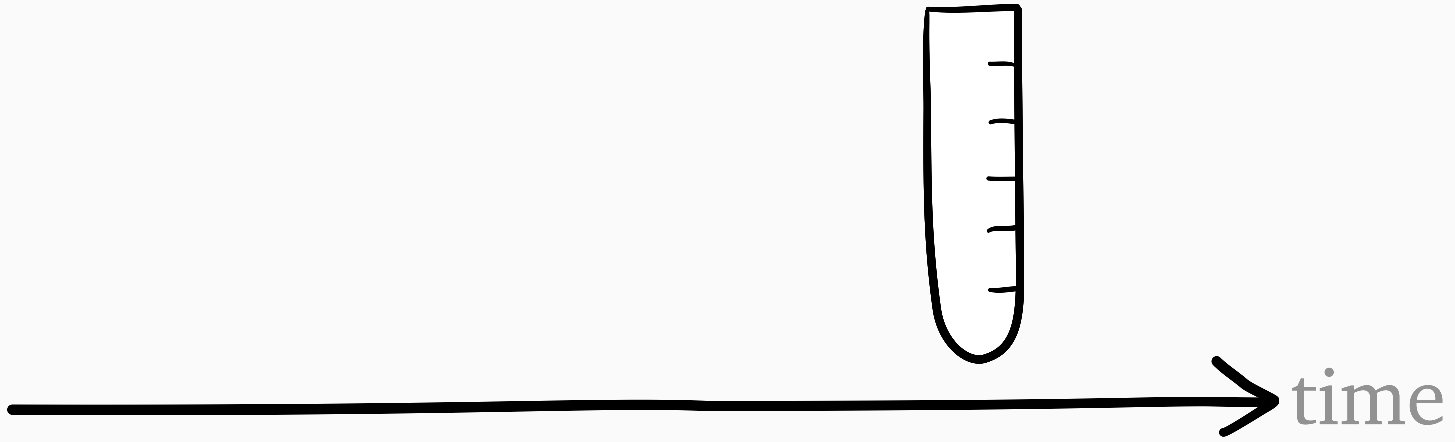
Key information: *arrival times*



Combining arrival time and size

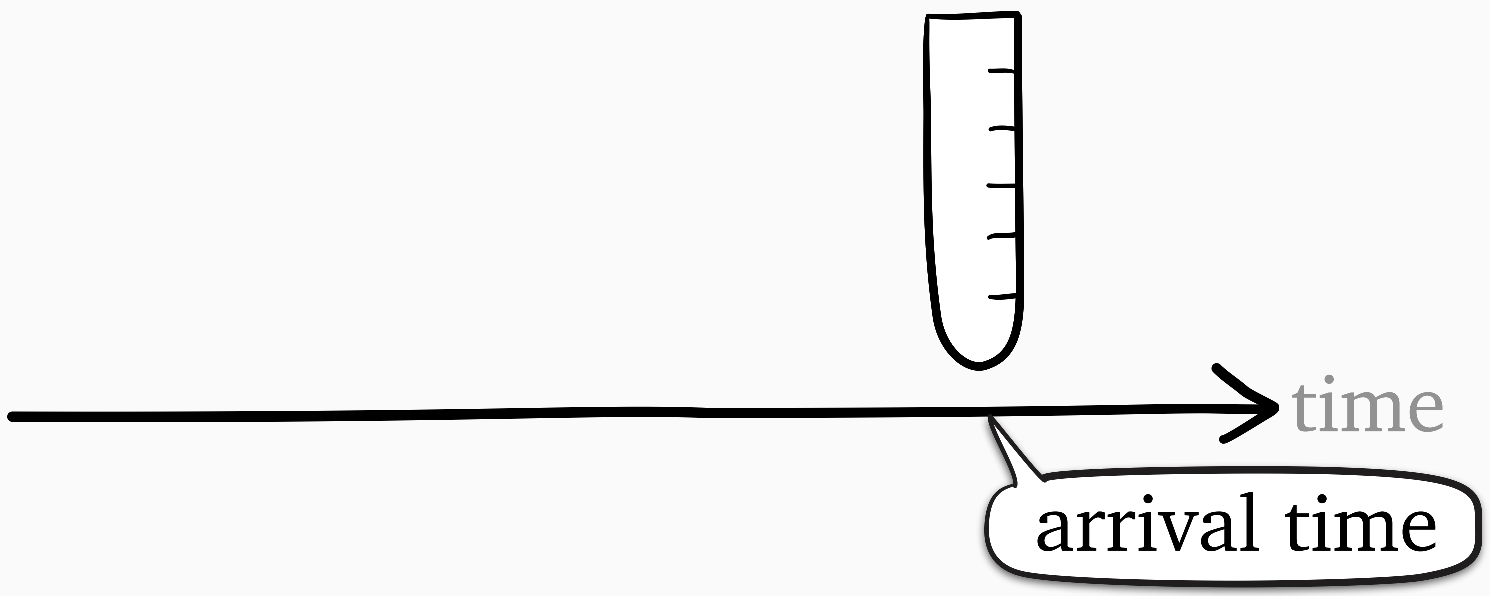


Combining arrival time and size



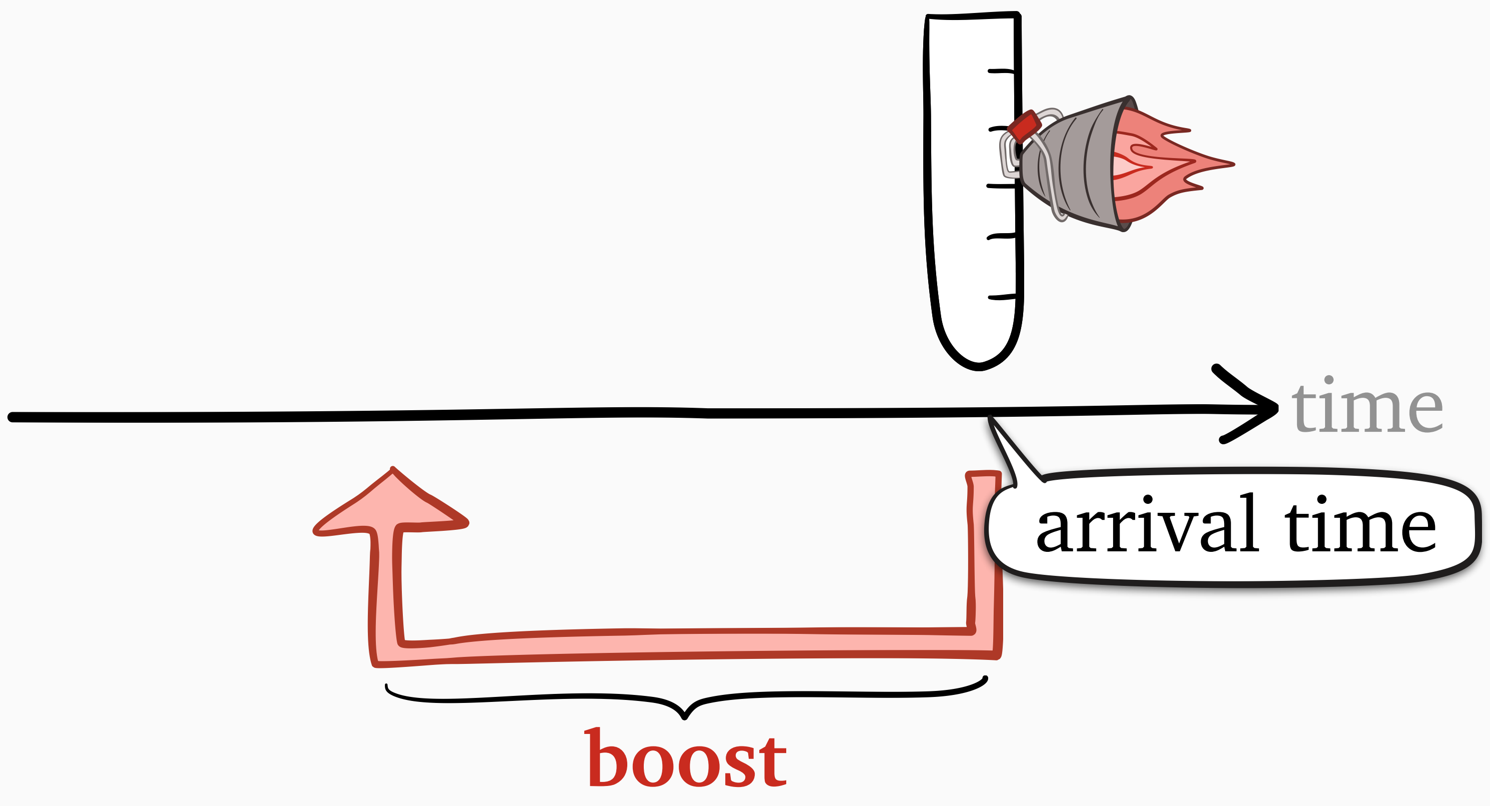
boosted arrival time
= arrival time − **boost**(size)

Combining arrival time and size



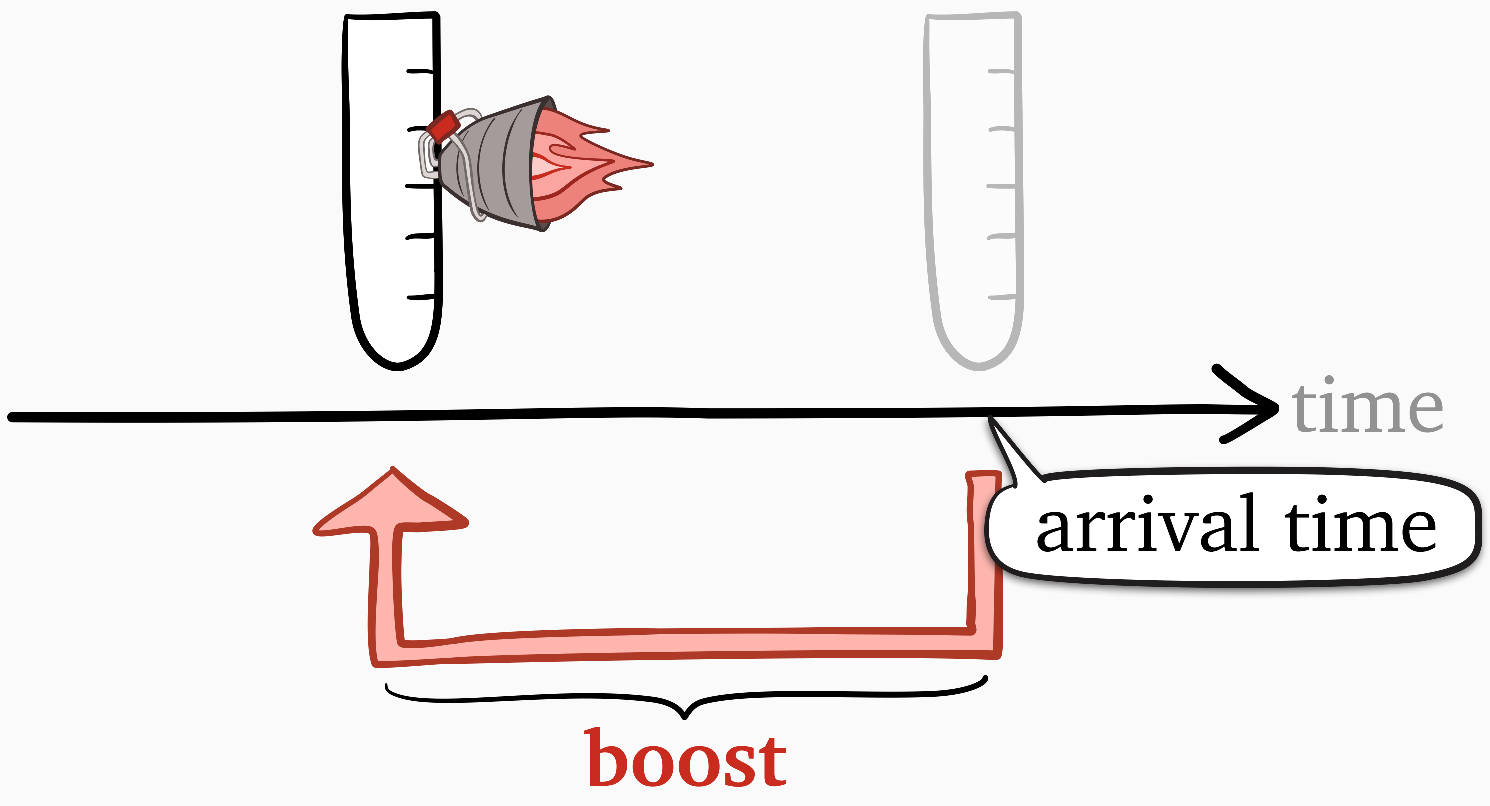
boosted arrival time
  = arrival time − **boost**(size)

Combining arrival time and size



boosted arrival time
= arrival time − **boost**(size)

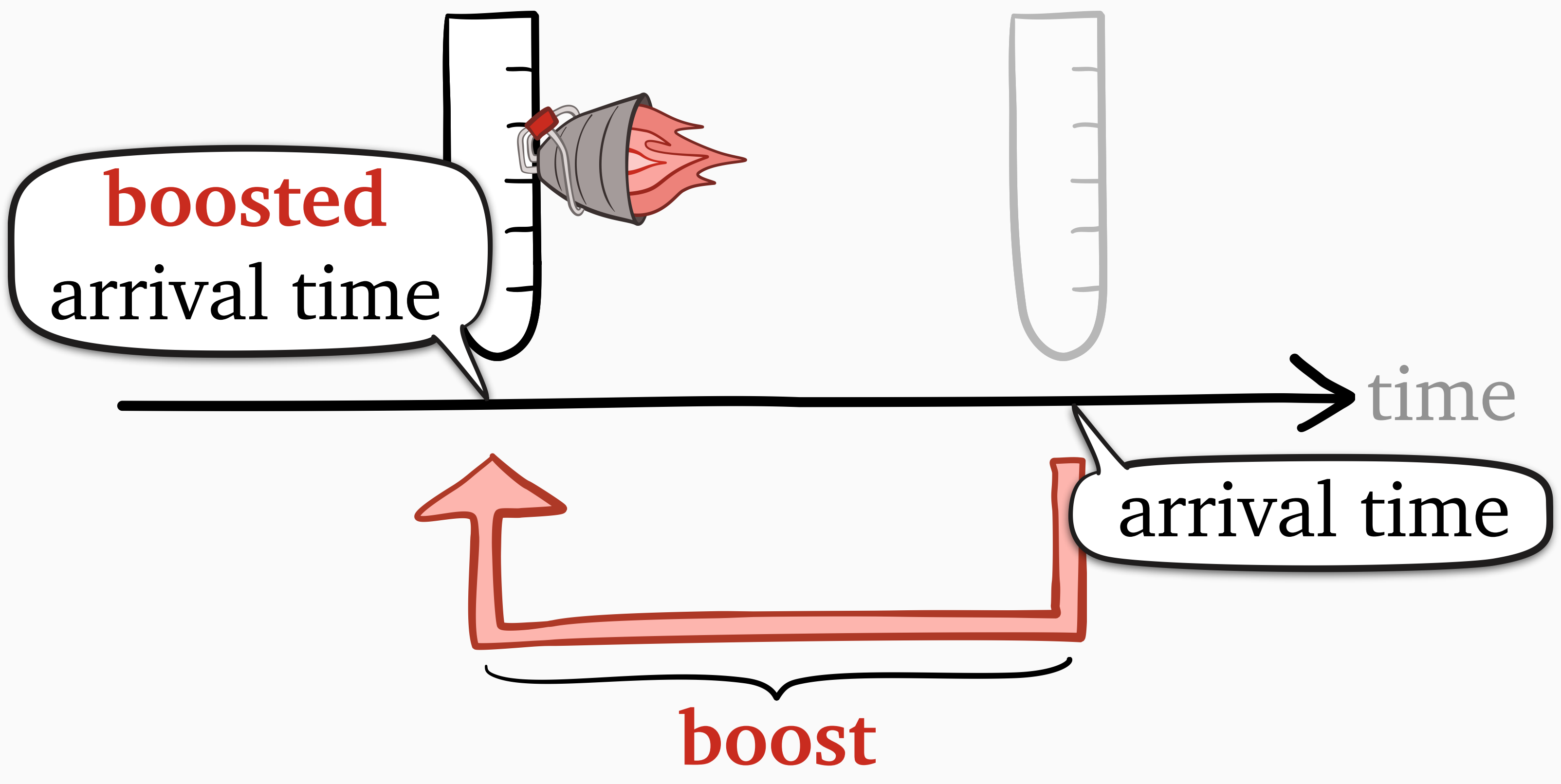
Combining arrival time and size



boosted arrival time

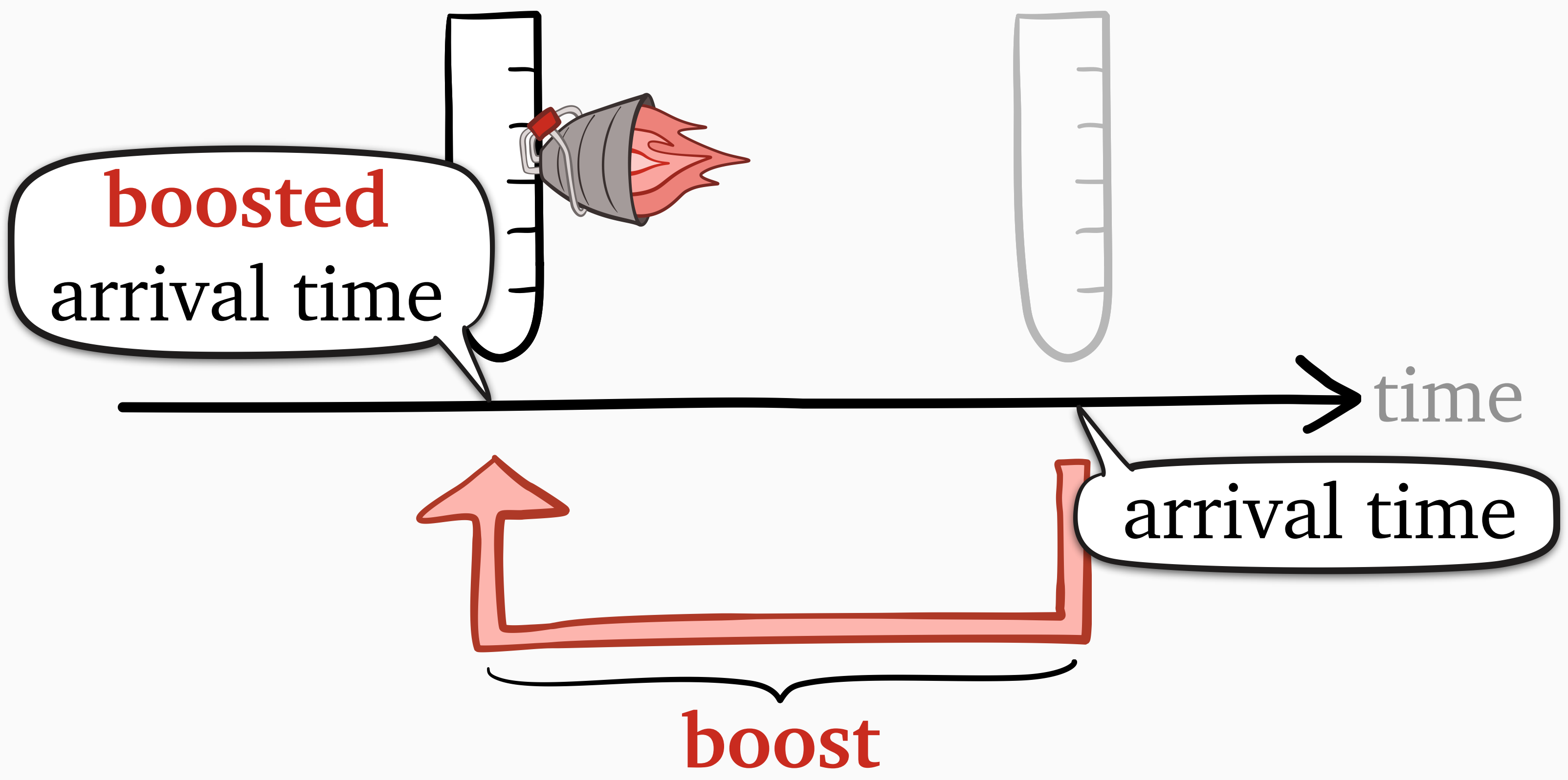
= arrival time − **boost**(size)

Combining arrival time and size



boosted arrival time
= arrival time − **boost**(size)

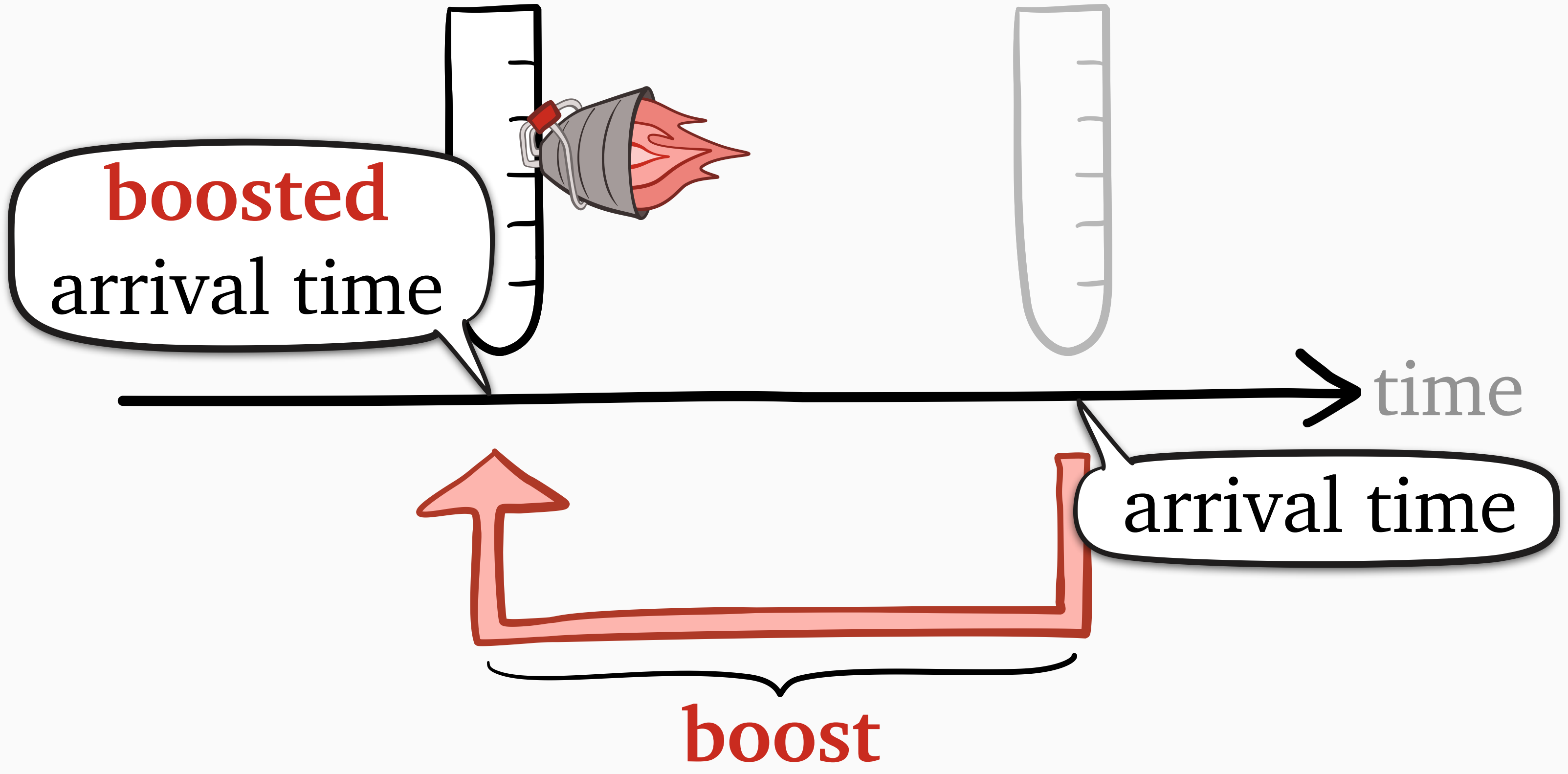
Combining arrival time and size

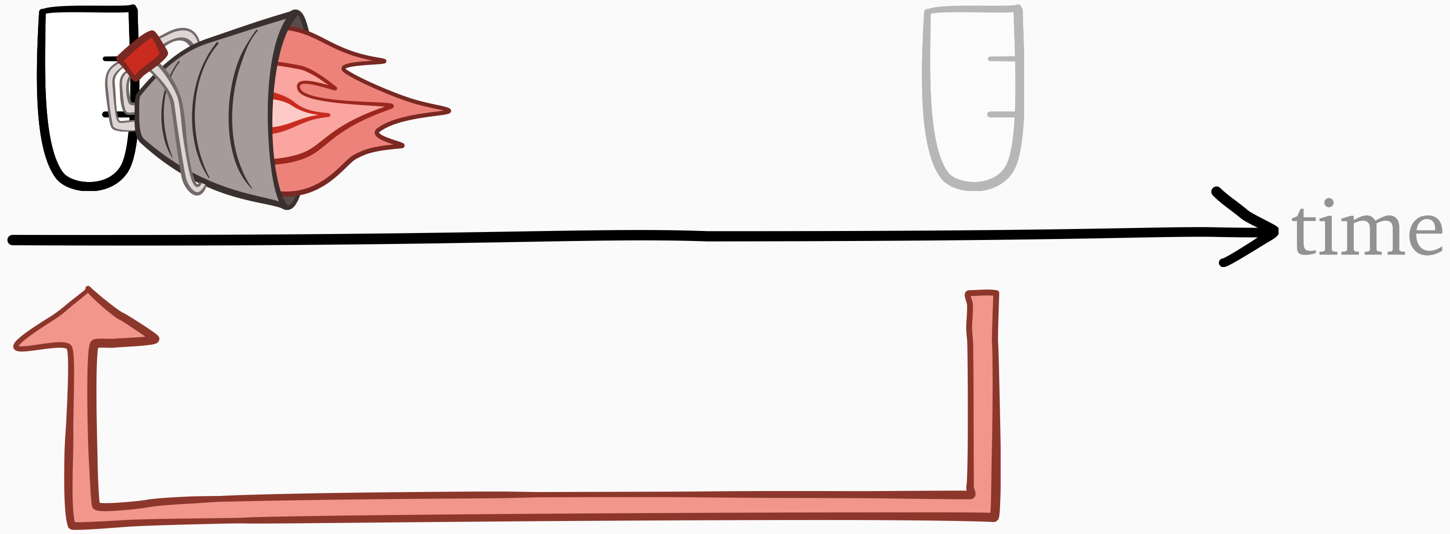


boosted arrival time
= arrival time − **boost**(size)

smaller sizes get bigger boosts

Combining arrival time and size

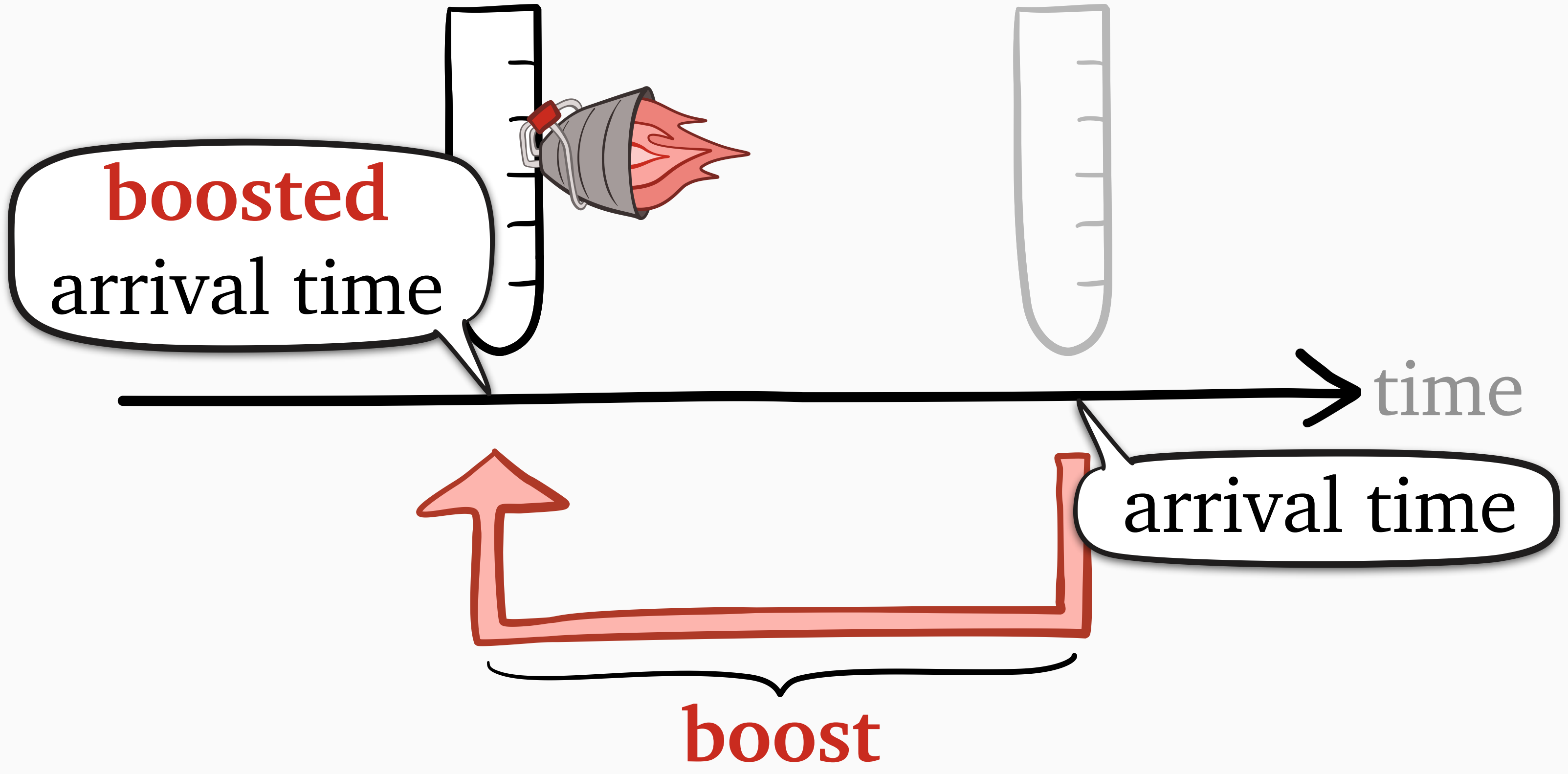




boosted arrival time
= arrival time − **boost**(size)

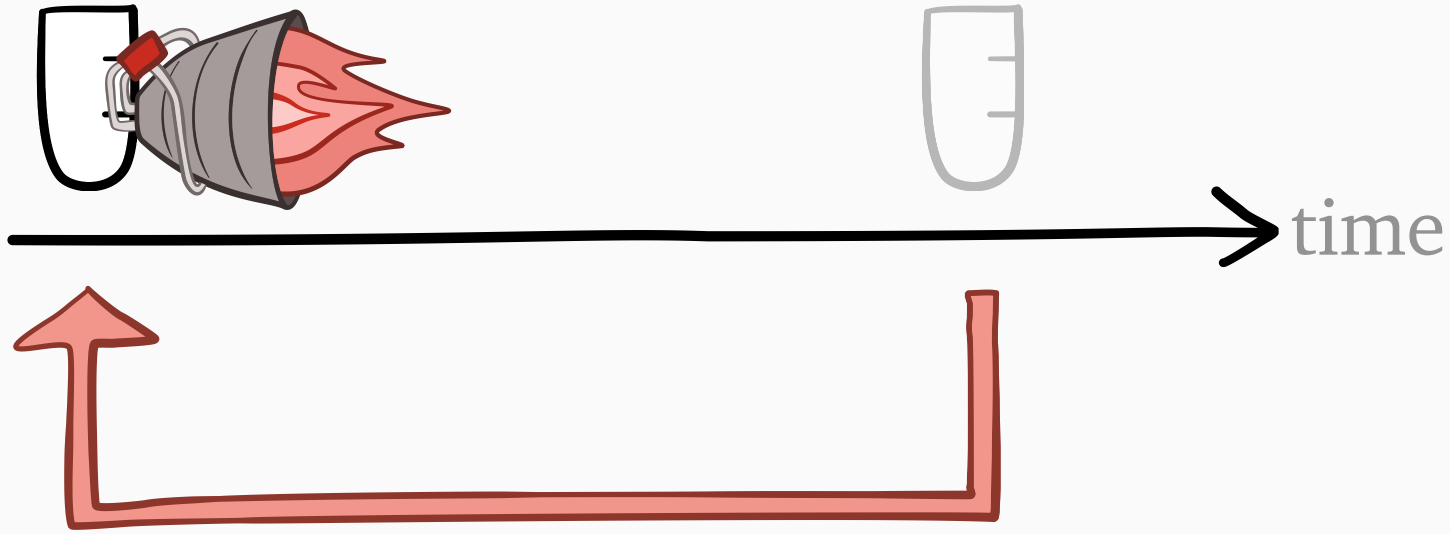
smaller sizes get bigger boosts

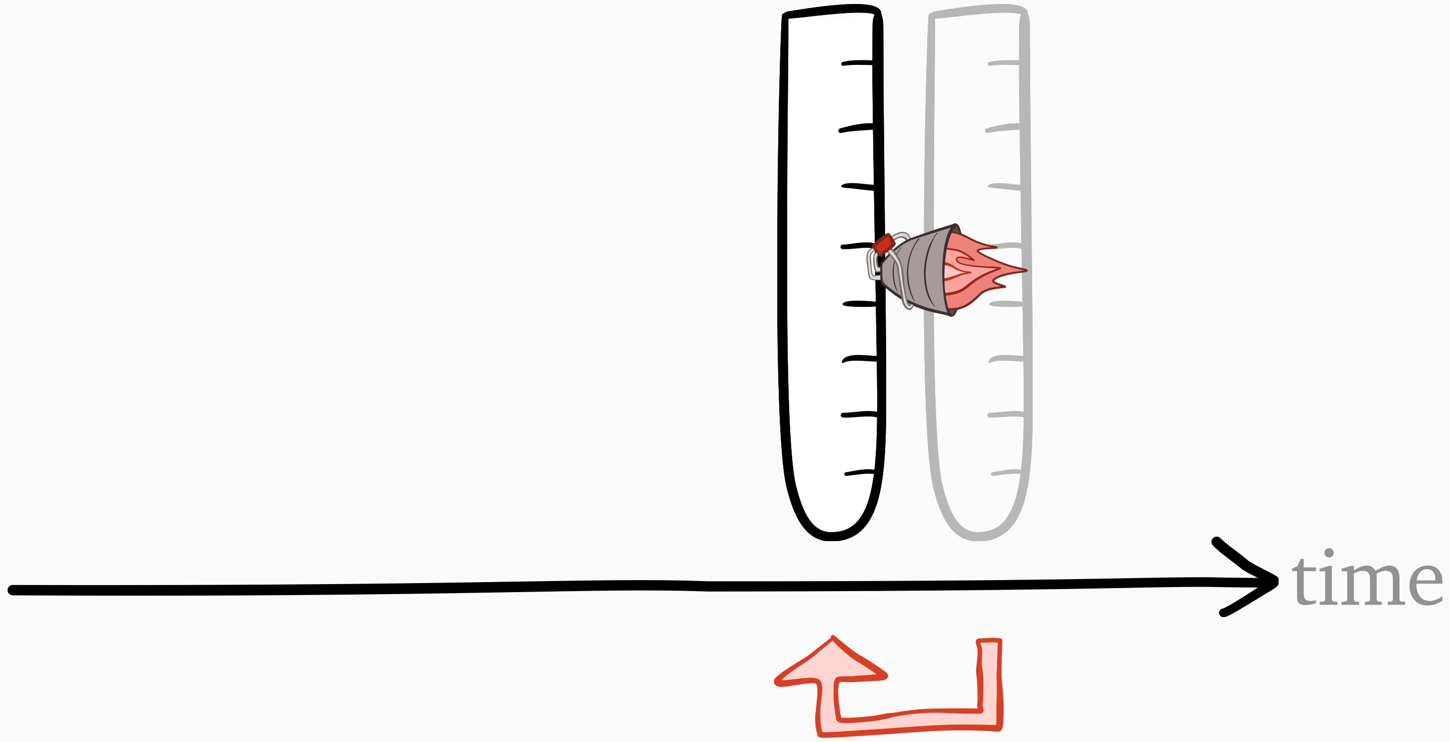
Combining arrival time and size



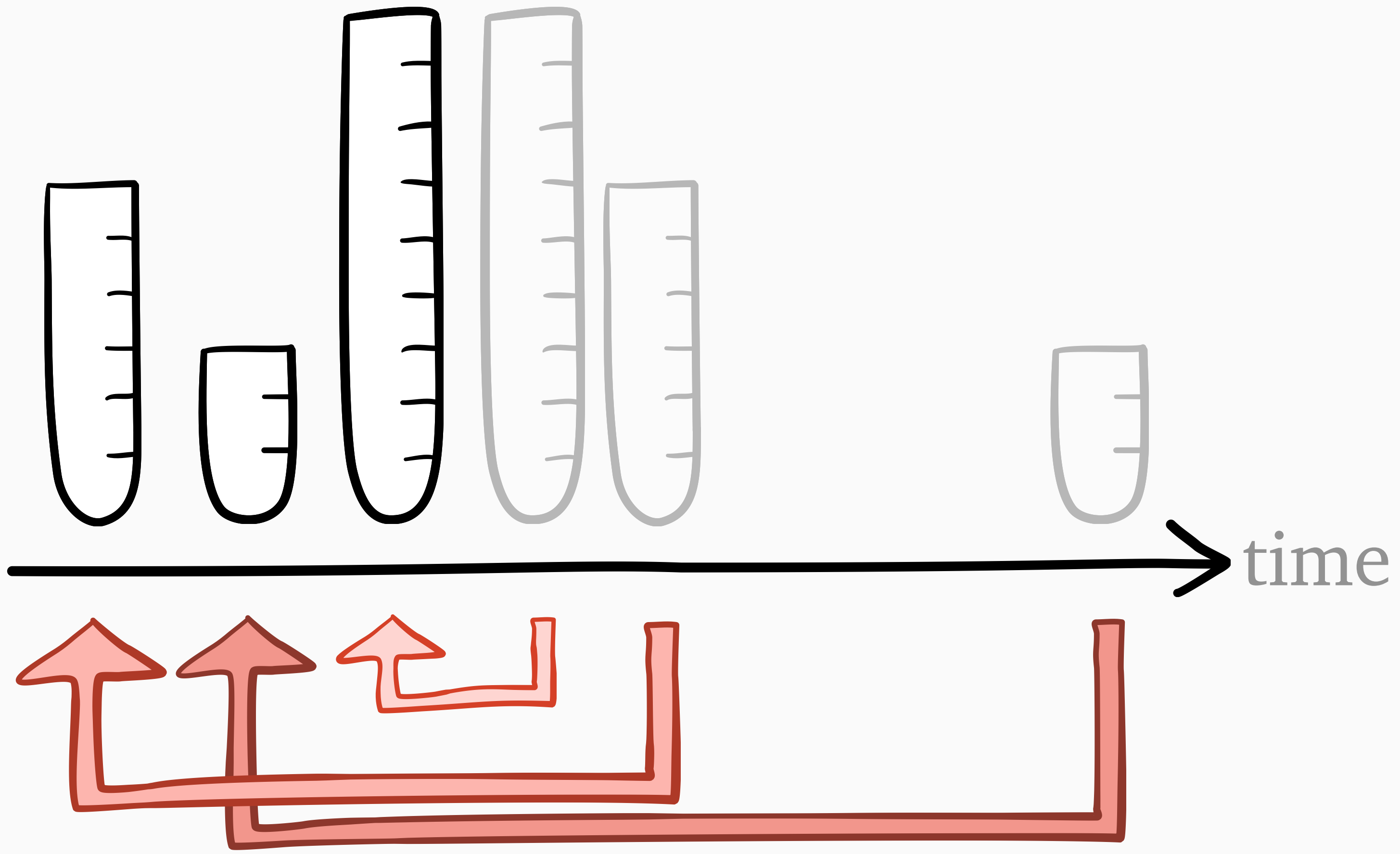
boosted arrival time
= arrival time − **boost**(size)

smaller sizes get bigger boosts



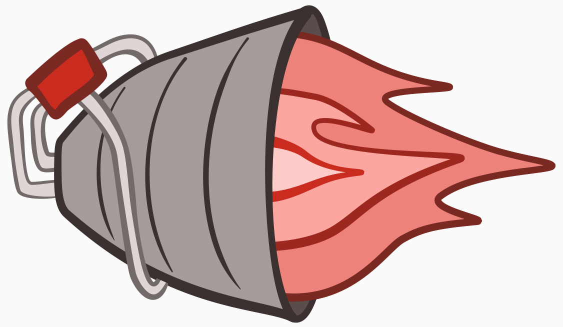


Boost policies

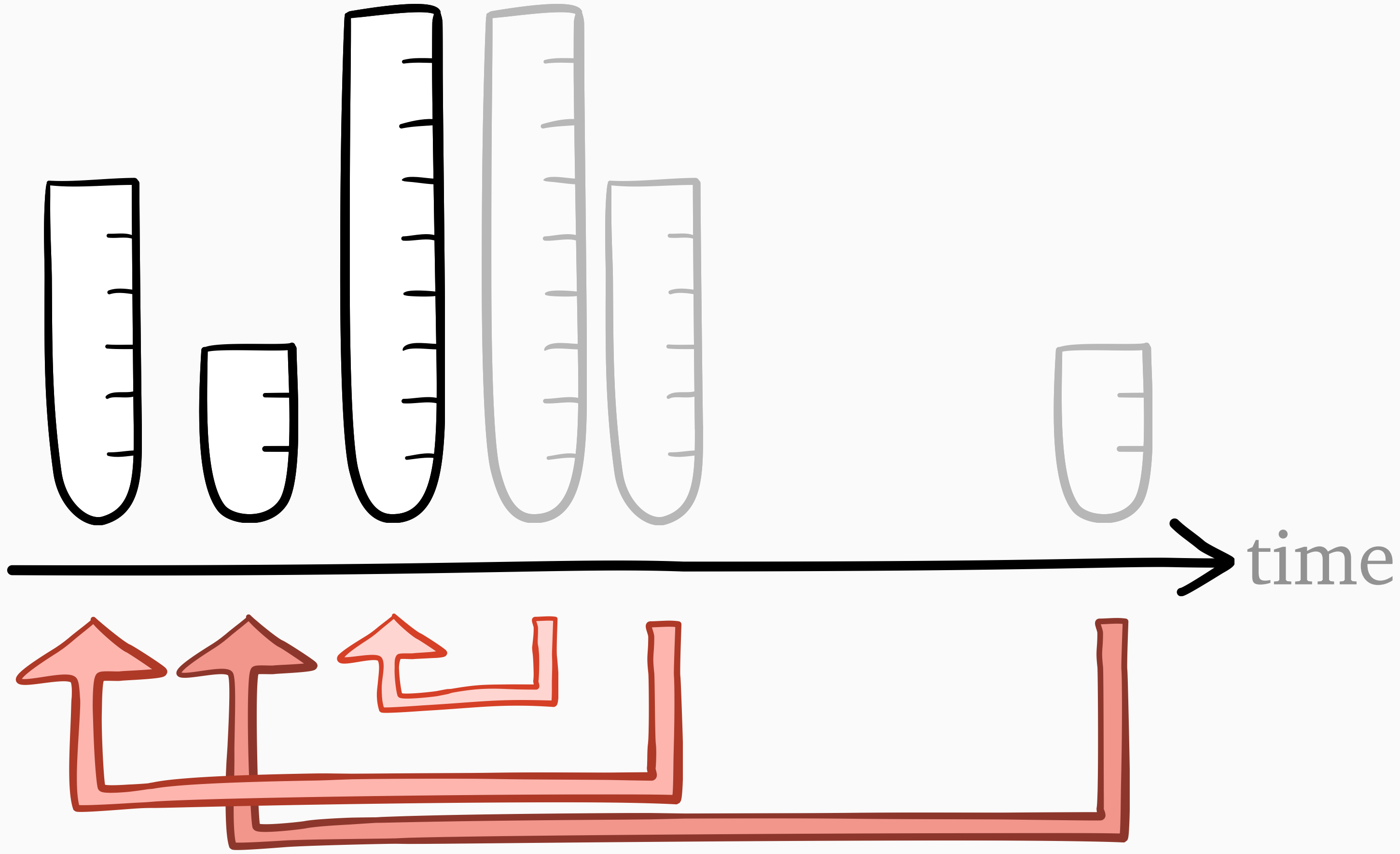


boosted arrival time
= arrival time − **boost**(size)

Boost policies

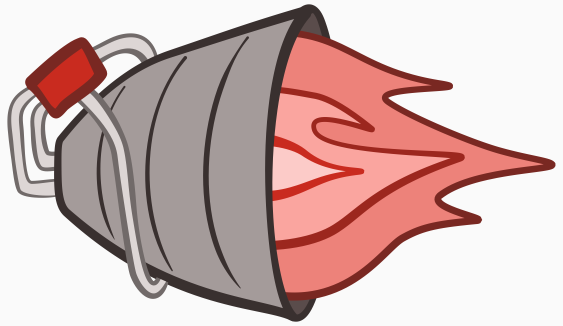


Scheduling rule: always serve job of *minimum* ***boosted*** *arrival time*

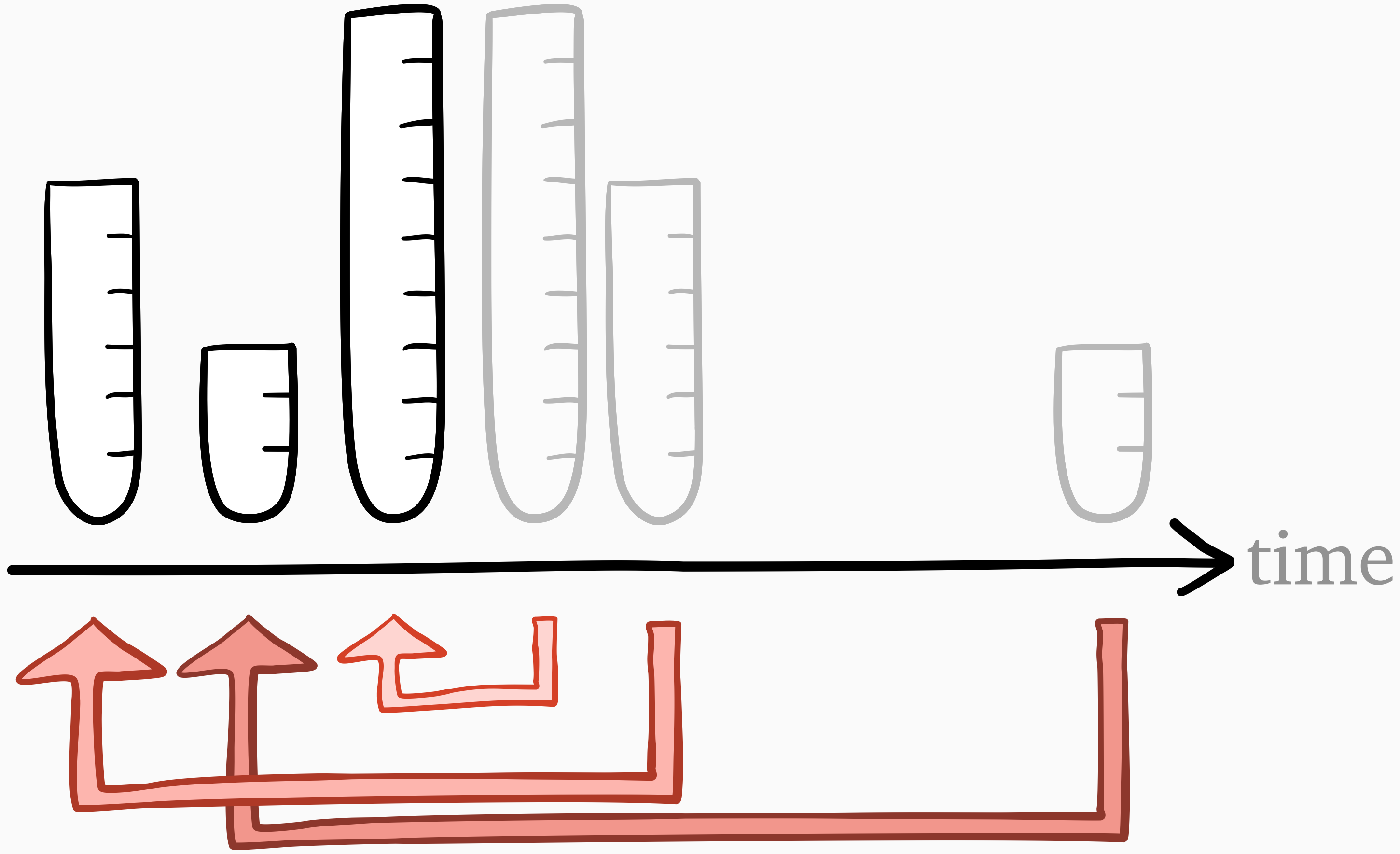


boosted arrival time
= arrival time − **boost**(size)

Boost policies



Scheduling rule: always serve job of
*minimum **boosted** arrival time*

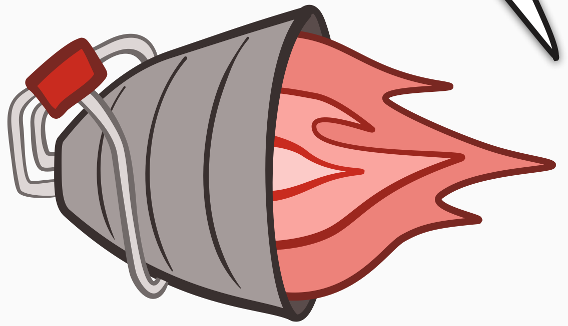


boosted arrival time
= arrival time − **boost**(size)

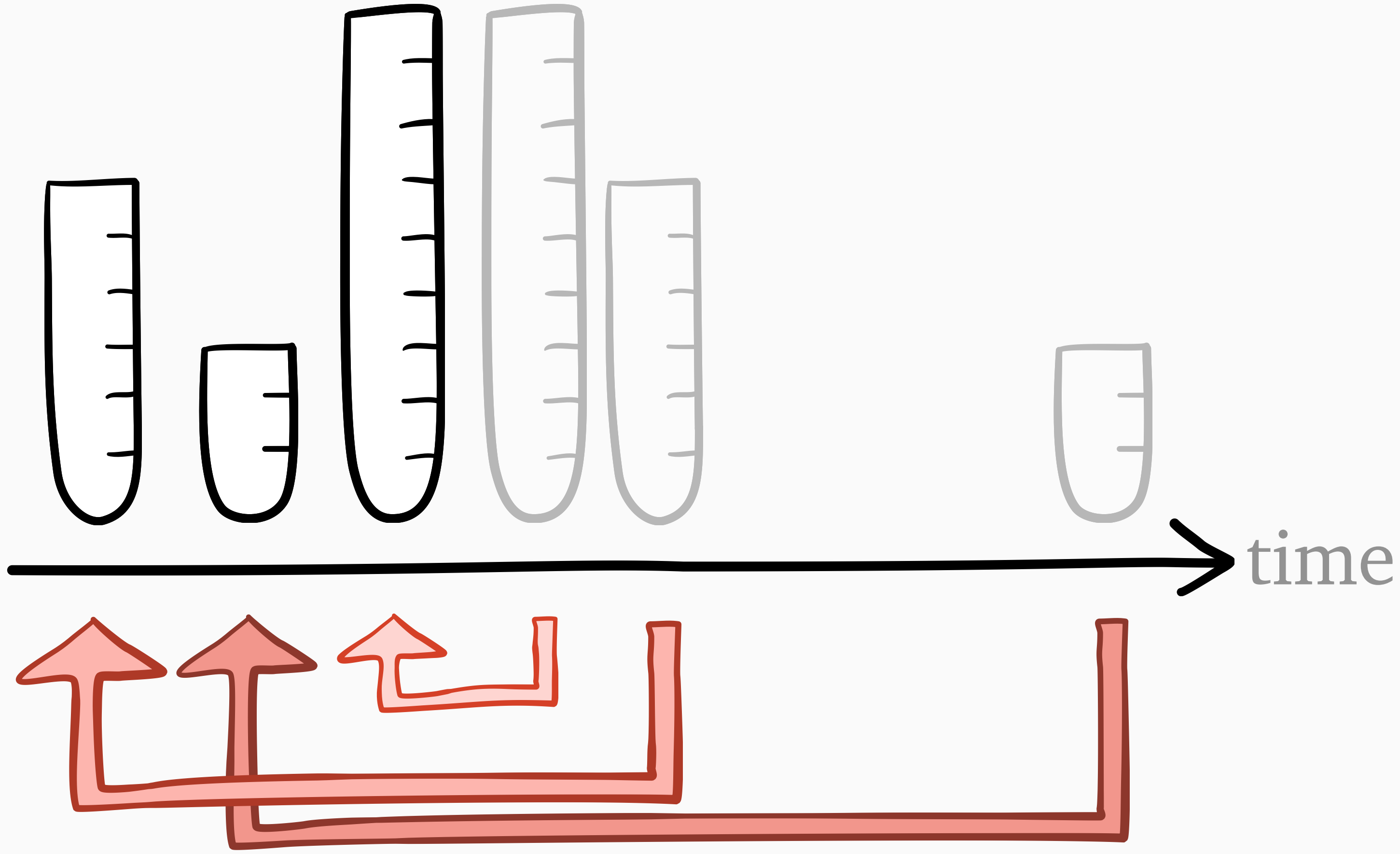
can vary choice of **boost** function

Boost policies

can be preemptive or nonpreemptive



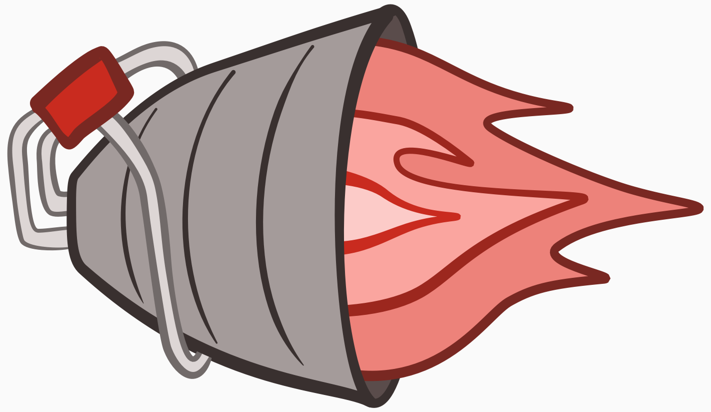
Scheduling rule: always serve job of *minimum **boosted** arrival time*



boosted arrival time
= arrival time − **boost**(size)

can vary choice of **boost** function

Boost

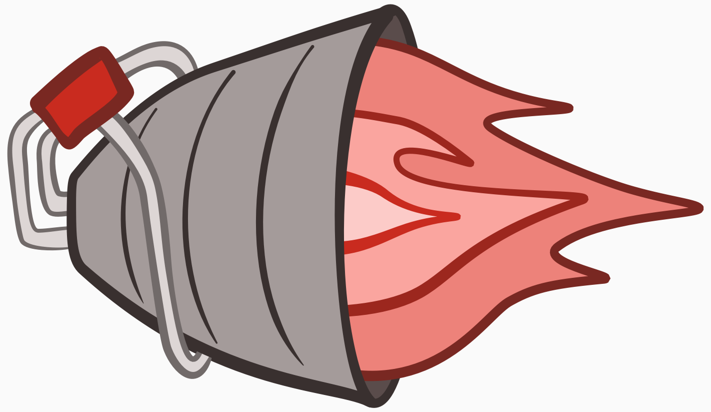


✓ Why is achieving strong tail optimality hard?

? How does the **Boost** policy family work?

? How do we achieve strong tail optimality?

Boost



- ✓ Why is achieving strong tail optimality hard?
- ✓ How does the **Boost** policy family work?
- ? How do we achieve strong tail optimality?

Boost

Why is achieving strong tail optimality hard?

How does the **Boost** policy family work?

How do we achieve strong tail optimality?

Why is **Boost** *weakly* tail optimal?

Why is **Boost** *weakly* tail optimal?

$\mathbf{P}[T > t] \sim Ce^{-\gamma t}$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t}$$

$$C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t]$$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Why is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \Rightarrow C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$\lim_{t \to \infty} e^{\gamma t} \mathbf{P}[W > t]$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\mathrm{FCFS}} = W + S$$

work

$$C_{\mathrm{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$$\lim_{t \to \infty} e^{\gamma t} \mathbf{P}[W > t]$$

Boost

$$T_{\mathrm{Boost}} \approx W + S - b(S) + V$$

Why is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

FCFS

$$T_{\text{FCFS}} = W + S$$

work

$$C_{\text{FCFS}} = C_W \mathbf{E}[e^{\gamma S}]$$

$$\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[W > t]$$

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

? finite?

Why is Boost *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t}$$

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

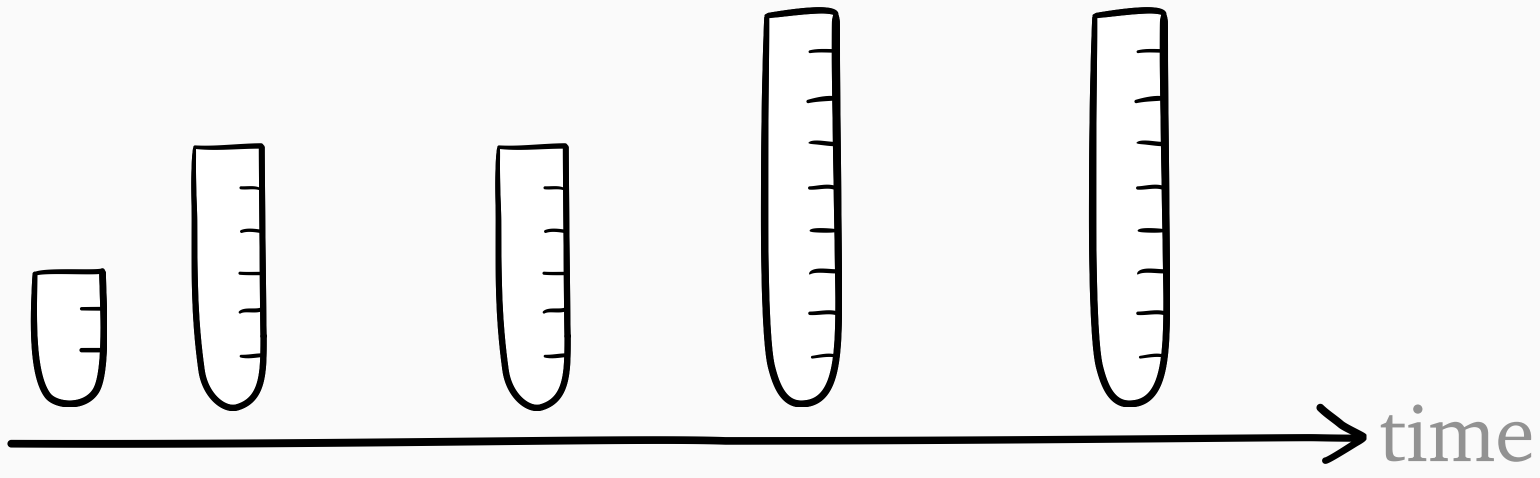
$$T_{\text{Boost}} \approx W + S - b(S) + V$$

crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\mathbf{E}[e^{\gamma V}]$$

finite?

V = **crossing work**



Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t}$$

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

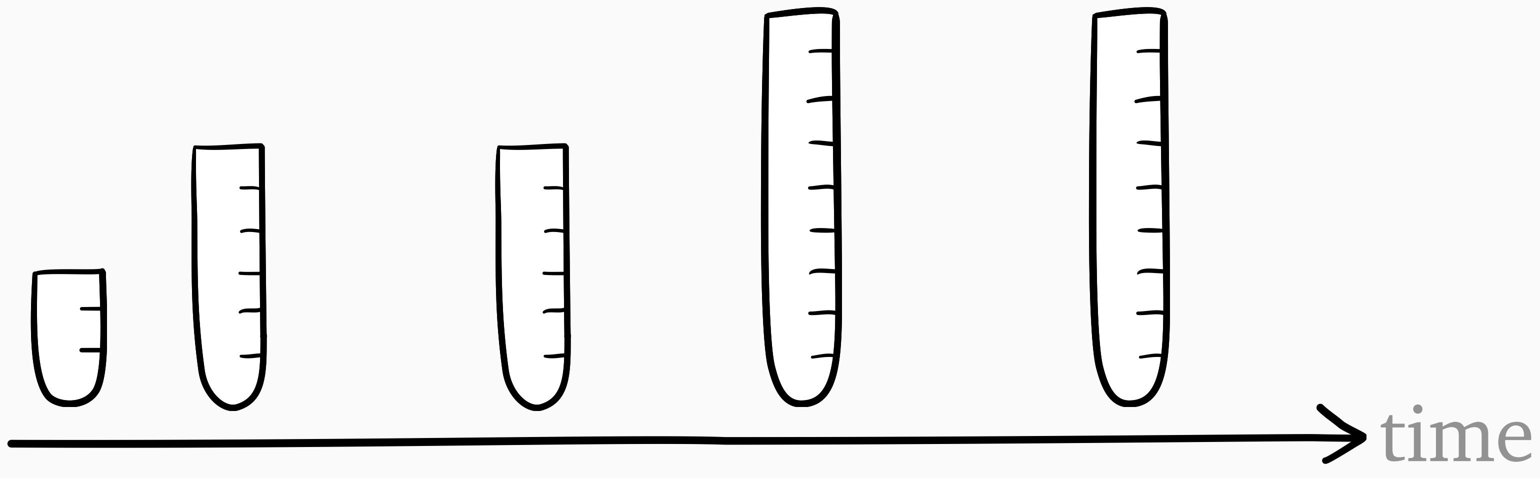
crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\mathbf{E}[e^{\gamma V}]$$

? finite?

V = crossing work

work that “boosts past” a given time



Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

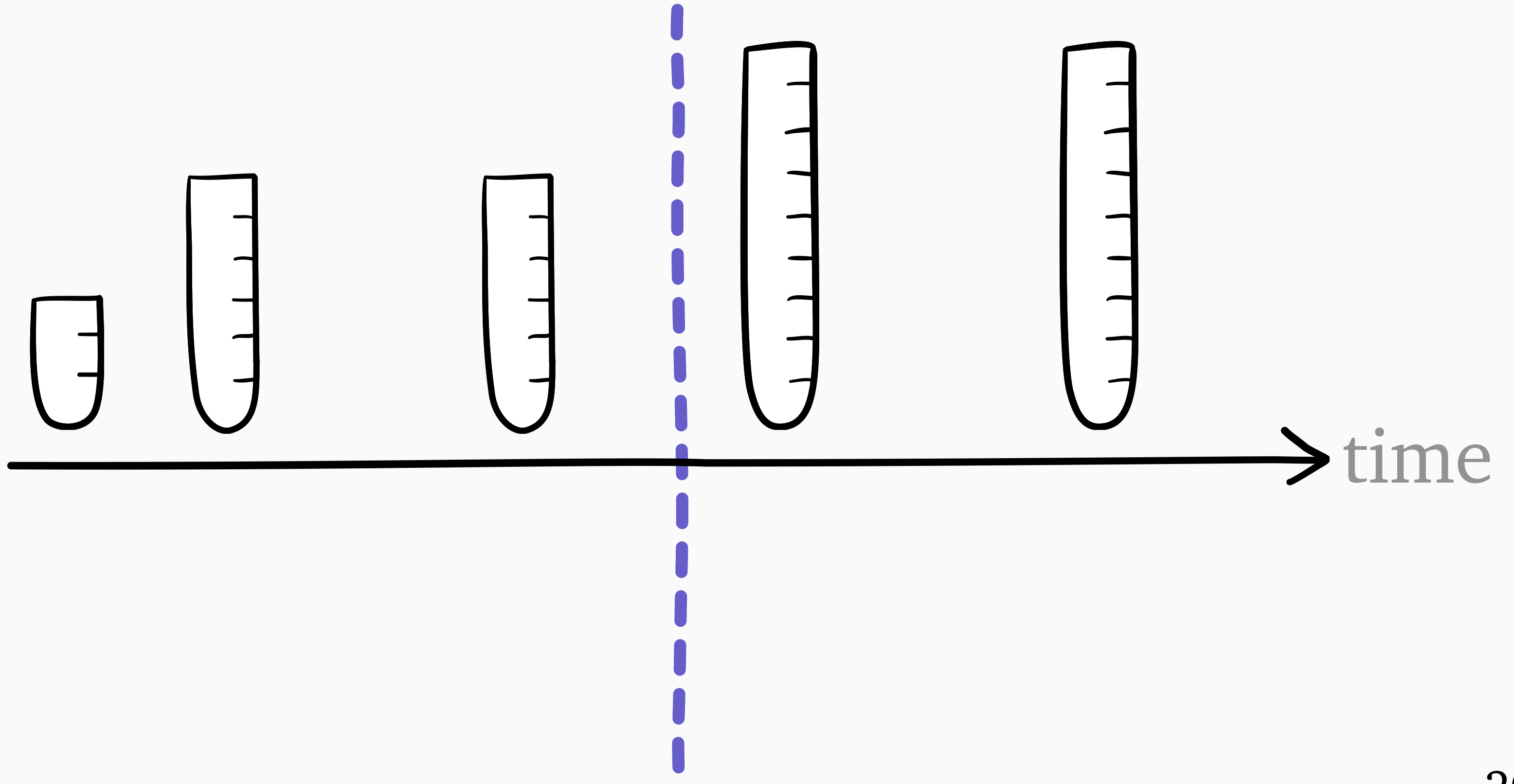
crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

finite?

V = **crossing work**

work that “boosts past” a given time



Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

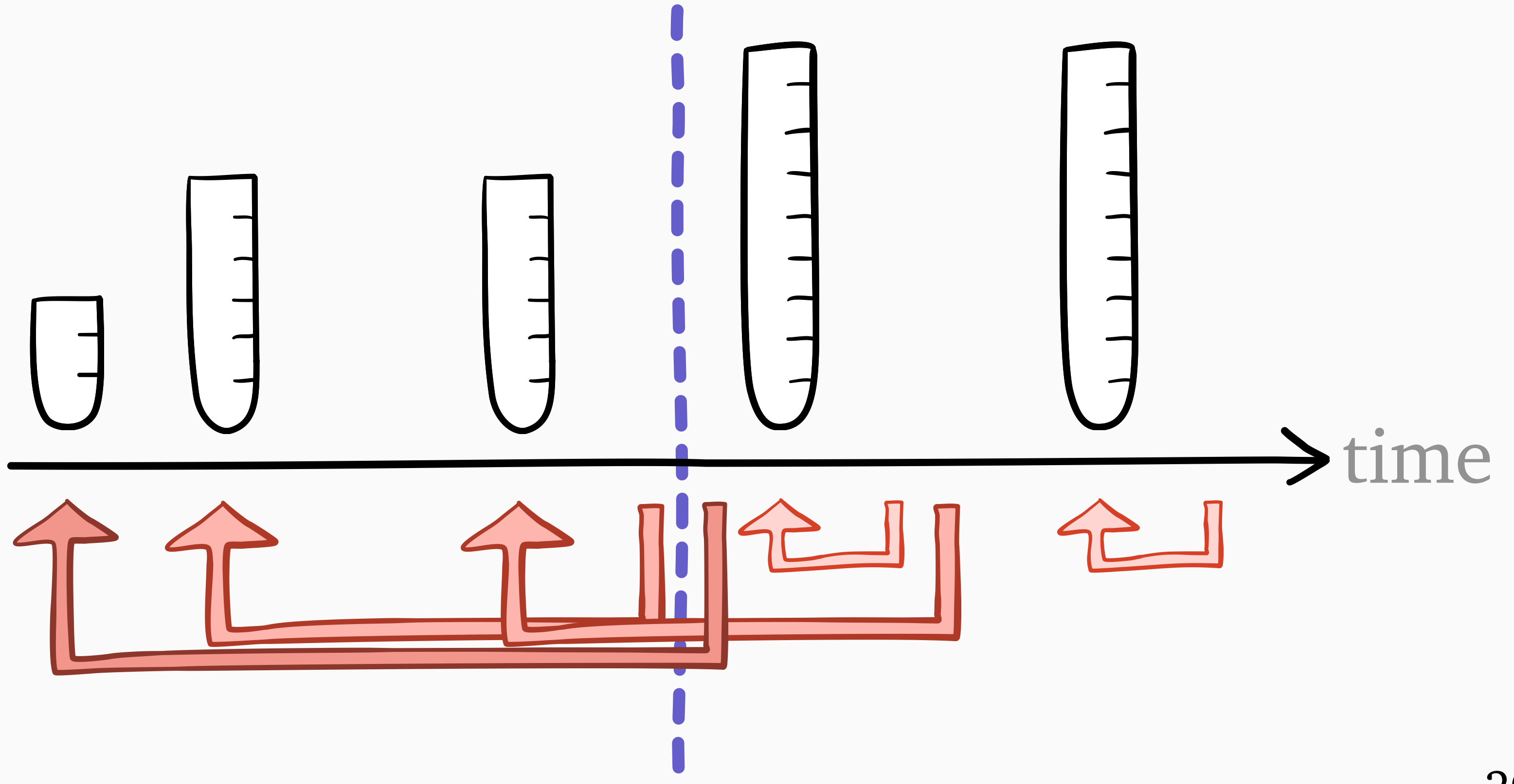
crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\mathbf{E}[e^{\gamma V}]$$

finite?

V = **crossing work**

work that “boosts past” a given time



Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \quad \Rightarrow \quad C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

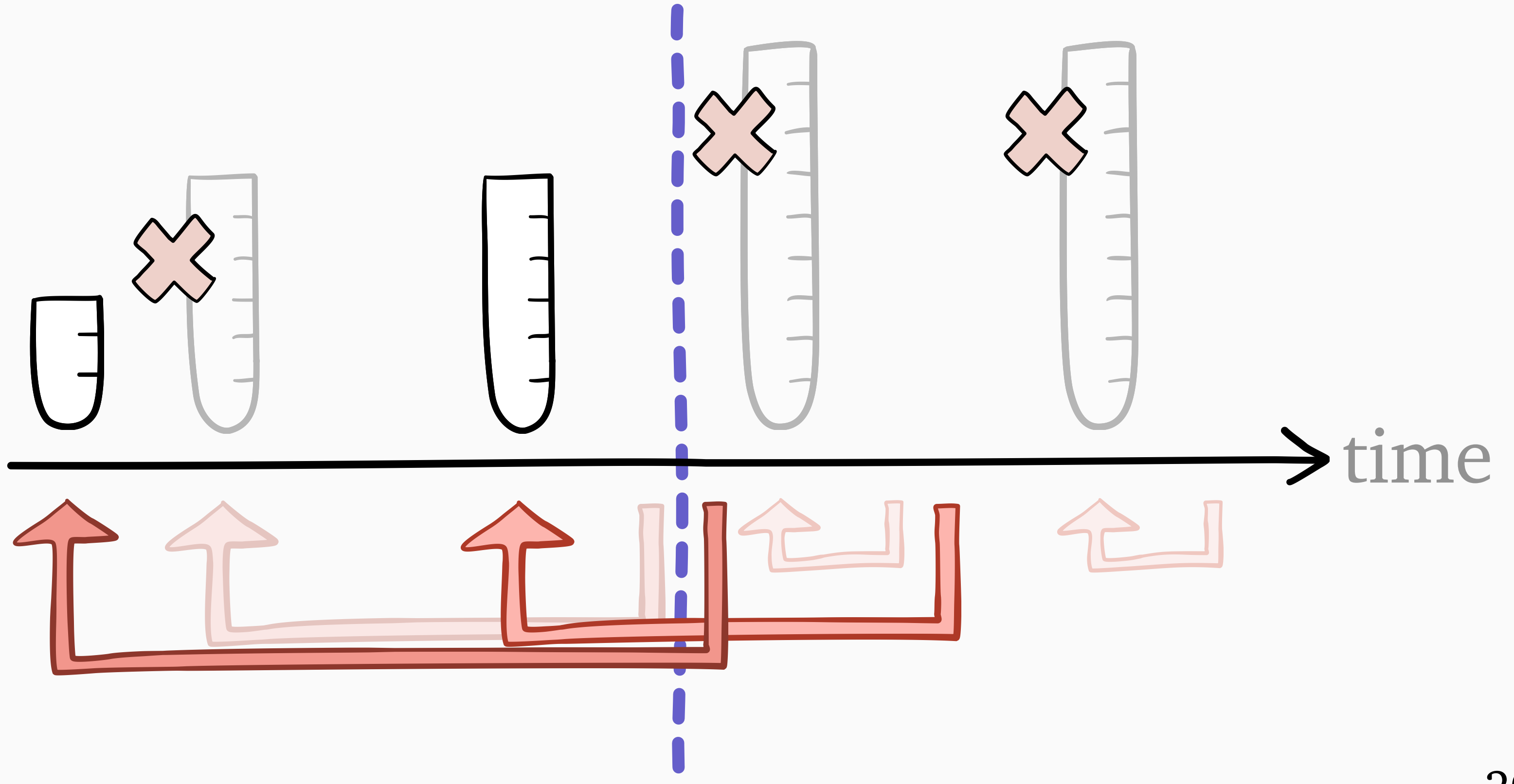
crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\mathbf{E}[e^{\gamma V}]$$

finite?

V = **crossing work**

work that "boosts past" a given time



Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim C e^{-\gamma t} \Rightarrow C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

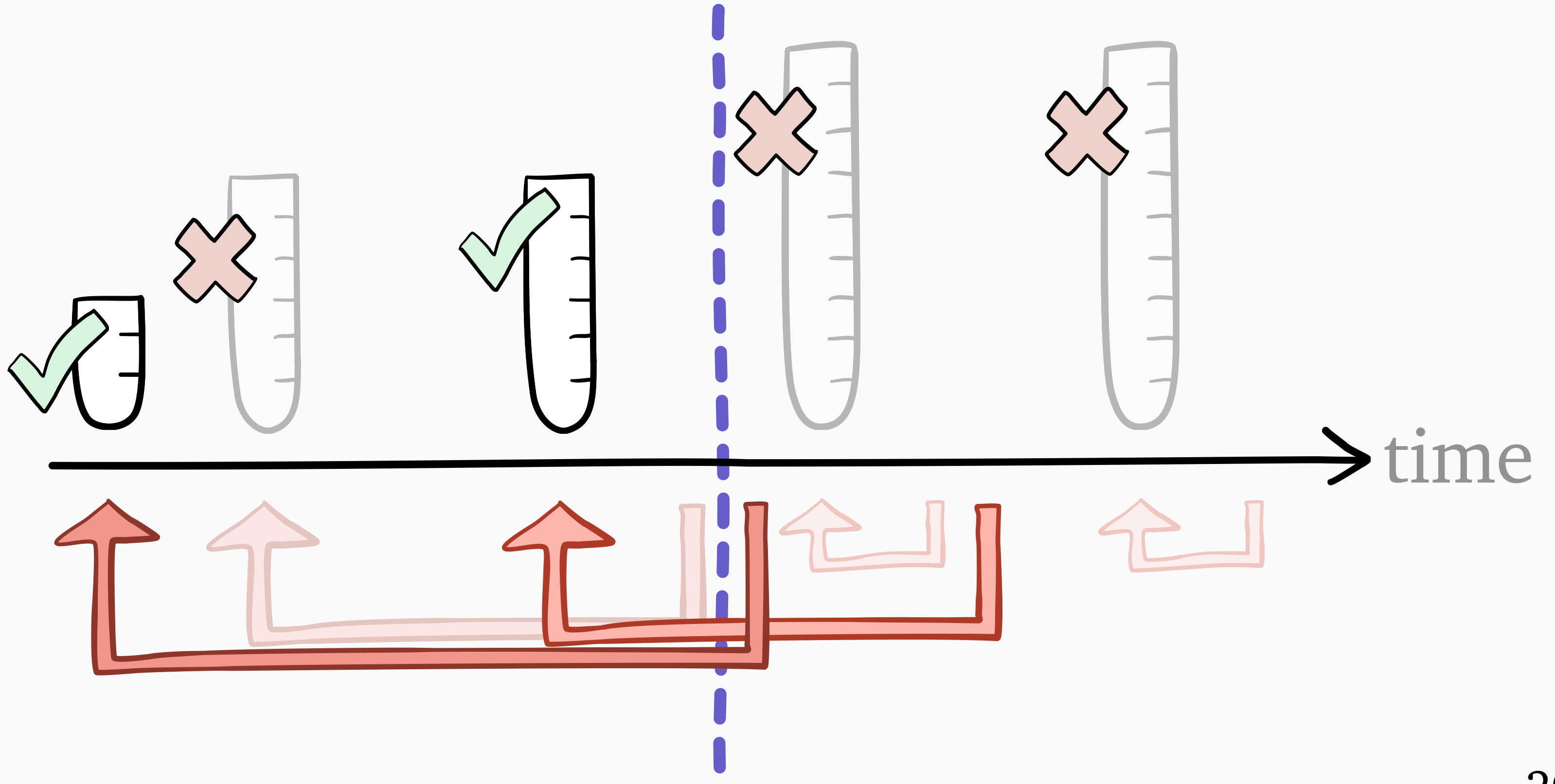
crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\,\mathbf{E}[e^{\gamma V}]$$

? finite?

V = **crossing work**

work that “boosts past” a given time



Why is **Boost** *weakly* tail optimal?

$$\mathbf{P}[T > t] \sim Ce^{-\gamma t} \Rightarrow C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

final value theorem

Boost

boost function

$$T_{\text{Boost}} \approx W + S - b(S) + V$$

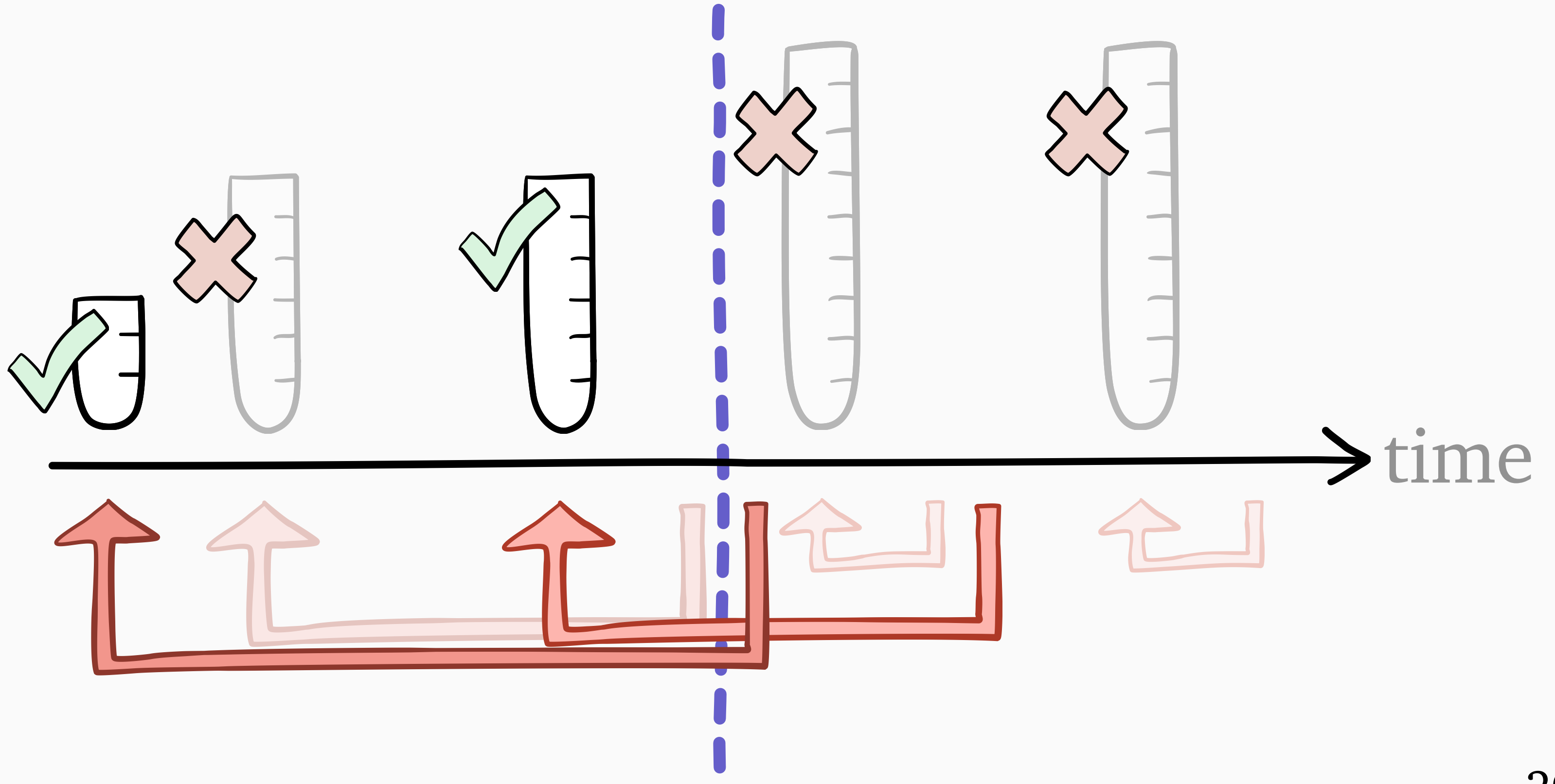
crossing work

$$C_{\text{Boost}} = C_W \mathbf{E}[e^{\gamma(S-b(S))}]\mathbf{E}[e^{\gamma V}]$$

Lemma: finite if $b(s) = O(1/s)$

V = crossing work

work that "boosts past" a given time



How to achieve *strong* tail optimality?

$$C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

How to achieve *strong* tail optimality?

$$C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

Finite batch problem?

How to achieve *strong* tail optimality?

$$C = \underbrace{\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t]}_{\text{“}\infty \cdot \mathbf{P}[T > \infty]\text{”}} = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

Finite batch problem?

How to achieve *strong* tail optimality?

$$C = \lim_{t\to\infty} \underbrace{e^{\gamma t}\mathbf{P}[T > t]}_{\text{“}\infty \cdot \mathbf{P}[T > \infty]\text{”}} = \lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

Always zero in batch setting

Finite batch problem?

How to achieve *strong* tail optimality?

$$C = \underbrace{\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t]}_{\text{“}\infty\cdot\mathbf{P}[T>\infty]\text{”}} = \underbrace{\lim_{\theta\to\gamma}\frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0\cdot\mathbf{E}[e^{\gamma T}]\text{”}}$$

Always zero in batch setting

Finite batch problem?

How to achieve *strong* tail optimality?

$$C = \underbrace{\lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t]}_{\text{“}\infty \cdot \mathbf{P}[T > \infty]\text{”}} = \underbrace{\lim_{\theta\to\gamma} \frac{\gamma - \theta}{\gamma}\mathbf{E}[e^{\theta T}]}_{\text{“}0 \cdot \mathbf{E}[e^{\gamma T}]\text{”}}$$

Finite batch problem?

Always zero in batch setting

Makes sense in batch setting!

How to achieve *strong* tail optimality?

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

"$\infty \cdot \mathbf{P}[T > \infty]$" — Always zero in batch setting

"$0 \cdot \mathbf{E}[e^{\gamma T}]$" — Makes sense in batch setting!

Finite batch problem?

$t_i = d_i - a_i$

$a_i =$ arrival time of job i

$d_i =$ departure time of job i

How to achieve *strong* tail optimality?

$$C = \lim_{t\to\infty} e^{\gamma t}\mathbf{P}[T > t] = \lim_{\theta\to\gamma} \frac{\gamma-\theta}{\gamma}\mathbf{E}[e^{\theta T}]$$

"$\infty \cdot \mathbf{P}[T > \infty]$" — Always zero in batch setting

"$0 \cdot \mathbf{E}[e^{\gamma T}]$" — Makes sense in batch setting!

Finite batch problem?

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

How to achieve *strong* tail optimality?

$$C = \lim_{t \to \infty} e^{\gamma t} \mathbf{P}[T > t] = \lim_{\theta \to \gamma} \frac{\gamma - \theta}{\gamma} \mathbf{E}[e^{\theta T}]$$

"$\infty \cdot \mathbf{P}[T > \infty]$" — Always zero in batch setting

"$0 \cdot \mathbf{E}[e^{\gamma T}]$" — Makes sense in batch setting!

Finite batch problem?

$t_i = d_i - a_i$

$a_i =$ arrival time of job i

$d_i =$ departure time of job i

Batch problem: minimize

almost classic problem

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n} \sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n} \sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$

$a_i =$ arrival time of job i

$d_i =$ departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

⚠ **Classic metric:** mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

$-\theta < 0$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$
a_i = arrival time of job i
d_i = departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

$-\theta < 0$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$

$a_i =$ arrival time of job i

$d_i =$ departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

can't start i before a_i

$\gamma > 0$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

$-\theta < 0$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$
a_i = arrival time of job i
d_i = departure time of job i

Batch problem: minimize

can't start i before a_i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

$-\theta < 0$

Relaxation solved by (sign-flipped) WDSPT, which is **Boost** with

$$b(s) = \frac{1}{\gamma}\log\frac{1}{1-e^{-\gamma s}}$$

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$

a_i = arrival time of job i

d_i = departure time of job i

Batch problem: minimize

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

can't start i before a_i

$\gamma > 0$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

$-\theta < 0$

Relaxation solved by (sign-flipped) WDSPT, which is **Boost** with

$$b(s) = \frac{1}{\gamma}\log\frac{1}{1-e^{-\gamma s}}$$

γ-Boost

How to achieve *strong* tail optimality?

$t_i = d_i - a_i$
a_i = arrival time of job i
d_i = departure time of job i

Batch problem: minimize

can't start i before a_i

$$\mathbf{E}[e^{\gamma T}] = \frac{1}{n}\sum_{i=1}^{n} e^{\gamma t_i} = \frac{1}{n}\sum_{i=1}^{n} e^{-\gamma a_i} e^{\gamma d_i}$$

$\gamma > 0$

Classic metric: mean weighted discounted departure time

$$\frac{1}{n}\sum_{i=1}^{n} w_i e^{-\theta d_i}$$

$-\theta < 0$

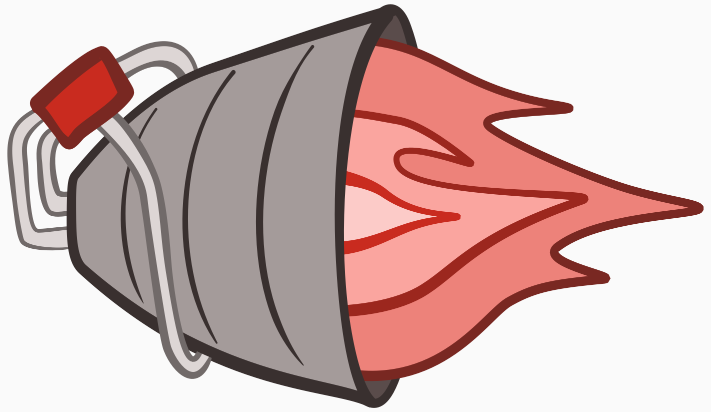
Relaxation solved by (sign-flipped) WDSPT, which is **Boost** with

$$b(s) = \frac{1}{\gamma}\log\frac{1}{1-e^{-\gamma s}}$$

γ-Boost

Unknown sizes:
swap WDSPT for Gittins

Boost



✓ Why is achieving strong tail optimality hard?

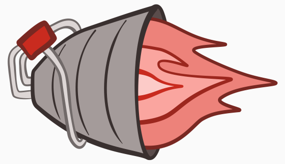
✓ How does the **Boost** policy family work?

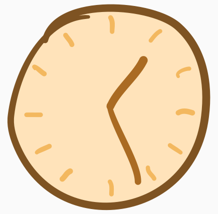
? How do we achieve strong tail optimality?

Boost

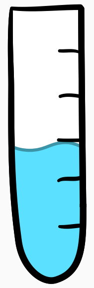
- Why is achieving strong tail optimality hard?
- How does the **Boost** policy family work?
- How do we achieve strong tail optimality?

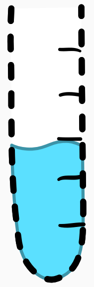
Our contributions:


Design the **Boost** scheduling policy

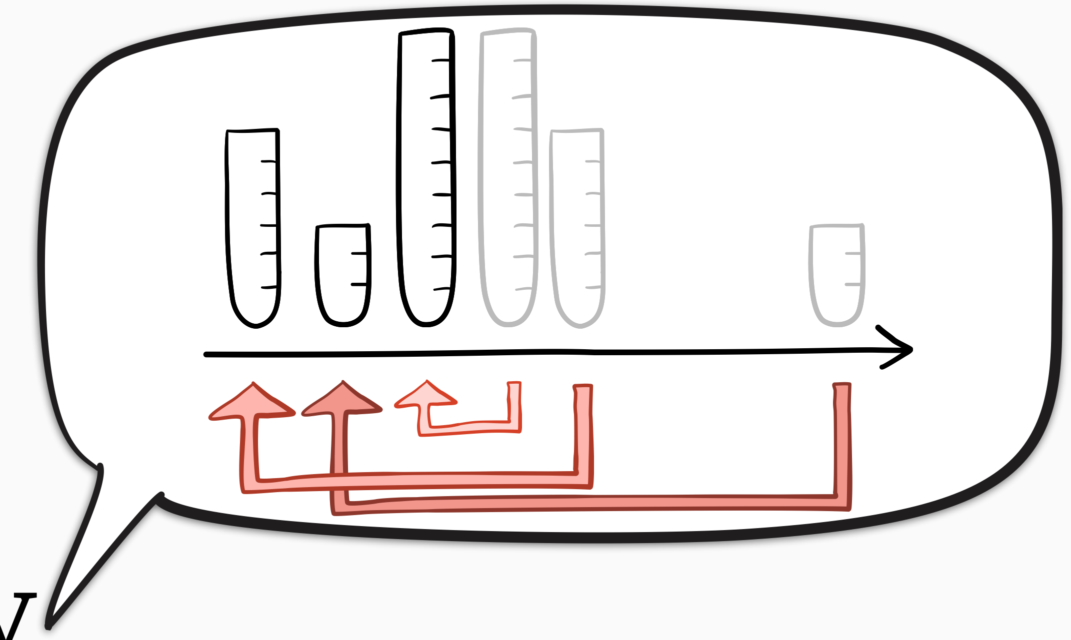

Analyze **Boost**'s performance

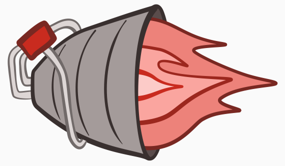

Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

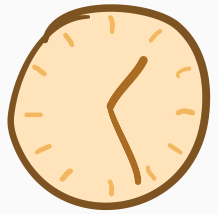

Known job sizes
Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.


Unknown job sizes
Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.

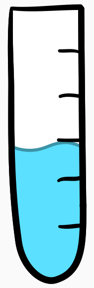
Our contributions:



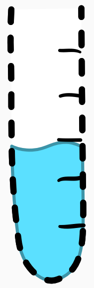
 Design the **Boost** scheduling policy

 Analyze **Boost**'s performance

 Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

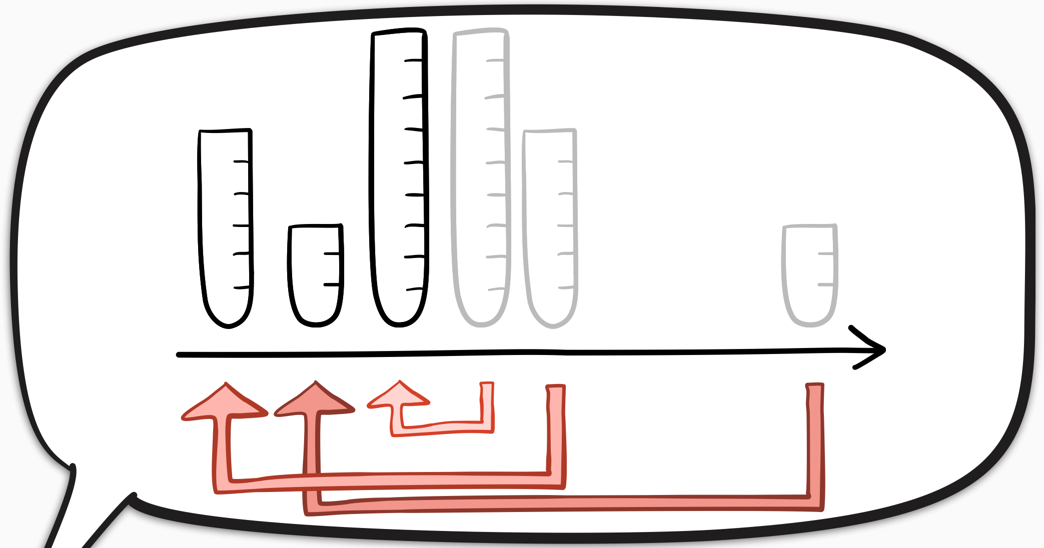
 **Known job sizes**

Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

 **Unknown job sizes**

Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.

Our contributions:



Design the **Boost** scheduling policy

compute C_{Boost}

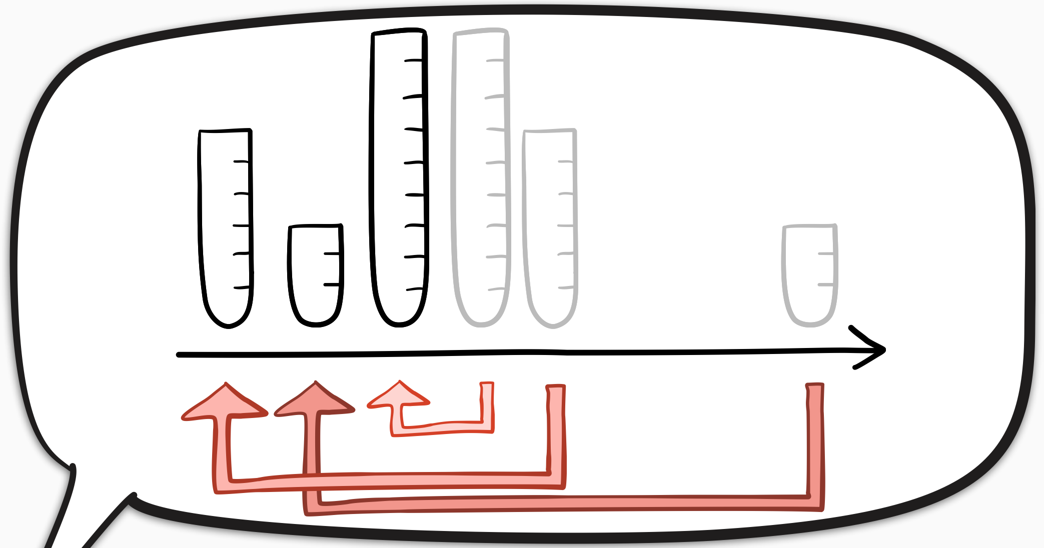
Analyze **Boost**'s performance

Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Known job sizes
Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

Unknown job sizes
Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.

Our contributions:



Design the **Boost** scheduling policy

compute C_{Boost}

Analyze **Boost**'s performance

γ-Boost:

$$b(s) = \frac{1}{\gamma} \log \frac{1}{1 - e^{-\gamma s}}$$

Prove **Boost** is *strongly tail-optimal* for light-tailed sizes

Known job sizes

Yu & Scully. *Strongly Tail-Optimal Scheduling in the Light-Tailed M/G/1.* SIGMETRICS 2024.

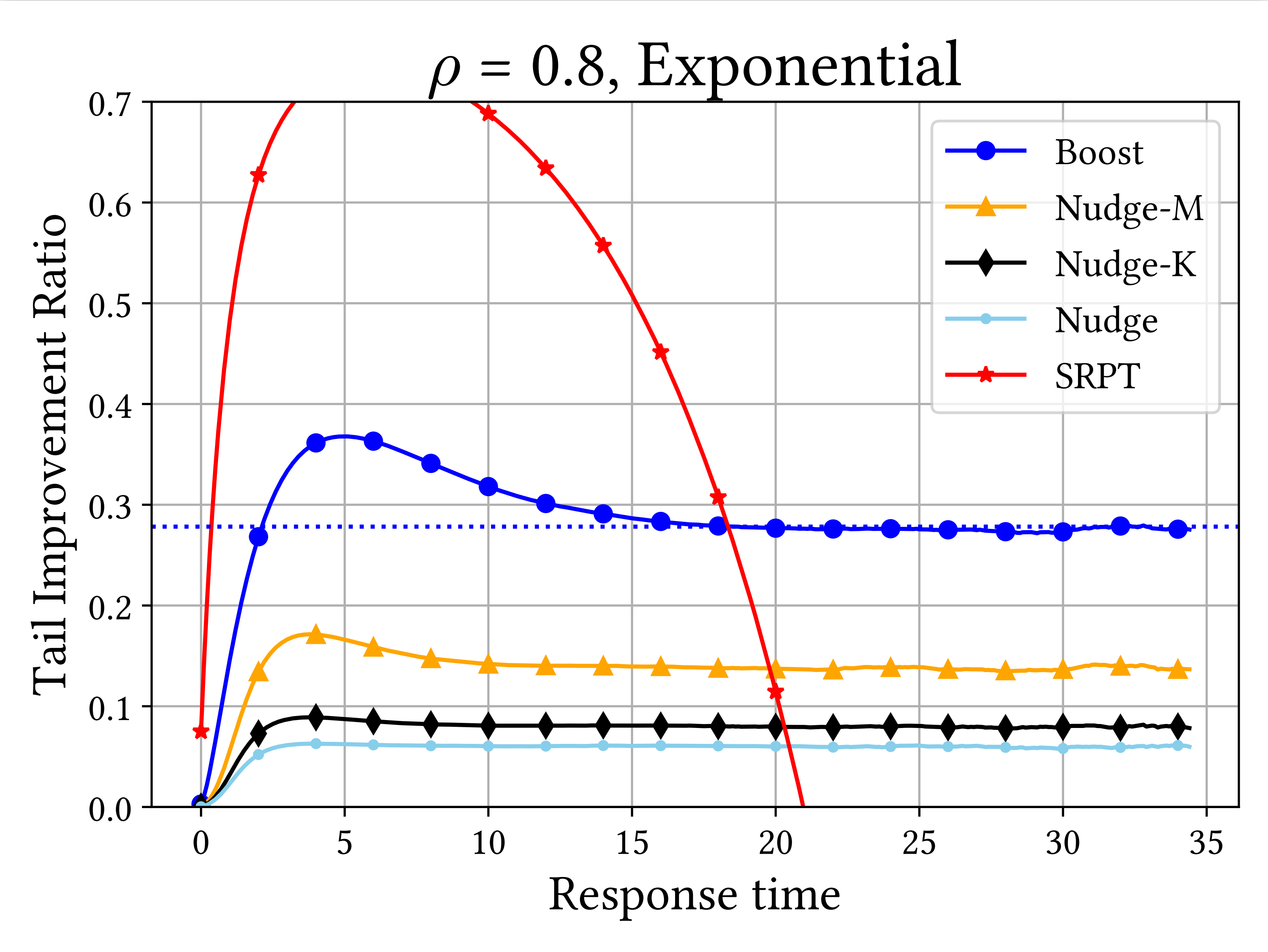
Unknown job sizes

Harlev, Yu, & Scully. *A Gittins Policy for Optimizing Tail Latency.* MAMA 2024.

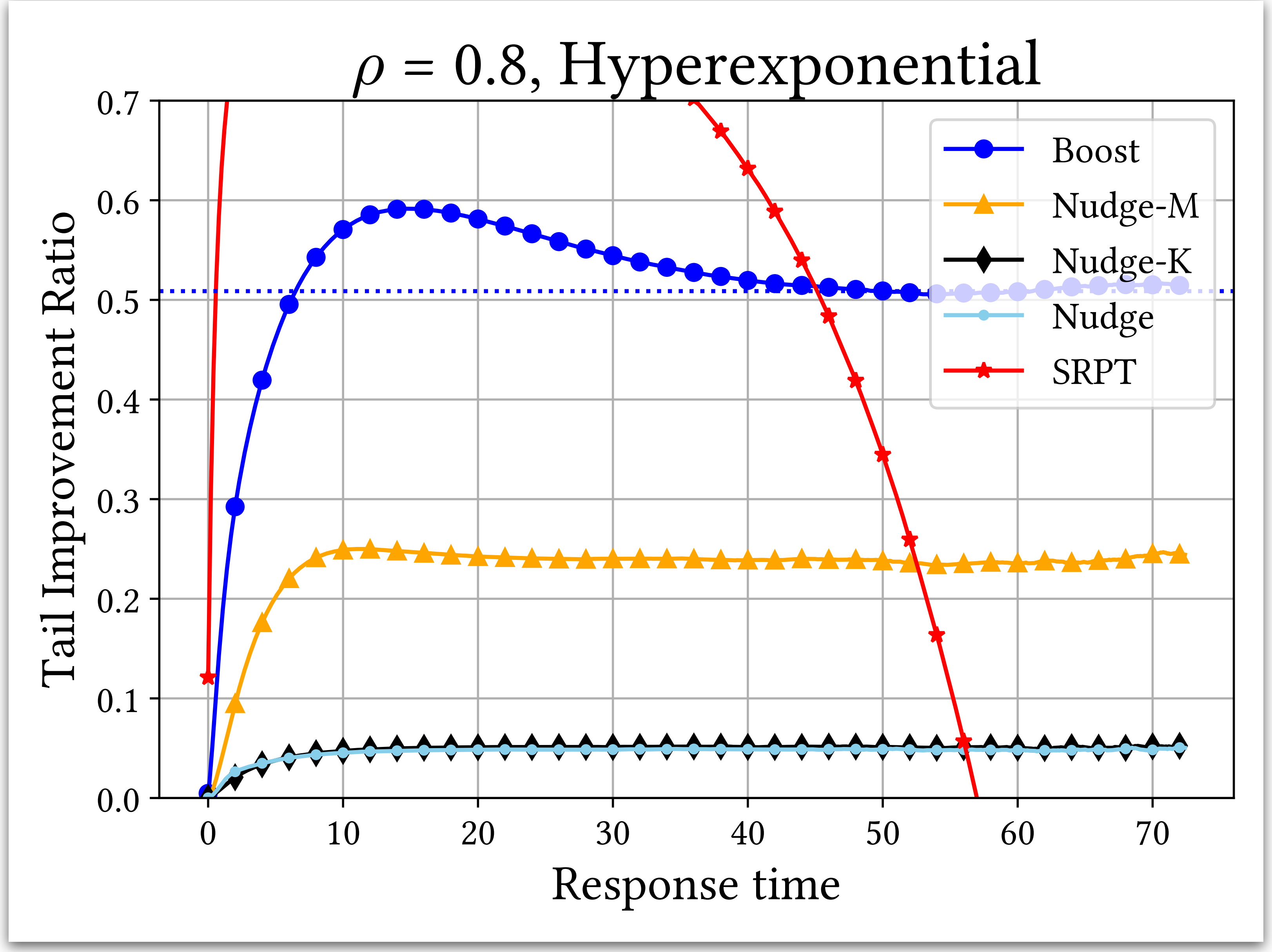
Bonus slides

Impact of job size variance

Low variance

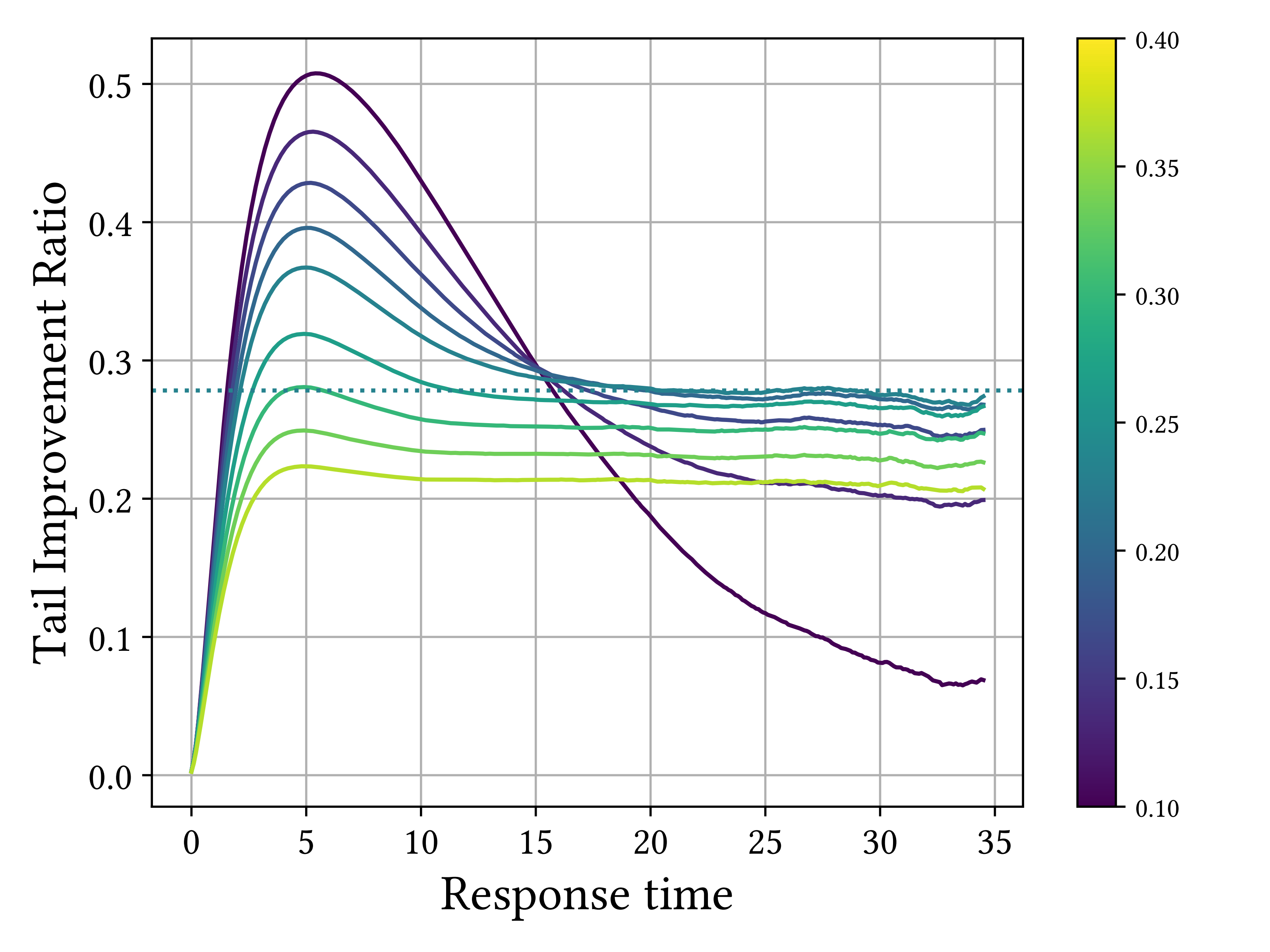


High variance

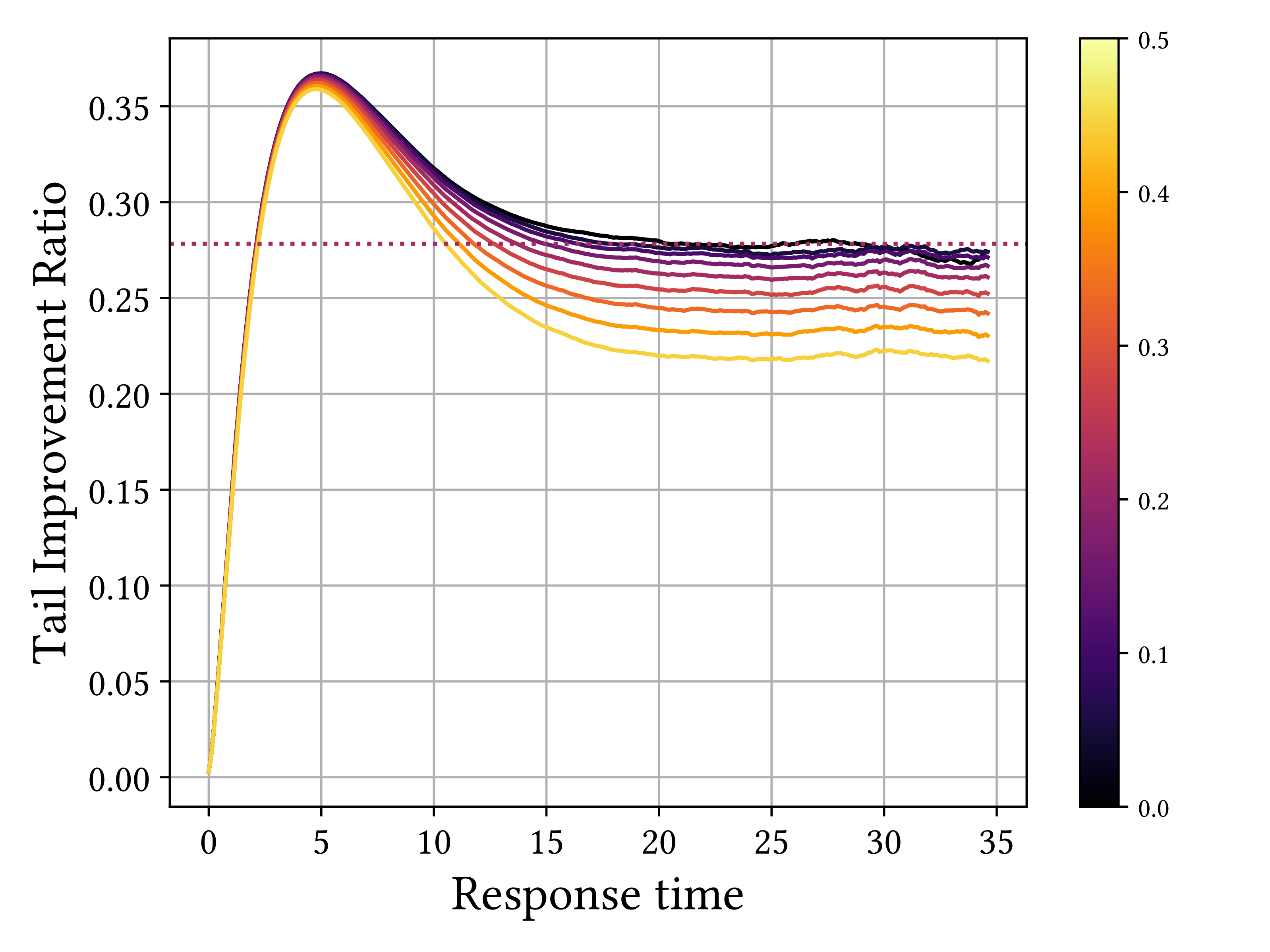


Sensitivity analysis

Misspecified γ



Noisy size information



Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim As^{-\alpha}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim Ae^{-\alpha s}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T > t] \sim C t^{-\gamma}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T > t] \sim C t^{-\gamma}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T > t] \sim C e^{-\gamma t}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim As^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim Ae^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

γ_π = *decay rate* of π

C_π = *tail constant* of π

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

γ_π = *decay rate of* π

C_π = *tail constant of* π

Weak optimality:

maximize γ_π

Heavy-tailed sizes

"S Pareto-ish" (regularly varying)

$$\mathbf{P}[S > s] \sim A s^{-\alpha}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi t^{-\gamma_\pi}$$

Light-tailed sizes

"S exponential-ish or lighter" (class I)

$$\mathbf{P}[S > s] \sim A e^{-\alpha s}$$

$$\mathbf{P}[T_\pi > t] \sim C_\pi e^{-\gamma_\pi t}$$

$\gamma_\pi = \textit{decay rate}$ of π

$C_\pi = \textit{tail constant}$ of π

Weak optimality:
maximize γ_π

Strong optimality:
maximize γ_π, minimize C_π

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)		
FCFS		

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	
FCFS		optimal γ

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?		

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?	**“Catastrophe”** one giant job	

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$ (I'm the giant job)	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?	**"Catastrophe"** one giant job	

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ
Main cause of large T?	**"Catastrophe"** one giant job	

Background on decay rates

	Heavy-tailed sizes	**Light-tailed sizes**
SRPT, LAS, etc. (least attained service)	**optimal** $\gamma = \alpha$ I'm the giant job	**pessimal** γ
FCFS	**pessimal** $\gamma = \alpha - 1$ I'm stuck behind the giant job	**optimal** γ
? Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ I see lots of *work* when I arrive
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$ I'm the giant job	pessimal γ I'm a very big job, lots of smaller jobs are passing me
FCFS	pessimal $\gamma = \alpha - 1$ I'm stuck behind the giant job	optimal γ I see lots of *work* when I arrive
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
Main cause of large T?	**“Catastrophe”** one giant job	**“Conspiracy”** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets		
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	
Main cause of large T?	**“Catastrophe”** one giant job	**“Conspiracy”** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	intermediate γ
Main cause of large T?	**“Catastrophe”** one giant job	**“Conspiracy”** lots of biggish jobs

Background on decay rates

	Heavy-tailed sizes	Light-tailed sizes
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	intermediate γ
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

I'm in bucket 2, lots of bucket 1 jobs are passing me

Background on decay rates

	Heavy-tailed sizes	Light-taile
SRPT, LAS, etc. (least attained service)	optimal $\gamma = \alpha$	pessimal γ
FCFS	pessimal $\gamma = \alpha - 1$	optimal γ
SRPT or LAS with just two buckets	pessimal $\gamma = \alpha - 1$	intermediate γ
Main cause of large T?	**"Catastrophe"** one giant job	**"Conspiracy"** lots of biggish jobs

I'm a very big job, lots of smaller jobs are passing me

I'm in bucket 2, lots of bucket 1 jobs are passing me

Background on decay rates

	Light-taile
SRPT, LAS, etc. (least attained service)	pessimal γ
FCFS	optimal γ
SRPT or LAS with just two buckets	intermediate γ
Main cause of large T?	**"Conspiracy"** lots of biggish jobs

I'm a very big job, lots of smaller jobs are passing me

I'm in bucket 2, lots of bucket 1 jobs are passing me

Background on decay rates

Light-taile[d]

I'm a very big job, lots of smaller jobs are passing me

SRPT, LAS, etc.
(least attained service) — pessimal γ

FCFS — optimal γ

I'm in bucket 2, lots of bucket 1 jobs are passing me

SRPT or LAS with just two buckets — intermediate γ

⚠ **Takeaway:**
for optimality, must *avoid strict priorities*

Main cause of large T? — **"Conspiracy"** lots of biggish jobs

Background on weak and strong optimality

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

Weakly optimal

$R_\pi < \infty$

Strongly optimal

$R_\pi = 1$

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$		
Strongly optimal $R_\pi = 1$		

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS SRPT PS (processor sharing) LAS (least attained service)	
Strongly optimal $R_\pi = 1$		

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

newish Corollary of prior work [Wierman & Zwart, 2012]

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	Light-tailed sizes
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

newish Corollary of prior work [Wierman & Zwart, 2012]

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$


	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Background on weak and strong optimality

Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served)
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served)
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierman & Zwart, 2012]

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierman & Zwart, 2012]

Background on weak and strong optimality

Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

	Heavy-tailed sizes ✓	Light-tailed sizes ?
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierma

Theorem: $R_{\text{Nudge}} < R_{\text{FCFS}}$

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	FCFS? [Wierma

Theorem:
$R_{\text{Nudge}} < R_{\text{FCFS}}$

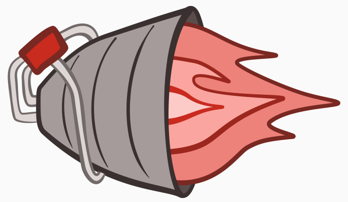
Background on weak and strong optimality

Asymptotic tail ratio: $$R_\pi = \sup_{\pi'} \limsup_{t \to \infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$$

	Heavy-tailed sizes	**Light-tailed sizes**
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	

Background on weak and strong optimality

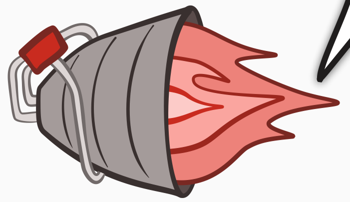
Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes 	Light-tailed sizes 
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et al., 2021]
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	 **Boost** 

Background on weak and strong optimality

Asymptotic tail ratio: $R_\pi = \sup_{\pi'} \limsup_{t\to\infty} \frac{\mathbf{P}[T_\pi > t]}{\mathbf{P}[T_{\pi'} > t]}$

	Heavy-tailed sizes ✓	Light-tailed sizes ✓
Weakly optimal $R_\pi < \infty$	Preemptive LCFS	FCFS (first-come first-served) Nudge [Grosof et
Strongly optimal $R_\pi = 1$	SRPT PS (processor sharing) LAS (least attained service)	NEW! **Boost**



Theorem:
$R_{\text{Boost}} = 1$