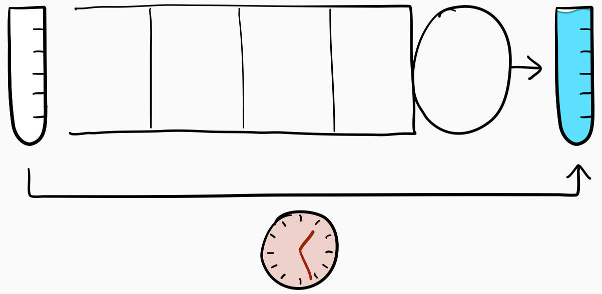
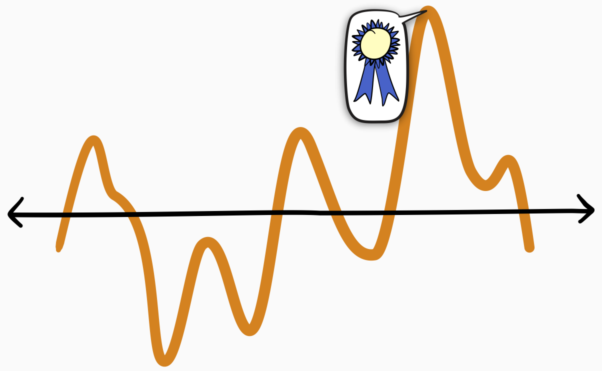
What problems does the **Gittins index** solve?

Ziv Scully
Cornell ORIE

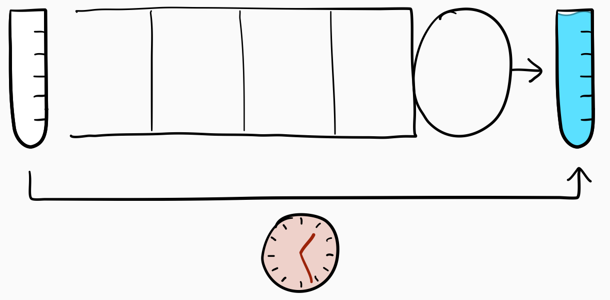


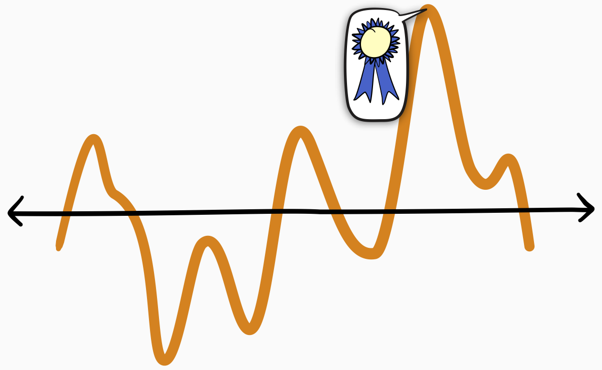


What problems does the **Gittins index** solve?

Isn't this old news?
[Gittins, 1979; Gittins, 1989]

Ziv Scully
Cornell ORIE



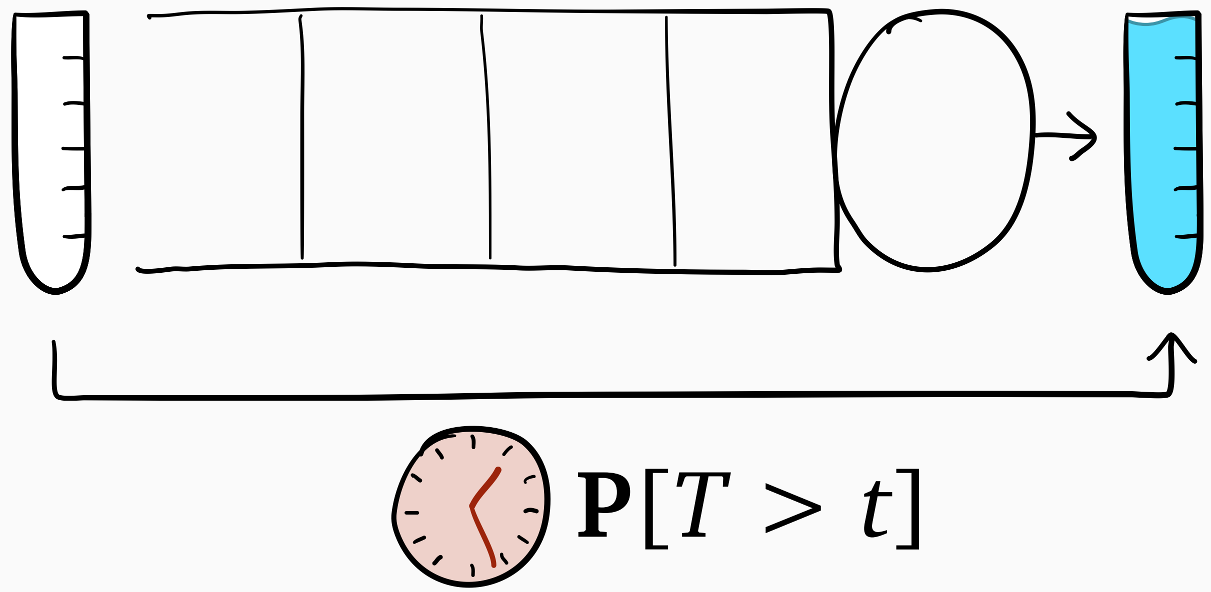


Two new **Gittins** applications

Two new Gittins applications

Tail scheduling

(in single-server queues)

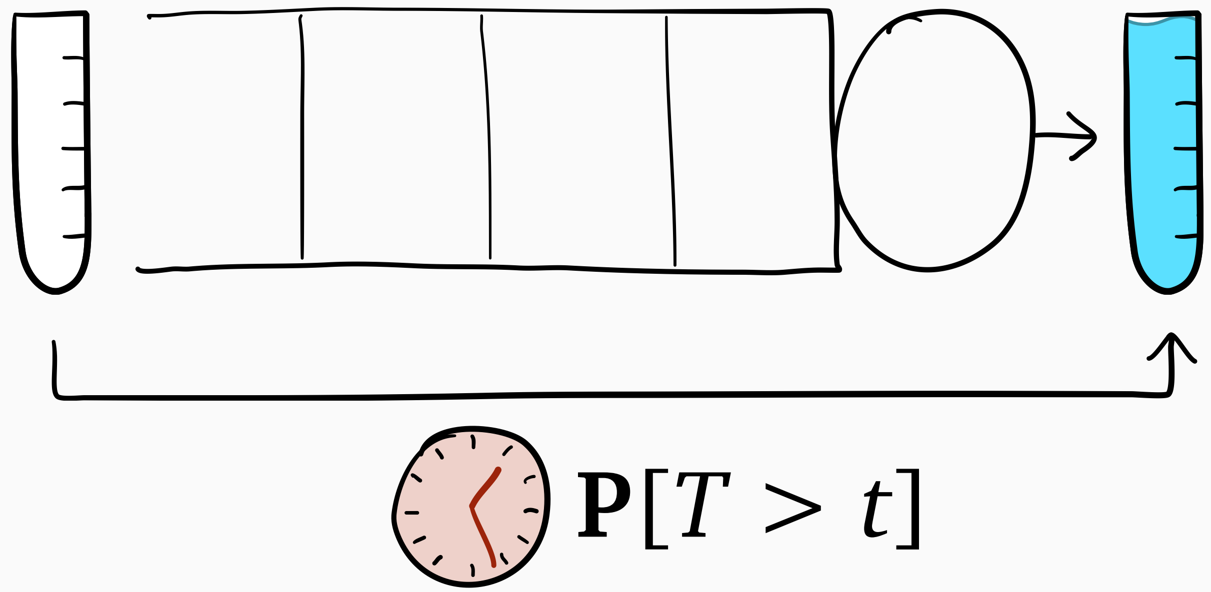


$\mathbf{P}[T > t]$

Two new **Gittins** applications

Tail scheduling

(in single-server queues)



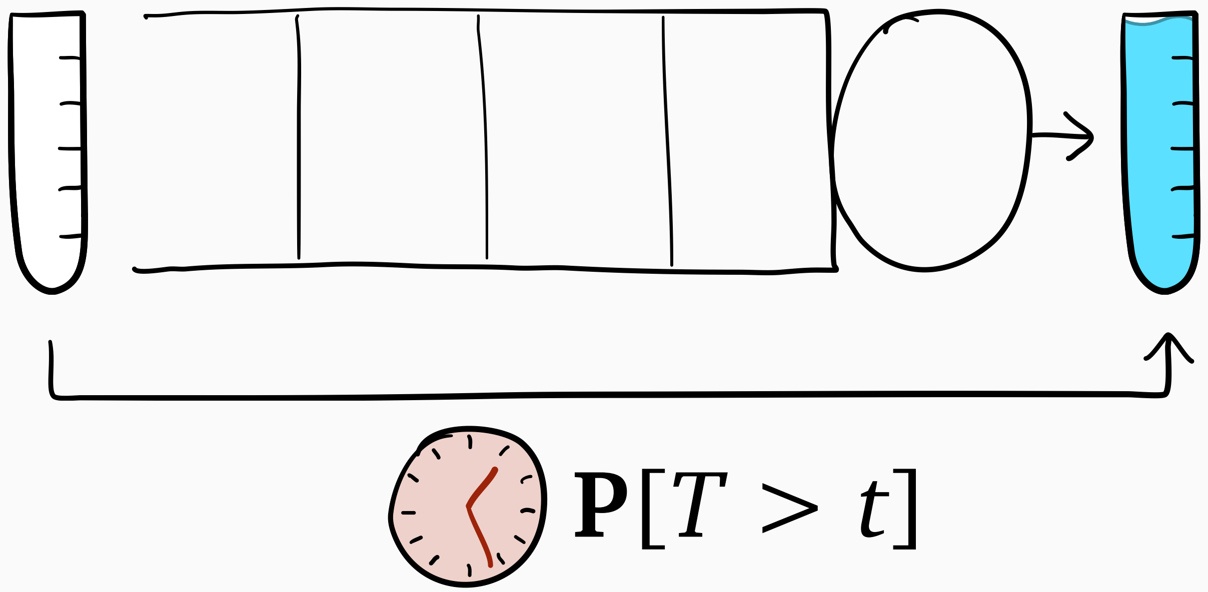
$\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

Two new Gittins applications

Tail scheduling

(in single-server queues)

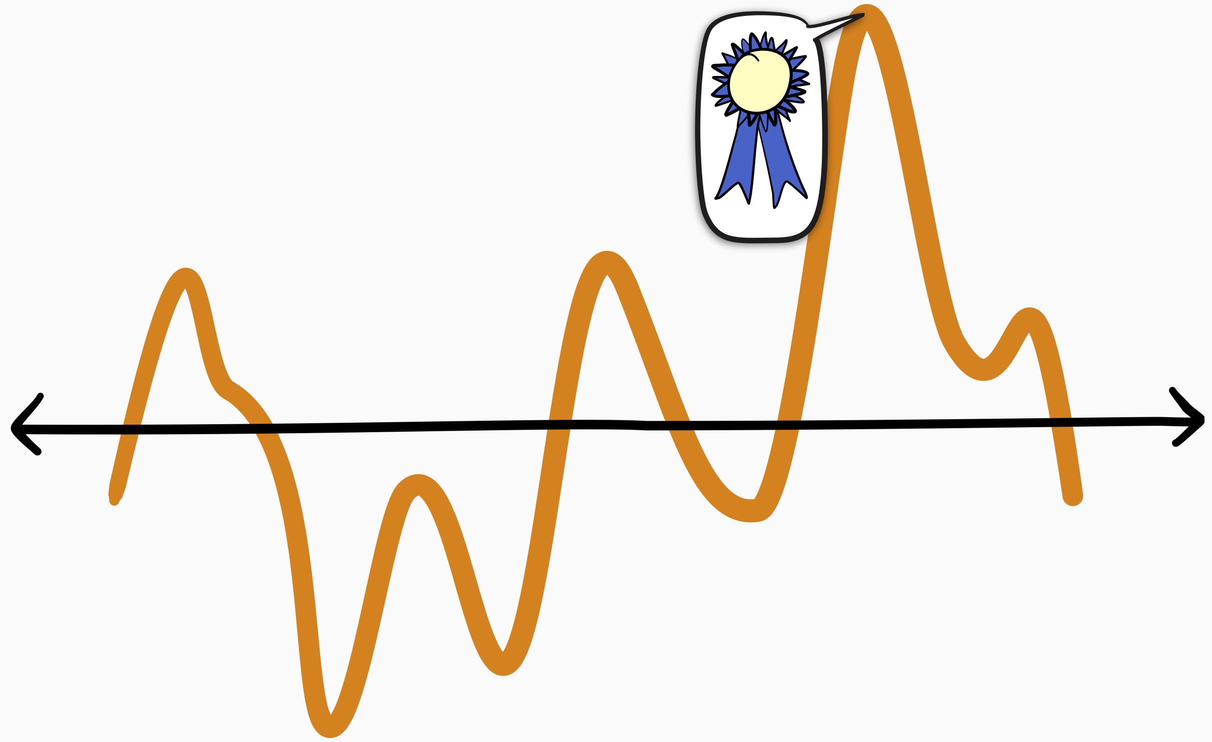


$\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

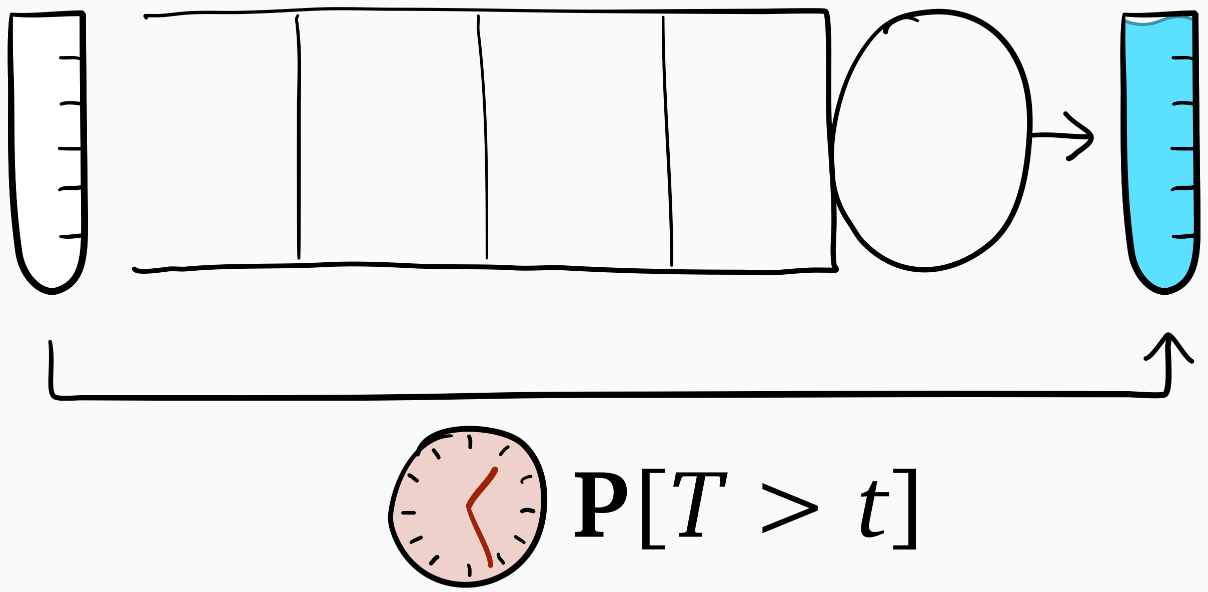
BayesOpt

(Bayesian optimization)



Two new Gittins applications

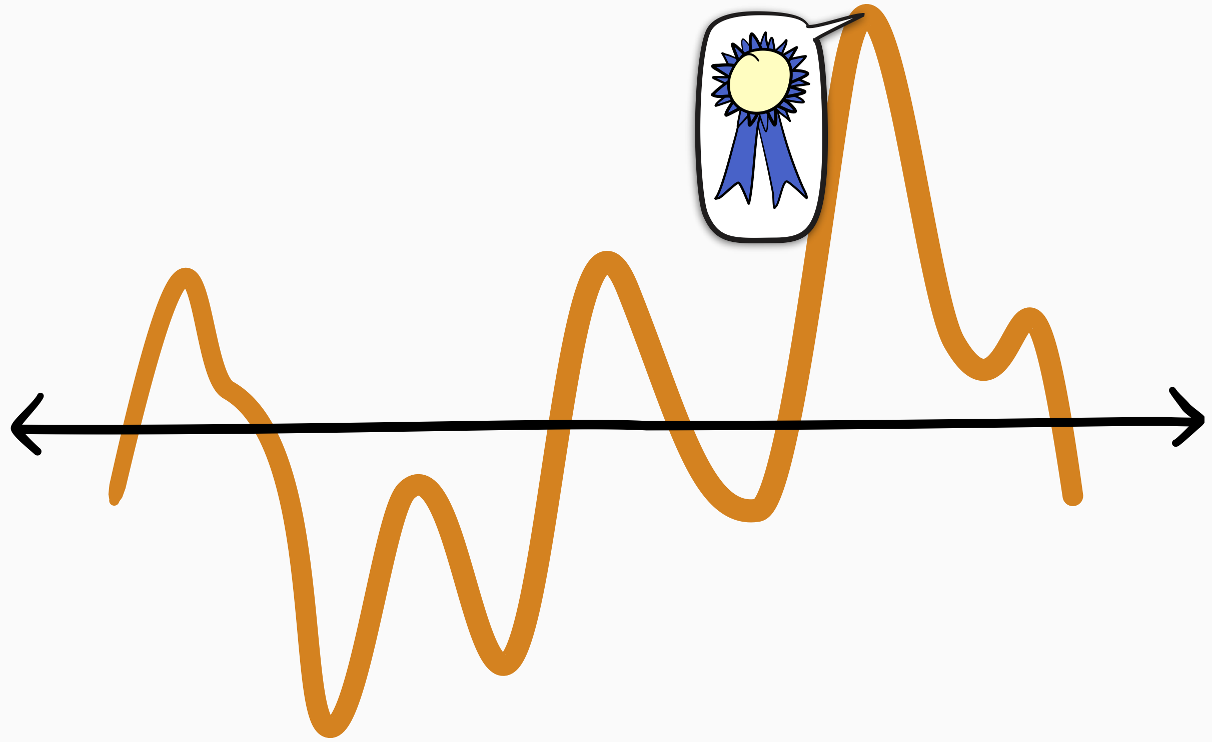
Tail scheduling
(in single-server queues)



$\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

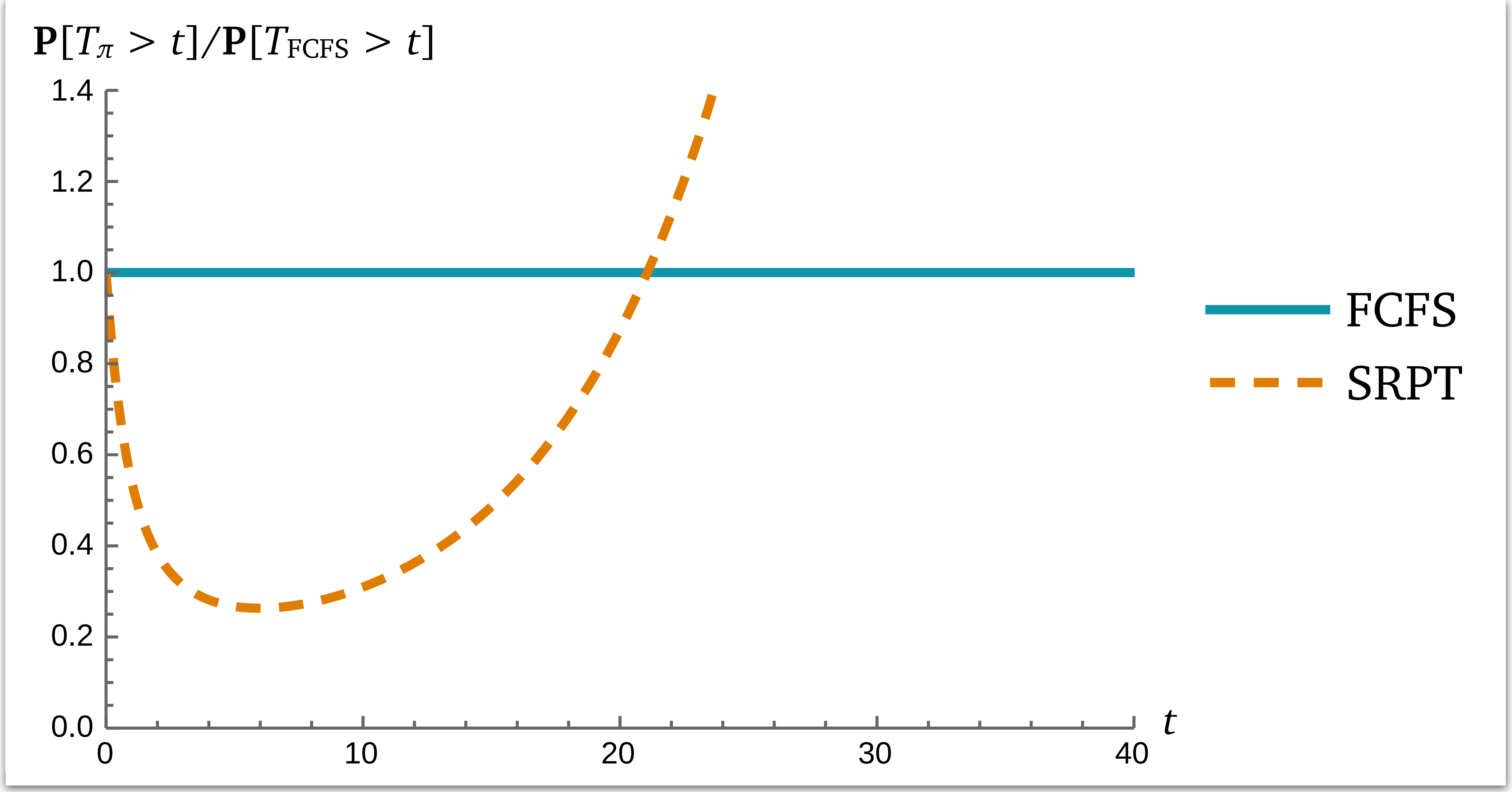
BayesOpt
(Bayesian optimization)



Goal: find large function value with few function evaluations

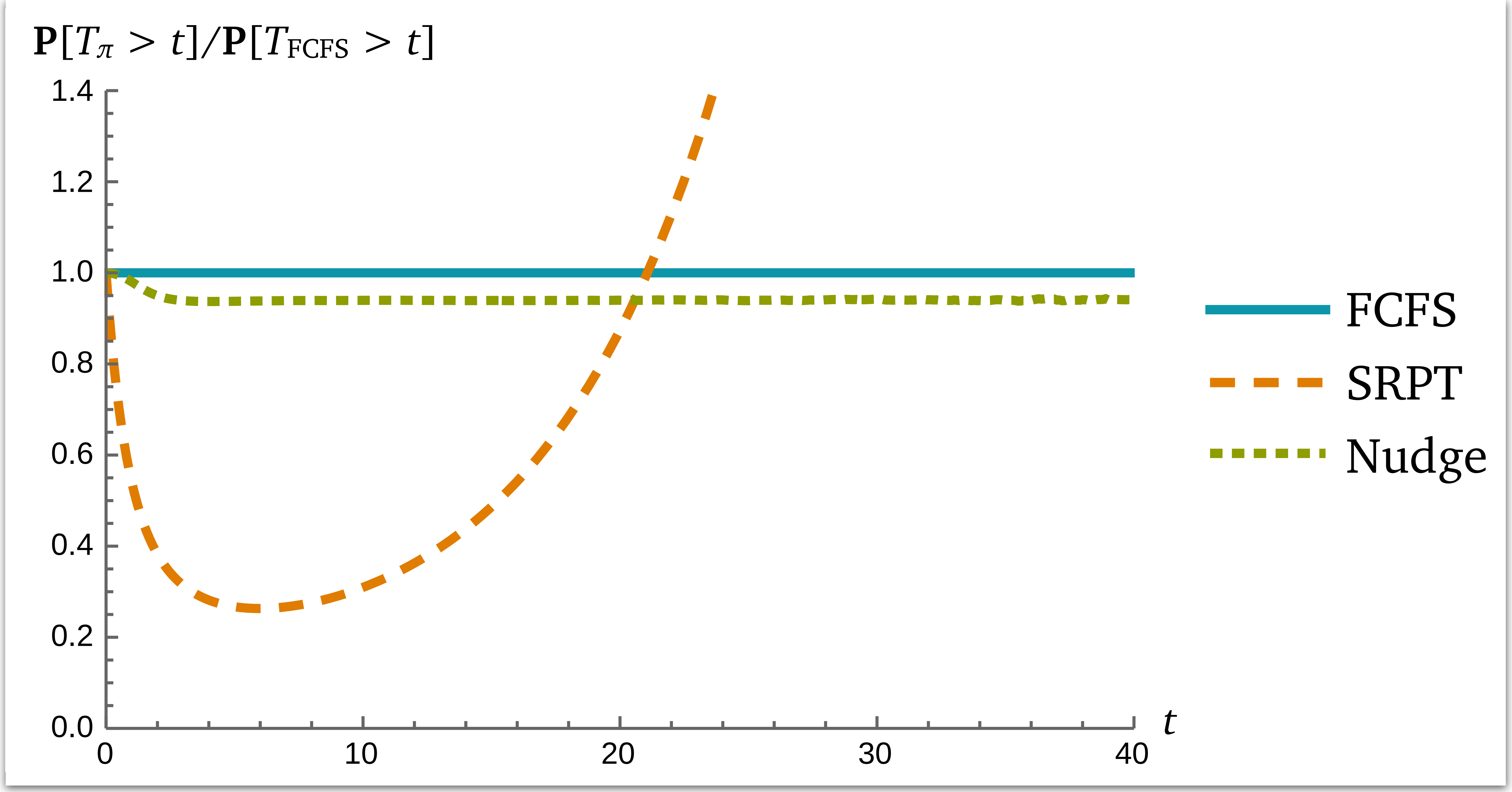
Tail scheduling empirical results

George Yu



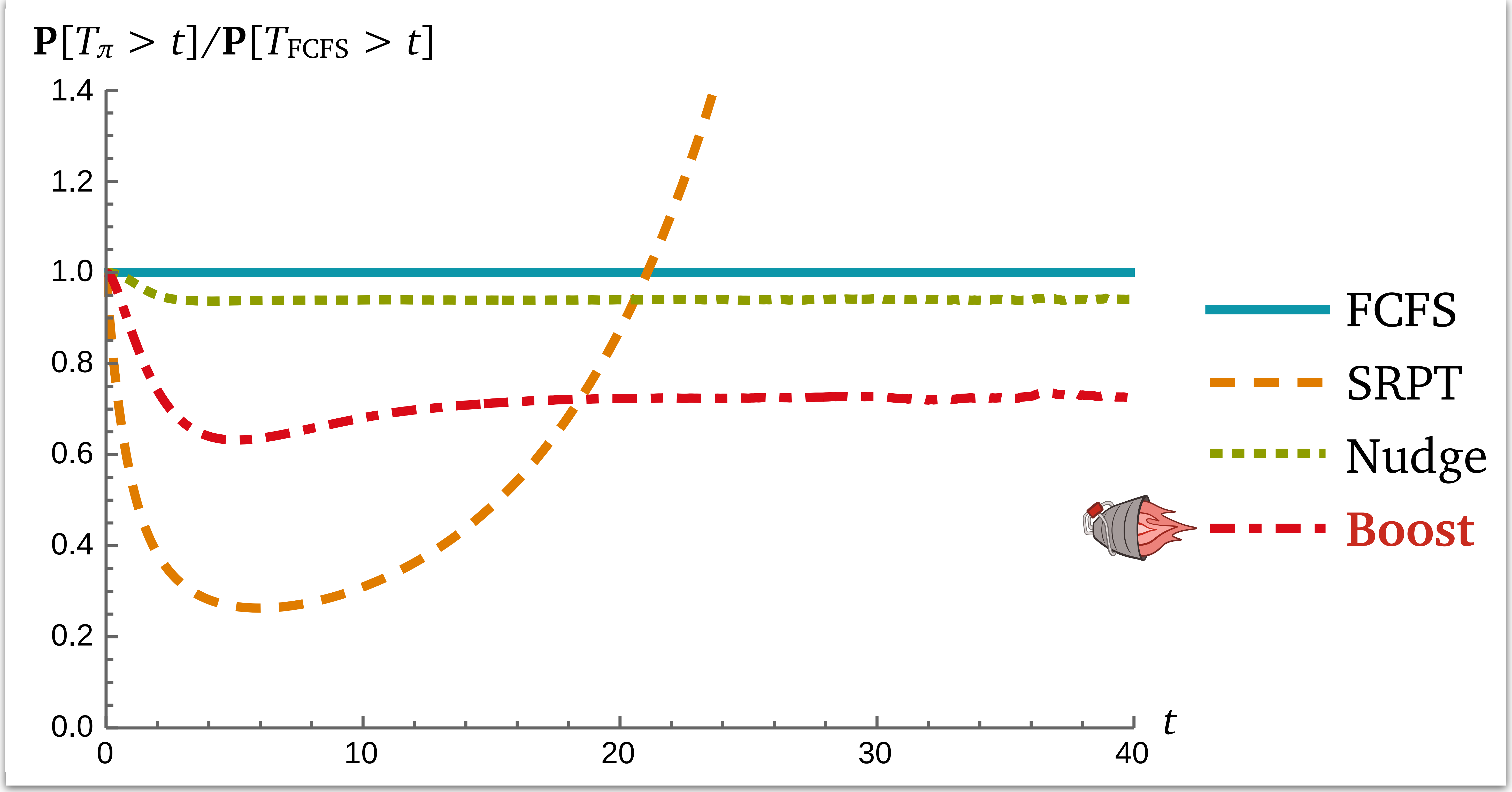
Tail scheduling empirical results

George Yu



Tail scheduling empirical results

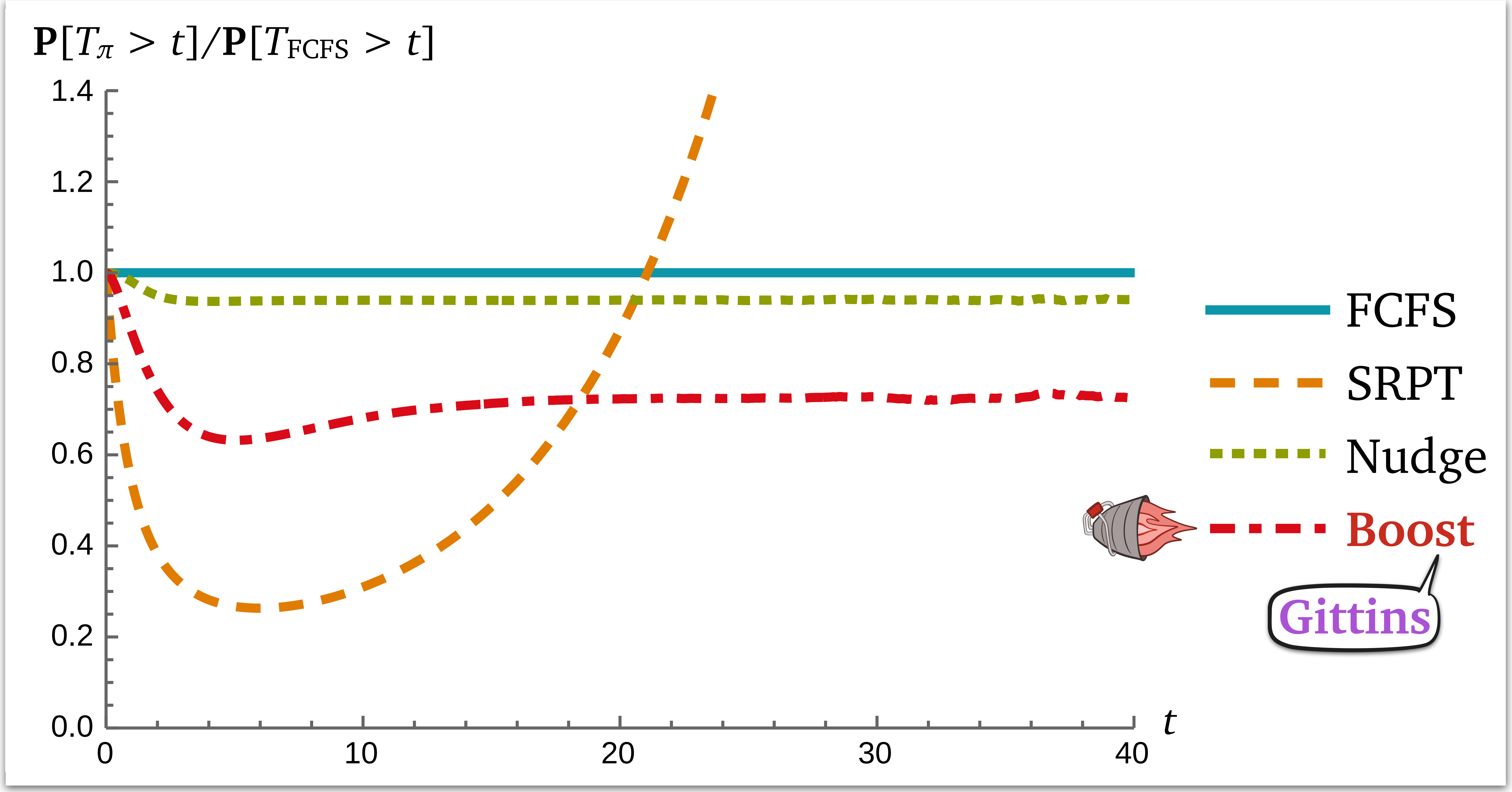
George Yu



Tail scheduling empirical results

George Yu

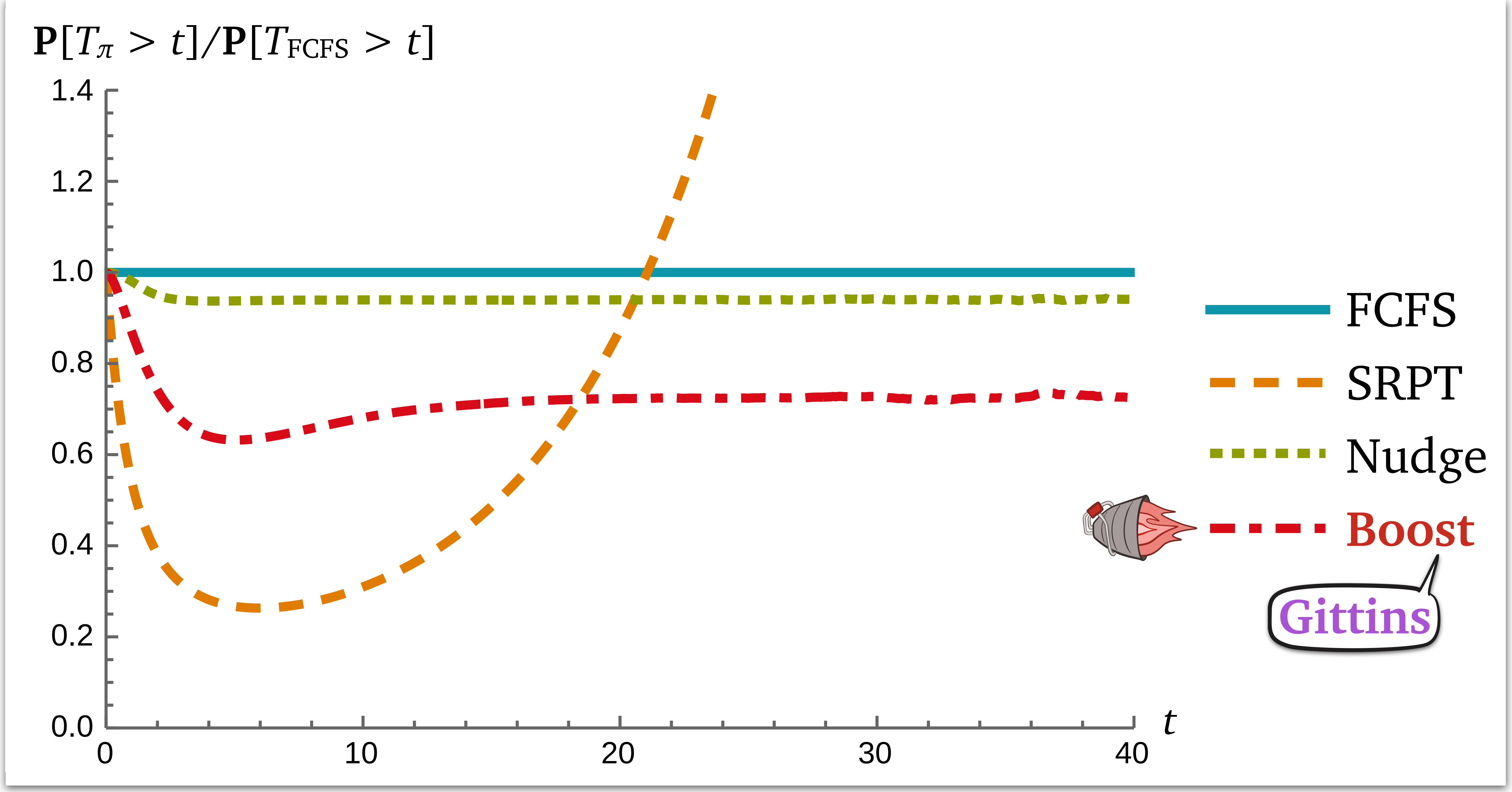
Amit Harlev



Tail scheduling empirical results

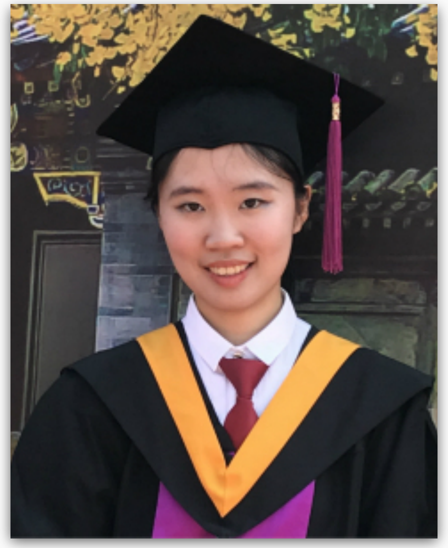
George Yu

Amit Harlev



See our SIGMETRICS paper [Yu & Scully, 2024] and MAMA abstract [Harlev et al., 2024]

BayesOpt empirical results

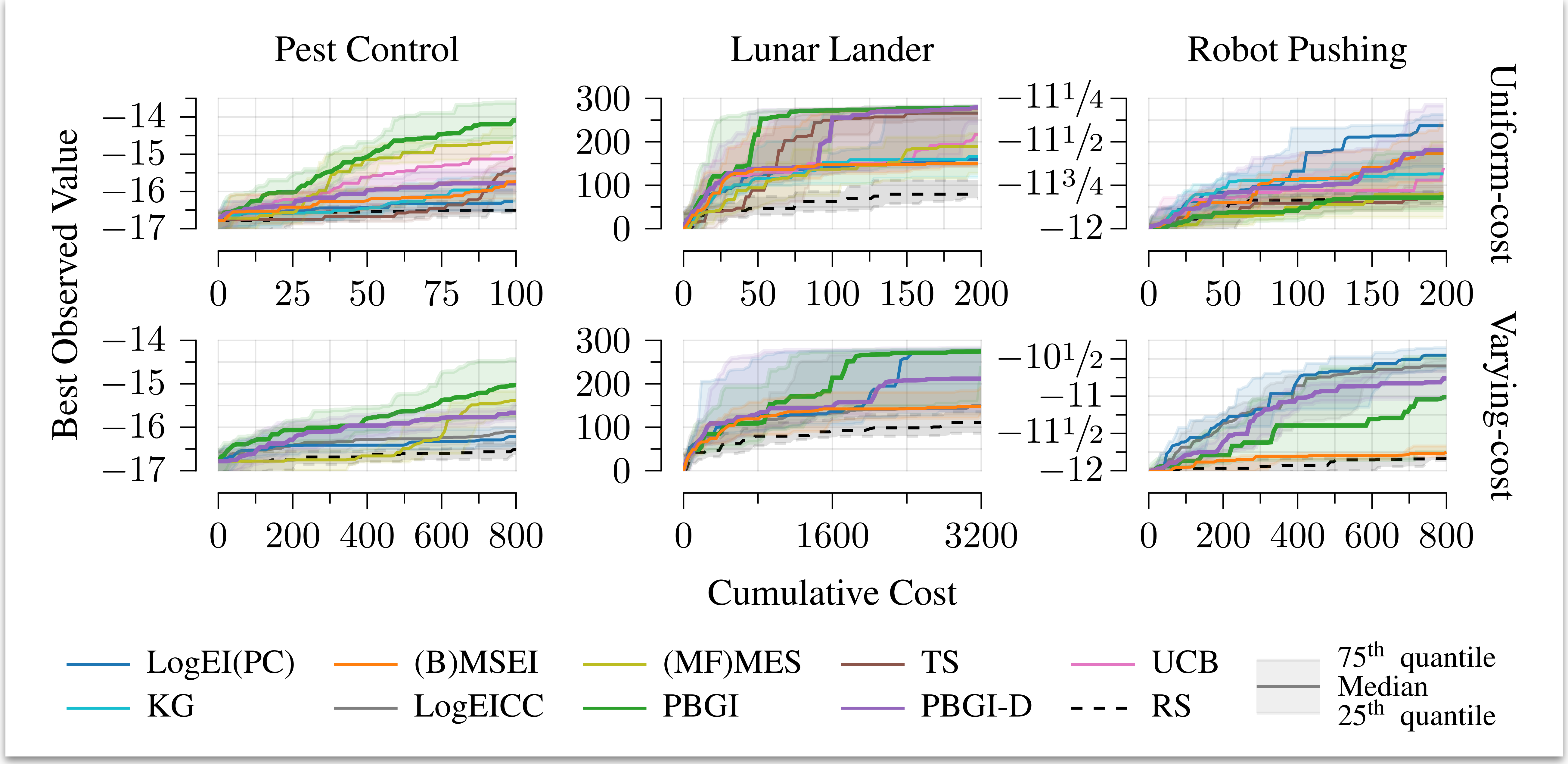


Qian Xie

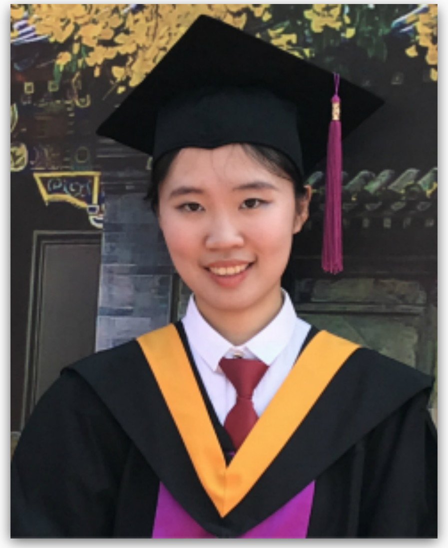
Alex Terenin

Raul Astudillo

Peter Frazier



BayesOpt empirical results

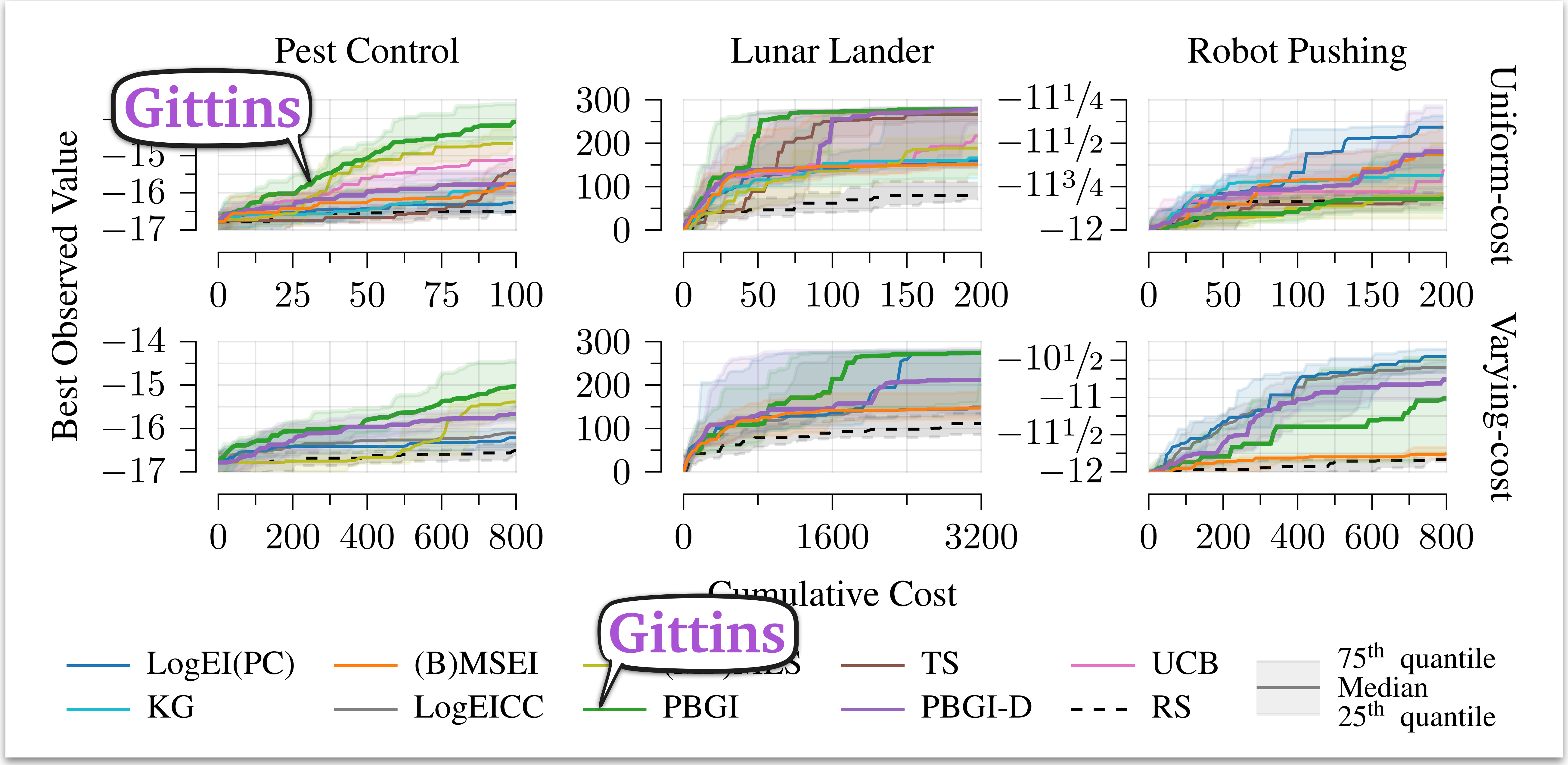


Qian Xie

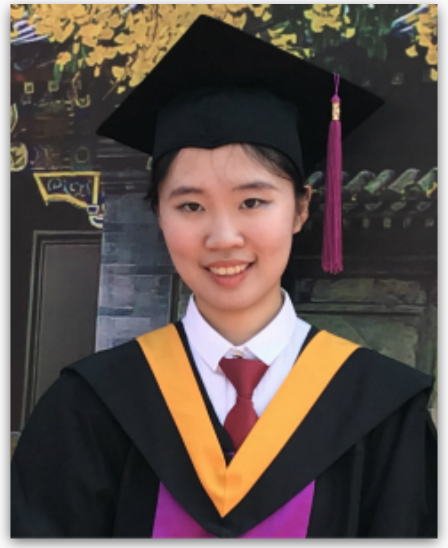
Alex Terenin

Raul Astudillo

Peter Frazier



BayesOpt empirical results

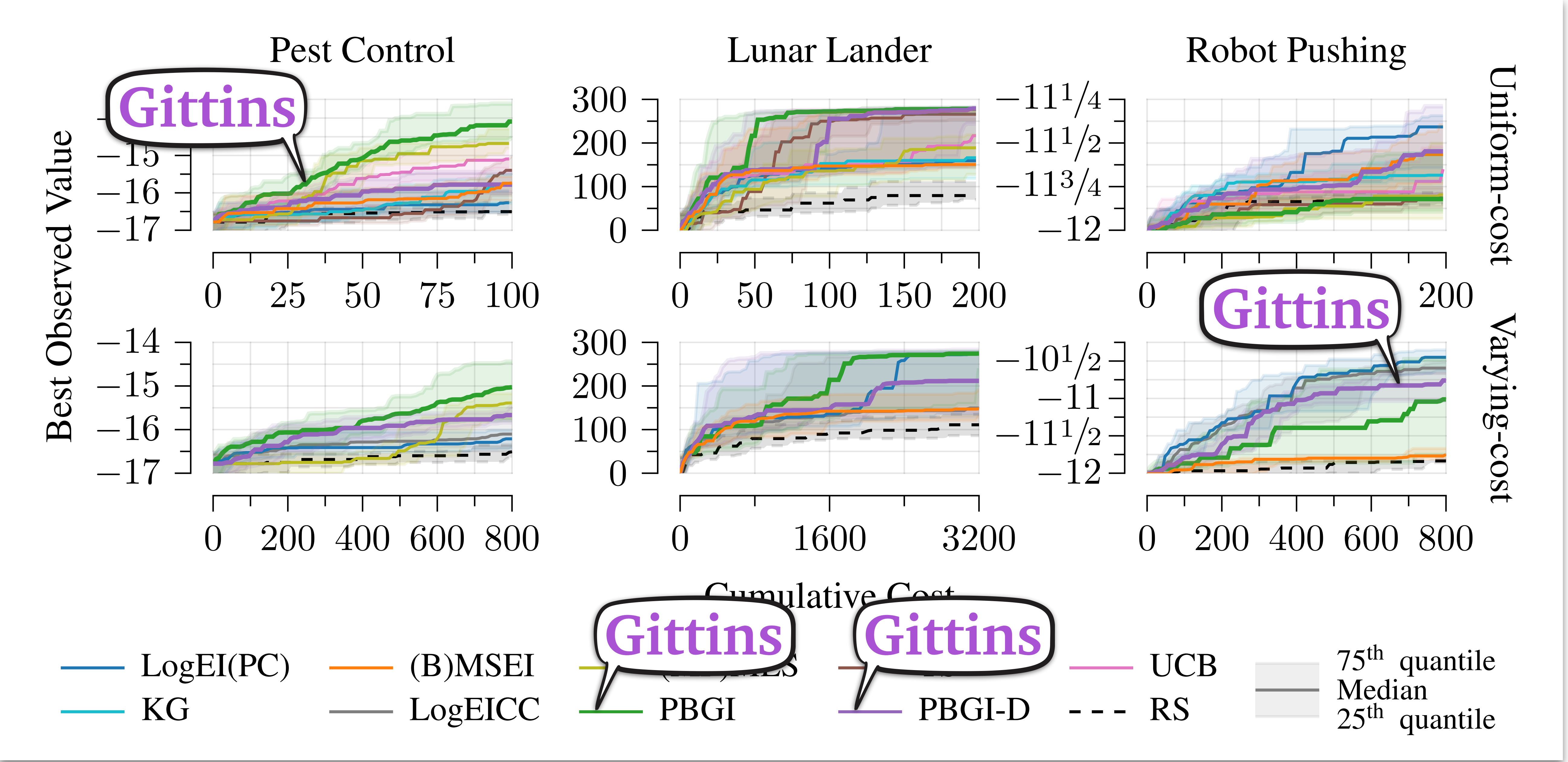


Qian Xie

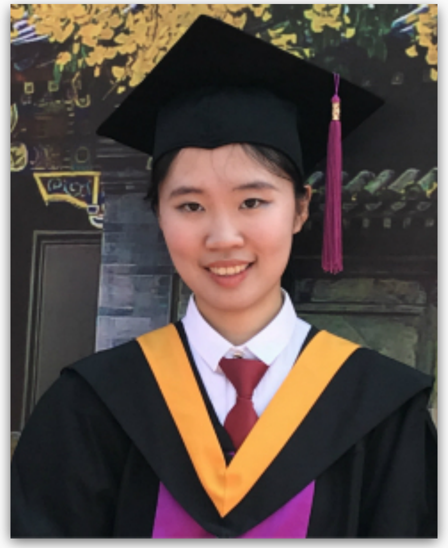
Alex Terenin

Raul Astudillo

Peter Frazier



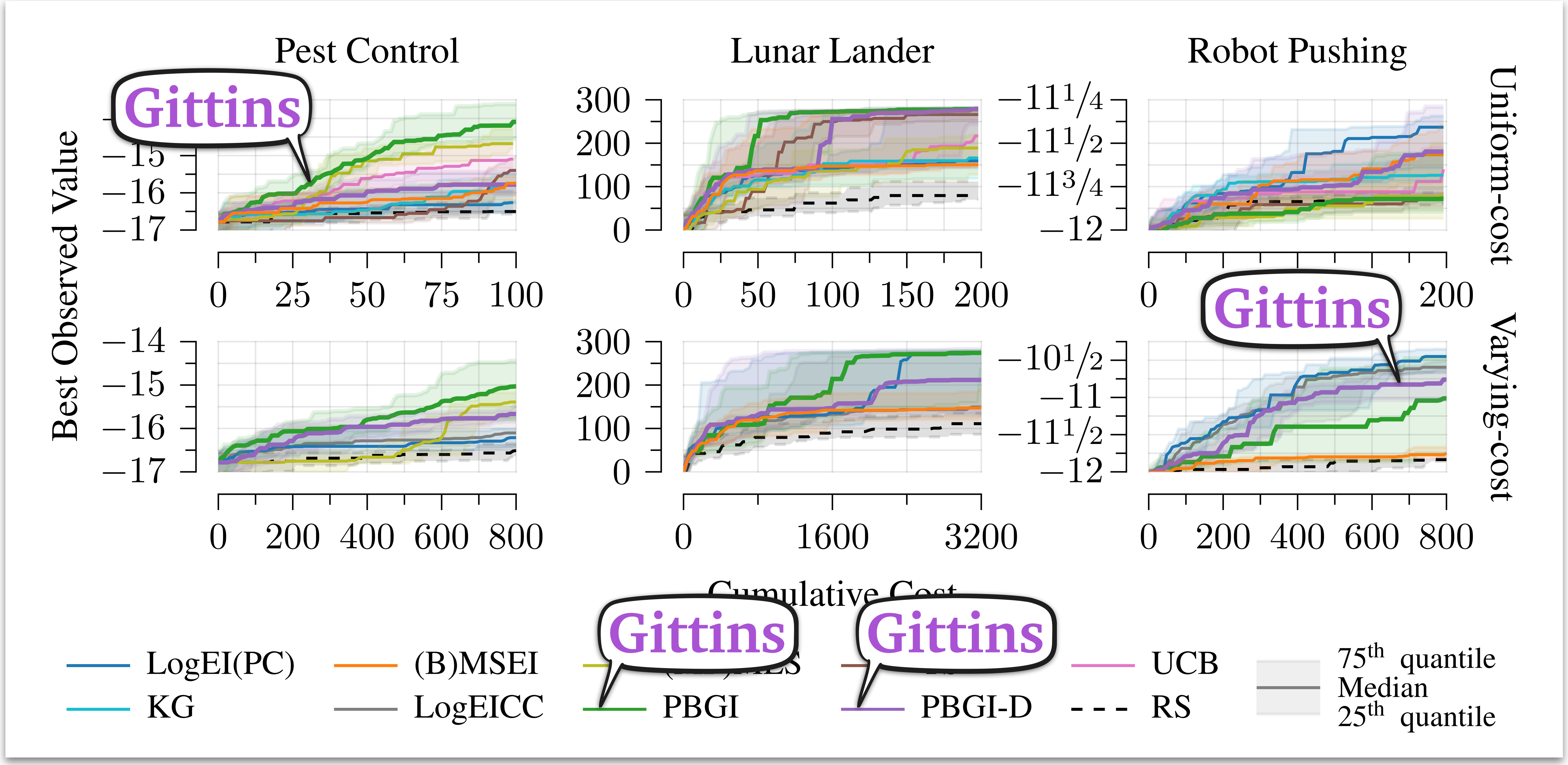
BayesOpt empirical results


Qian Xie


Alex Terenin


Raul Astudillo


Peter Frazier

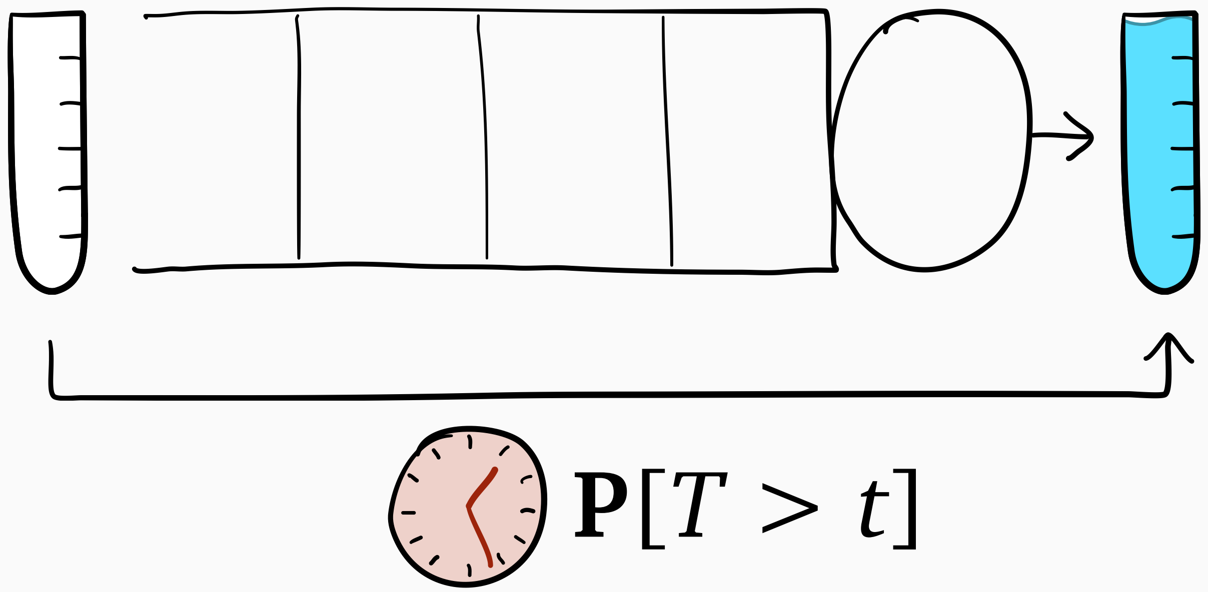


See our NeurIPS 2024 paper [Xie et al., 2024]

Two new Gittins applications

Tail scheduling

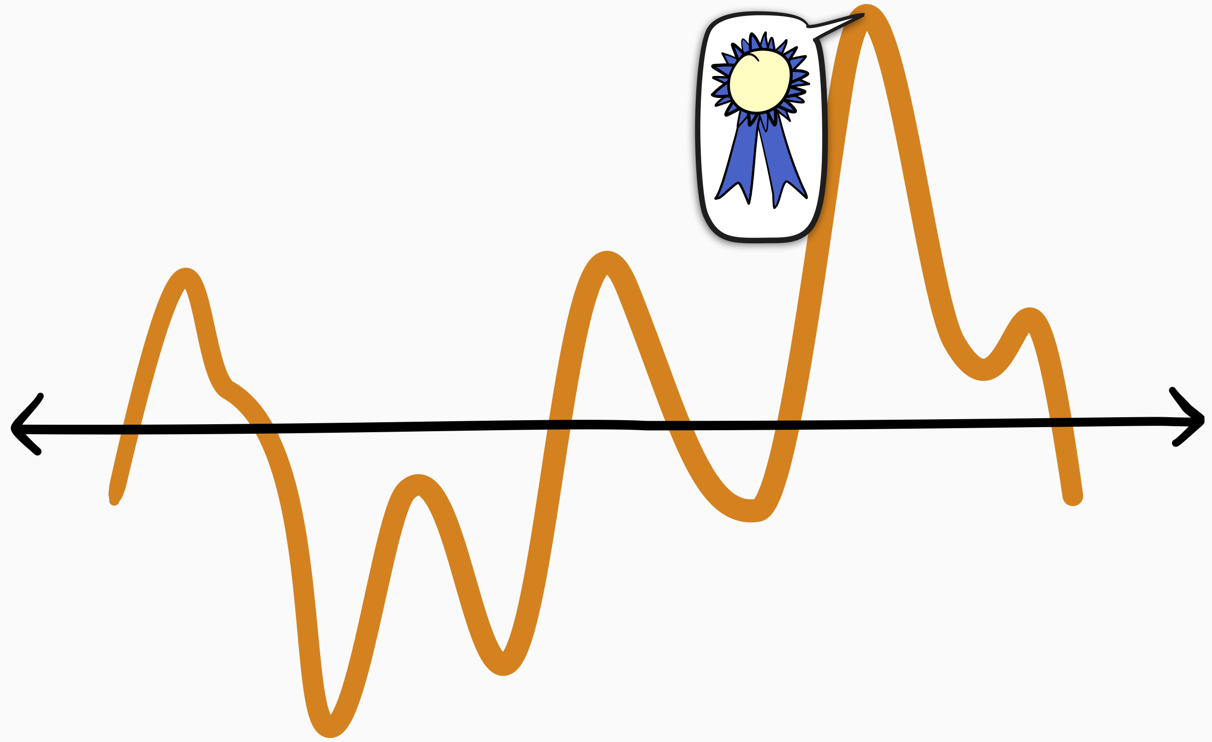
(in single-server queues)



Goal: minimize probability of very long response time

BayesOpt

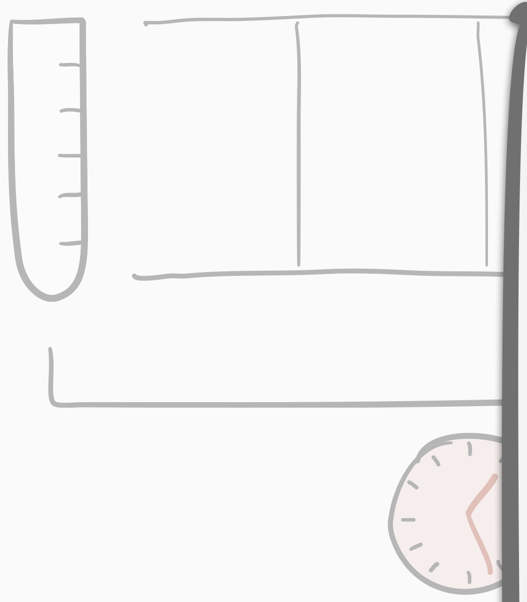
(Bayesian optimization)



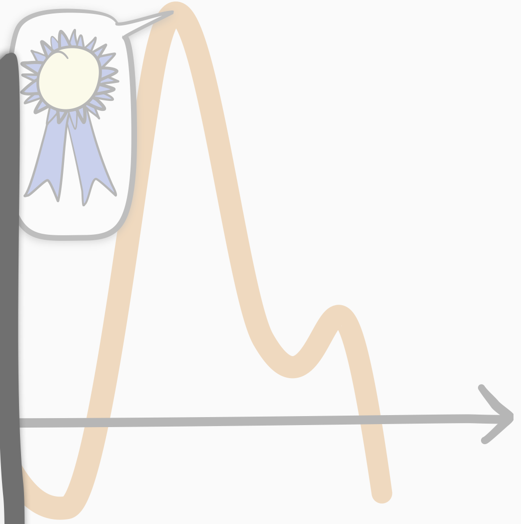
Goal: find large function value with few function evaluations

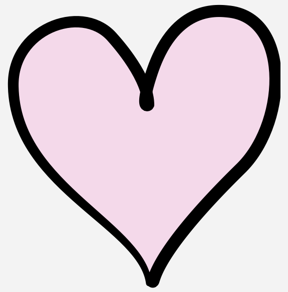
Two new **Gittins** applications

Tail scheduling
(in single-server queues)



BayesOpt
(Bayesian optimization)





Potential for *direct impact*

Goal: mini

of very long

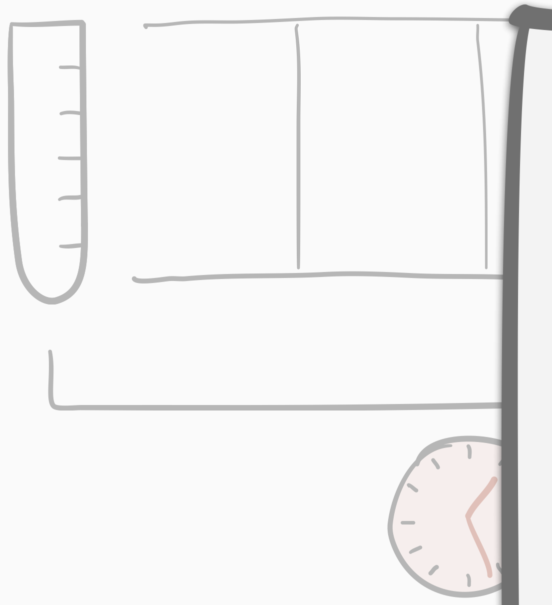
unction value

evaluations

Two new Gittins applications

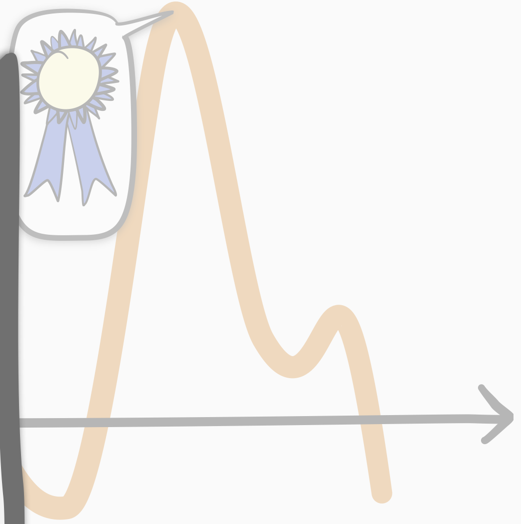
Tail scheduling

(in single-server queues)



BayesOpt

(Bayesian optimization)



Potential for *direct impact*

Out of scope of classic theory

Goal: mini

of very long

unction value

evaluations

This talk

This talk

What is the **Gittins index**?

This talk

- What is the **Gittins index**?
- Why is **Gittins** optimal?

This talk

What is the **Gittins index**?

Why is **Gittins** optimal?

What is *(and isn't)* covered by classical **Gittins** theory?

This talk

What is the **Gittins index**?

Why is **Gittins** optimal?

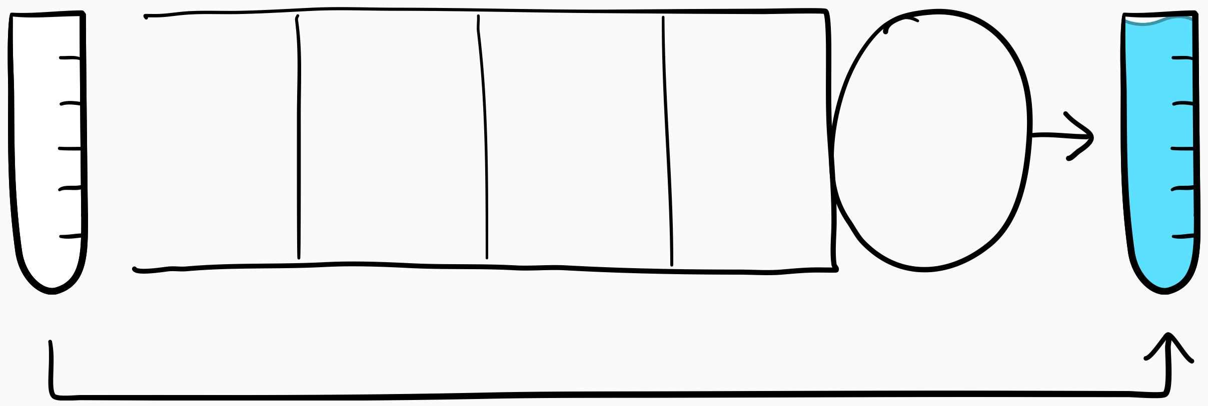
What is *(and isn't)* covered by classical **Gittins** theory?

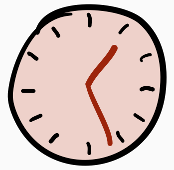
How might we apply **Gittins** *beyond* the classical theory?

Two new Gittins applications

Tail scheduling

(in single-server queues)

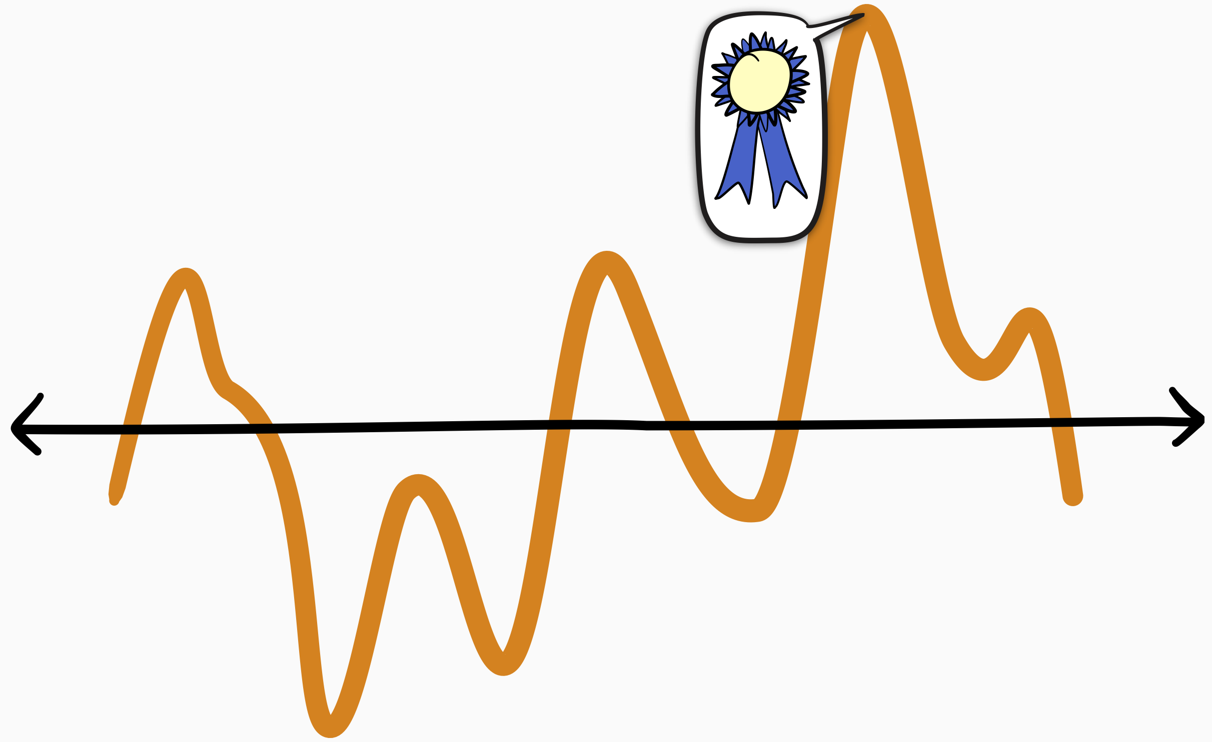


 $\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

BayesOpt

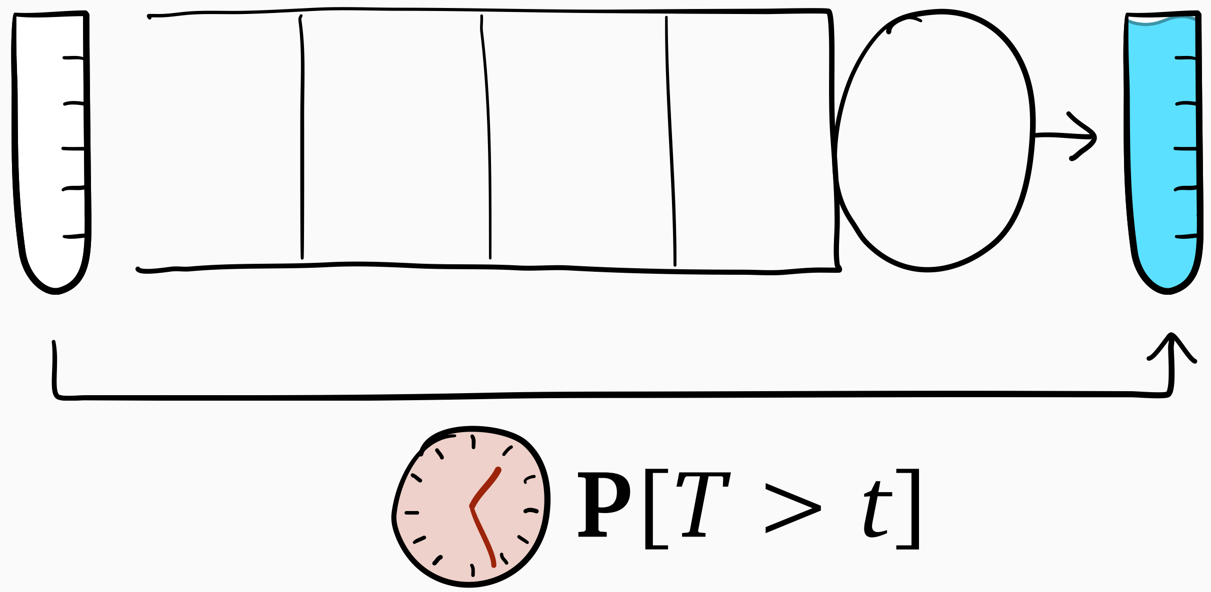
(Bayesian optimization)



Goal: find large function value with few function evaluations

Two *classical* **Gittins** applications

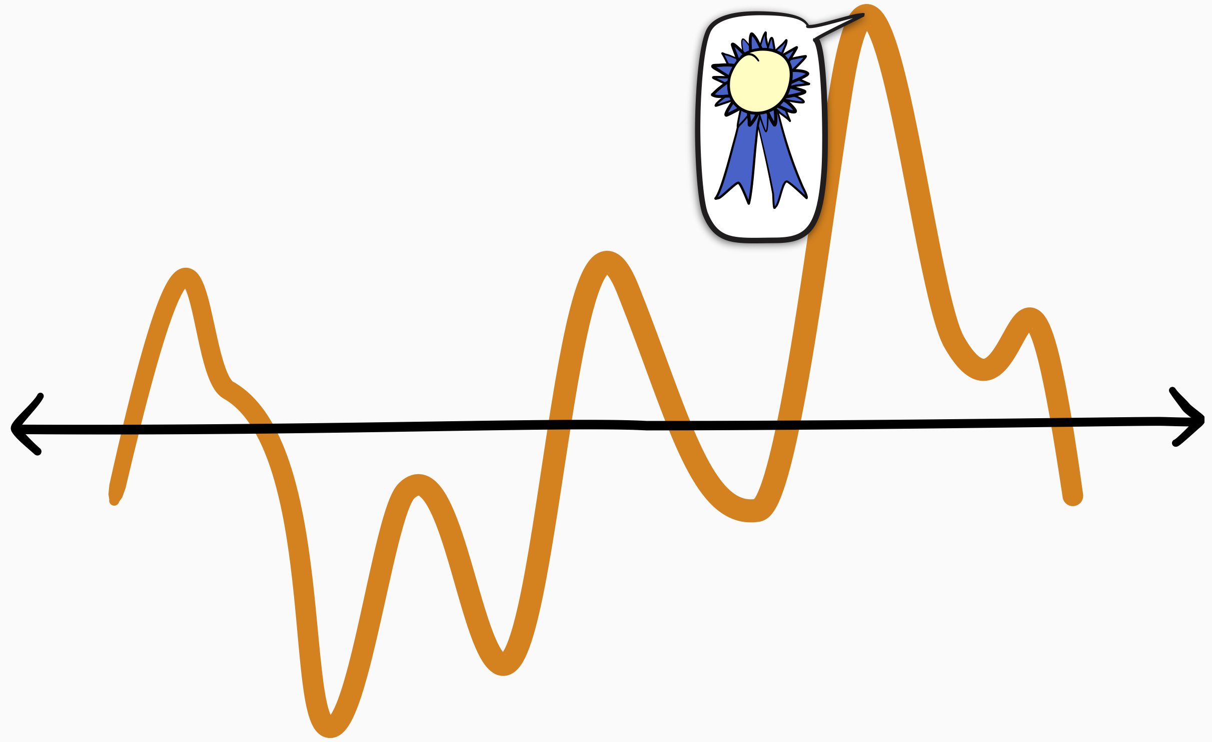
Tail scheduling
(in single-server queues)



$\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

BayesOpt
(Bayesian optimization)

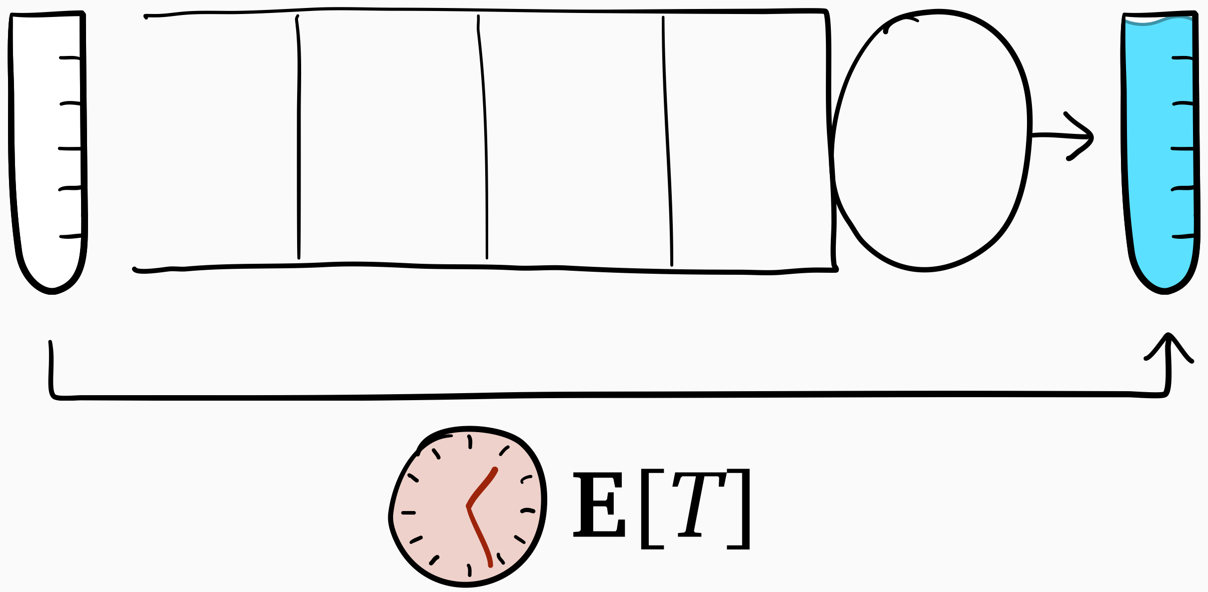


Goal: find large function value with few function evaluations

Two *classical* **Gittins** applications

Mean scheduling

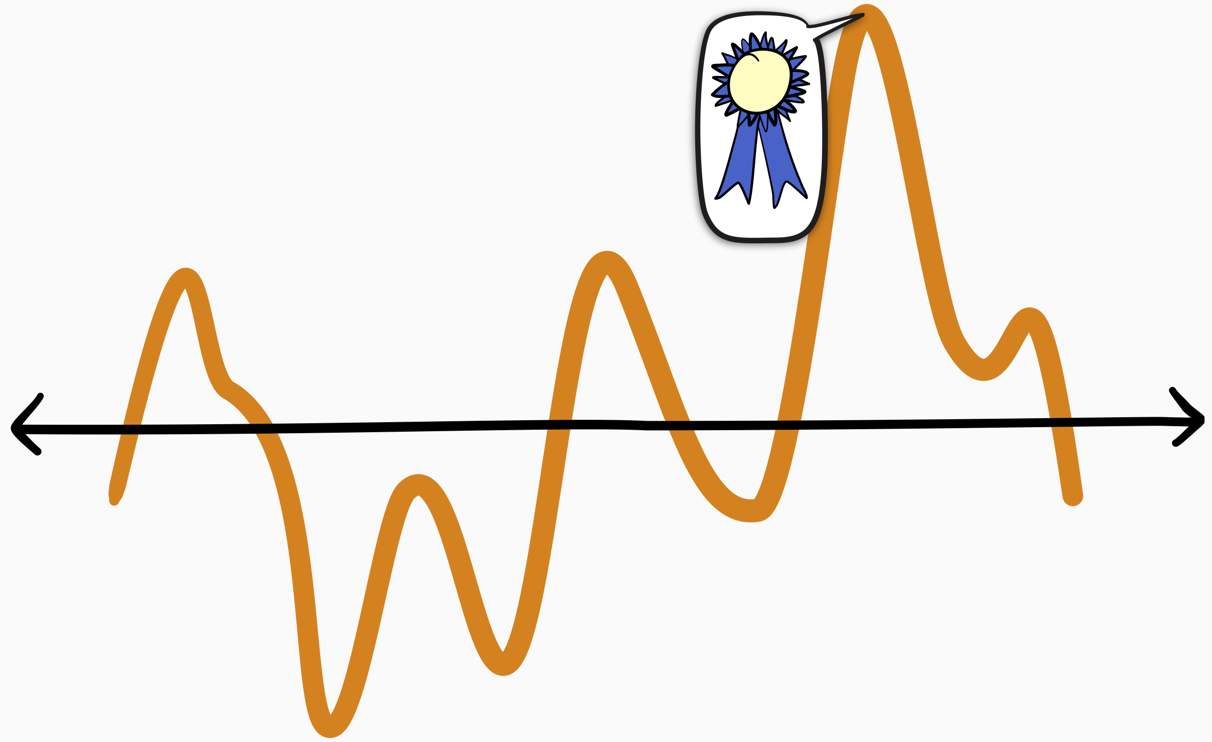
(in single-server queues)



Goal: minimize probability of very long response time

BayesOpt

(Bayesian optimization)

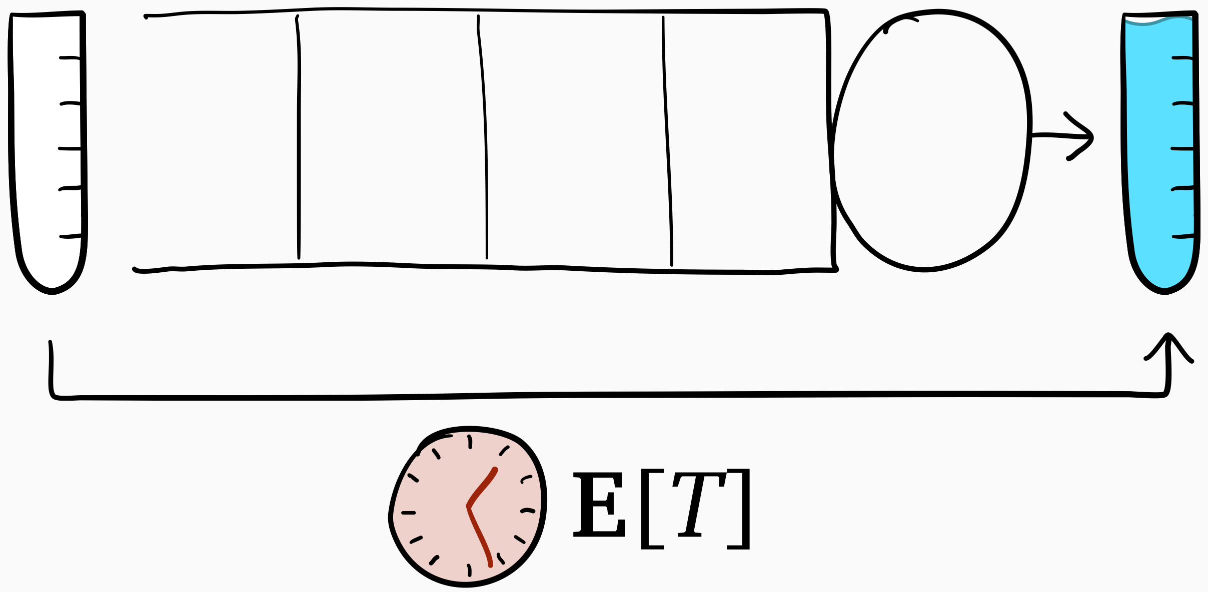


Goal: find large function value with few function evaluations

Two *classical* **Gittins** applications

Mean scheduling

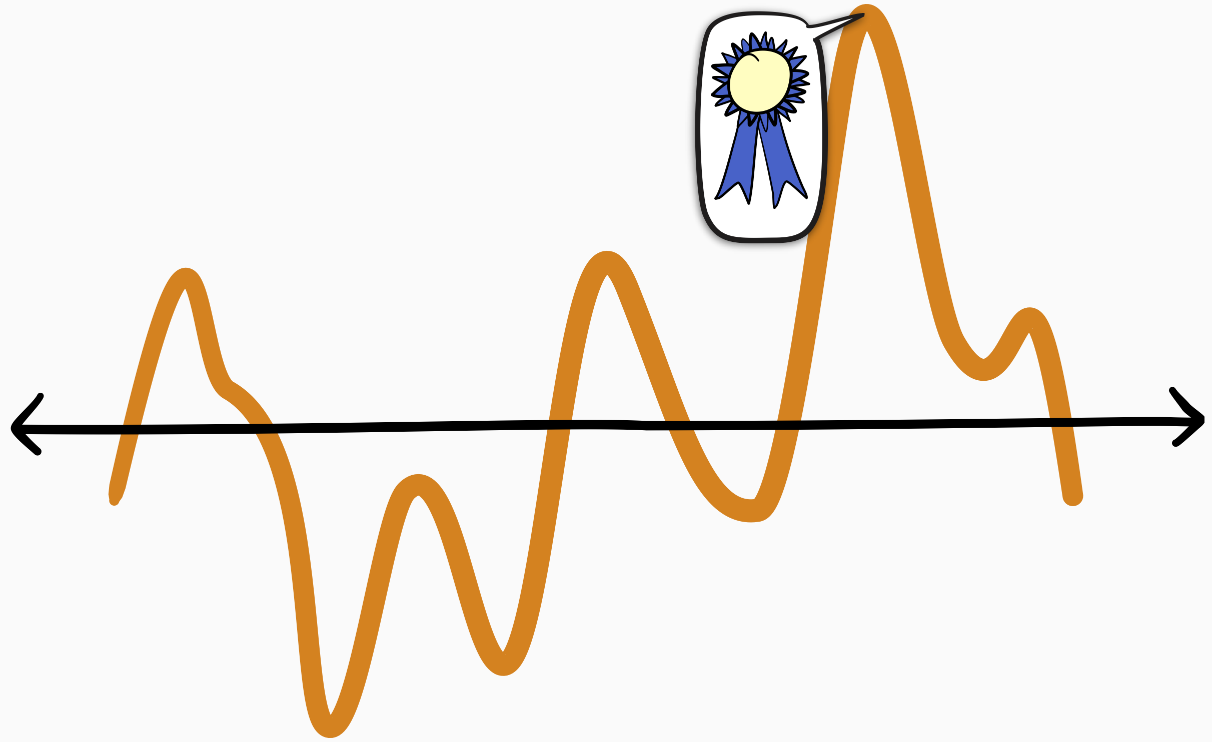
(in single-server queues)



Goal: minimize mean of the response time distribution

BayesOpt

(Bayesian optimization)

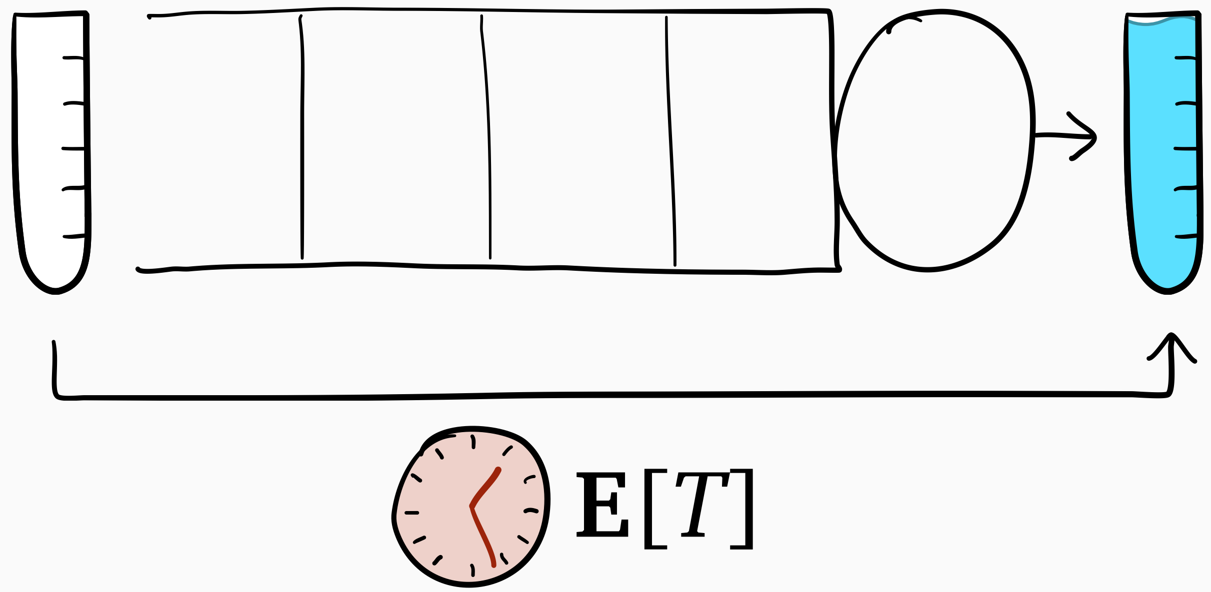


Goal: find large function value with few function evaluations

Two *classical* **Gittins** applications

Mean scheduling

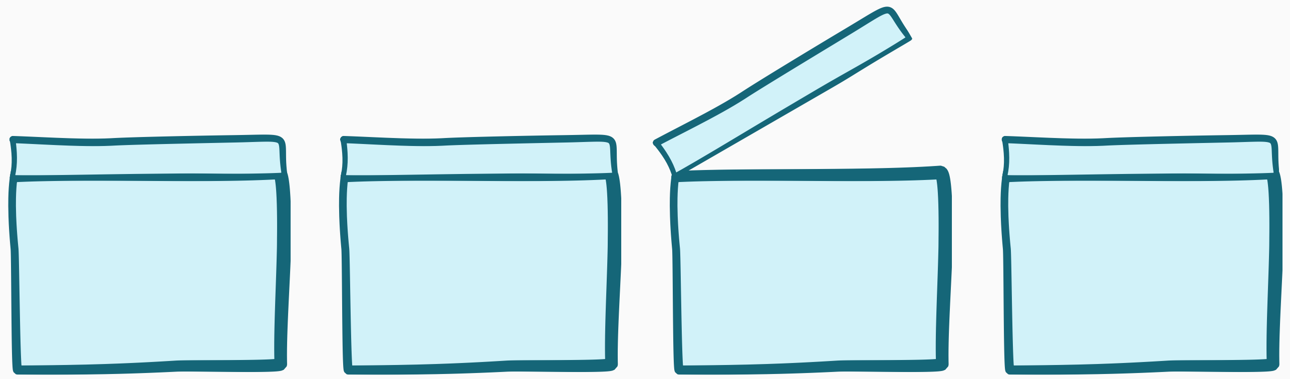
(in single-server queues)



Goal: minimize mean of the response time distribution

Pandora's box

(search for best alternative)

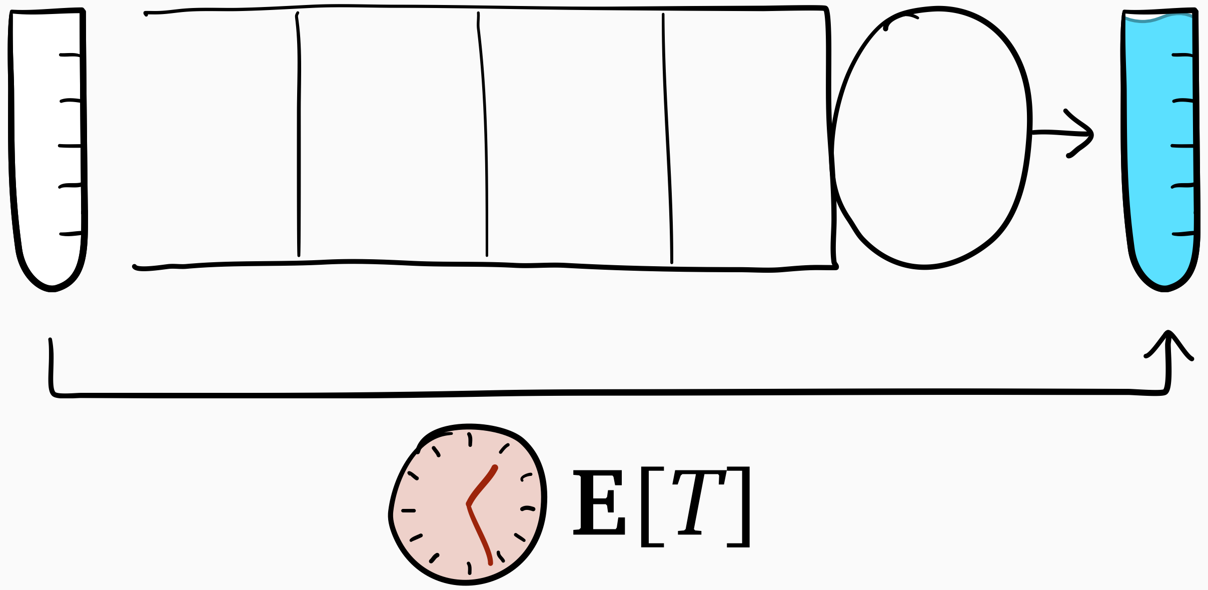


Goal: find large function value with few function evaluations

Two *classical* **Gittins** applications

Mean scheduling

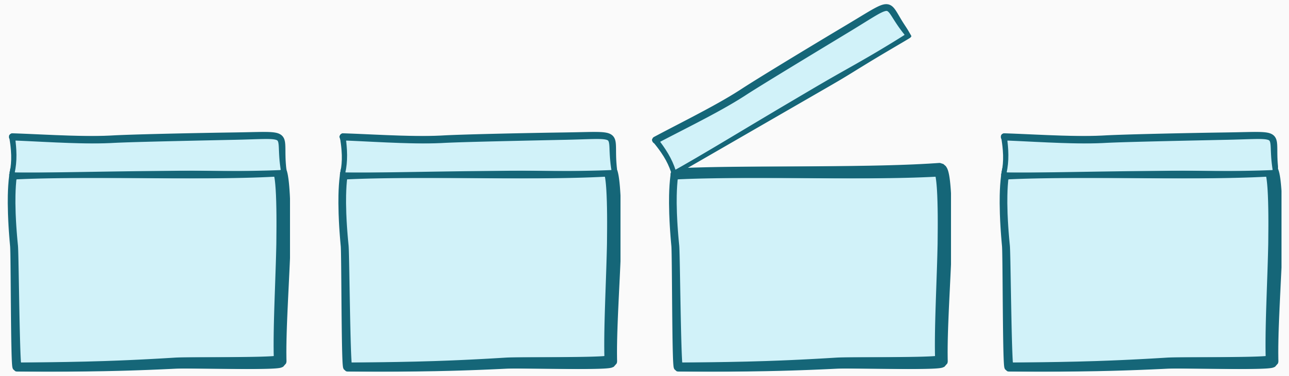
(in single-server queues)



Goal: minimize mean of the response time distribution

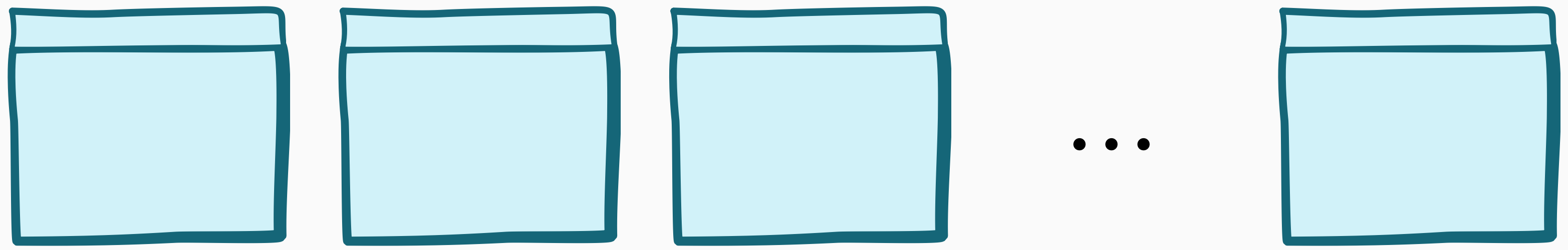
Pandora's box

(search for best alternative)

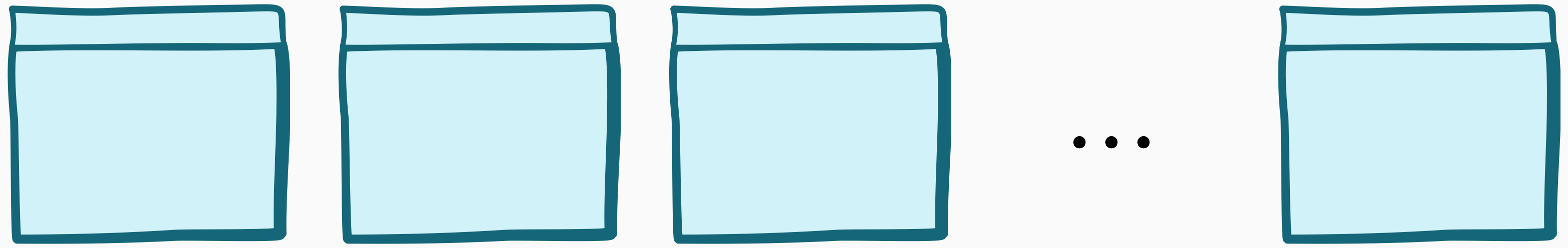


Goal: find box with large reward without opening too many

Pandora's box problem

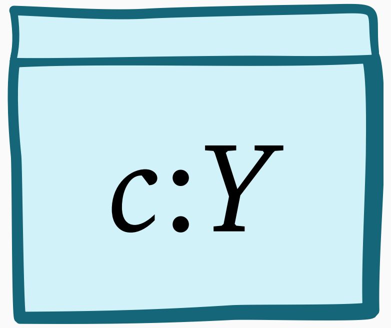


Pandora's box problem

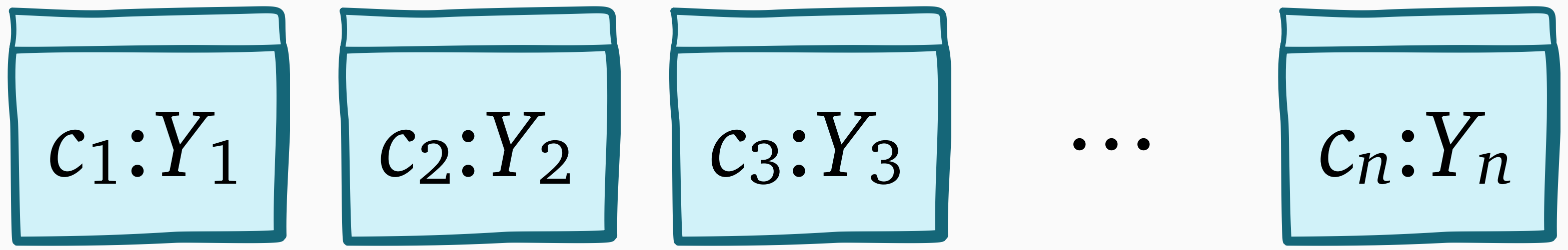


Each box:

- Opening cost c
- Hidden reward Y

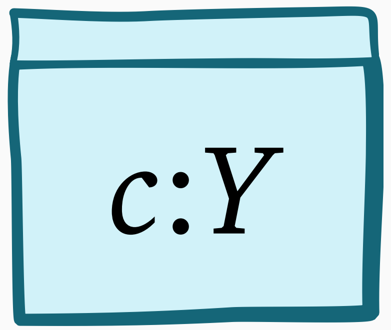


Pandora's box problem

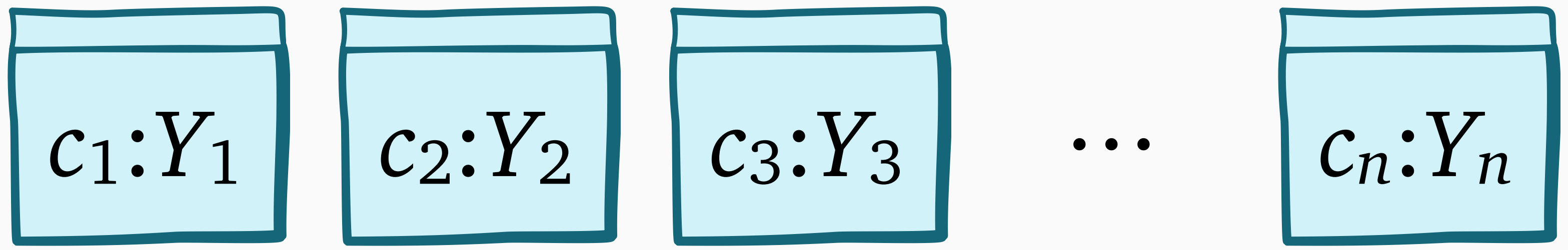


Each box:

- Opening cost c
- Hidden reward Y



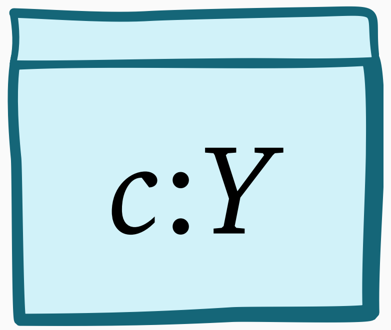
Pandora's box problem



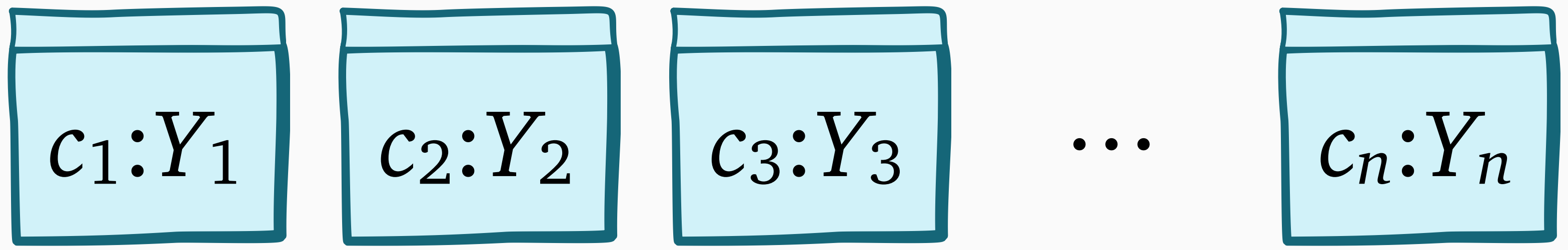
Each box:

- Opening cost c
- Hidden reward Y

independent



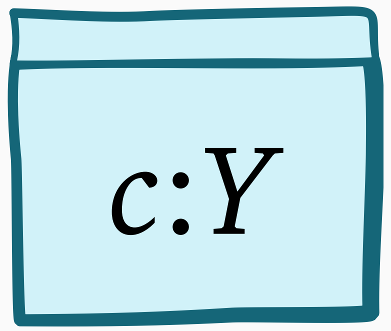
Pandora's box problem



Each box:

- Opening cost c
- Hidden reward Y

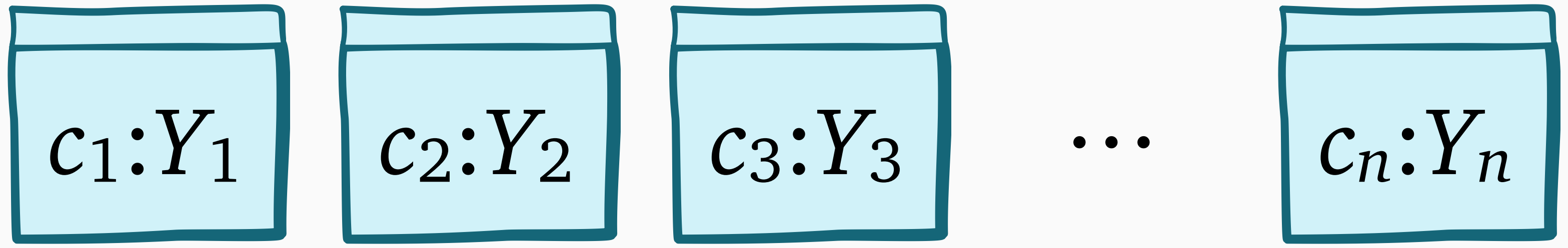
independent



Decision process:

- Open boxes one at a time
- Stop by selecting open box

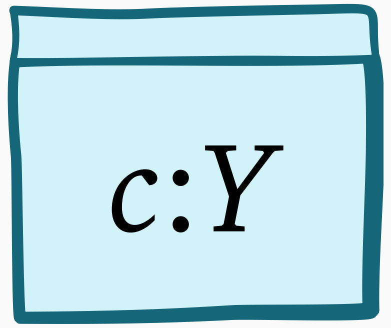
Pandora's box problem



Each box:

- Opening cost c
- Hidden reward Y

independent



Decision process:

- Open boxes one at a time
- Stop by selecting open box

Goal: maximize $\mathbf{E}\left[Y_{\text{selected}} - \sum_{i \text{ opened}} c_i\right]$

Pandora's box problem

$c_1 : Y_1$ $c_2 : Y_2$ $c_3 : Y_3$ $\cdots$ $c_n : Y_n$

Each box:

- Opening cost c
- Hidden reward Y

independent

$c : Y$

Decision process:

- Open boxes one at a time
- Stop by selecting open box

Which box to open?

Goal: maximize $\mathbf{E}\left[Y_{\text{selected}} - \sum_{i \text{ opened}} c_i\right]$

Pandora's box problem

$c_1{:}Y_1$ $c_2{:}Y_2$ $c_3{:}Y_3$ $\cdots$ $c_n{:}Y_n$

Each box:

- Opening cost c
- Hidden reward Y

independent

$c{:}Y$

Decision process:

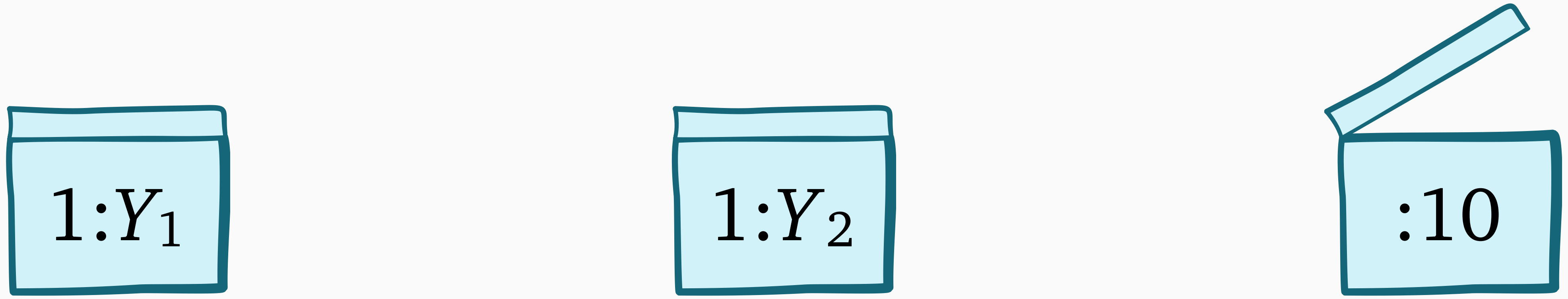
- Open boxes one at a time
- Stop by selecting open box

Goal: maximize $\mathbf{E}\left[Y_{\text{selected}} - \sum_{i \text{ opened}} c_i\right]$

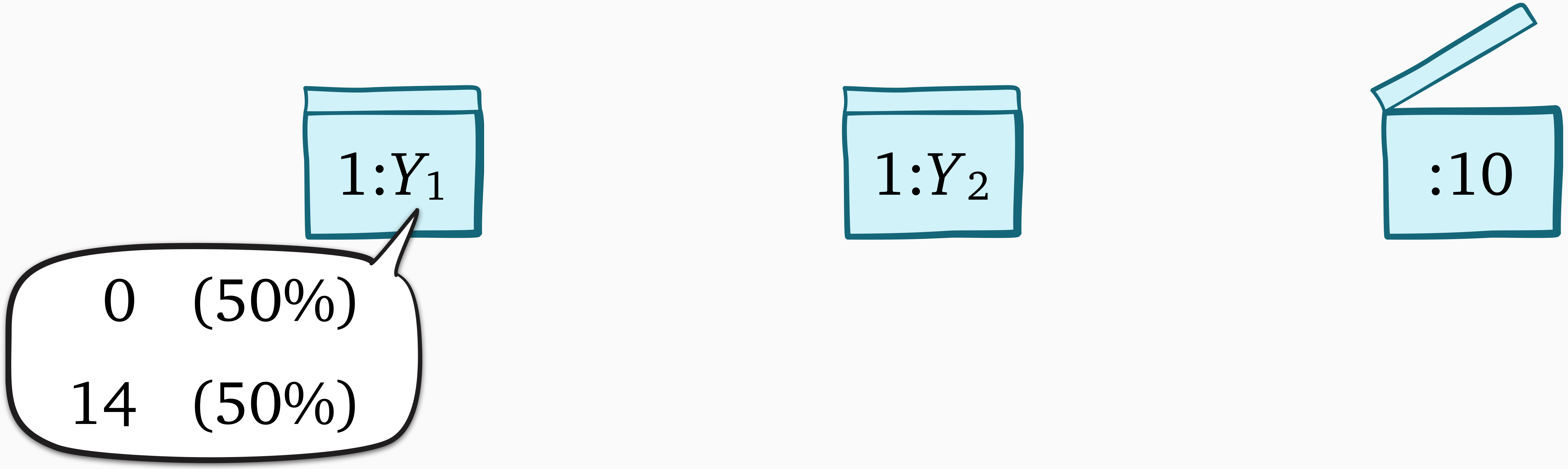
Which box to open?

Is it time to stop?

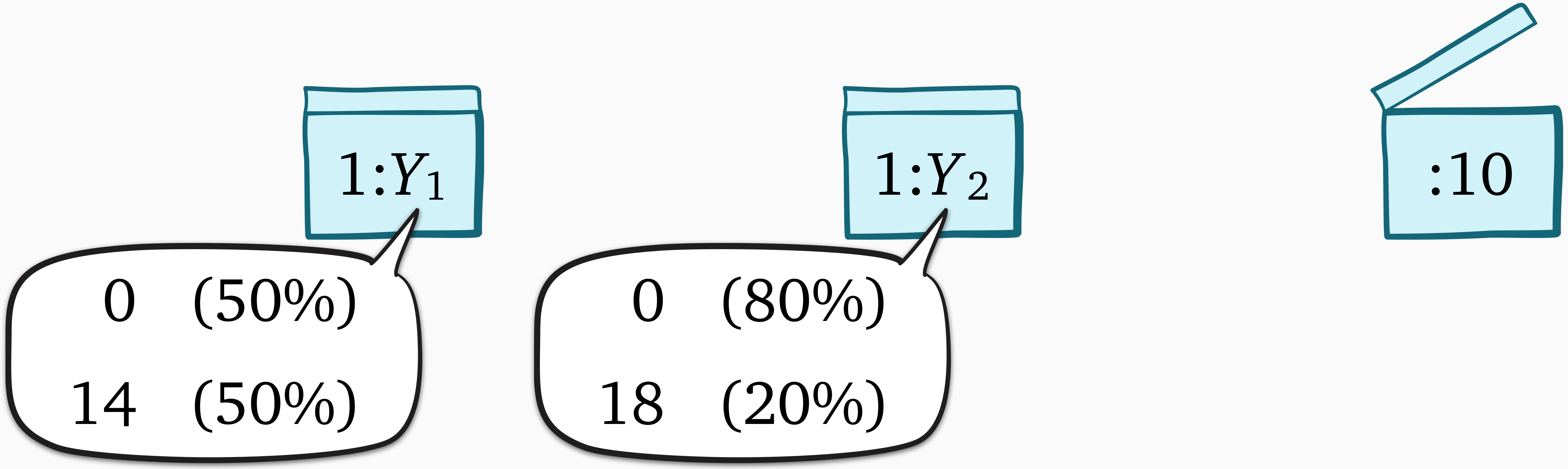
Why is **Pandora's box** hard?



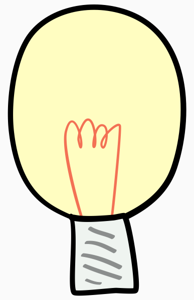
Why is Pandora's box hard?



Why is Pandora's box hard?

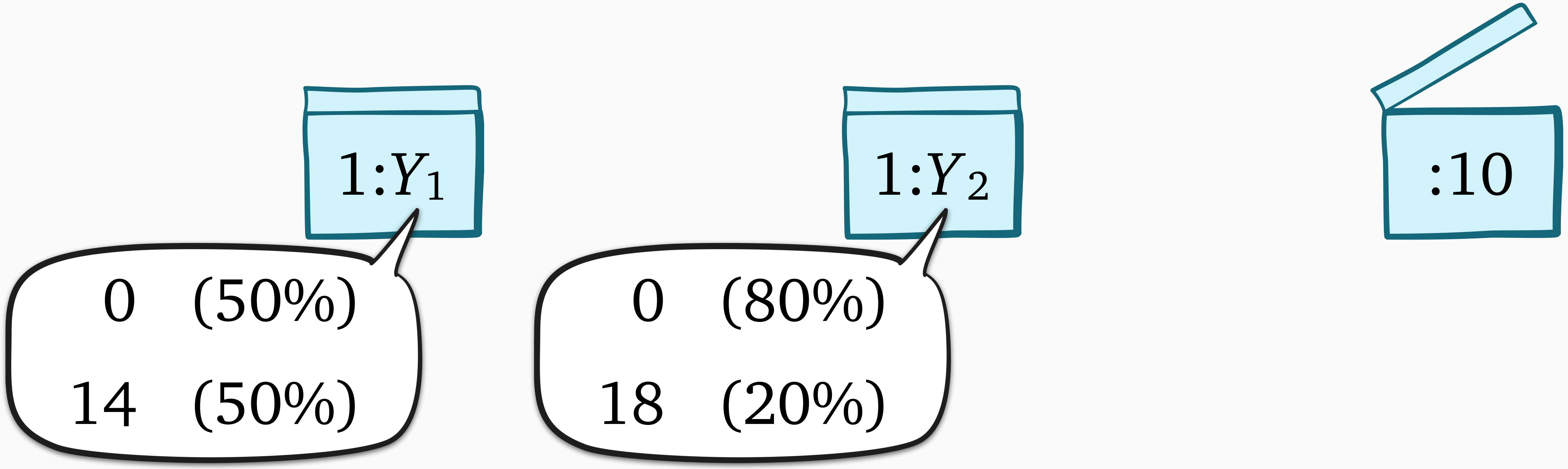


Why is Pandora's box hard?



Expected improvement of Y over r:

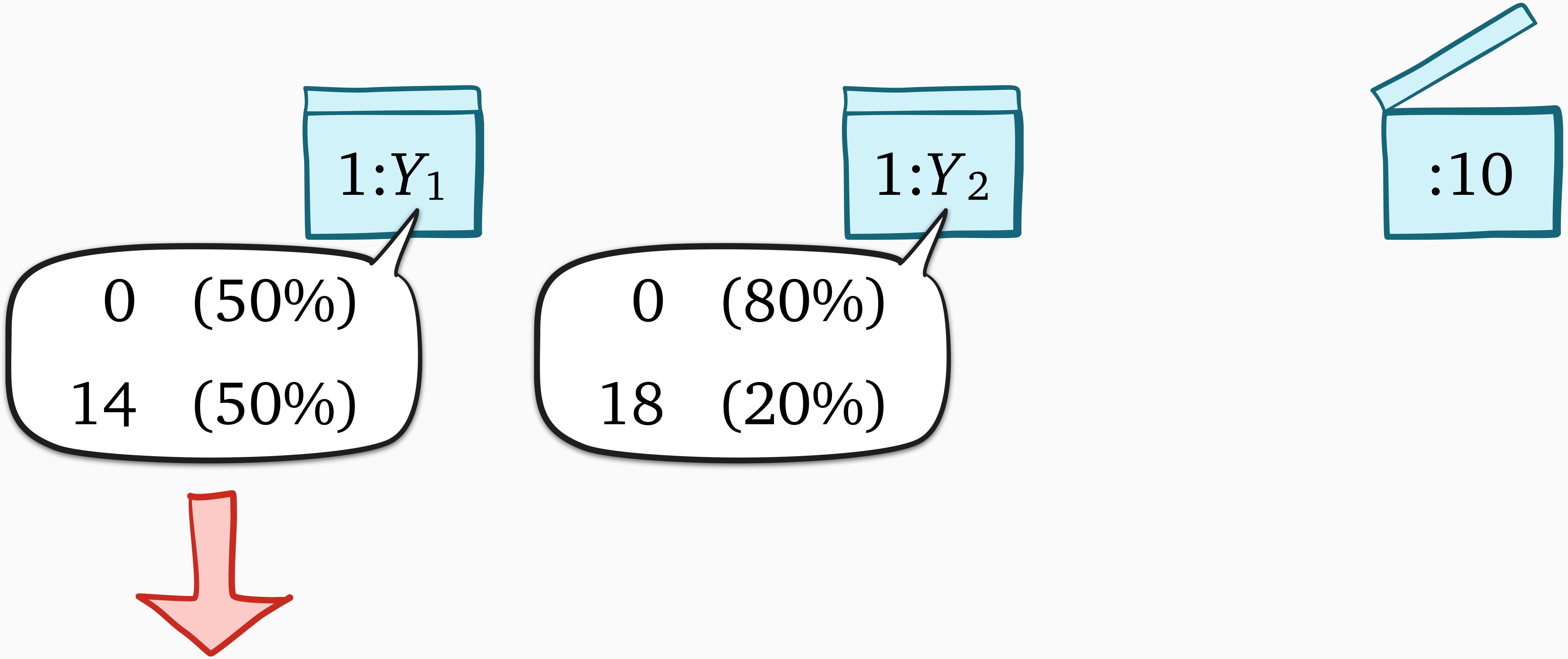
$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+]$$



Why is Pandora's box hard?

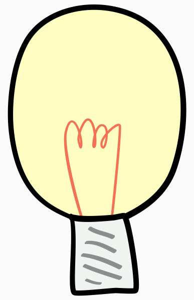
Expected improvement of Y over r:

$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+]$$



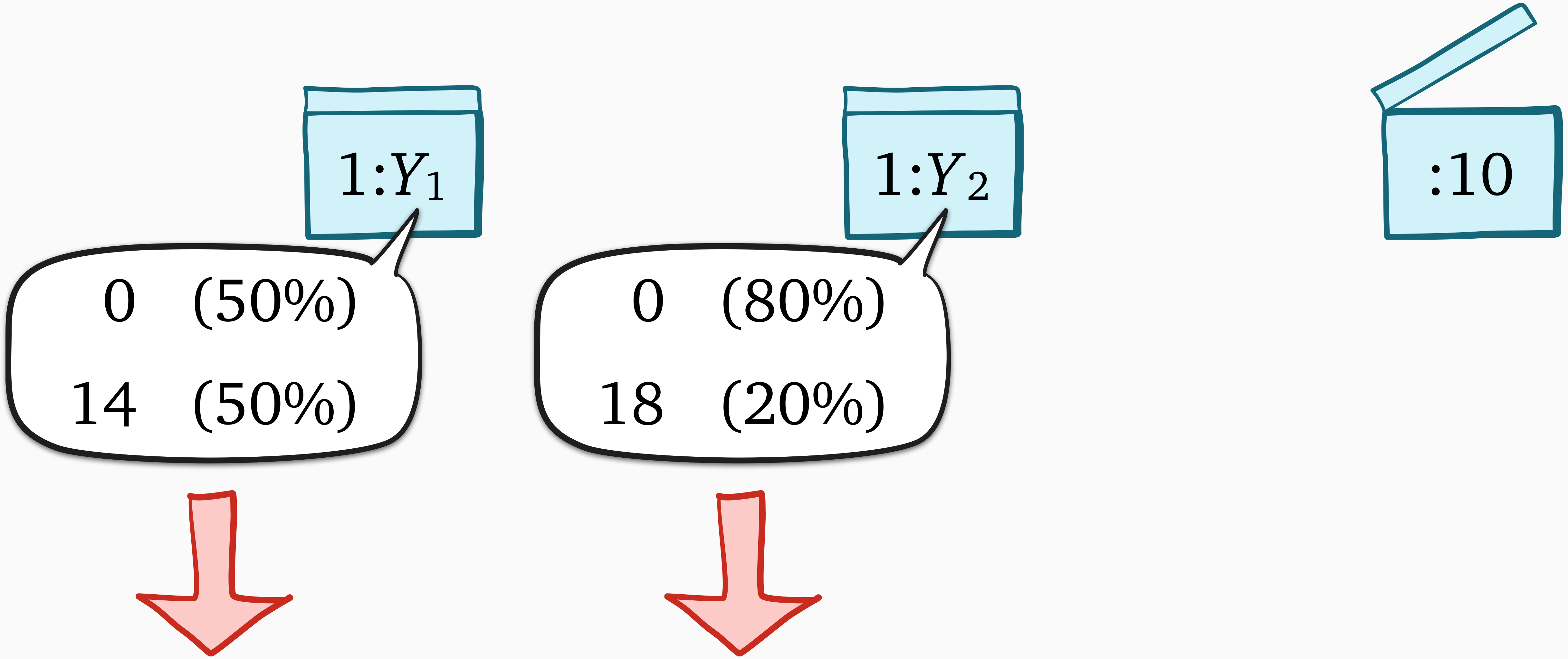
$\mathrm{EI}(Y_1, 10) = 2$

Why is **Pandora's box** hard?



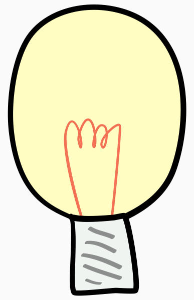
Expected improvement of Y over r:

$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+]$$



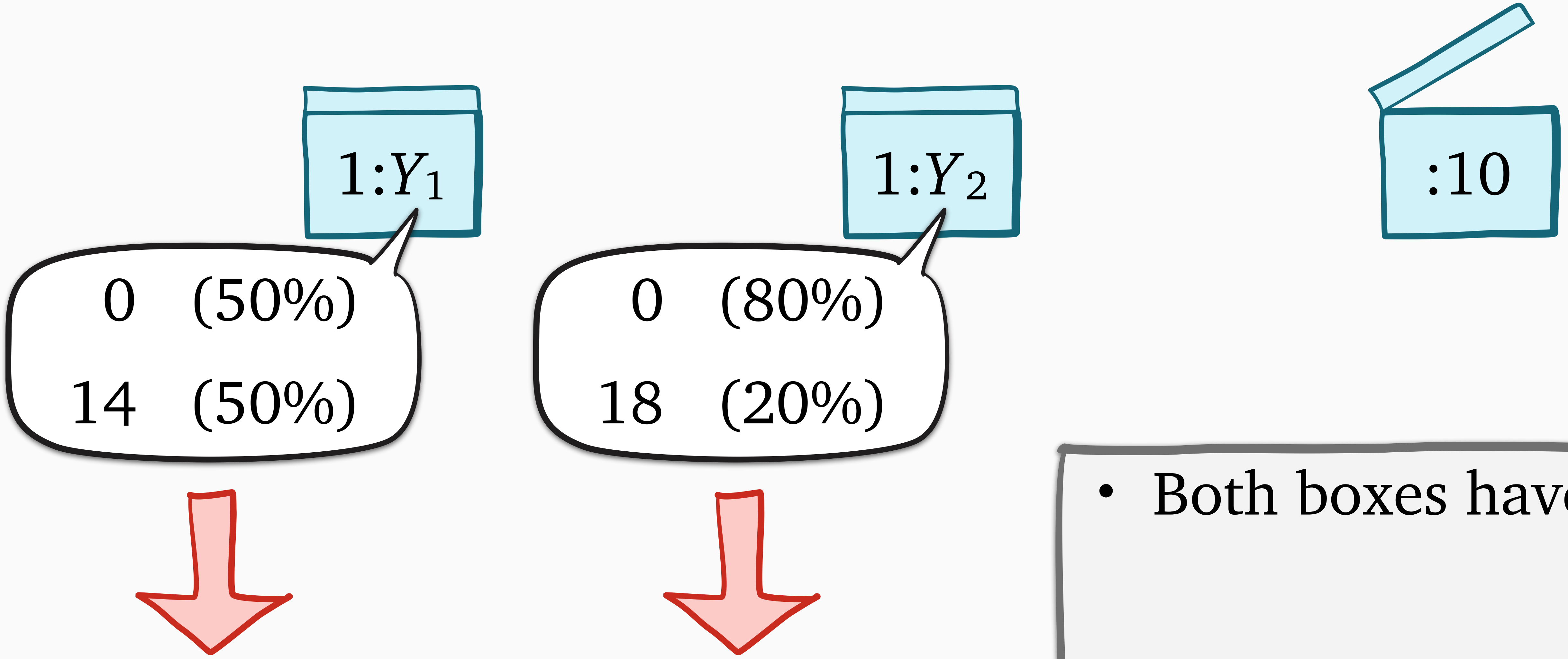
$\mathrm{EI}(Y_1, 10) = 2$ $\mathrm{EI}(Y_2, 10) = 1.6$

Why is **Pandora's box** hard?



Expected improvement of Y over r:

$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+]$$

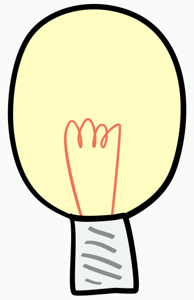


$\mathrm{EI}(Y_1, 10) = 2$

$\mathrm{EI}(Y_2, 10) = 1.6$

- Both boxes have $\mathrm{EI}(Y_i, 10) > c_i$

Why is **Pandora's box** hard?



Expected improvement of Y over r:

$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+]$$

$1{:}Y_1$

$1{:}Y_2$

$:10$

0	(50%)
14	(50%)

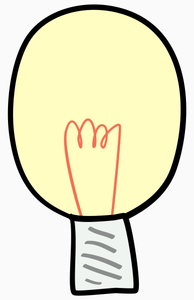
0	(80%)
18	(20%)

$\mathrm{EI}(Y_1, 10) = 2$

$\mathrm{EI}(Y_2, 10) = 1.6$

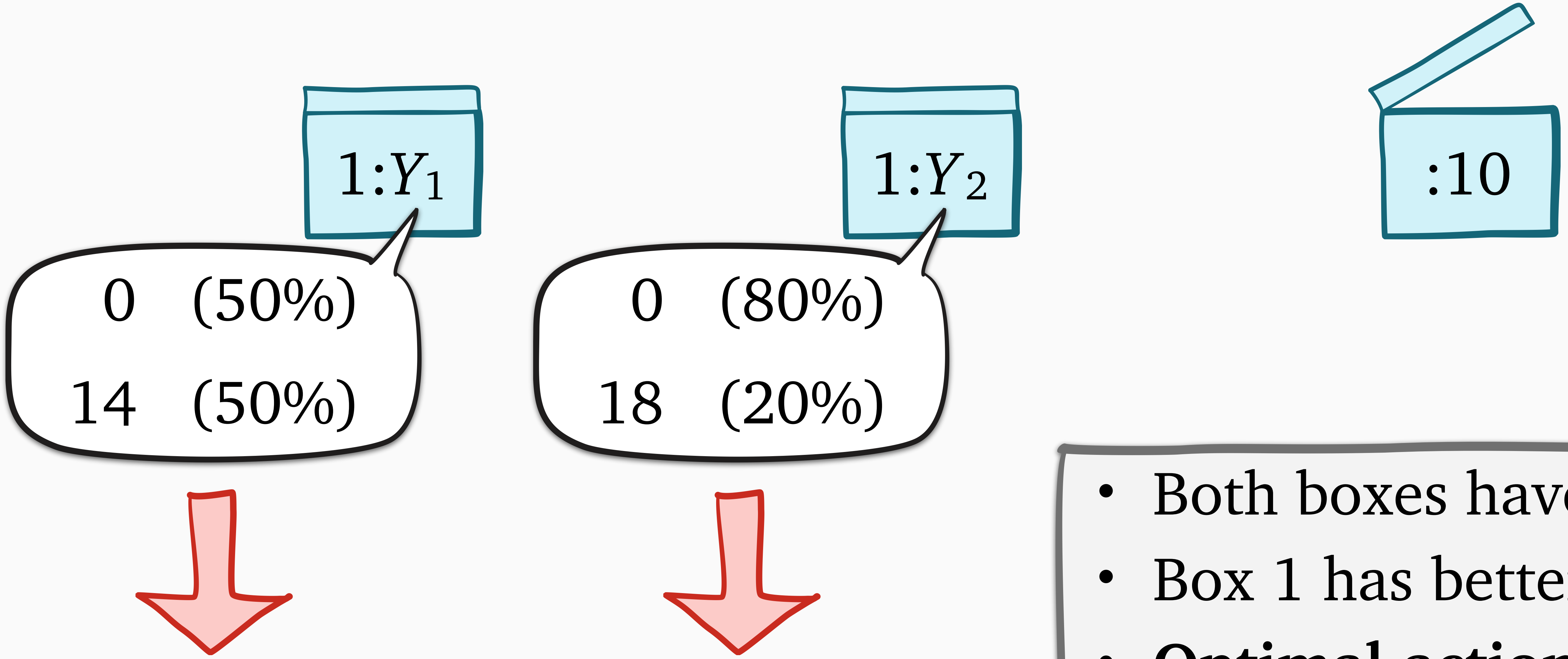
- Both boxes have $\mathrm{EI}(Y_i, 10) > c_i$
- Box 1 has better EI

Why is Pandora's box hard?



Expected improvement of Y over r:

$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+]$$



$\mathrm{EI}(Y_1, 10) = 2$

$\mathrm{EI}(Y_2, 10) = 1.6$

- Both boxes have $\mathrm{EI}(Y_i, 10) > c_i$
- Box 1 has better EI
- **Optimal action:** *open box 2!*

Optimal policy: **Gittins**

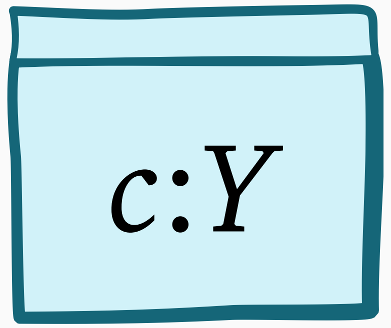
Optimal policy: **Gittins**

Step 1: *rate* each box separately

Step 2: *act* on box of best rating

Optimal policy: Gittins

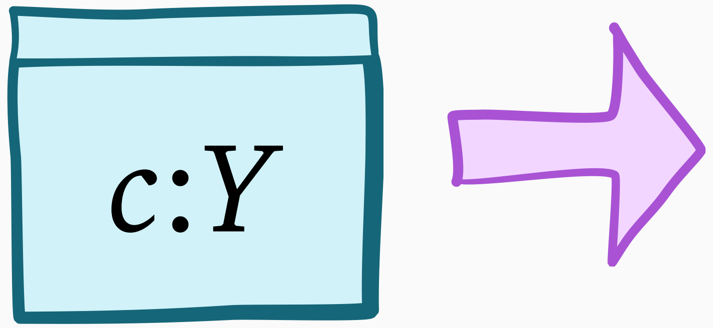
Step 1: *rate* each box separately



Step 2: *act* on box of best rating

Optimal policy: Gittins

Step 1: *rate* each box separately



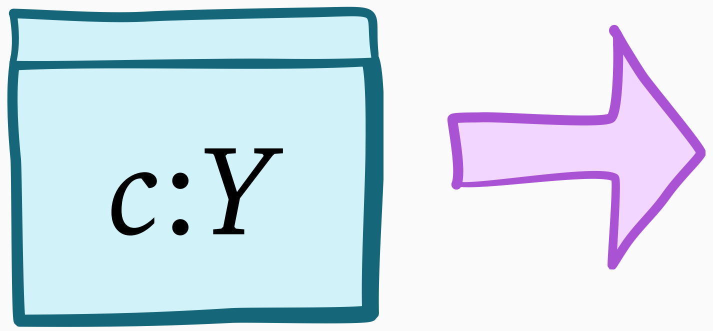
Gittins index:

$$g(c{:}Y)$$

Step 2: *act* on box of best rating

Optimal policy: Gittins

Step 1: *rate* each box separately



Gittins index:

$g(c{:}Y)$

higher is better

Step 2: *act* on box of best rating

Optimal policy: Gittins

Step 1: *rate* each box separately

$c{:}Y$

Gittins index:

$g(c{:}Y)$

higher is better

Step 2: *act* on box of best rating

$c_1{:}Y_1$ $c_2{:}Y_2$ $c_3{:}Y_3$ $\cdots$ $c_n{:}Y_n$

Optimal policy: Gittins

Step 1: *rate* each box separately

$c{:}Y$

Gittins index:

$g(c{:}Y)$

higher is better

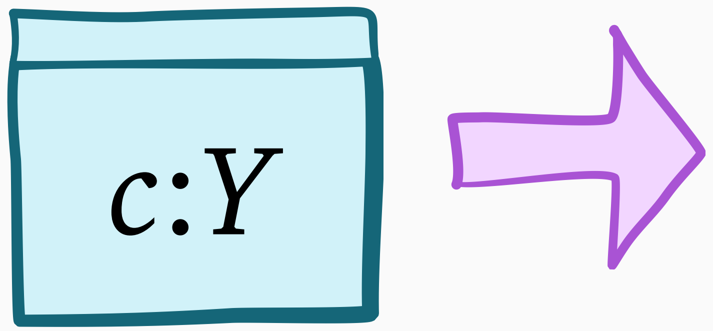
Step 2: *act* on box of best rating

$c_1{:}Y_1$ $c_2{:}Y_2$ $c_3{:}Y_3$ $\cdots$ $c_n{:}Y_n$

$g(c_1{:}Y_1)$ $g(c_2{:}Y_2)$ $g(c_3{:}Y_3)$ $\cdots$ $g(c_n{:}Y_n)$

Optimal policy: Gittins

Step 1: *rate* each box separately

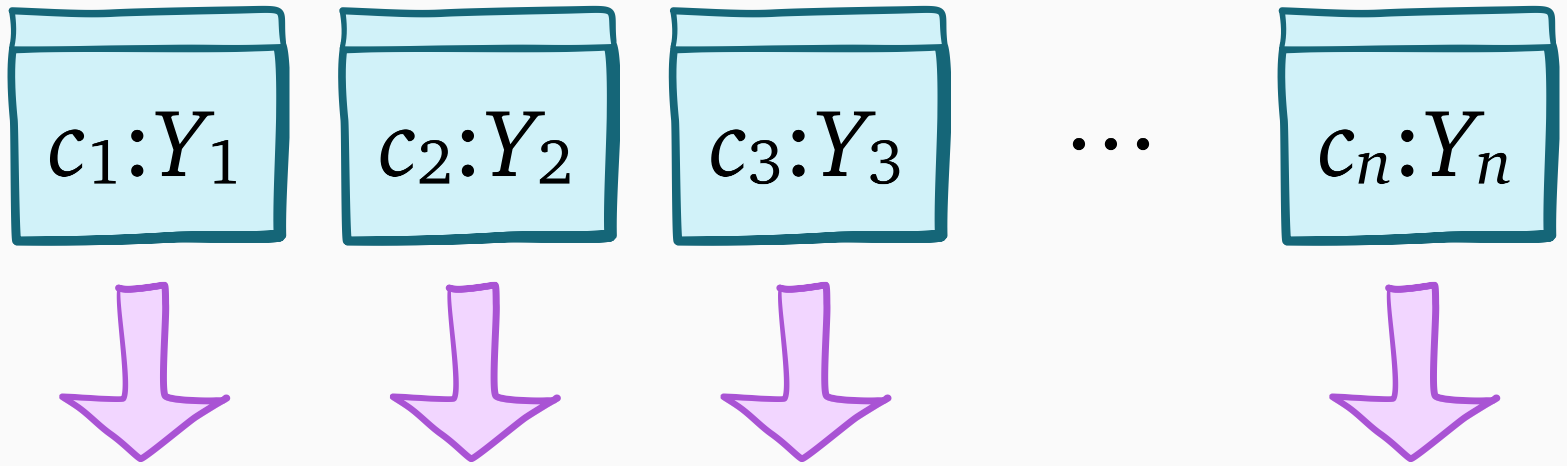


Gittins index:

$g(c{:}Y)$

higher is better

Step 2: *act* on box of best rating



$g(c_1{:}Y_1)$ $g(c_2{:}Y_2)$ $g(c_3{:}Y_3)$ $\cdots$ $g(c_n{:}Y_n)$

Gittins policy: if box of max Gittins index is…

- *closed:* open it
- *open:* select it

Optimal policy: Gittins

Step 1: *rate* each box separately

$c{:}Y$

Gittins index:

$g(c{:}Y)$

higher is better

Step 2: *act* on box of best rating

$c_1{:}Y_1$ $c_2{:}Y_2$ $c_3{:}Y_3$ $\cdots$ $c_n{:}Y_n$

$g(c_1{:}Y_1)$ $g(c_2{:}Y_2)$ $g(c_3{:}Y_3)$ $\cdots$ $g(c_n{:}Y_n)$

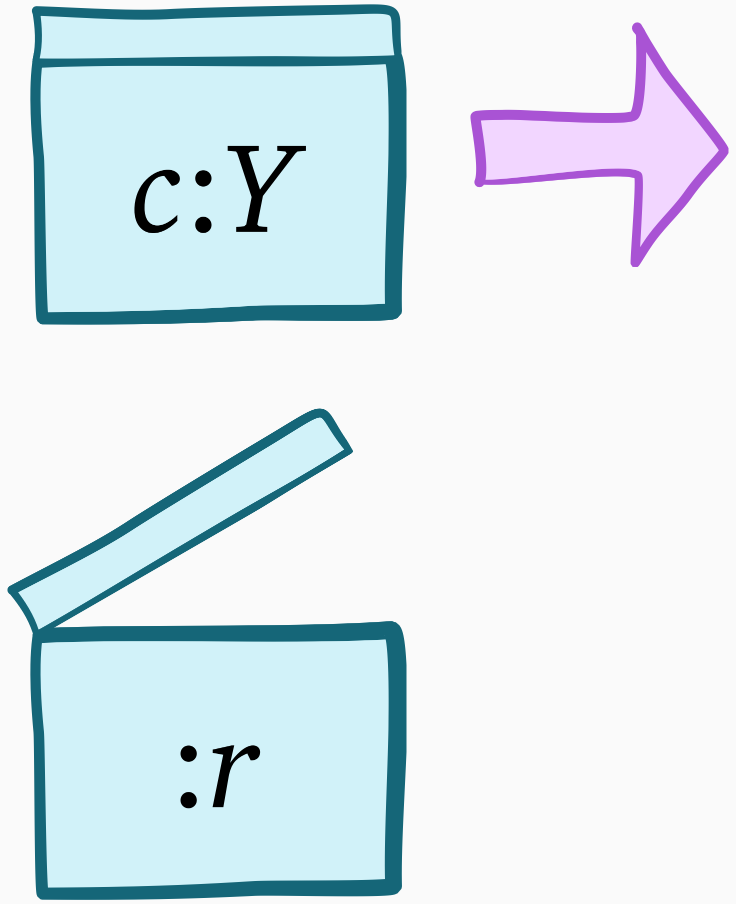
Gittins policy: if box of max Gittins index is…

- *closed:* open it
- *open:* select it

$\Big\}$ *act* on it

Optimal policy: Gittins

Step 1: *rate* each box separately

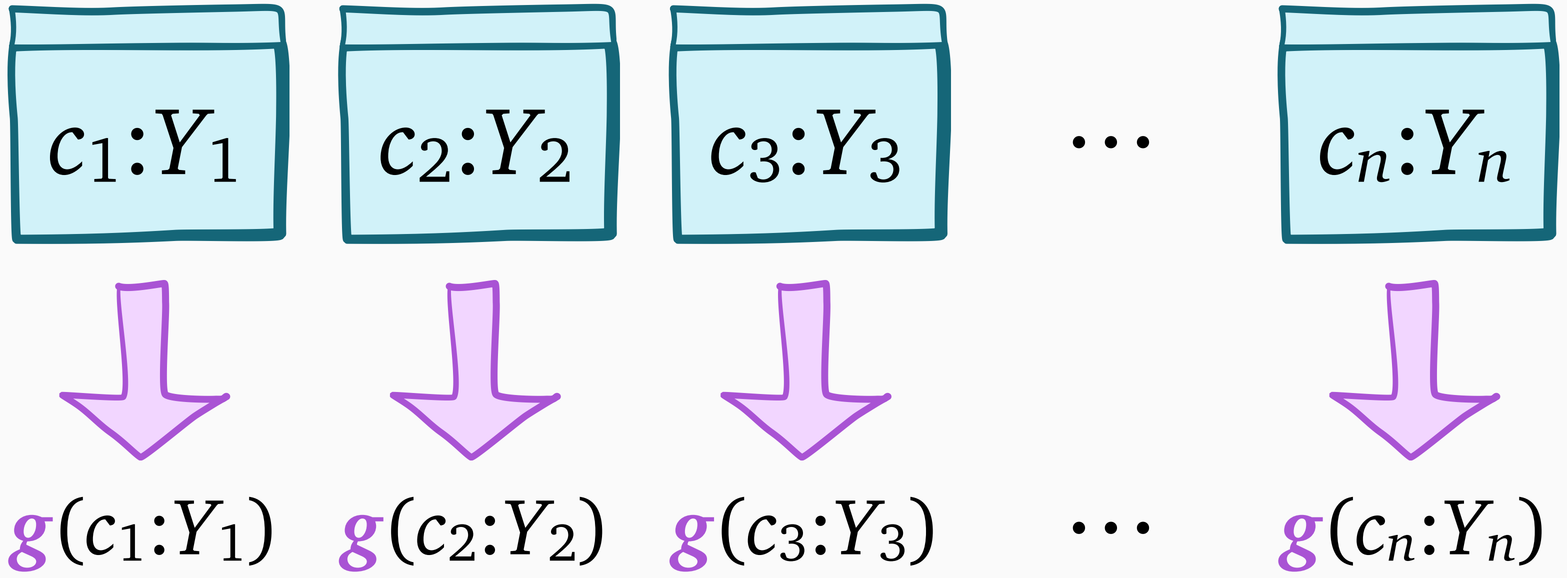


Gittins index:

$g(c{:}Y)$

higher is better

Step 2: *act* on box of best rating



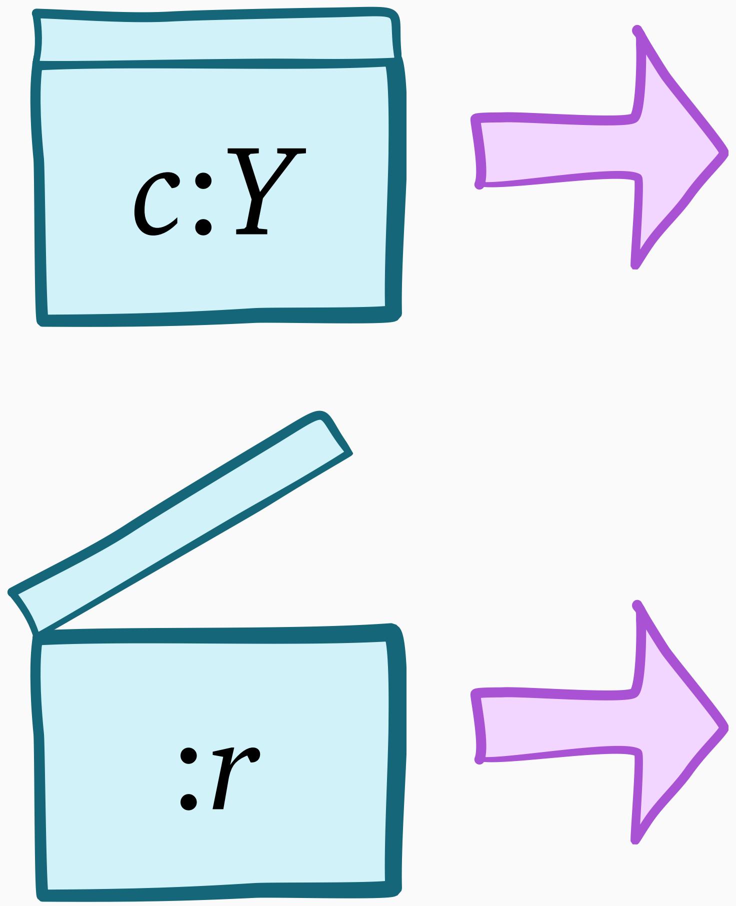
Gittins policy: if box of max Gittins index is…

- *closed:* open it
- *open:* select it

} *act* on it

Optimal policy: Gittins

Step 1: *rate* each box separately



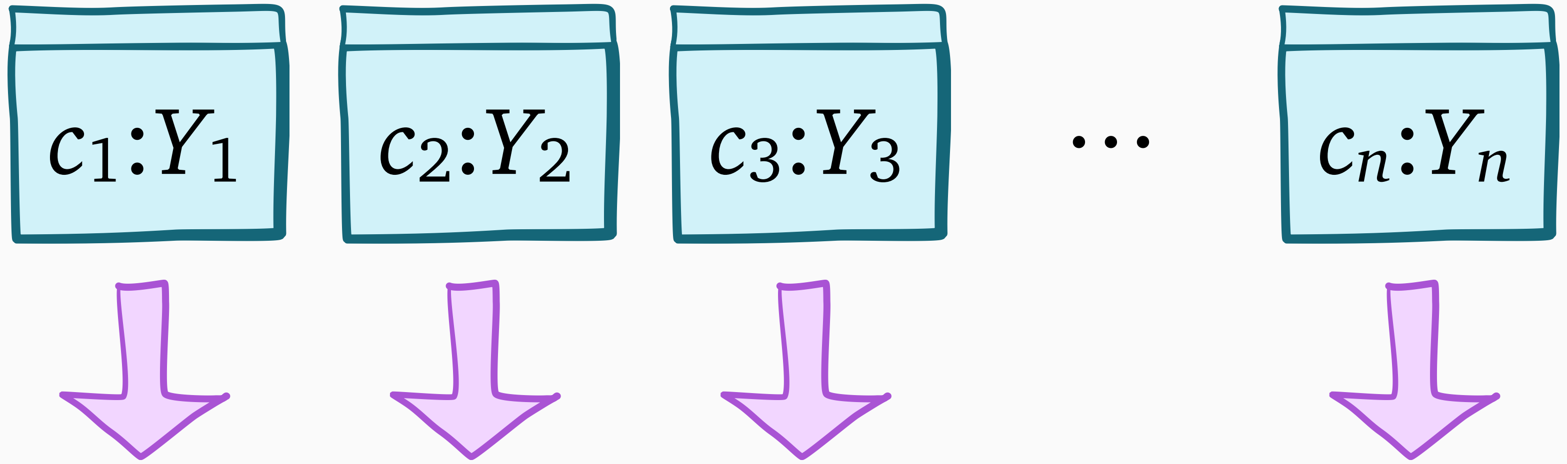
Gittins index:

$g(c{:}Y)$

higher is better

$g(:r) = r$

Step 2: *act* on box of best rating



$g(c_1{:}Y_1)$ $g(c_2{:}Y_2)$ $g(c_3{:}Y_3)$ $\cdots$ $g(c_n{:}Y_n)$

Gittins policy: if box of max Gittins index is…

- *closed:* open it
- *open:* select it

$\left.\begin{array}{l} \\ \\ \end{array}\right\}$ *act* on it

Optimal policy: Gittins

Step 1: *rate* each box separately

$c{:}Y$ → **Gittins index:** $g(c{:}Y)$

higher is better

$:r$ → $g(:r) = r$

Theorem: [Weitzman, 1979]
the Gittins policy is optimal

Step 2: *act* on box of best rating

$c_1{:}Y_1$ $c_2{:}Y_2$ $c_3{:}Y_3$ $\cdots$ $c_n{:}Y_n$

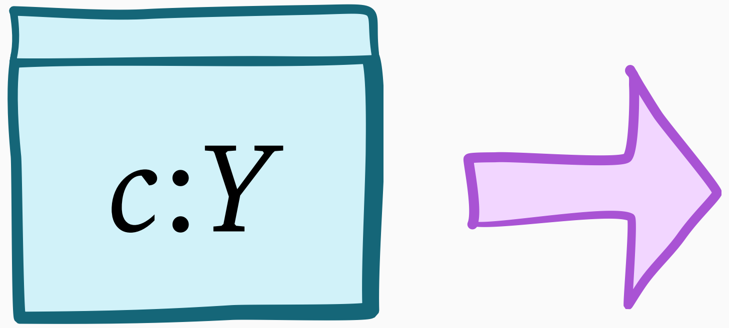
$g(c_1{:}Y_1)$ $g(c_2{:}Y_2)$ $g(c_3{:}Y_3)$ $\cdots$ $g(c_n{:}Y_n)$

Gittins policy: if box of max Gittins index is…

- *closed:* open it
- *open:* select it

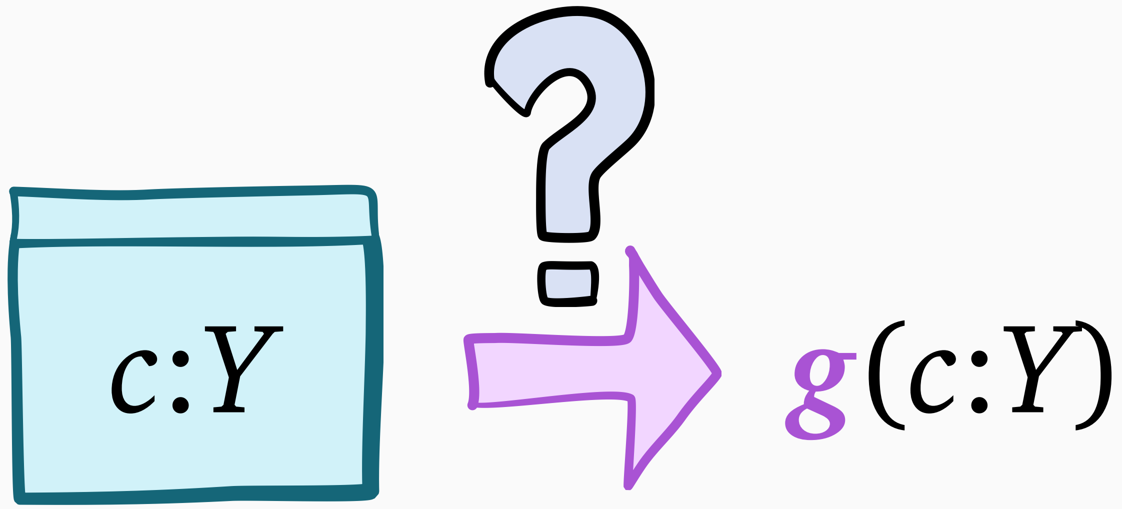
} *act* on it

Gittins index of a box

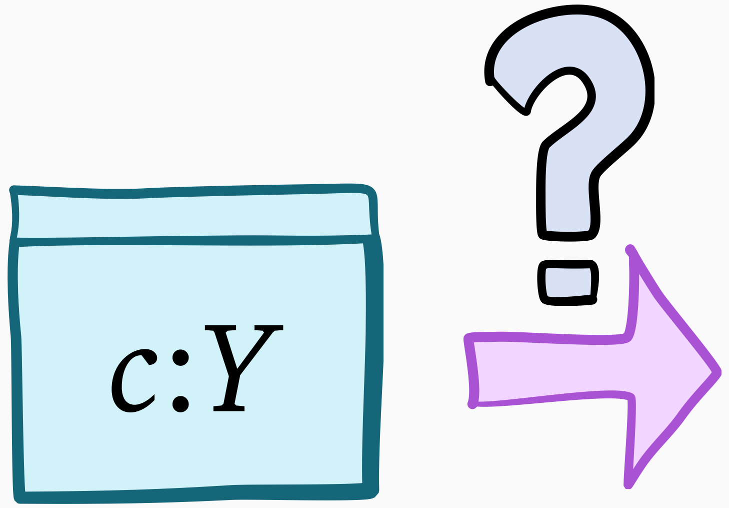


$g(c{:}Y)$

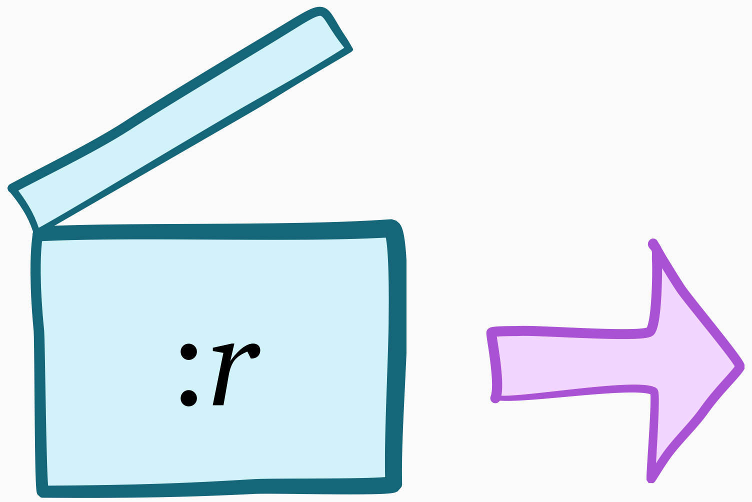
Gittins index of a box



Gittins index of a box

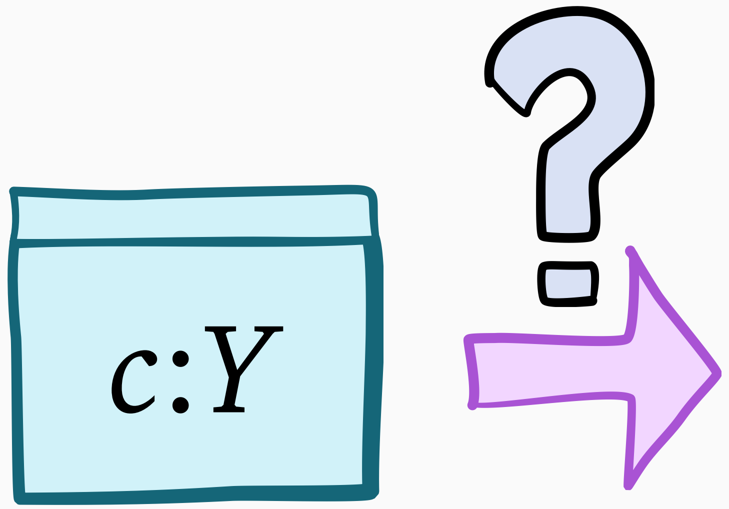


$g(c{:}Y)$

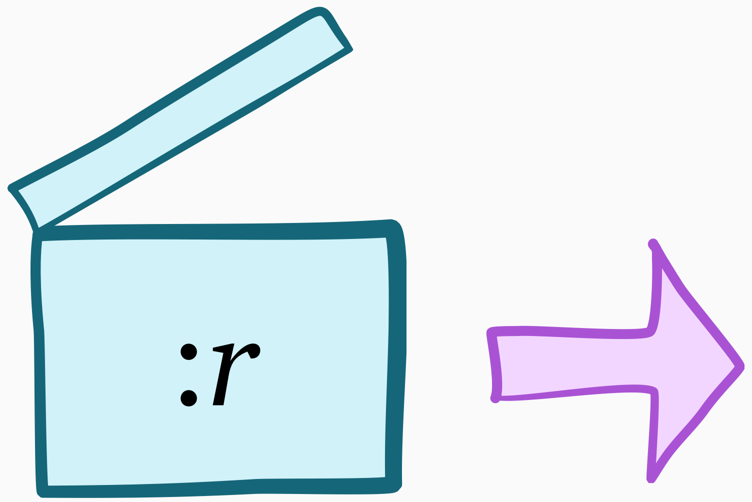


$g({:}r) = r$

Gittins index of a box

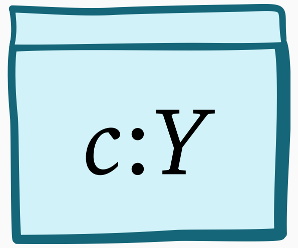


$g(c{:}Y)$

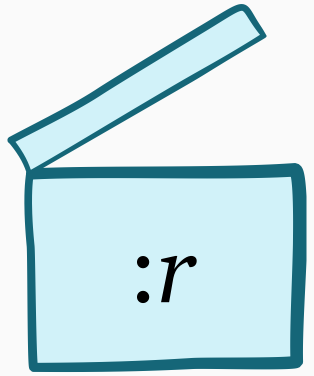


$g(:r) = r$

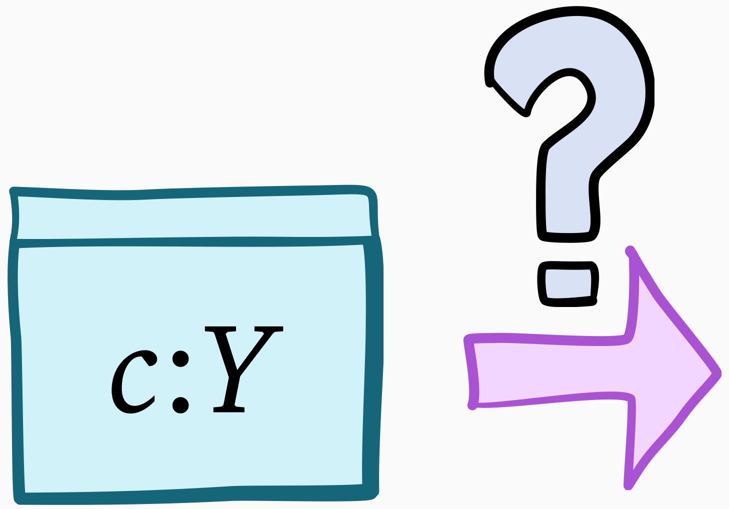
1.5-box problem



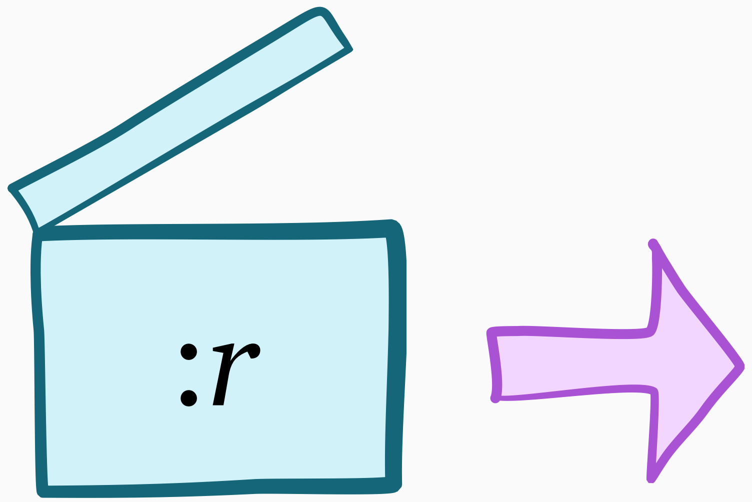
vs.



Gittins index of a box

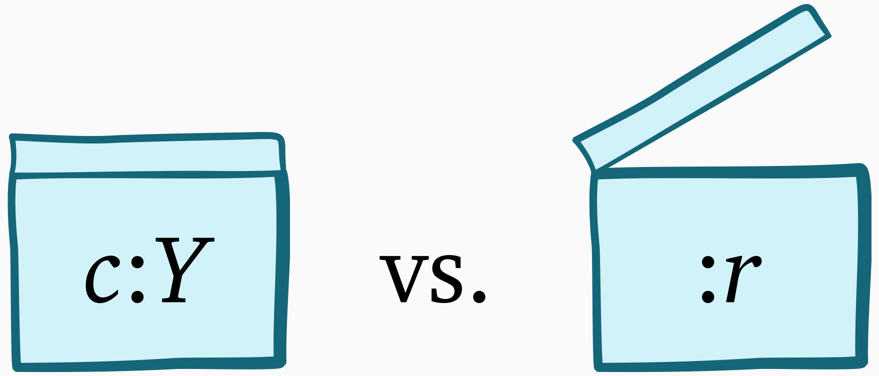


$g(c{:}Y)$



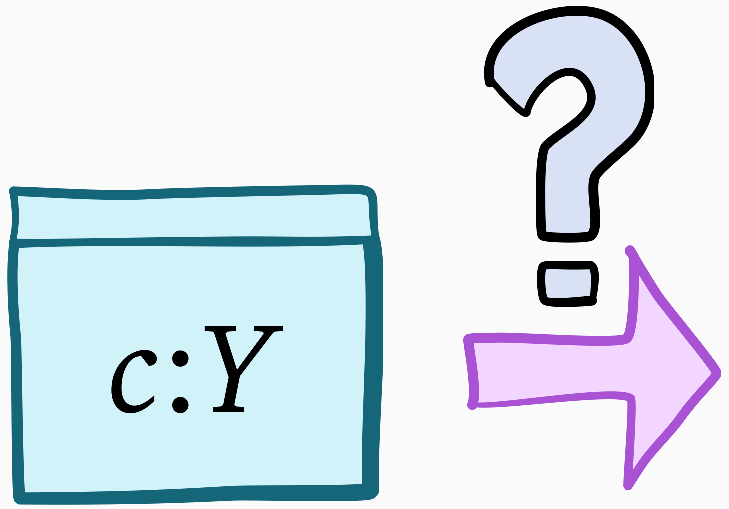
$g({:}r) = r$

1.5-box problem

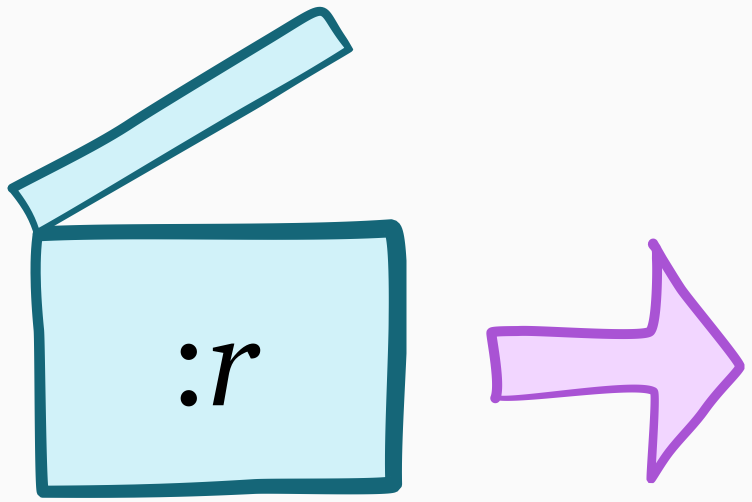


Key question: what to do in 1.5-box problem?

Gittins index of a box

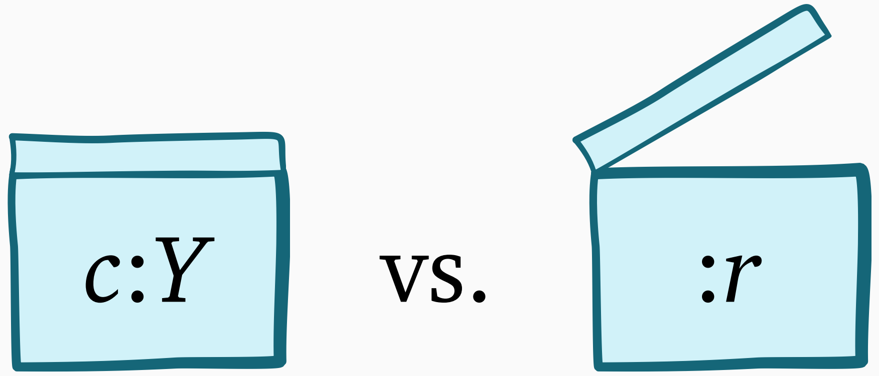


$g(c{:}Y)$

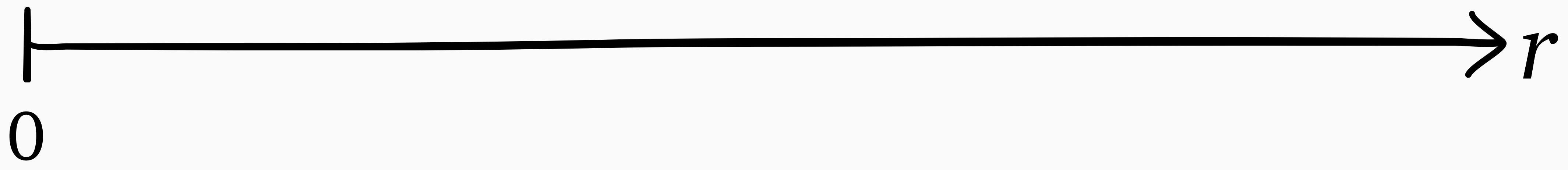


$g(:r) = r$

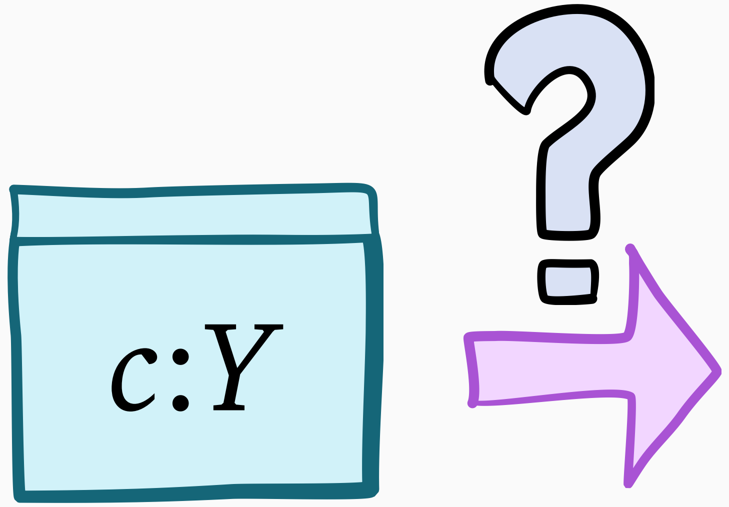
1.5-box problem



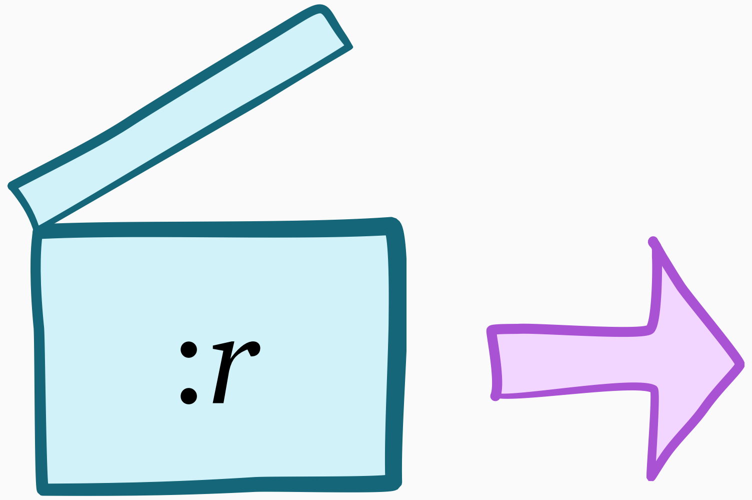
Key question: what to do in 1.5-box problem?



Gittins index of a box

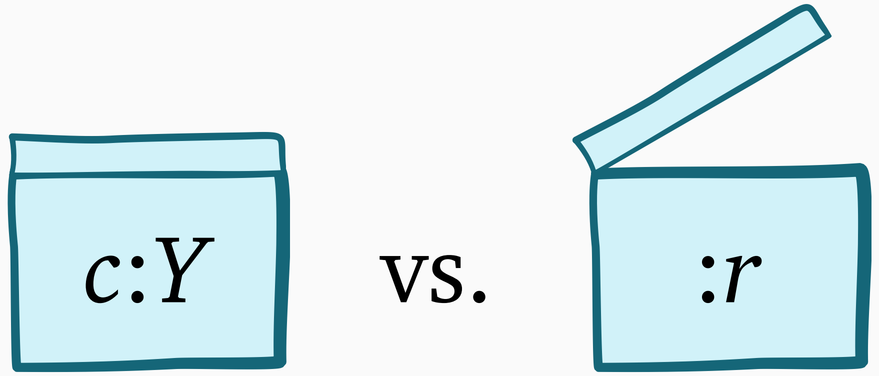


$g(c{:}Y)$

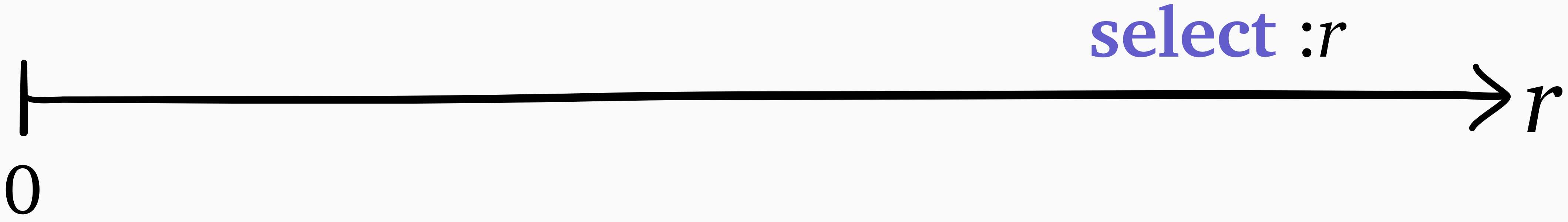


$g({:}r) = r$

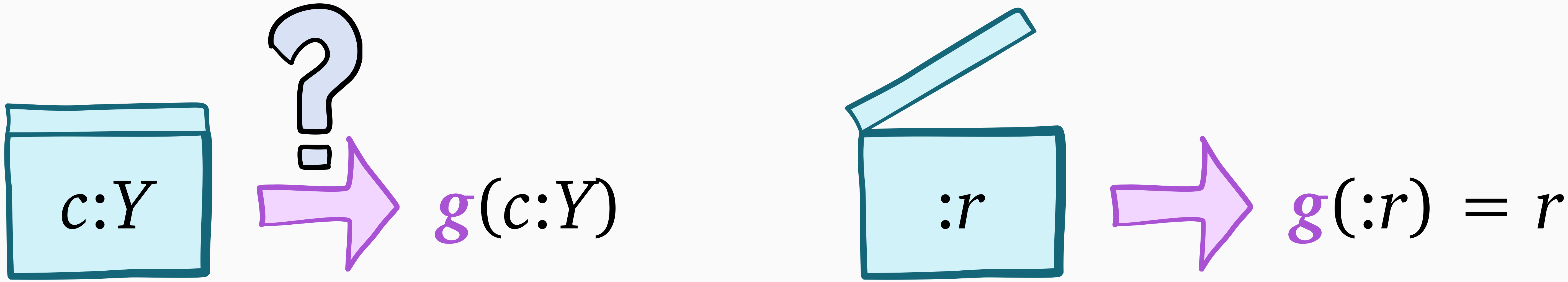
1.5-box problem



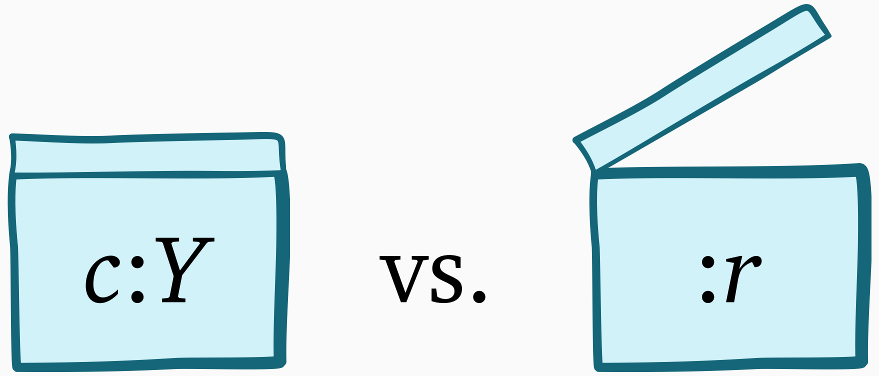
Key question: what to do in 1.5-box problem?



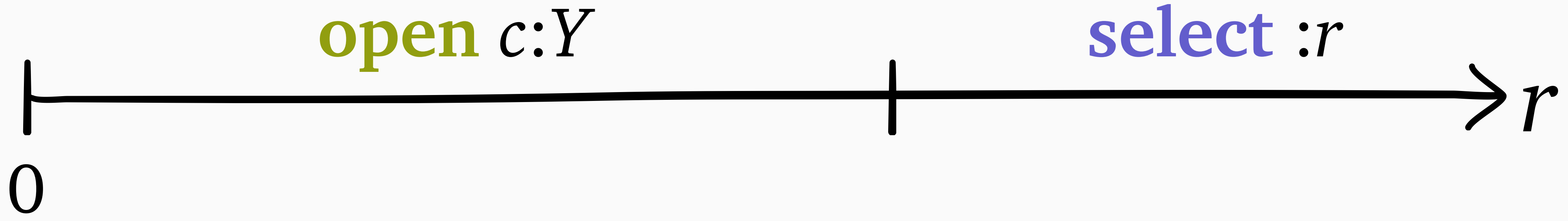
Gittins index of a box



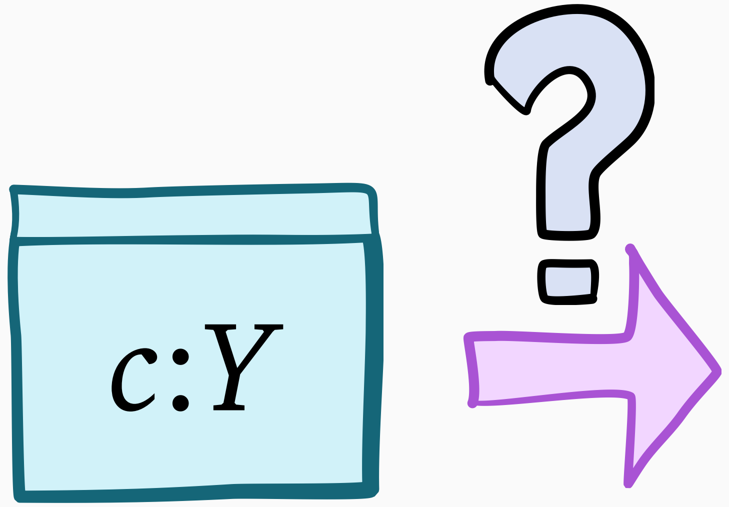
1.5-box problem



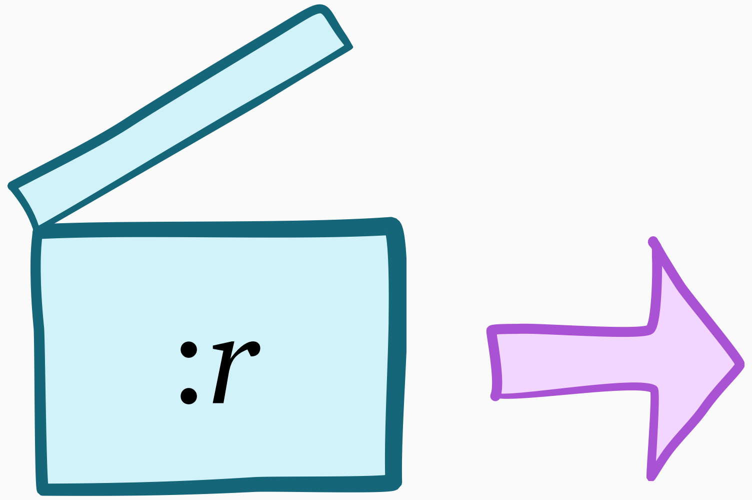
Key question: what to do in 1.5-box problem?



Gittins index of a box

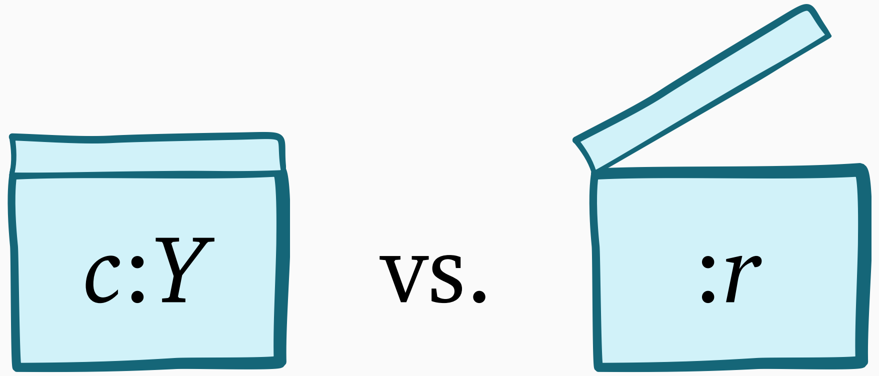


$g(c{:}Y)$

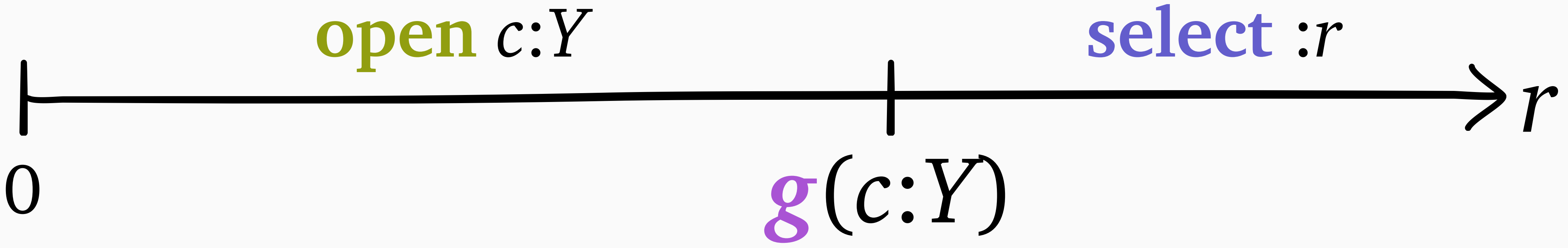


$g({:}r) = r$

1.5-box problem

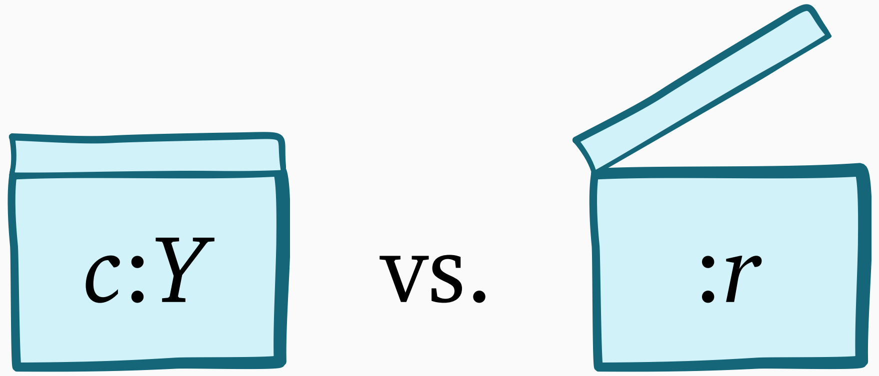


Key question: what to do in 1.5-box problem?

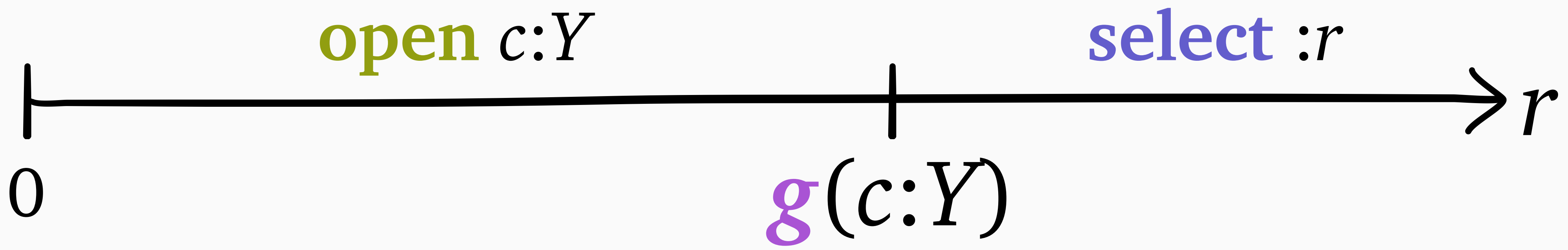


Gittins index of a box

1.5-box problem

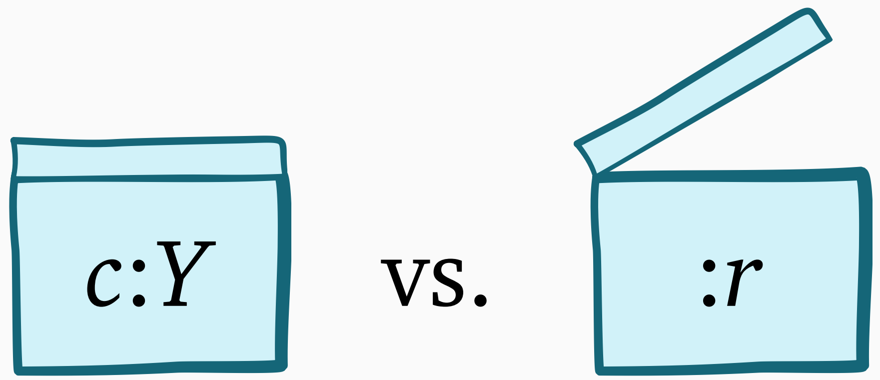


Key question: what to do in 1.5-box problem?

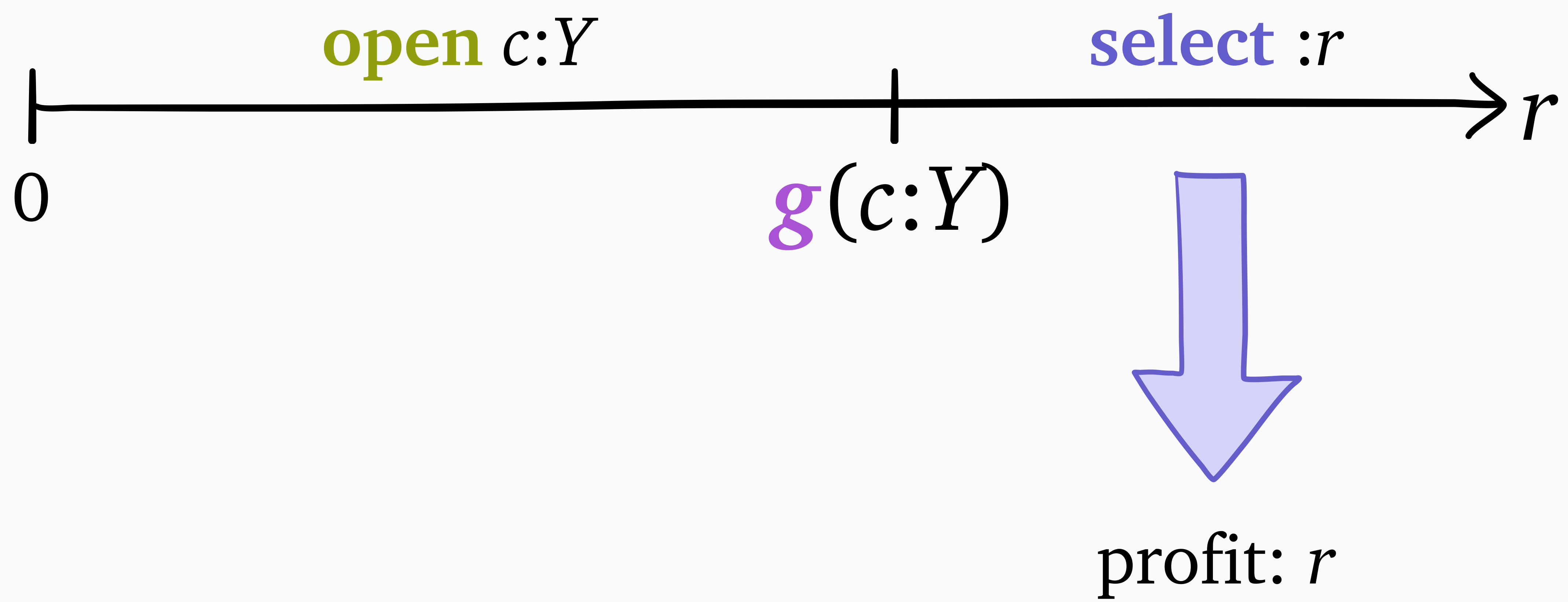


Gittins index of a box

1.5-box problem

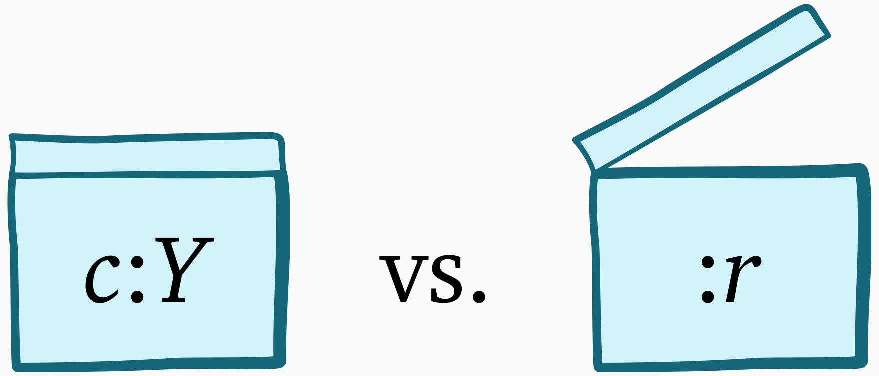


Key question: what to do in 1.5-box problem?

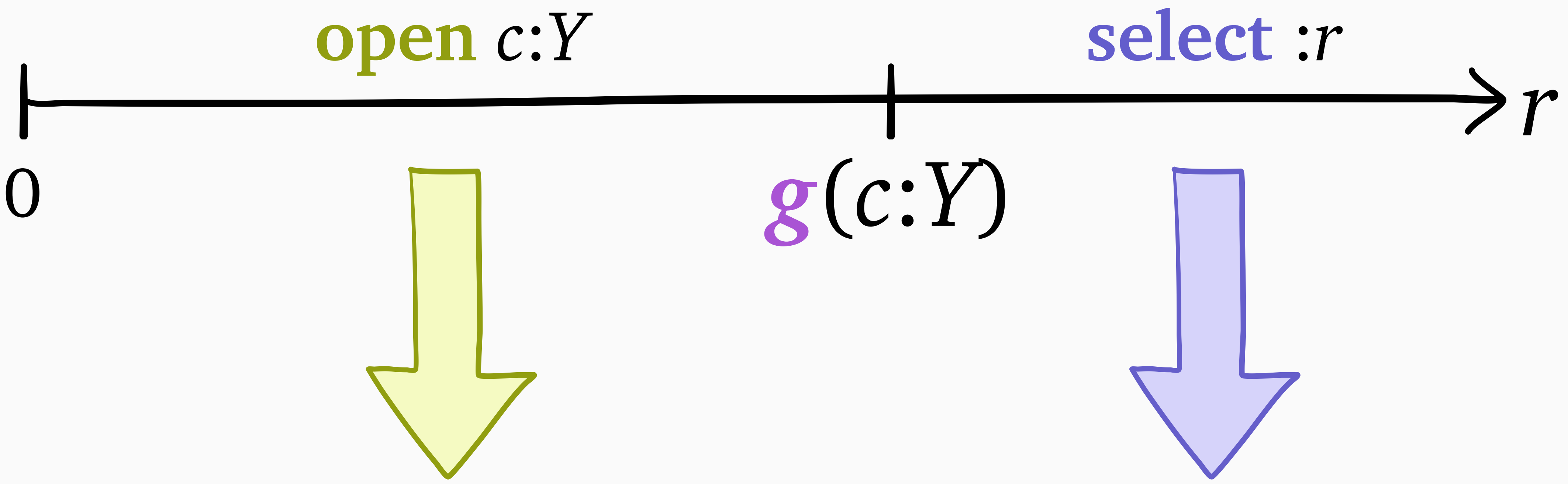


Gittins index of a box

1.5-box problem



Key question: what to do in 1.5-box problem?

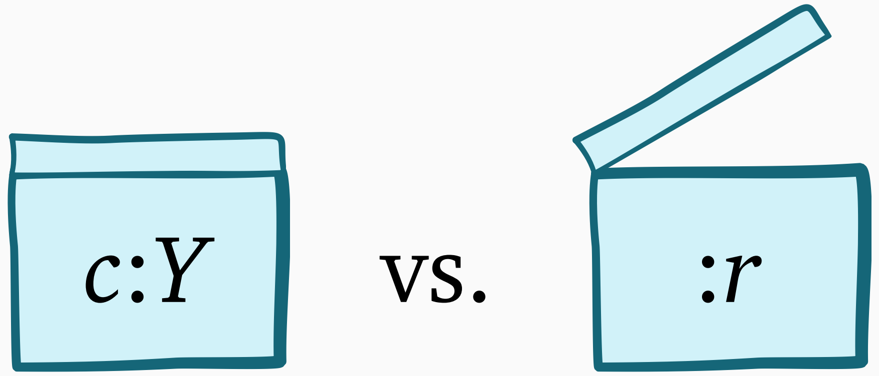


profit: $\mathbf{E}[\max\{Y, r\}] - c$

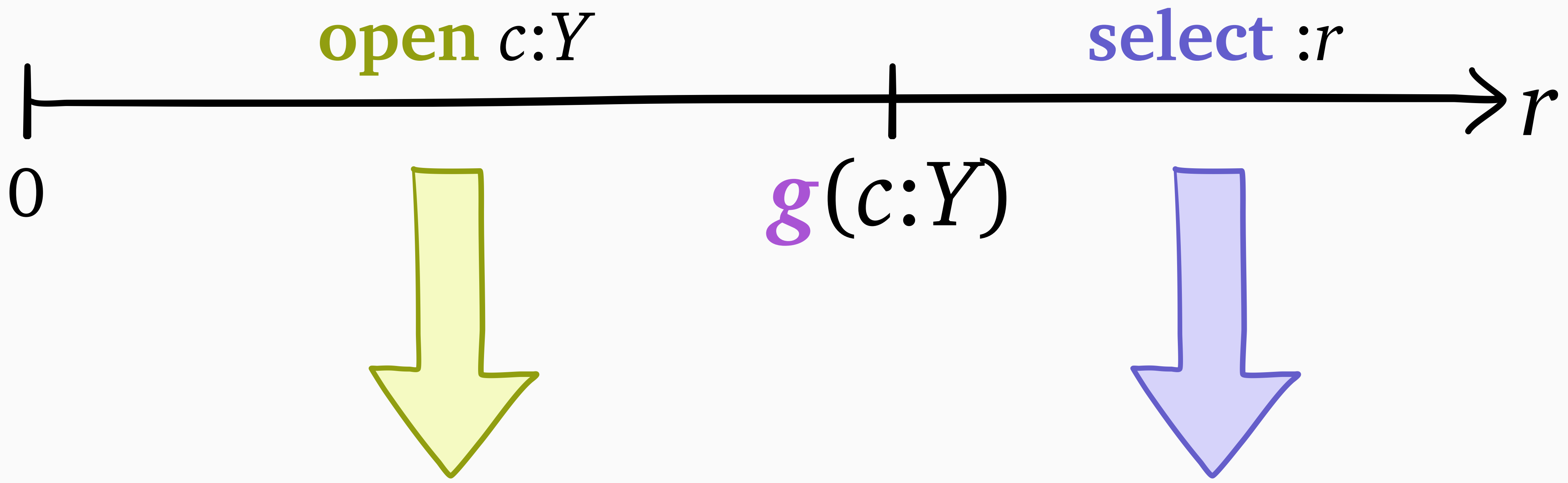
profit: r

Gittins index of a box

1.5-box problem



Key question: what to do in 1.5-box problem?



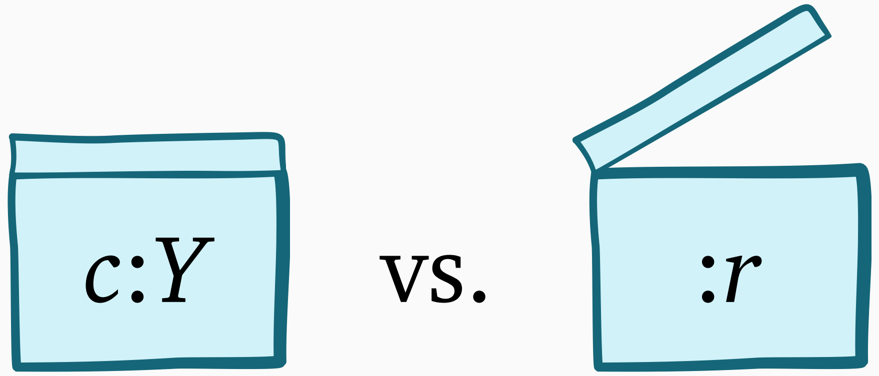
profit: $\mathbf{E}[\max\{Y, r\}] - c$

profit: r

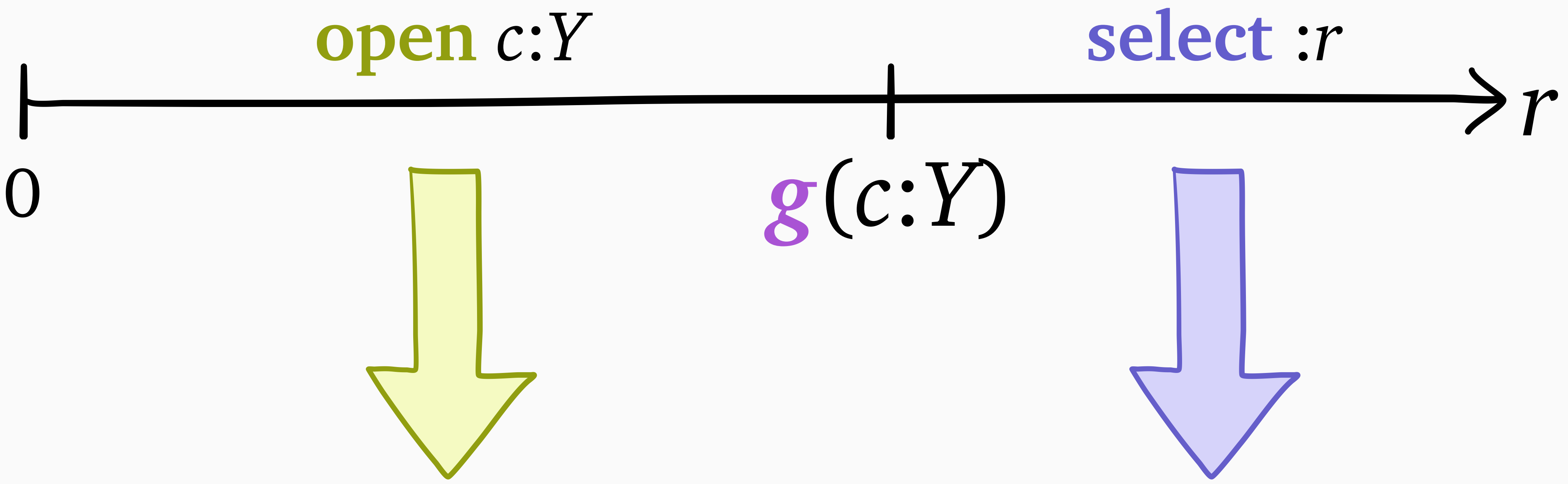
When are these equal?

Gittins index of a box

1.5-box problem



Key question: what to do in 1.5-box problem?



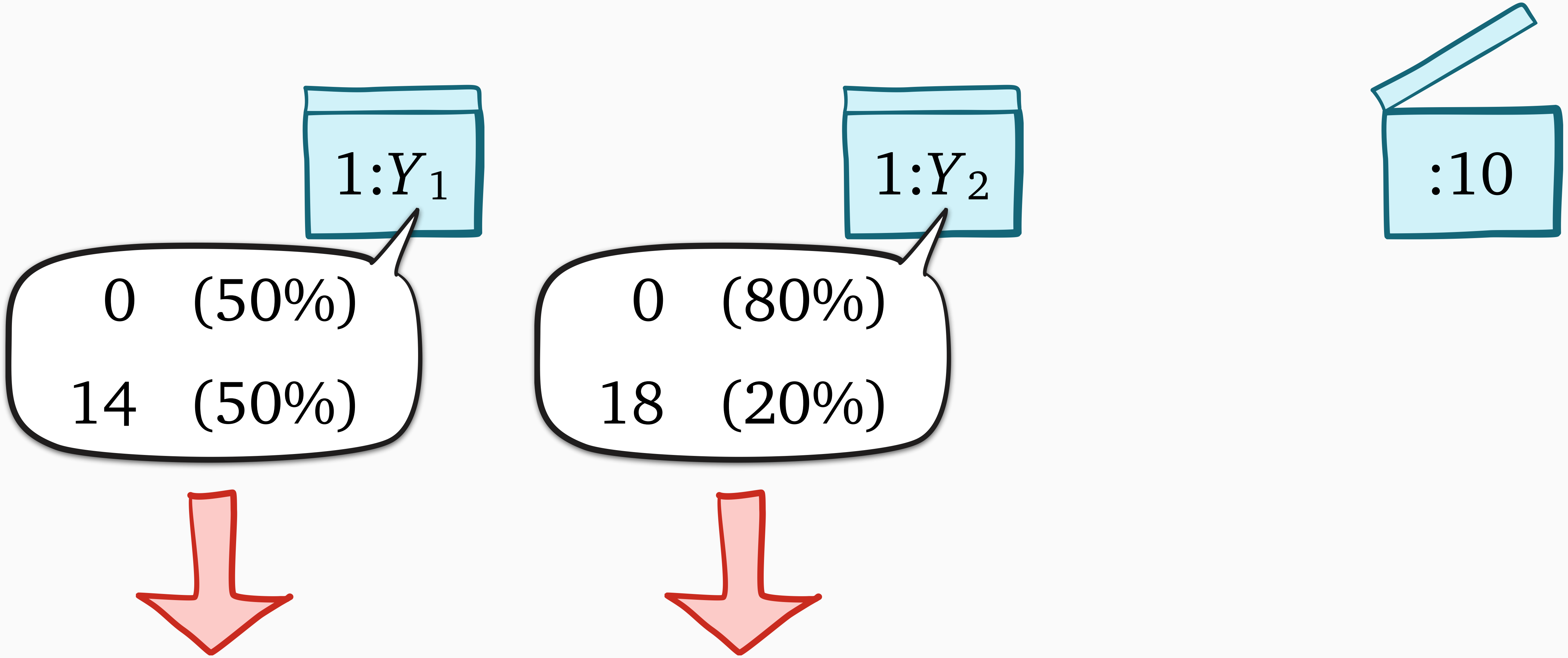
profit: $\mathbf{E}[\max\{Y, r\}] - c$

profit: r

When are these equal?

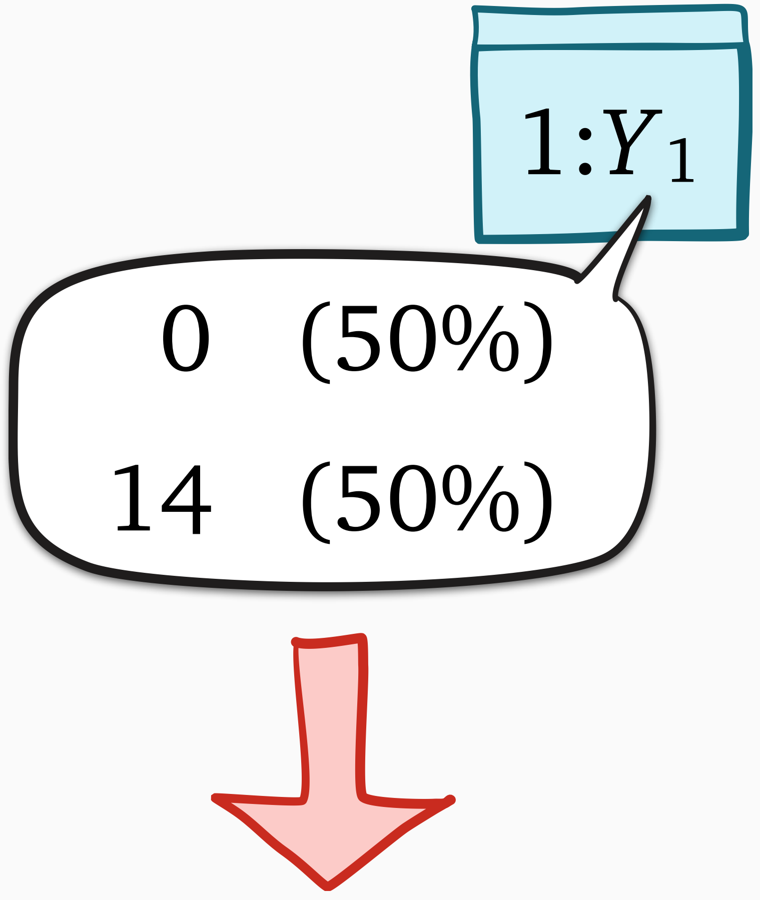
Defn: $g(c{:}Y)$ is solution r to
$$\mathrm{EI}(Y, r) = \mathbf{E}[(Y - r)^+] = c$$

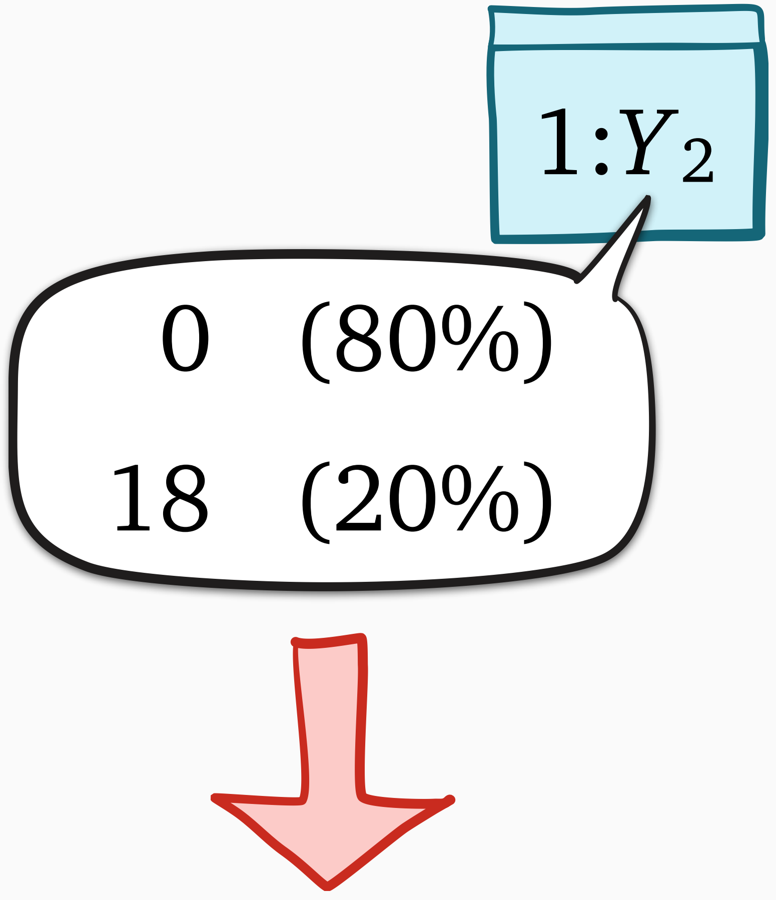
Expected improvement vs. **Gittins**



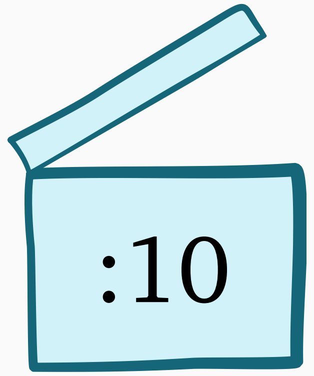
$\mathrm{EI}(Y_1, 10) = 2$ $\mathrm{EI}(Y_2, 10) = 1.6$

Expected improvement vs. Gittins



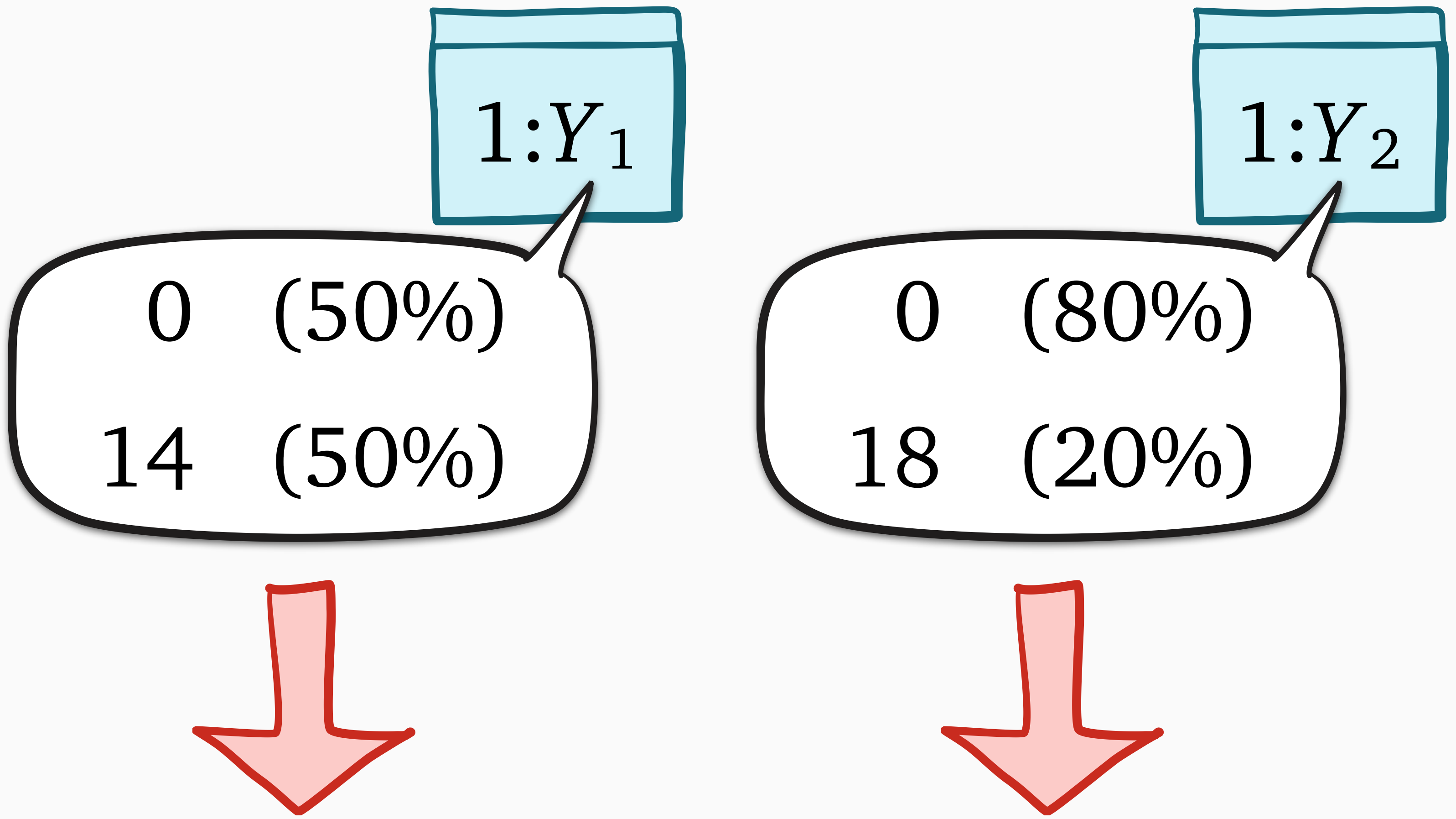


$\mathrm{EI}(Y_1, 10) = 2$ $\mathrm{EI}(Y_2, 10) = 1.6$

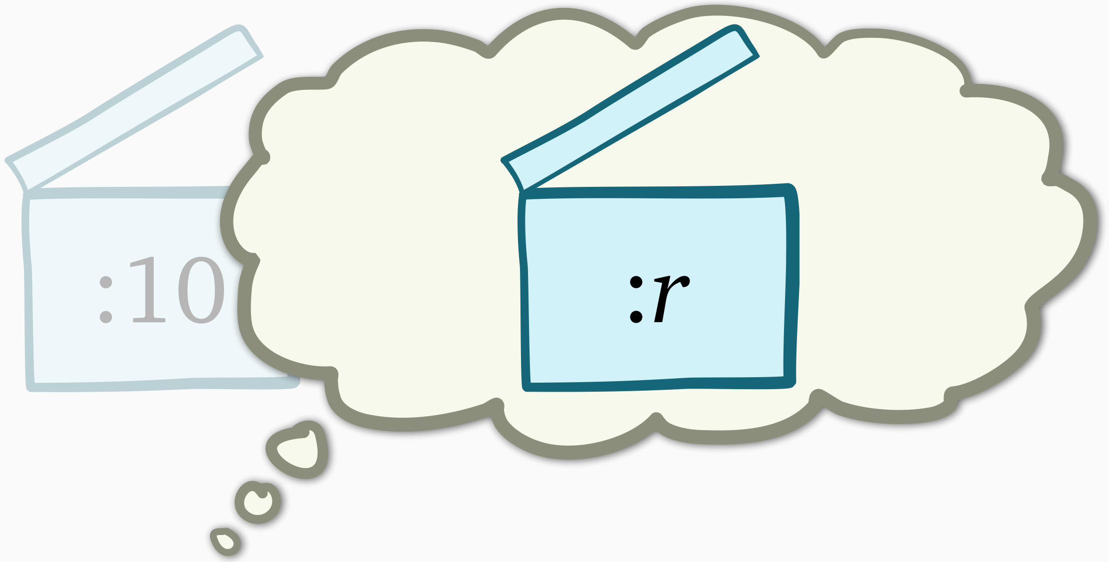


With $r = 10$, what is $\mathrm{EI}(Y, r) - c$?

Expected improvement vs. Gittins



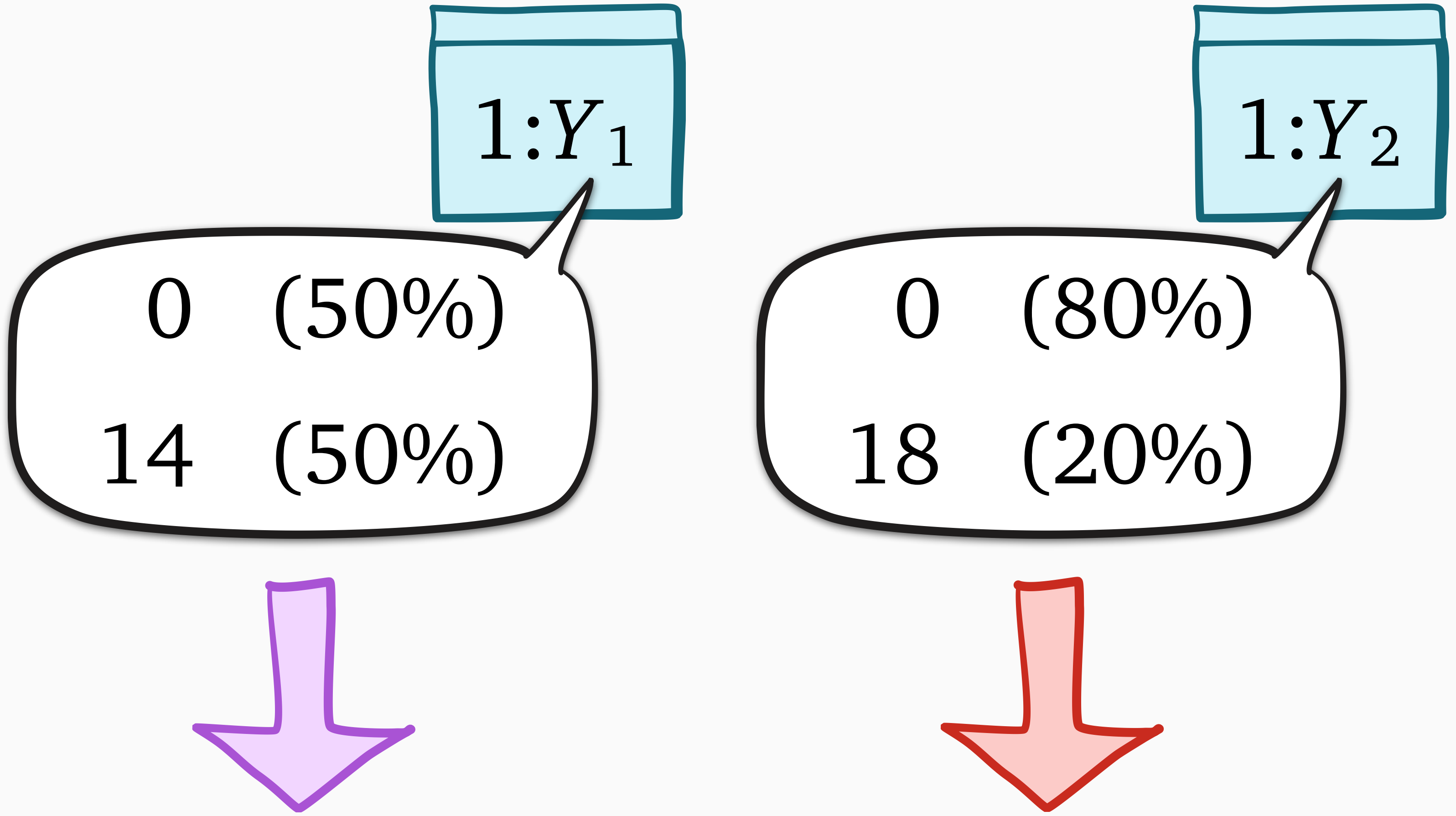
$\mathrm{EI}(Y_1, 10) = 2$ $\mathrm{EI}(Y_2, 10) = 1.6$



With $r = 10$, what is $\mathrm{EI}(Y, r) - c$?

For what r does $\mathrm{EI}(Y, r) = c$?

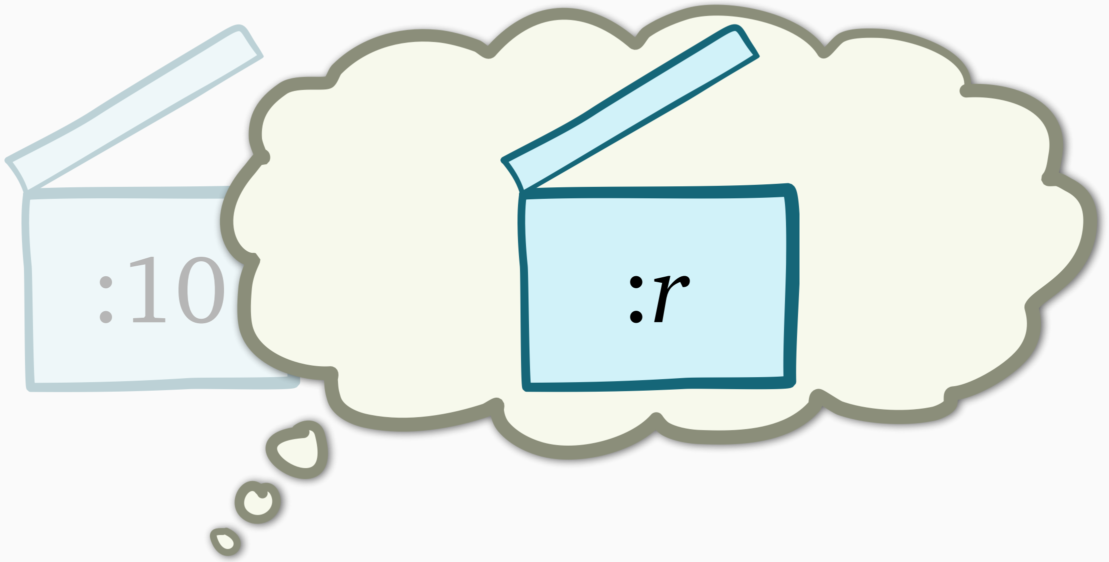
Expected improvement vs. Gittins



$\mathrm{EI}(Y_1, 10) = 2$

$g(1{:}Y_1) = 12$

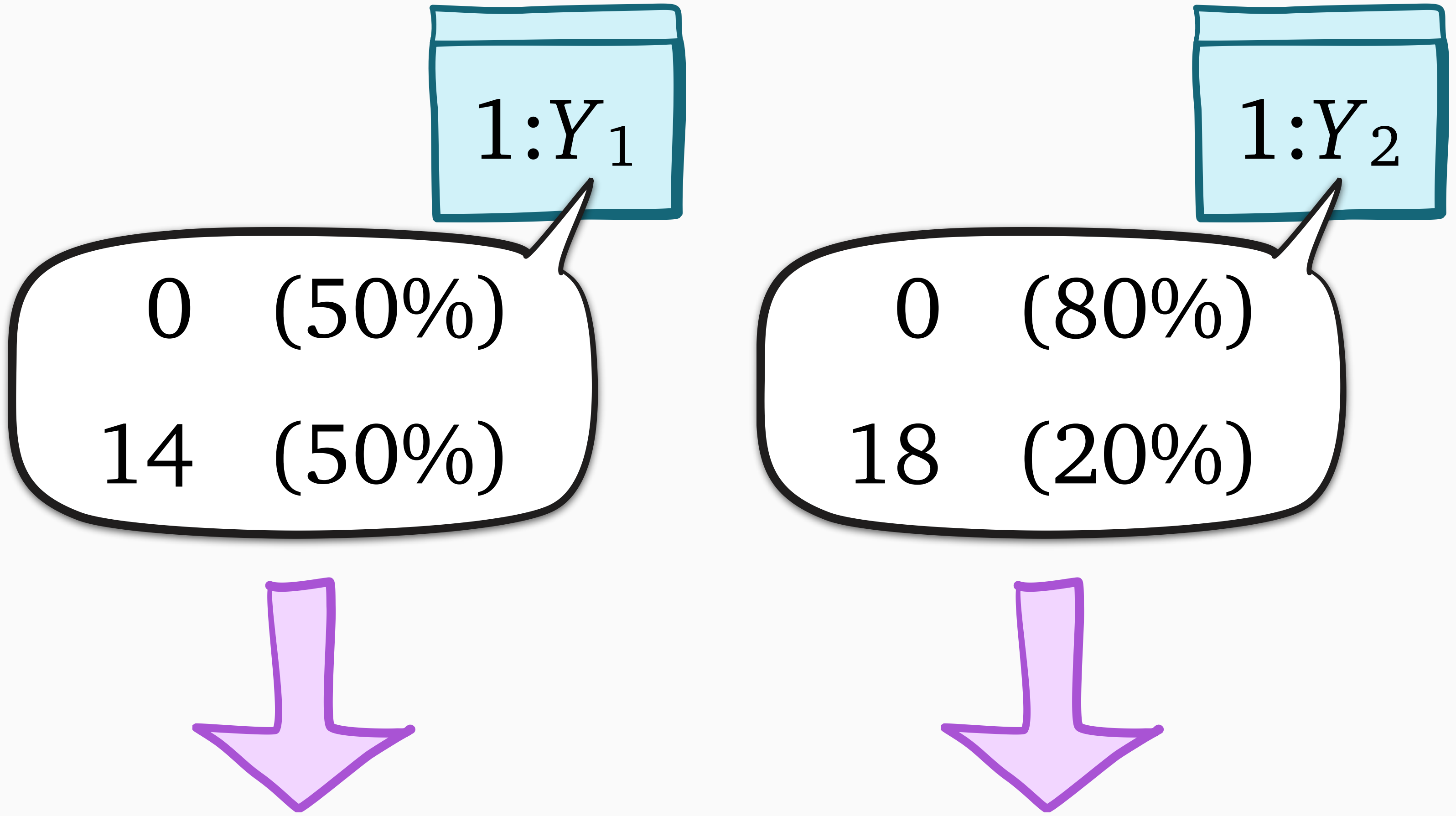
$\mathrm{EI}(Y_2, 10) = 1.6$



With $r = 10$, what is $\mathrm{EI}(Y, r) - c$?

For what r does $\mathrm{EI}(Y, r) = c$?

Expected improvement vs. **Gittins**

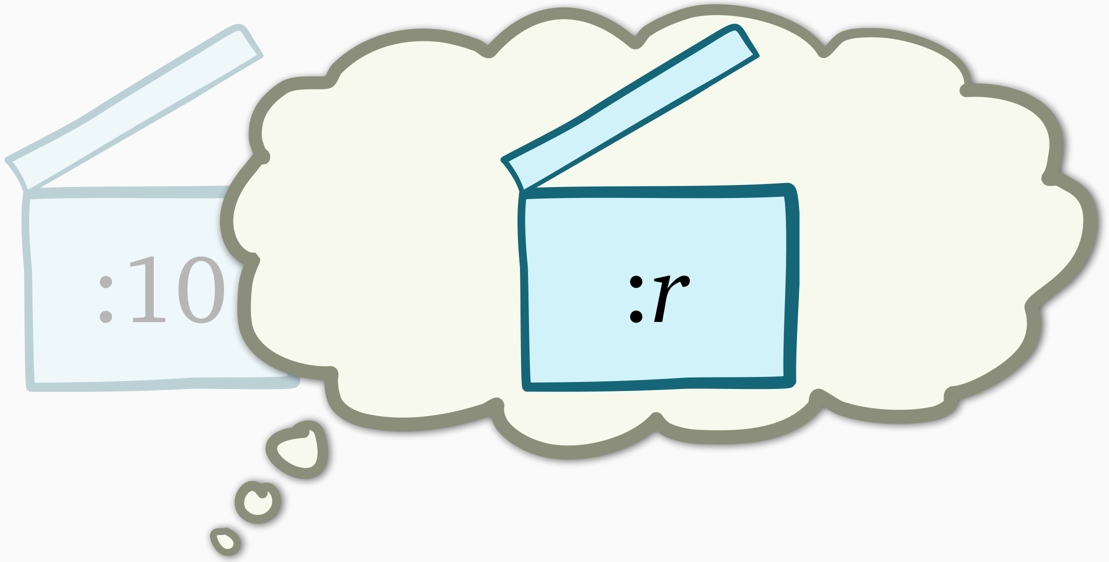


$\mathrm{EI}(Y_1, 10) = 2$

$g(1{:}Y_1) = 12$

$\mathrm{EI}(Y_2, 10) = 1.6$

$g(1{:}Y_2) = 13$

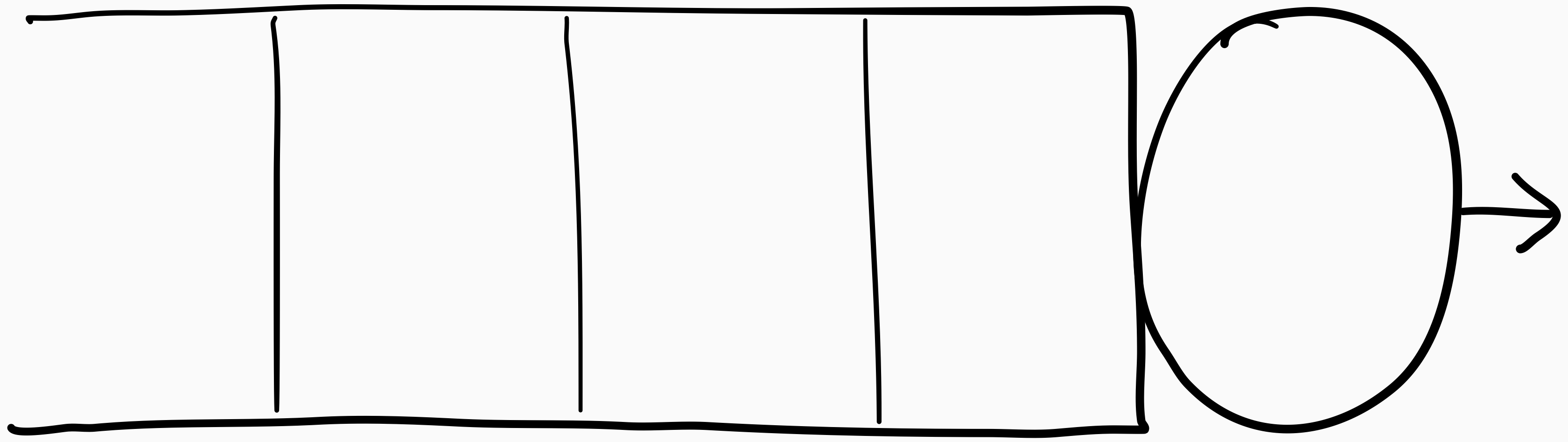


✗ With $r = 10$, what is $\mathrm{EI}(Y, r) - c$?

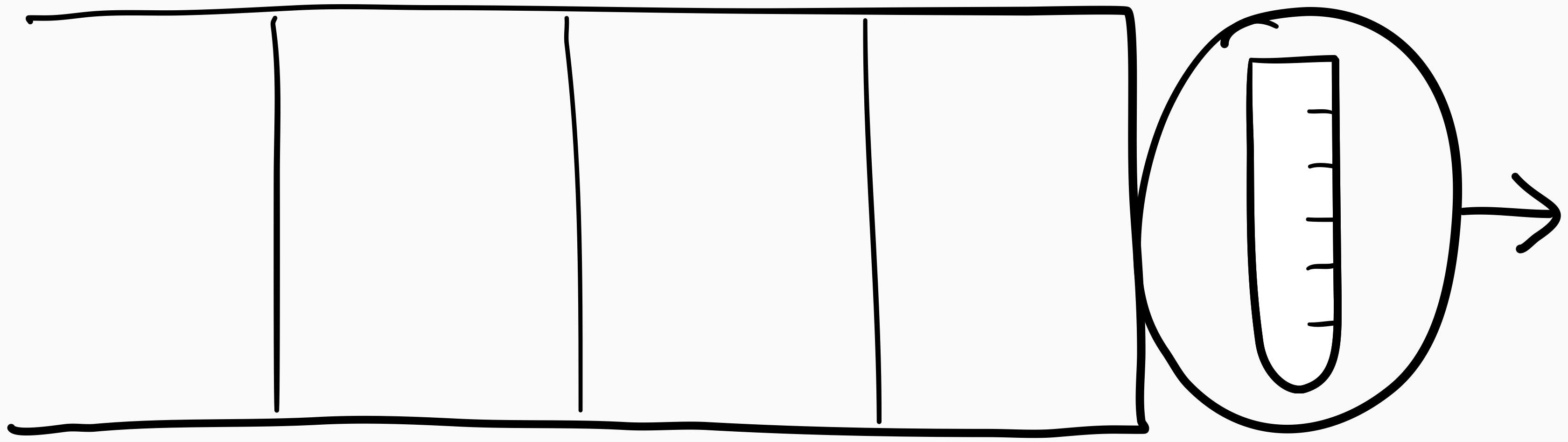
✓ For what r does $\mathrm{EI}(Y, r) = c$?

Mean scheduling in the M/G/1

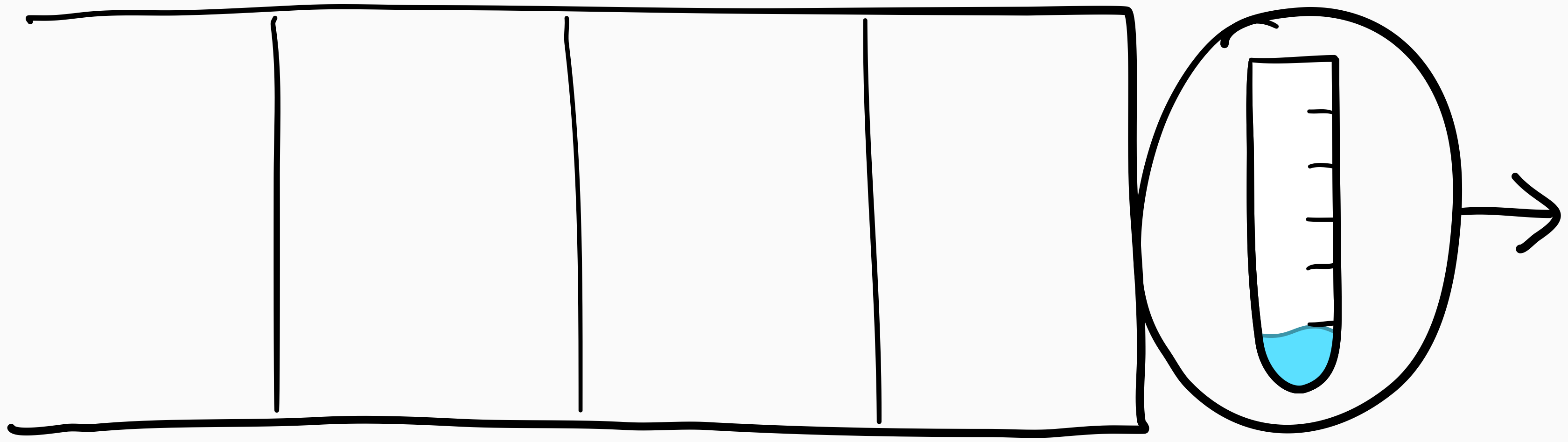
Mean scheduling in the M/G/1



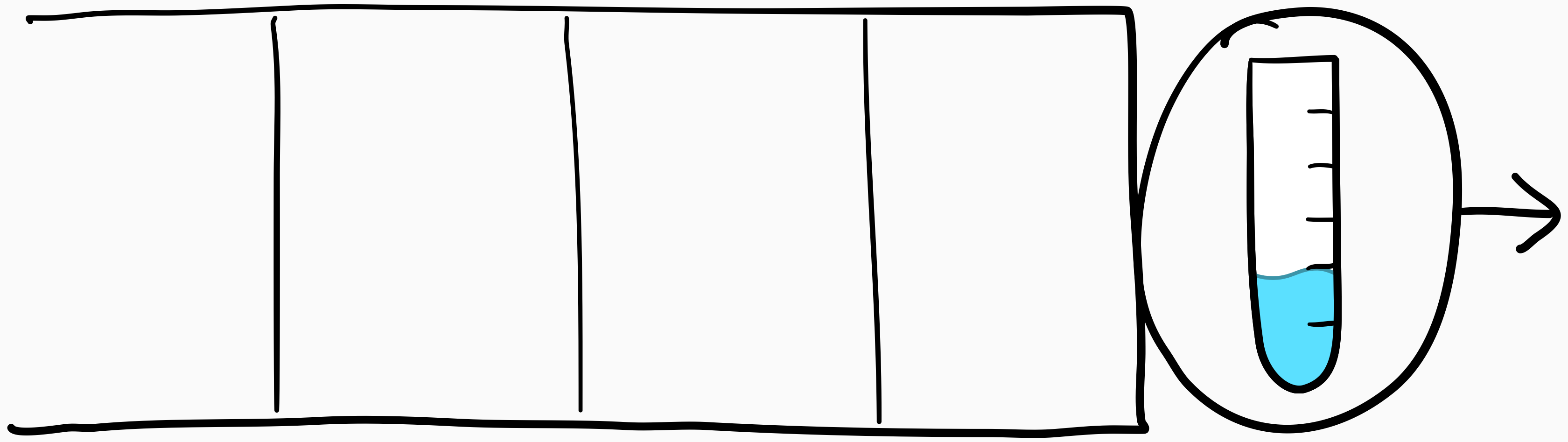
Mean scheduling in the M/G/1



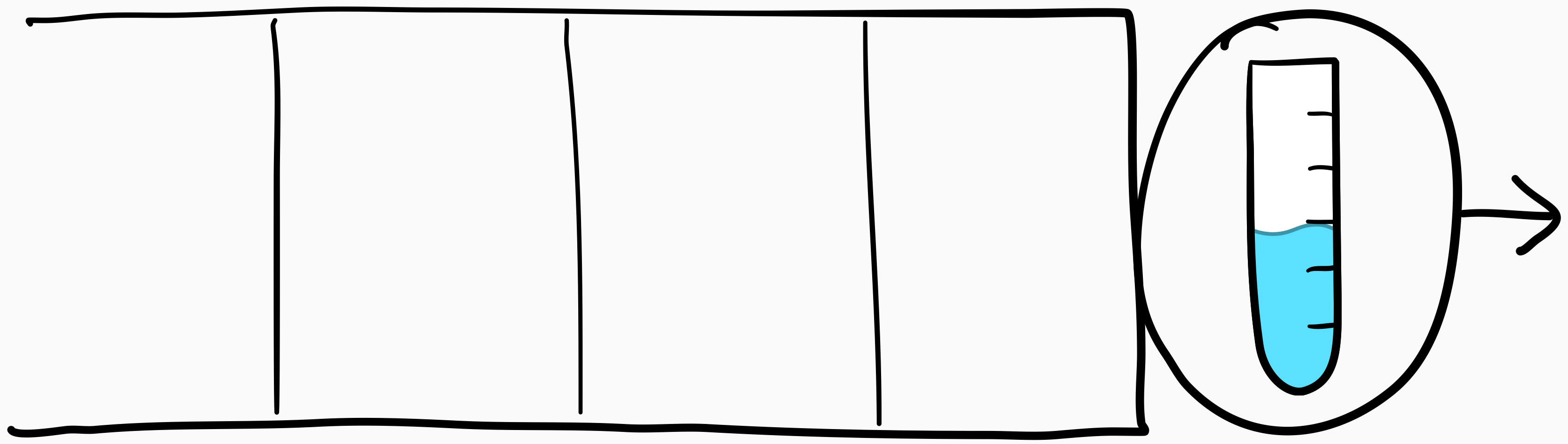
Mean scheduling in the M/G/1



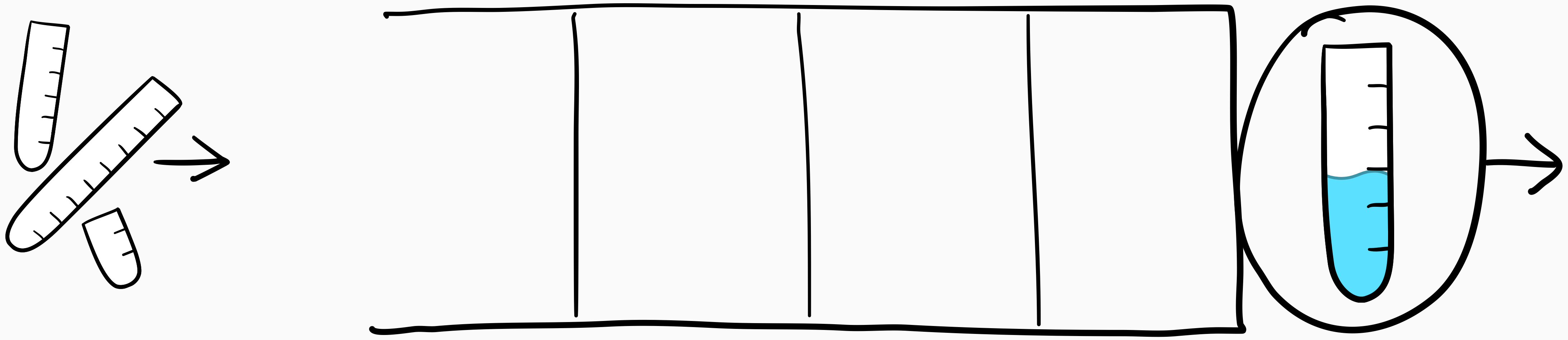
Mean scheduling in the M/G/1



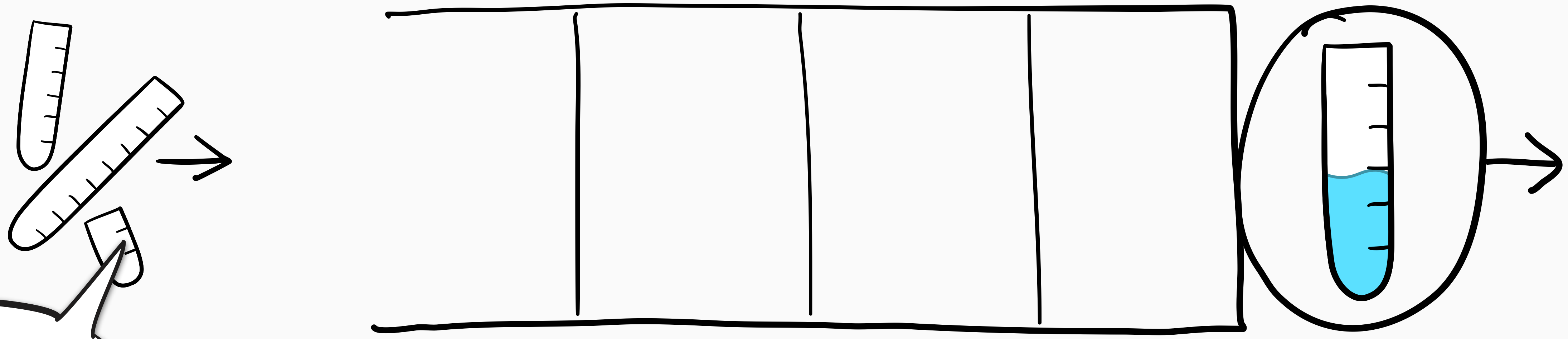
Mean scheduling in the M/G/1



Mean scheduling in the M/G/1



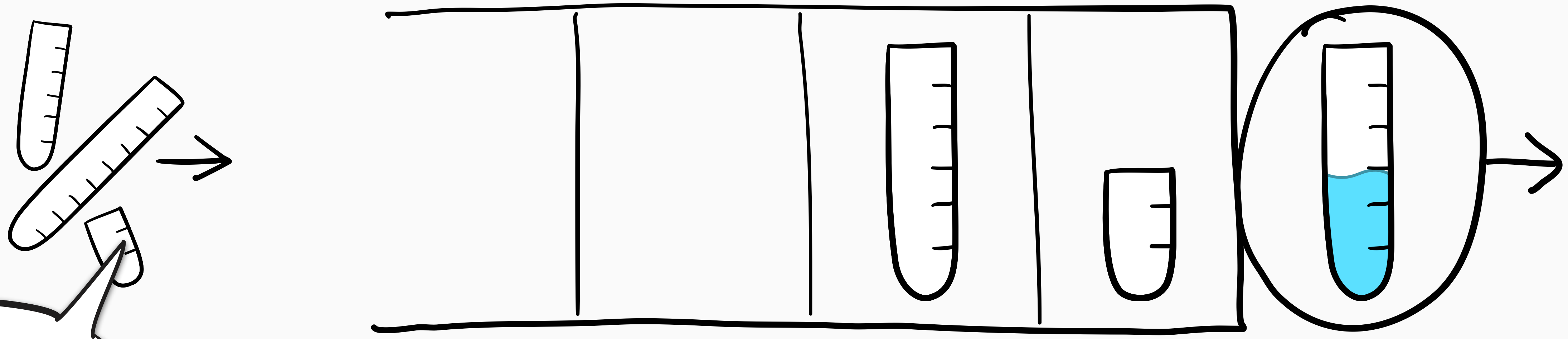
Mean scheduling in the M/G/1



M/G arrivals:

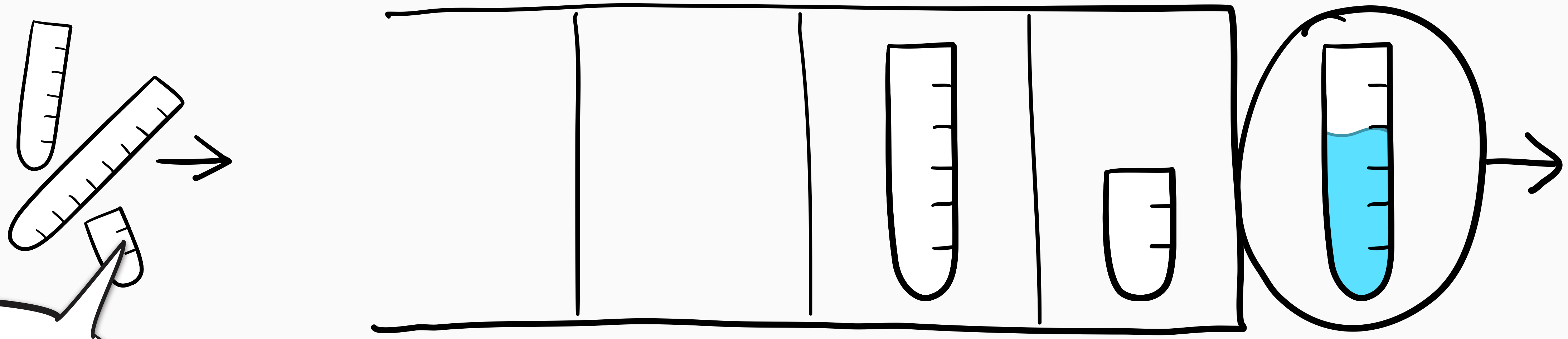
- arrival rate λ
- size distribution S

Mean scheduling in the M/G/1



M/G arrivals:
- arrival rate λ
- size distribution S

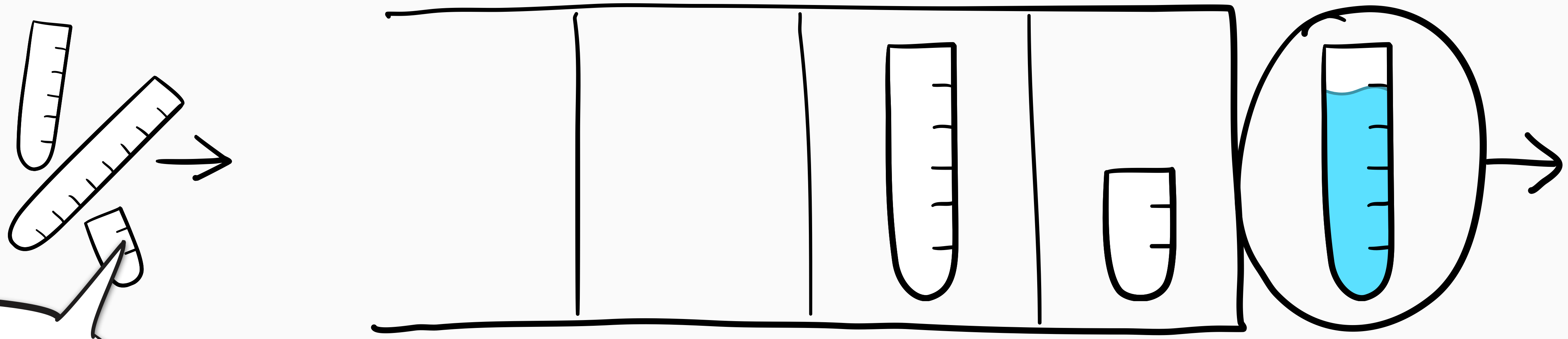
Mean scheduling in the M/G/1



M/G arrivals:

- arrival rate λ
- size distribution S

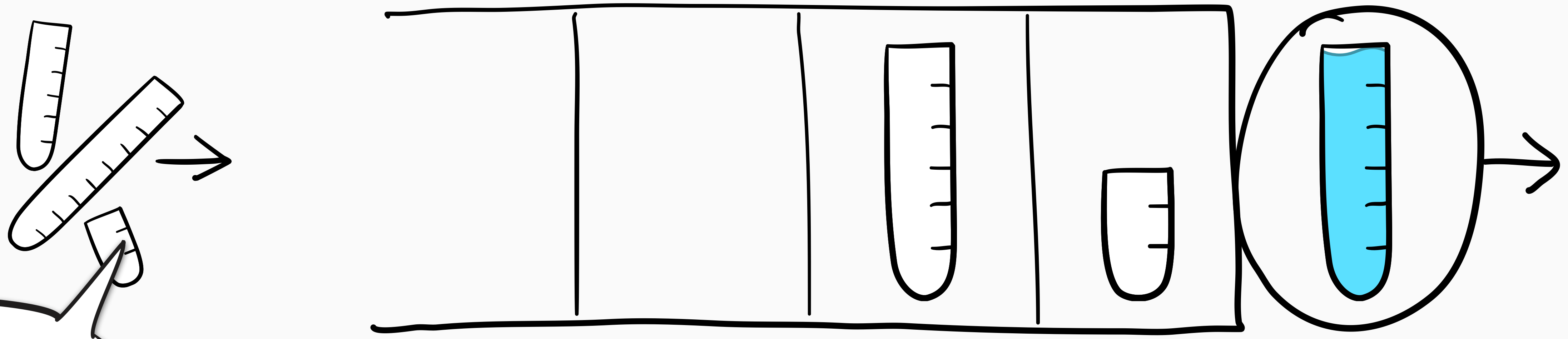
Mean scheduling in the M/G/1



M/G arrivals:

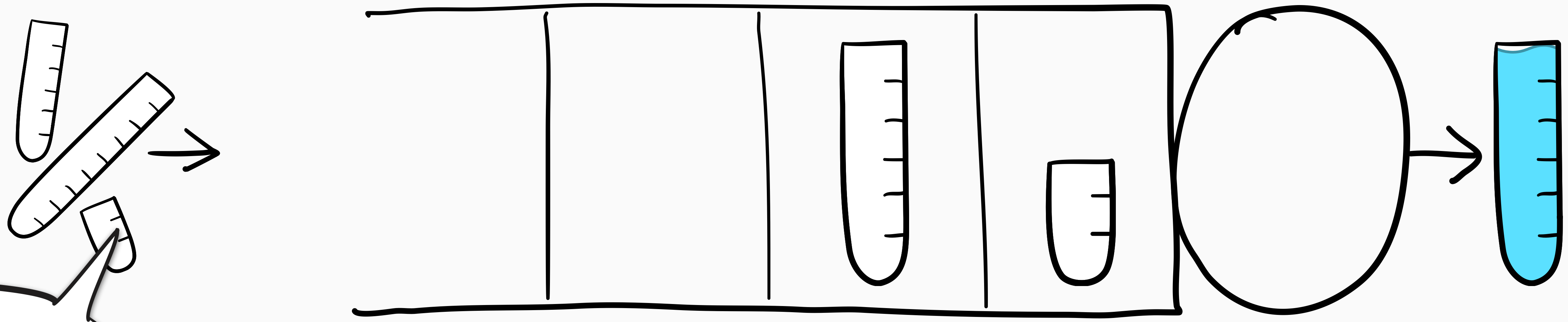
- arrival rate λ
- size distribution S

Mean scheduling in the M/G/1



M/G arrivals:
- arrival rate λ
- size distribution S

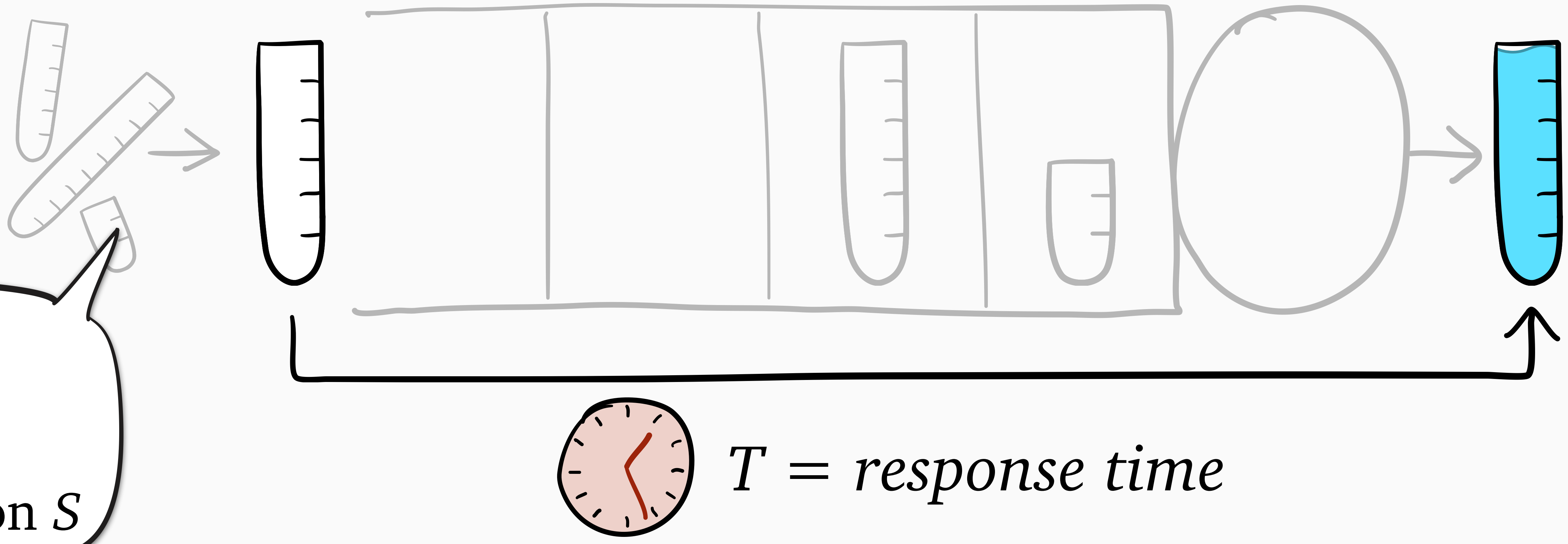
Mean scheduling in the M/G/1



M/G arrivals:

- arrival rate λ
- size distribution S

Mean scheduling in the M/G/1

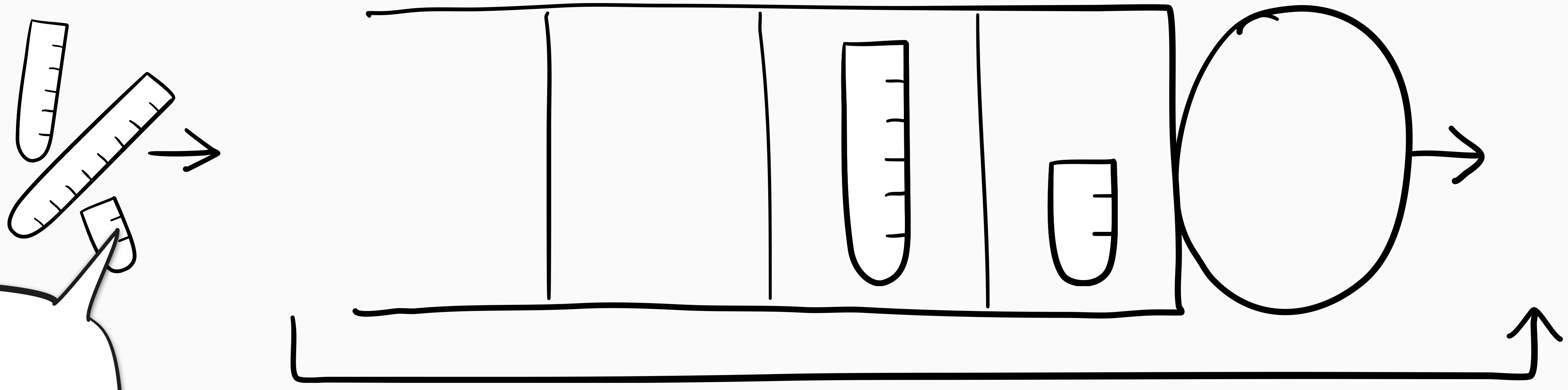


M/G arrivals:

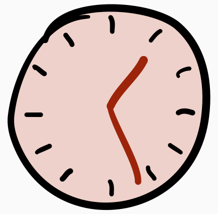
- arrival rate λ
- size distribution S

T = *response time*

Mean scheduling in the M/G/1

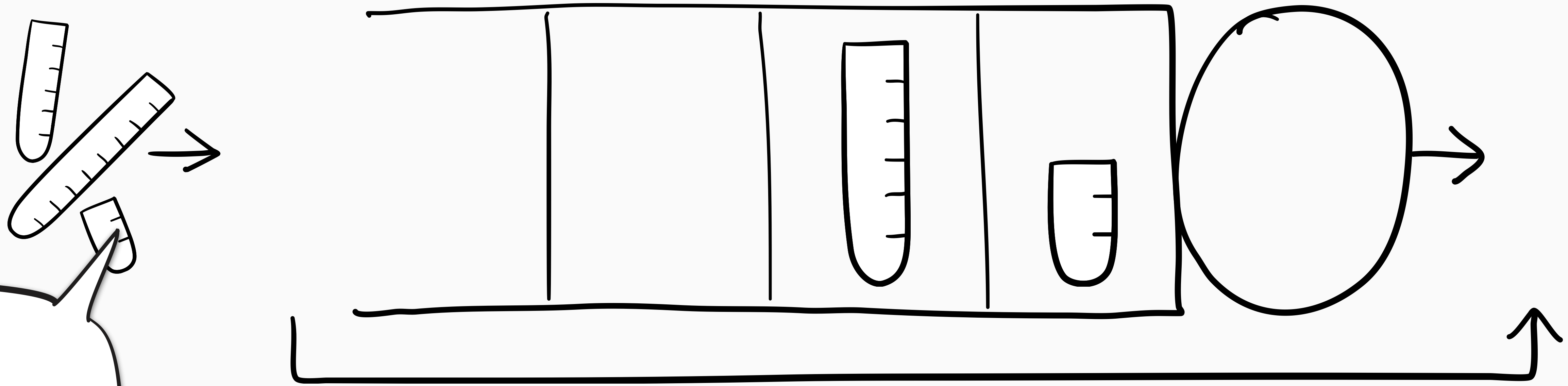


M/G arrivals:
- arrival rate λ
- size distribution S

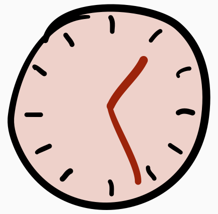


T = *response time*

Mean scheduling in the M/G/1



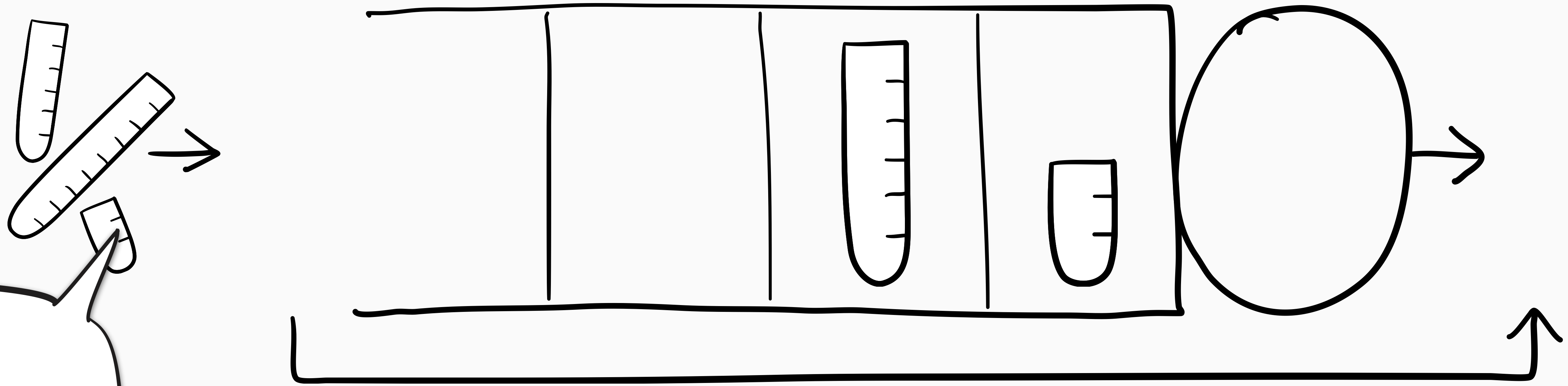
M/G arrivals:
- arrival rate λ
- size distribution S



T = *response time*

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

Mean scheduling in the M/G/1



M/G arrivals:

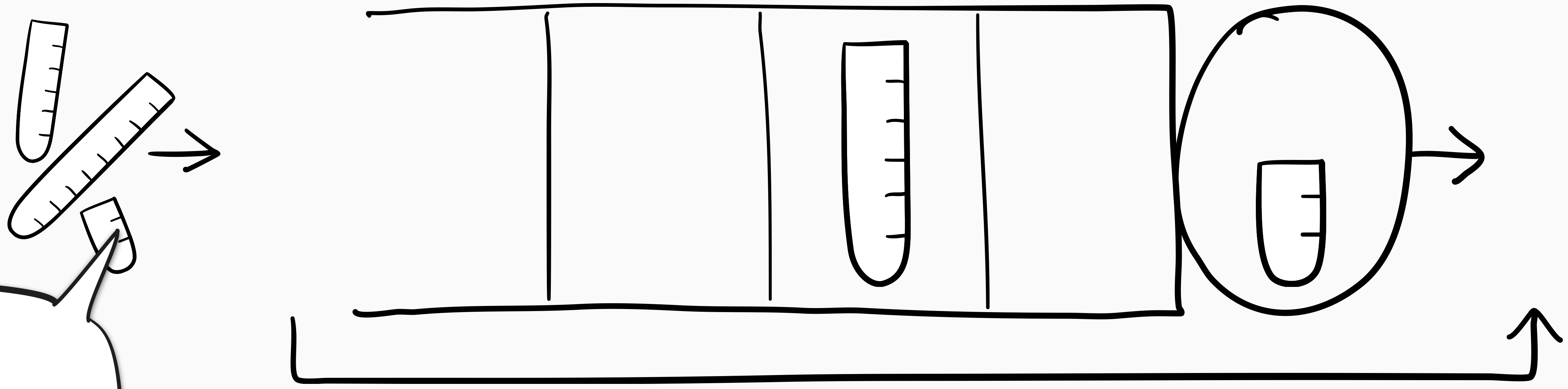
- arrival rate λ
- size distribution S

T = *response time*

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

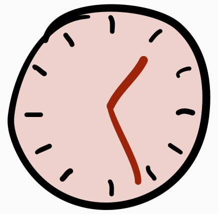
Serve short jobs before long jobs

Mean scheduling in the M/G/1



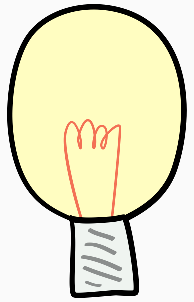
M/G arrivals:

- arrival rate λ
- size distribution S



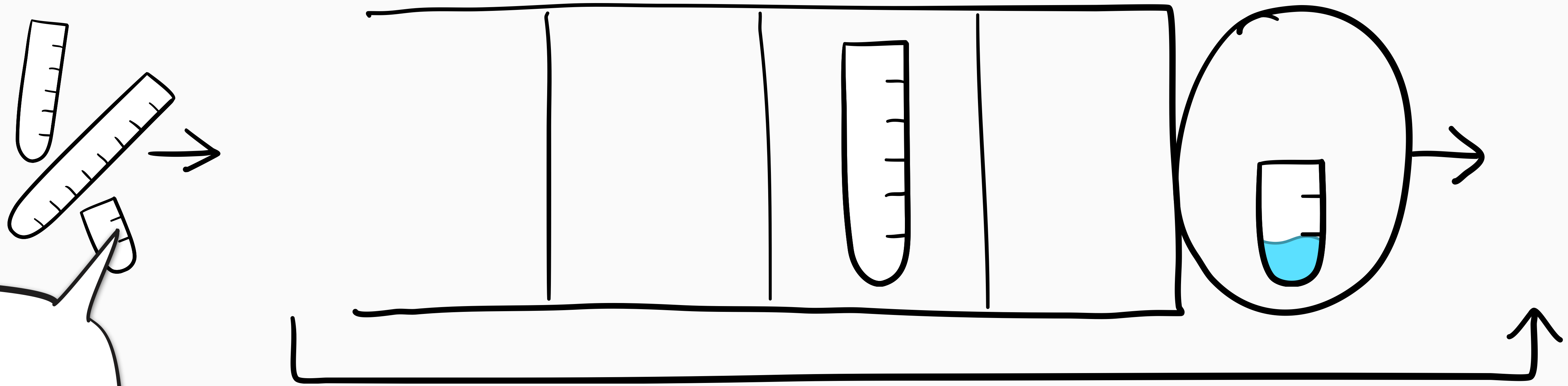
T = *response time*

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



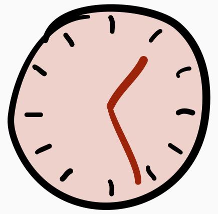
Serve short jobs
before long jobs

Mean scheduling in the M/G/1



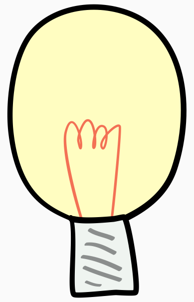
M/G arrivals:

- arrival rate λ
- size distribution S



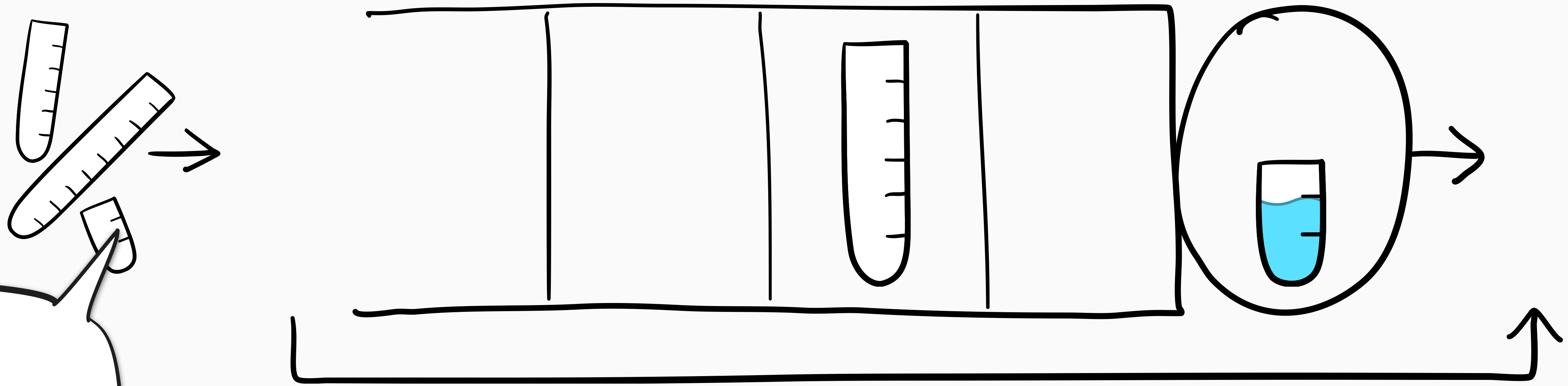
T = *response time*

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



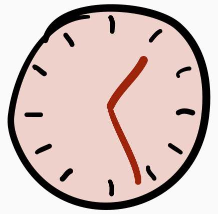
Serve short jobs
before long jobs

Mean scheduling in the M/G/1



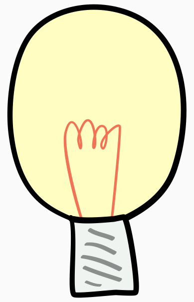
M/G arrivals:

- arrival rate λ
- size distribution S



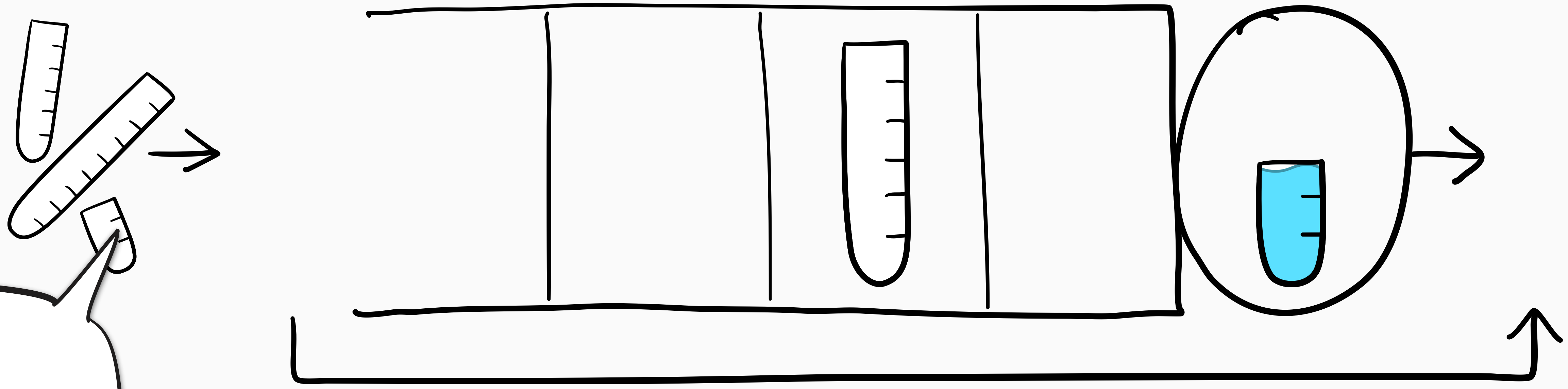
T = *response time*

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



Serve short jobs
before long jobs

Mean scheduling in the M/G/1



M/G arrivals:

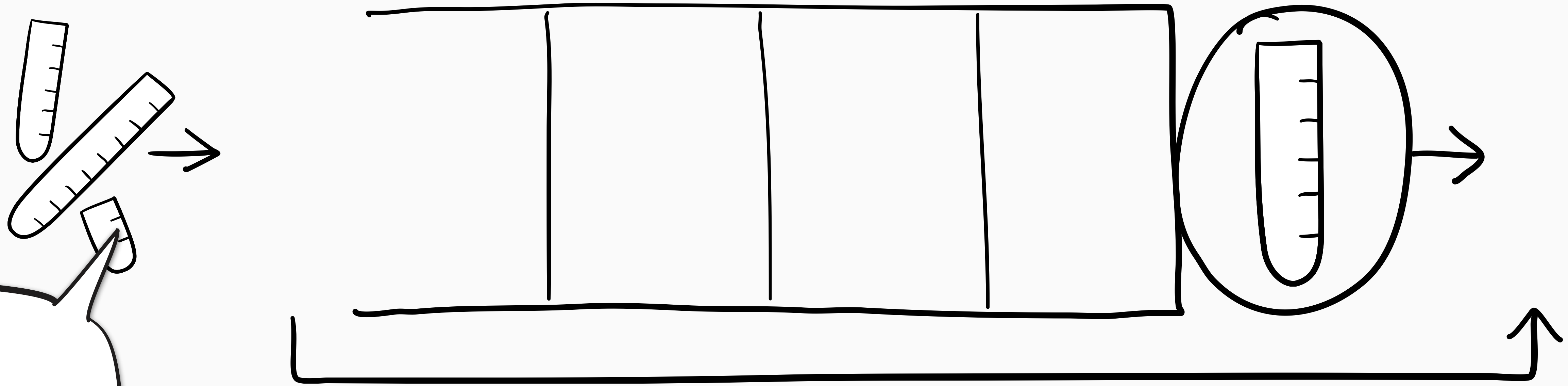
- arrival rate λ
- size distribution S

T = *response time*

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

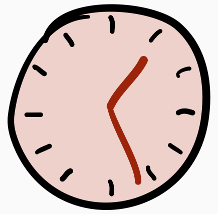
Serve short jobs
before long jobs

Mean scheduling in the M/G/1



M/G arrivals:

- arrival rate λ
- size distribution S

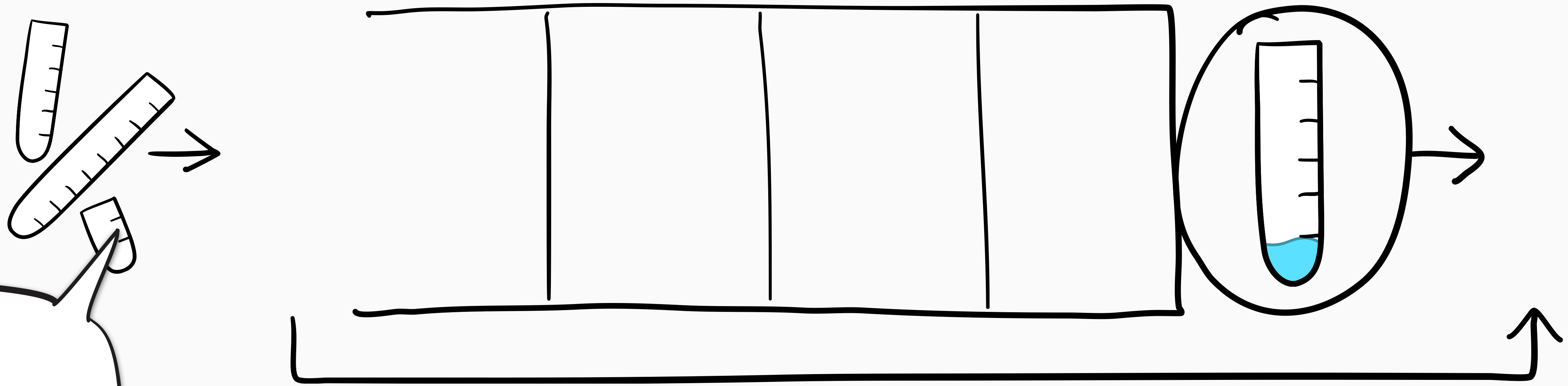


T = *response time*

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

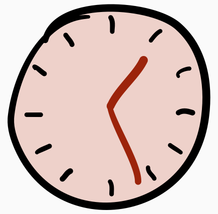
Serve short jobs
before long jobs

Mean scheduling in the M/G/1



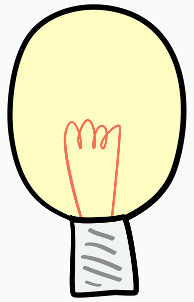
M/G arrivals:

- arrival rate λ
- size distribution S



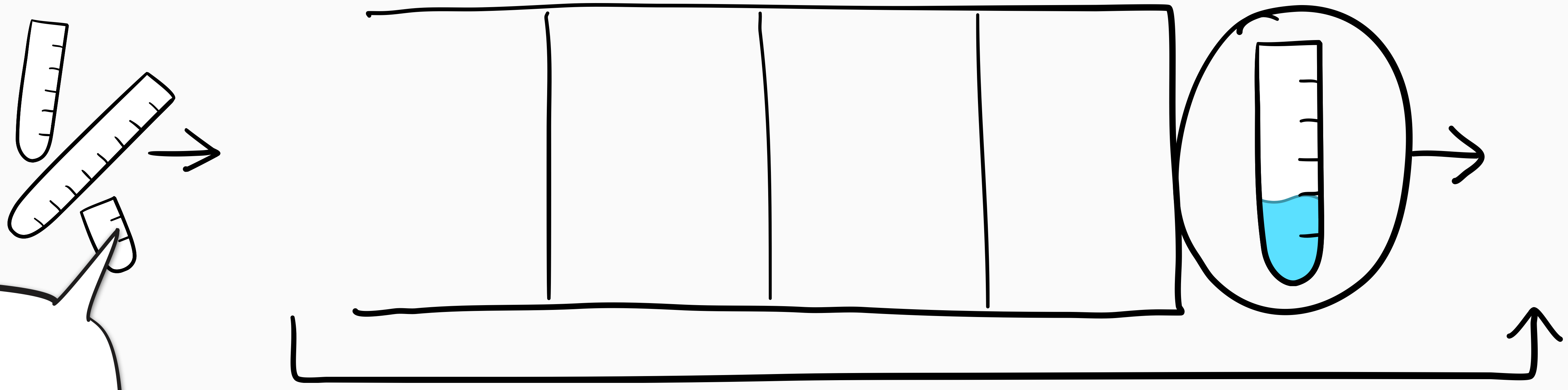
T = *response time*

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



Serve short jobs
before long jobs

Mean scheduling in the M/G/1



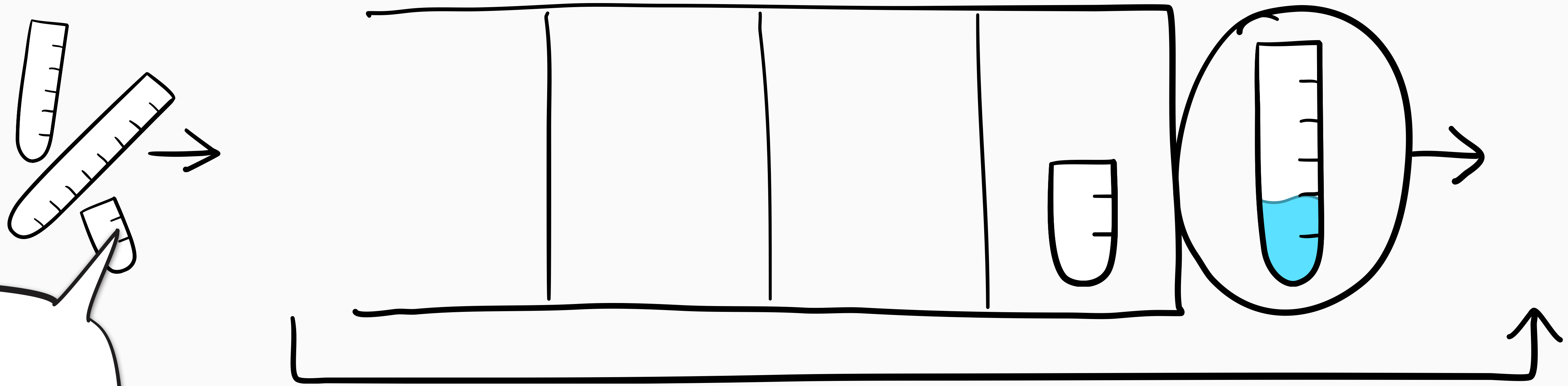
M/G arrivals:
- arrival rate λ
- size distribution S

T = *response time*

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

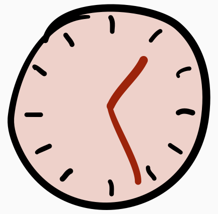
Serve short jobs
before long jobs

Mean scheduling in the M/G/1



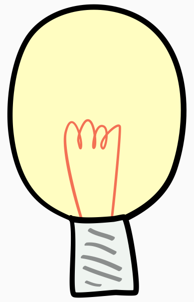
M/G arrivals:

- arrival rate λ
- size distribution S



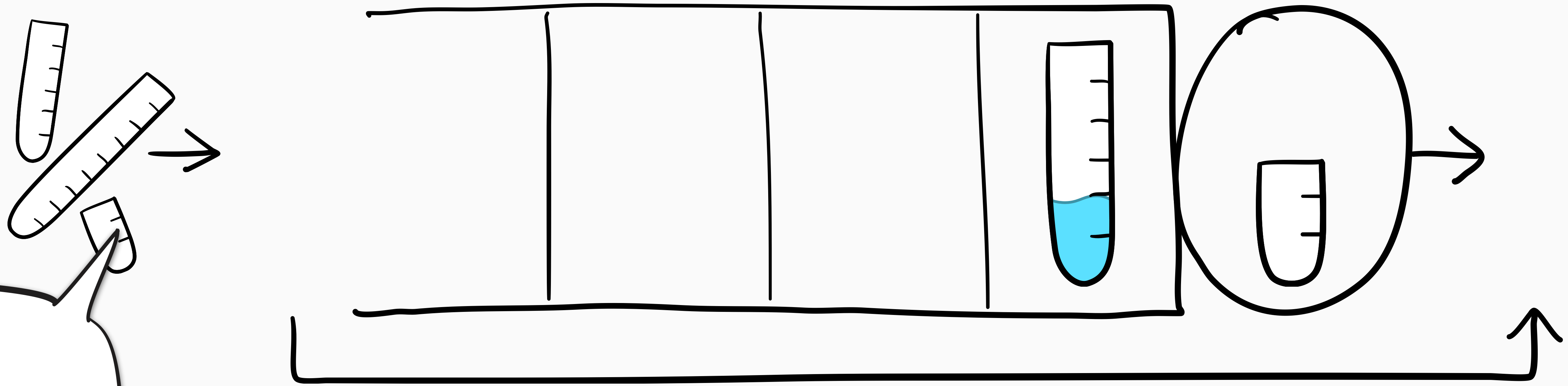
T = *response time*

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



Serve short jobs
before long jobs

Mean scheduling in the M/G/1



M/G arrivals:

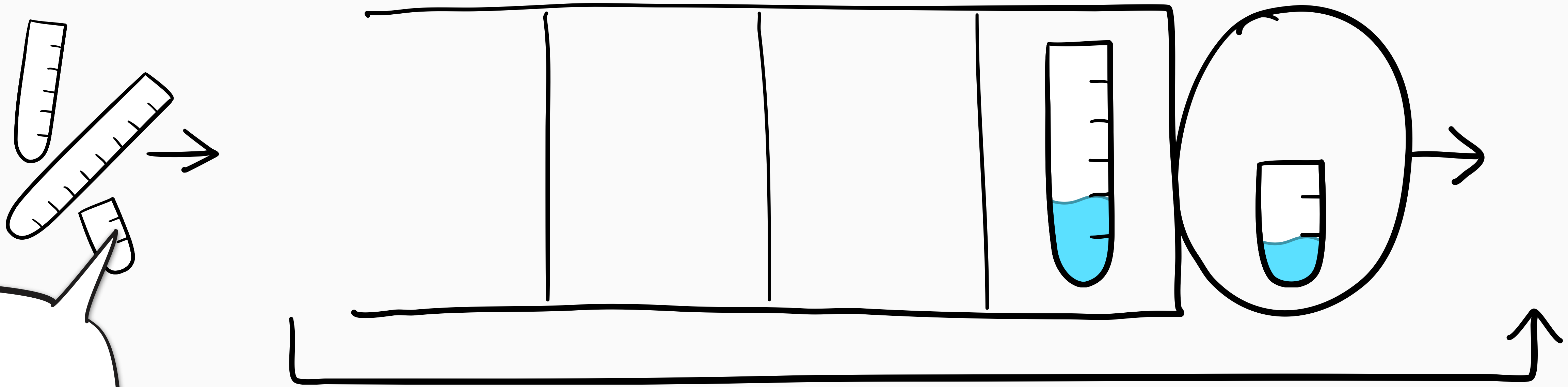
- arrival rate λ
- size distribution S

T = *response time*

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

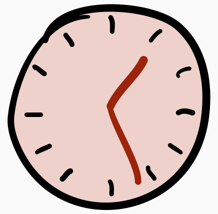
Serve short jobs before long jobs

Mean scheduling in the M/G/1



M/G arrivals:

- arrival rate λ
- size distribution S

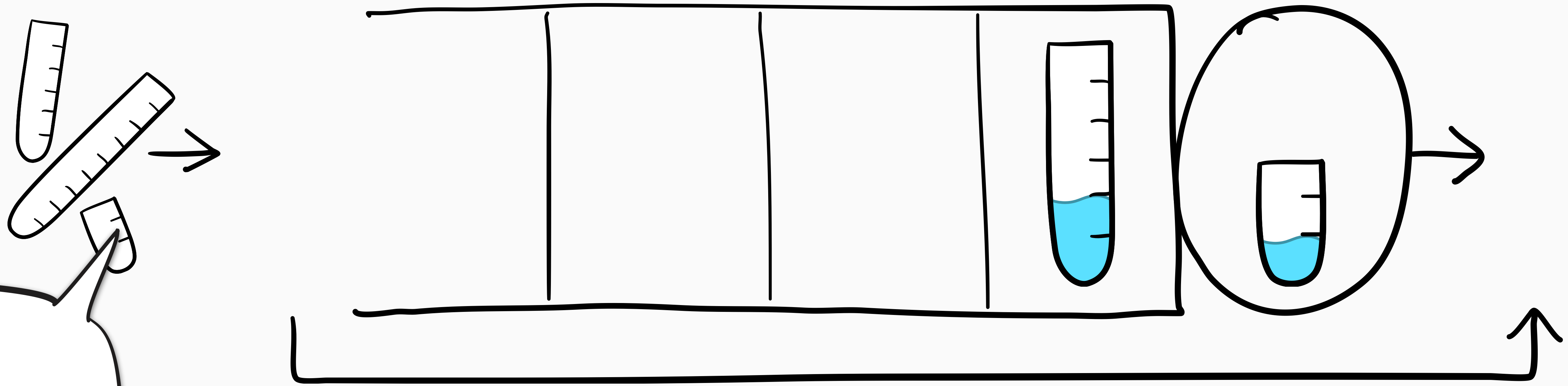


T = *response time*

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

Serve short jobs
before long jobs

Mean scheduling in the M/G/1



M/G arrivals:
- arrival rate λ
- size distribution S

T = *response time*

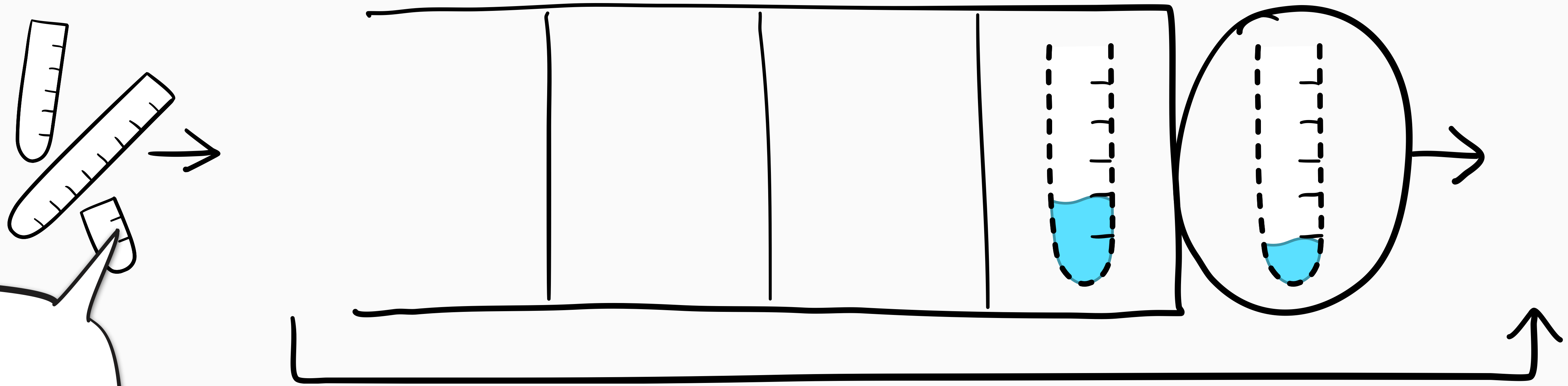
Minimize $\mathbf{E}[T]$?

Serve short jobs before long jobs

shortest remaining processing time

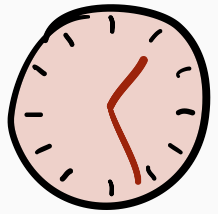
SRPT: minimizes $\mathbf{E}[T]$

Mean scheduling with *unknown sizes*



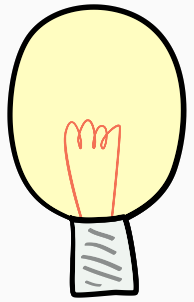
M/G arrivals:

- arrival rate λ
- size distribution S



T = *response time*

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

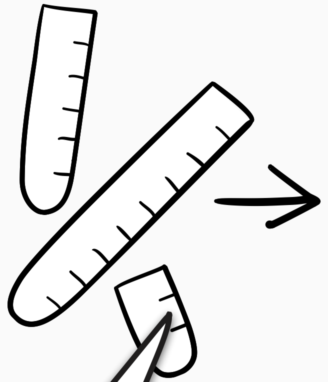


Serve short jobs before long jobs

shortest remaining processing time

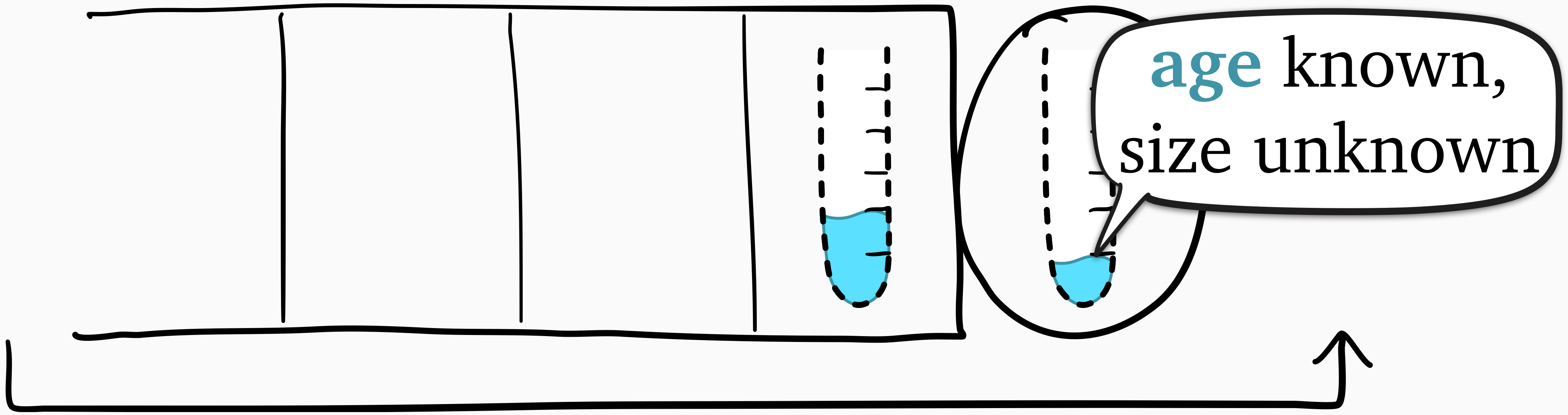
SRPT: minimizes $\mathbf{E}[T]$

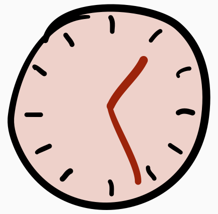
Mean scheduling with *unknown sizes*



M/G arrivals:

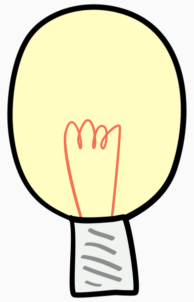
- arrival rate λ
- size distribution S





T = *response time*

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



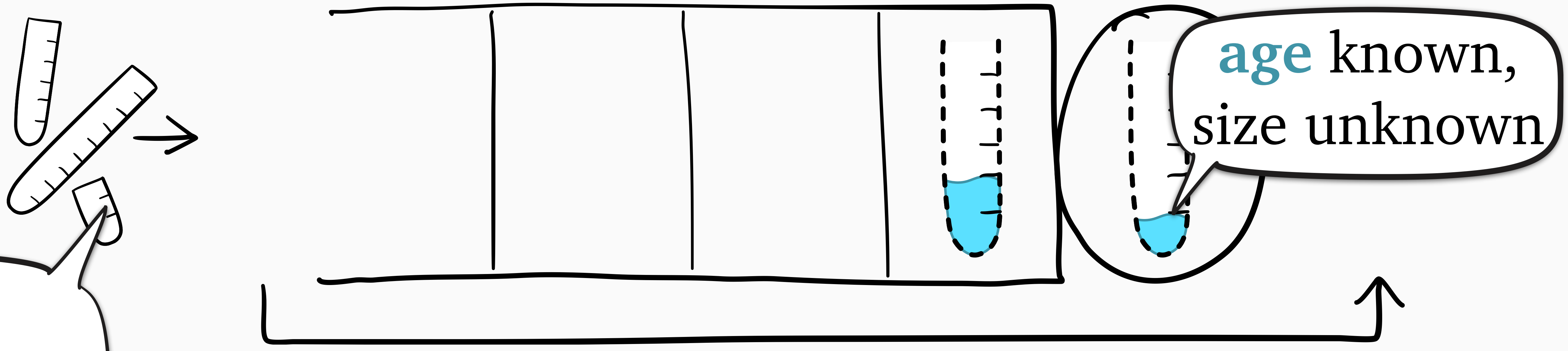
Serve short jobs before long jobs

shortest remaining processing time

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

Mean scheduling with *unknown sizes*

M/G arrivals:
- arrival rate λ
- size distribution S



T = *response time*

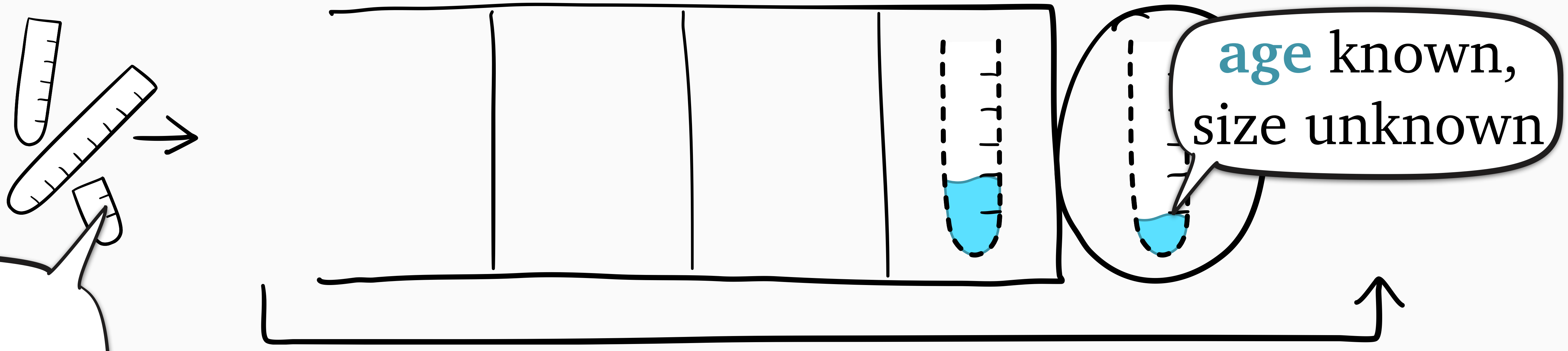
Minimize $\mathbf{E}[T]$?

Serve short jobs before long jobs

shortest remaining processing time

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

Mean scheduling with *unknown sizes*



M/G arrivals:

- arrival rate λ
- size distribution S

T = *response time*

… using *ages* and *size distribution* S

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

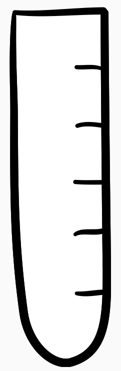
Serve short jobs before long jobs

shortest remaining processing time

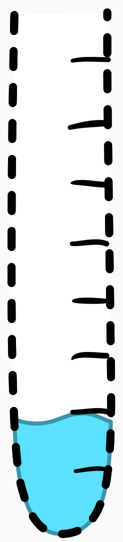
SRPT: minimizes $\mathbf{E}[T]$

Gittins index of a job

1.5-job problem



vs.

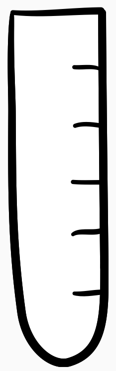


known size r ***unknown*** size

$(S \mid S > \textbf{age})$

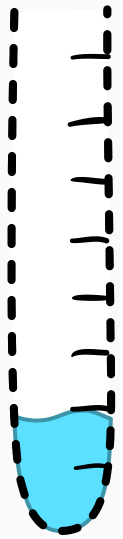
Gittins index of a job

1.5-job problem



known size r

vs.



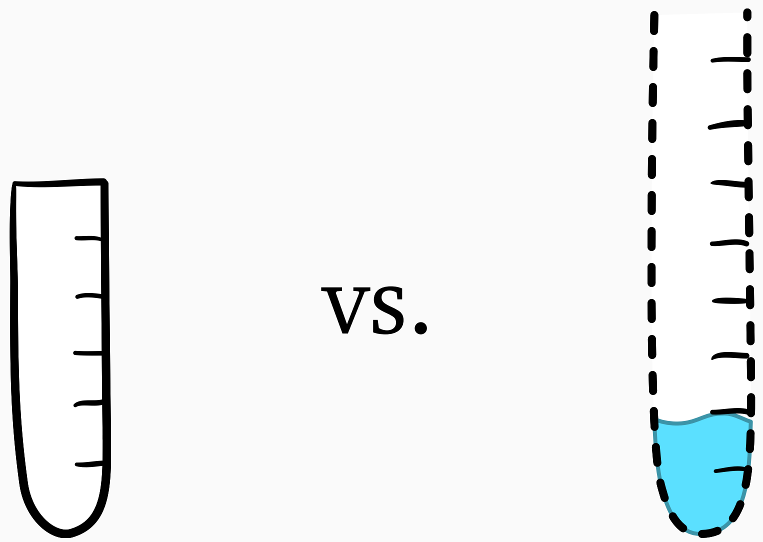
unknown size

$(S \mid S > \textbf{age})$

Key question: what to do in 1.5-job problem?

Gittins index of a job

1.5-job problem



vs.

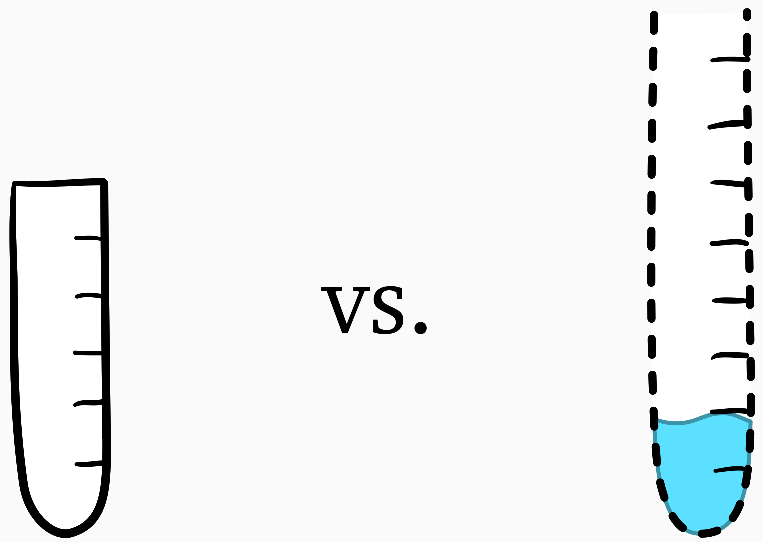
known size r *unknown* size

$(S \mid S > \text{age})$

Key question: what to do in 1.5-job problem?

Gittins index of a job

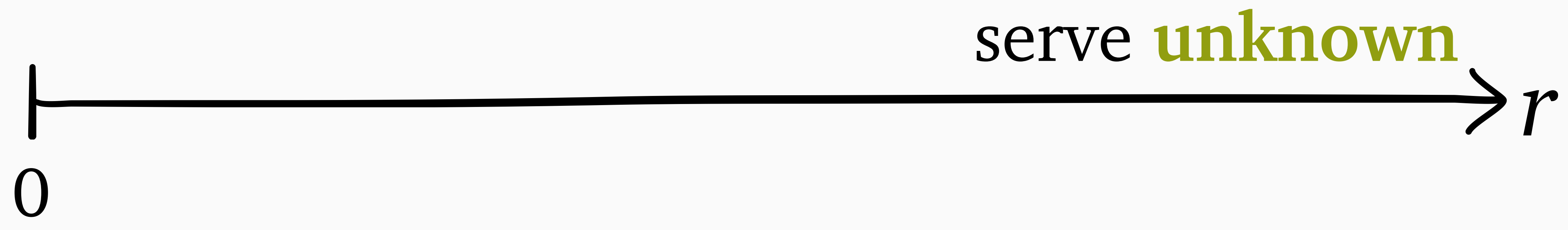
1.5-job problem



known size r *unknown* size

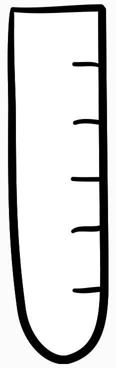
$(S \mid S > \textbf{age})$

Key question: what to do in 1.5-job problem?

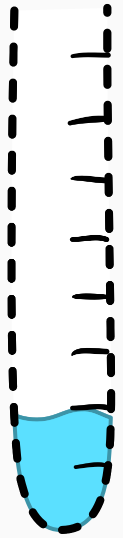


Gittins index of a job

1.5-job problem



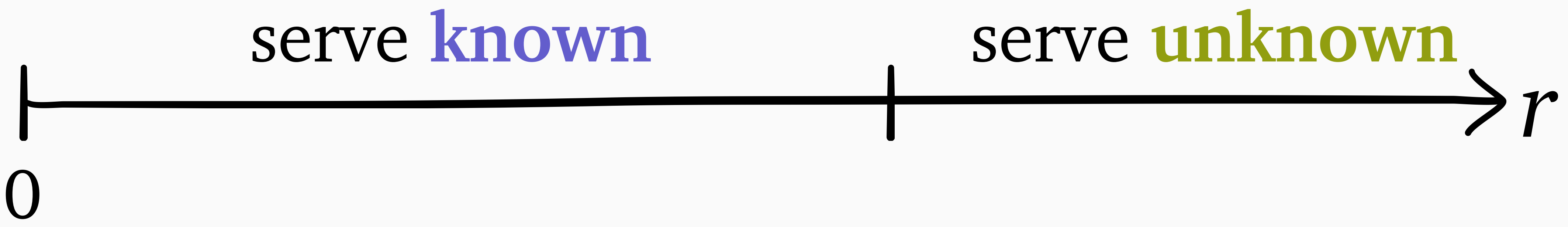
vs.



known size r *unknown* size

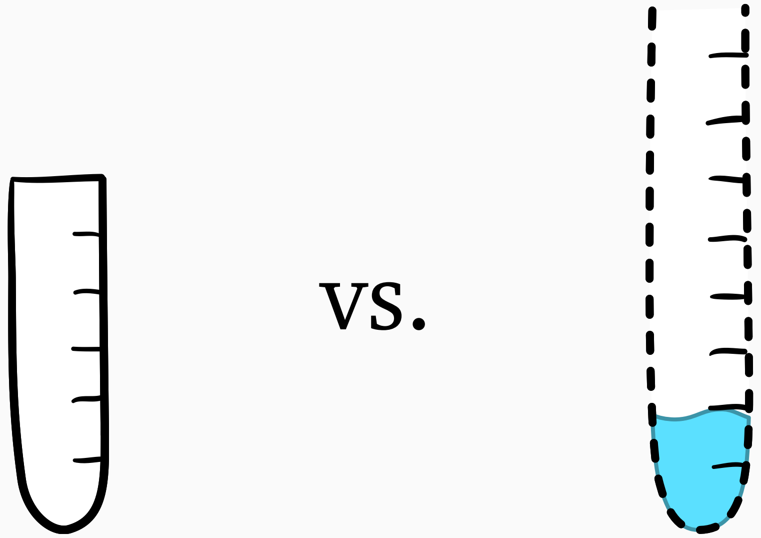
$(S \mid S > \textbf{age})$

Key question: what to do in 1.5-job problem?



Gittins index of a job

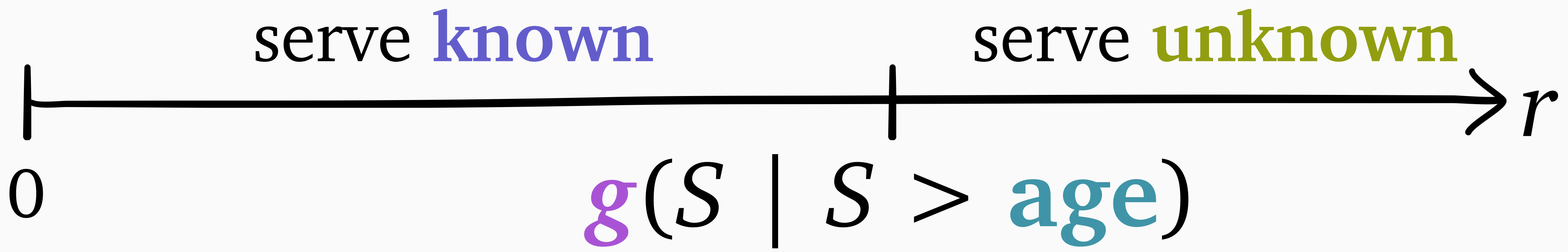
1.5-job problem



known size r *unknown* size

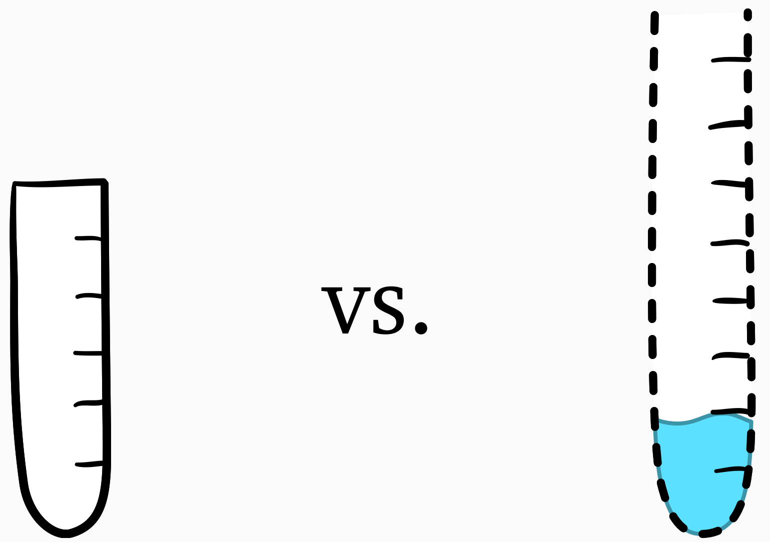
$(S \mid S > \textbf{age})$

Key question: what to do in 1.5-job problem?



Gittins index of a job

1.5-job problem



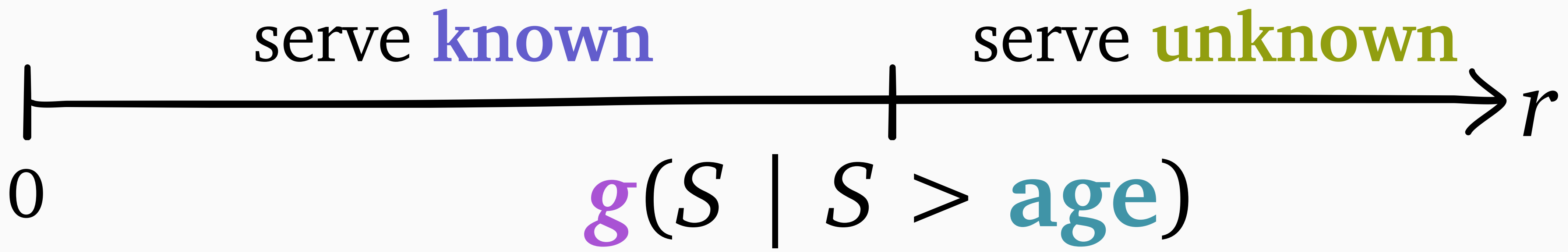
vs.

known size r — *unknown* size $(S \mid S > \text{age})$

1 (50%)

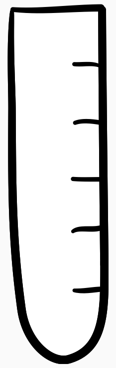
19 (50%)

Key question: what to do in 1.5-job problem?

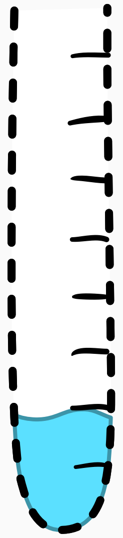


Gittins index of a job

1.5-job problem



vs.



known size r *unknown* size

$(S \mid S > \textbf{age})$

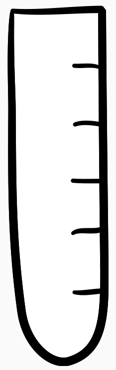
1 (50%)

19 (50%)

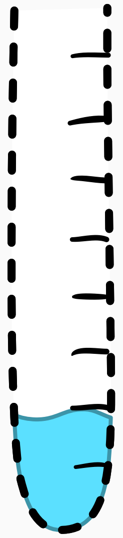
Key question: what to do in 1.5-job problem?

Gittins index of a job

1.5-job problem



vs.



known size r *unknown* size

$(S \mid S > \textbf{age})$

1 (50%)

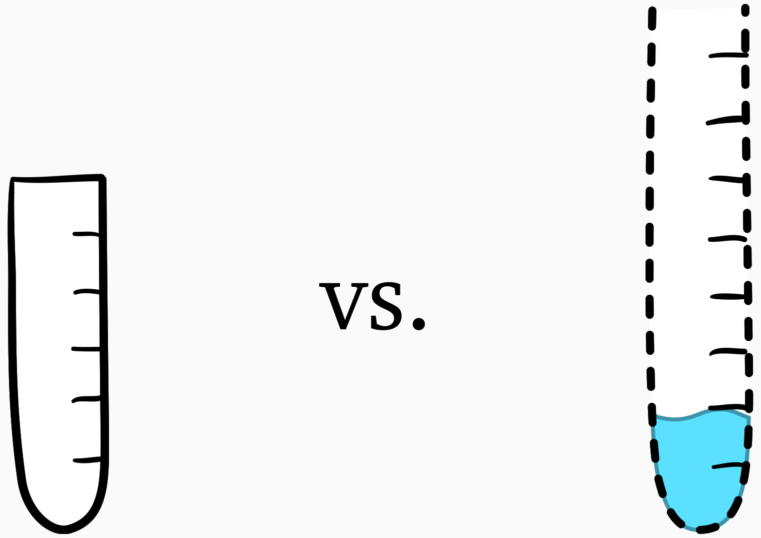
19 (50%)

Key question: what to do in 1.5-job problem?

Serve unknown?

Gittins index of a job

1.5-job problem



known size r vs. *unknown* size

$(S \mid S > \textbf{age})$

1 (50%)

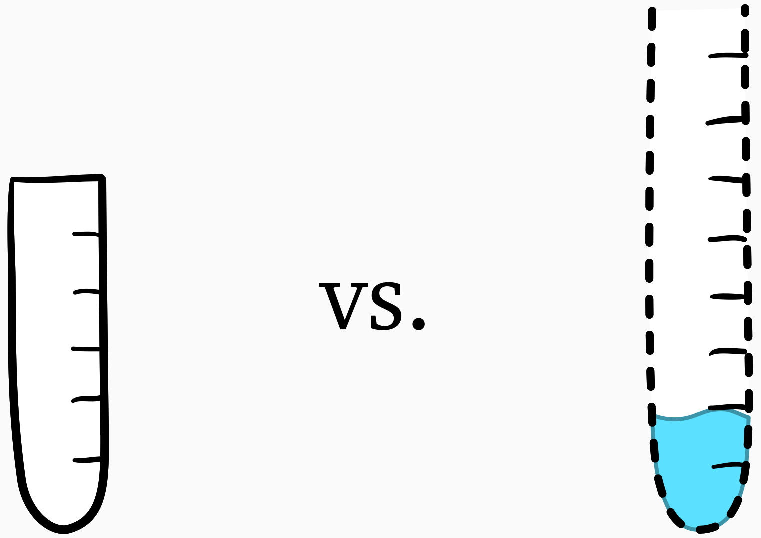
19 (50%)

Key question: what to do in 1.5-job problem?

Serve **unknown**?

Gittins index of a job

1.5-job problem

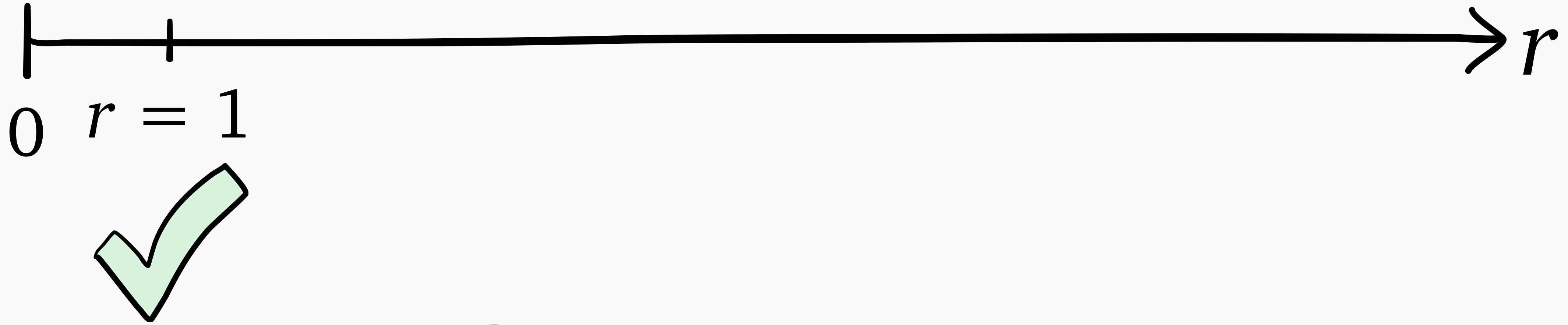


known size r ***unknown*** size

$(S \mid S > \textbf{age})$

1	(50%)
19	(50%)

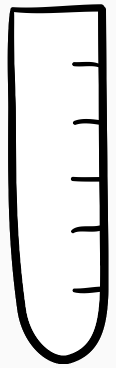
Key question: what to do in 1.5-job problem?



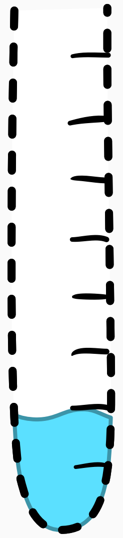
Serve **unknown**?

Gittins index of a job

1.5-job problem



vs.



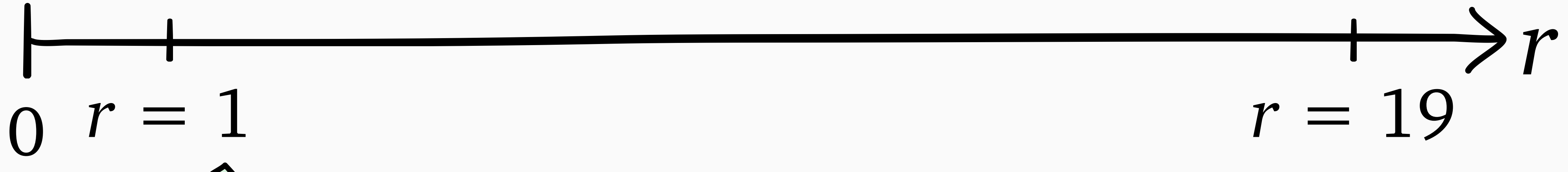
known size r *unknown* size

$(S \mid S > \textbf{age})$

1 (50%)

19 (50%)

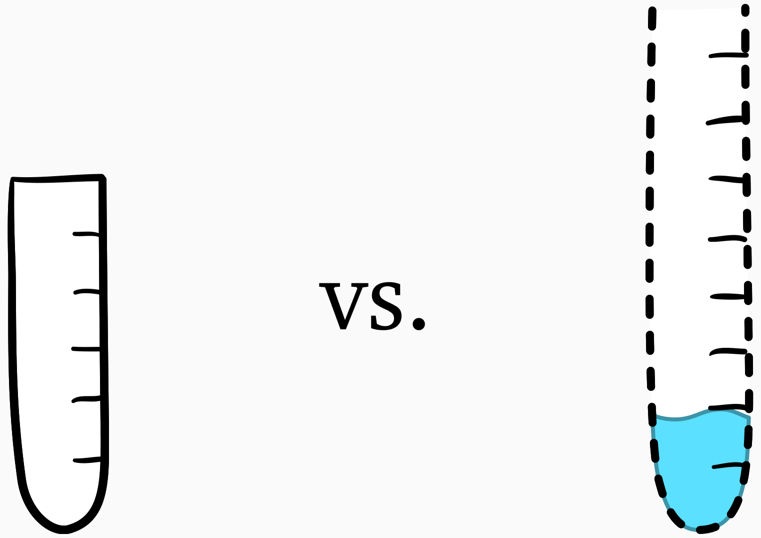
Key question: what to do in 1.5-job problem?



Serve unknown?

Gittins index of a job

1.5-job problem



known size r vs. *unknown* size

$(S \mid S > \text{age})$

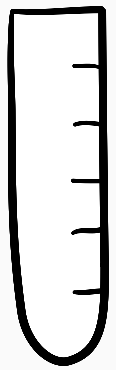
1	(50%)
19	(50%)

Key question: what to do in 1.5-job problem?

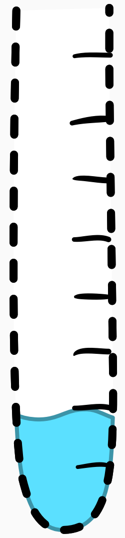
Serve unknown?

Gittins index of a job

1.5-job problem



vs.

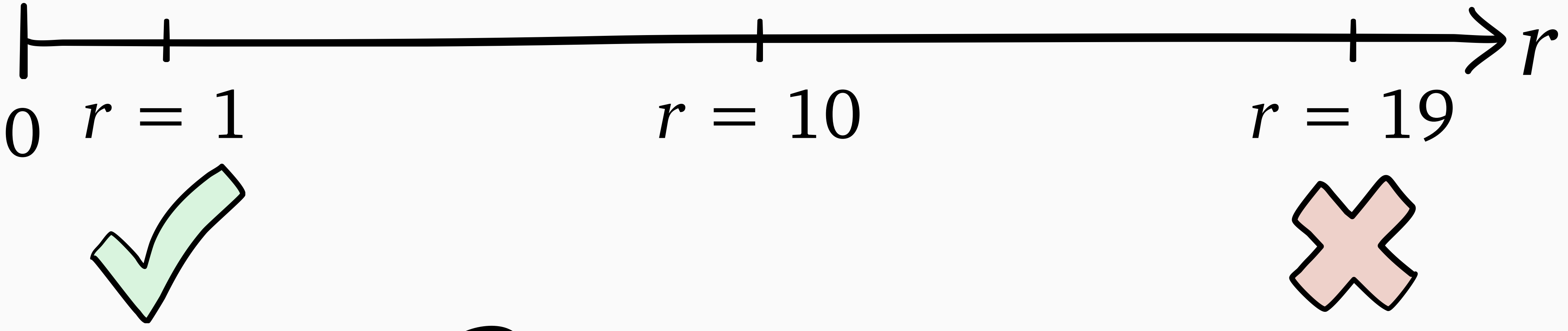


known size r *unknown* size

$(S \mid S > \mathbf{age})$

1	(50%)
19	(50%)

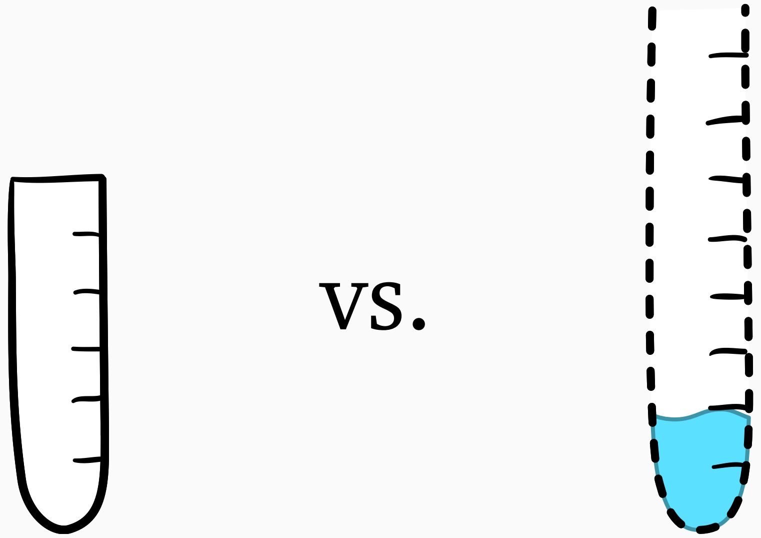
Key question: what to do in 1.5-job problem?



Serve unknown?

Gittins index of a job

1.5-job problem



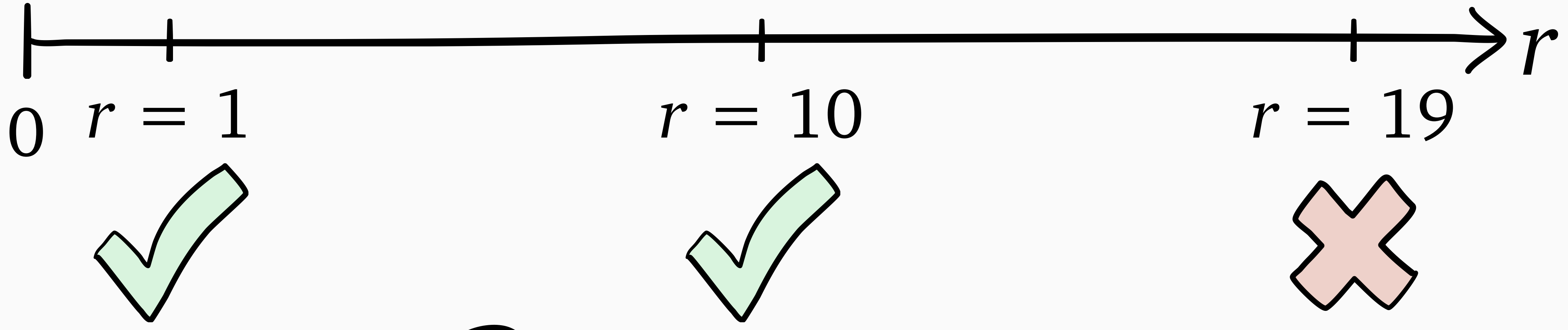
known size r *unknown* size

$(S \mid S > \text{age})$

1 (50%)

19 (50%)

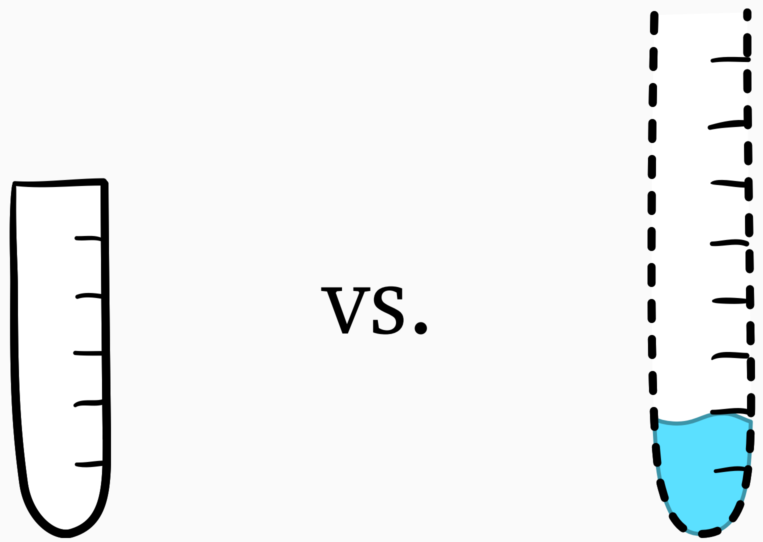
Key question: what to do in 1.5-job problem?



Serve unknown?

Gittins index of a job

1.5-job problem



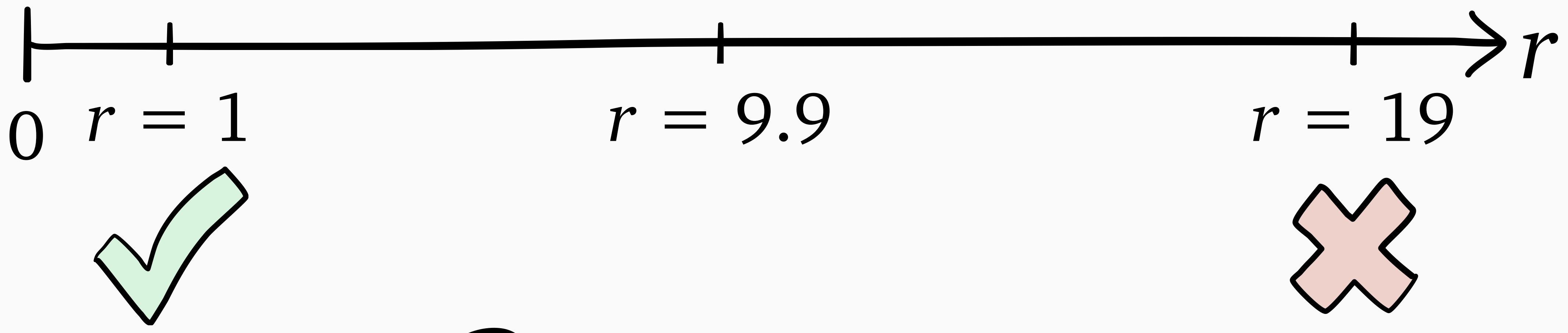
known size r *unknown* size

$(S \mid S > \textbf{age})$

1 (50%)

19 (50%)

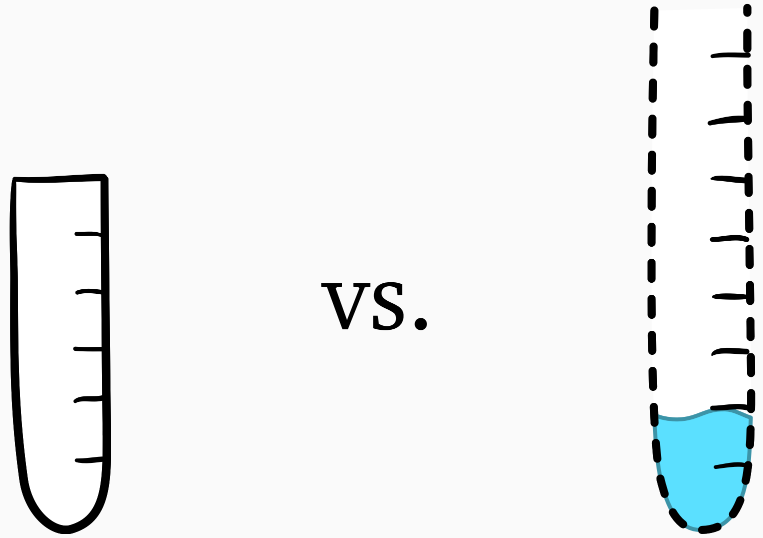
Key question: what to do in 1.5-job problem?



Serve **unknown**?

Gittins index of a job

1.5-job problem



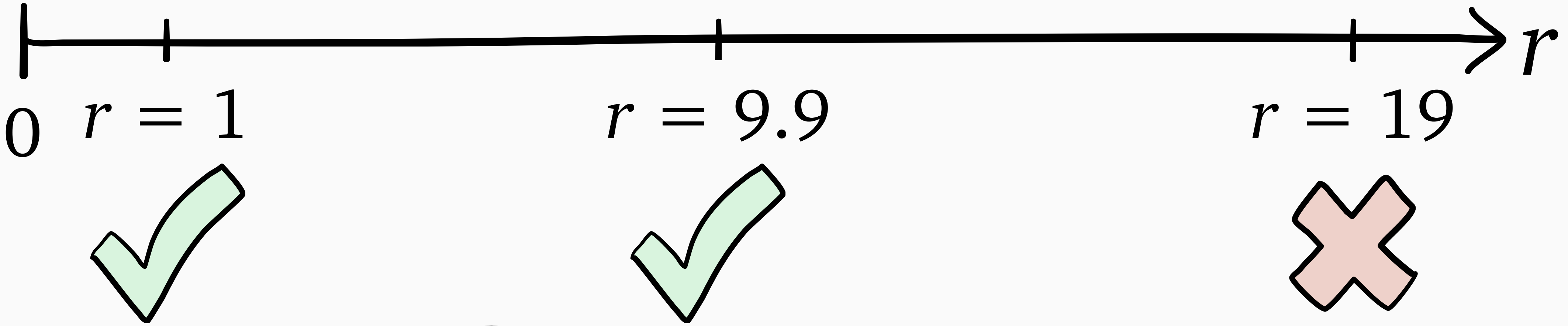
vs.

known size r — *unknown* size

$(S \mid S > \textbf{age})$

1 (50%)
19 (50%)

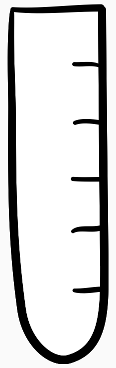
Key question: what to do in 1.5-job problem?



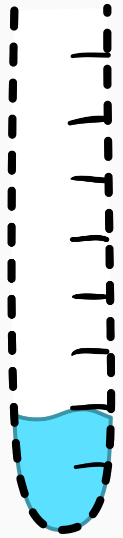
Serve **unknown**?

Gittins index of a job

1.5-job problem



vs.



known size r *unknown* size

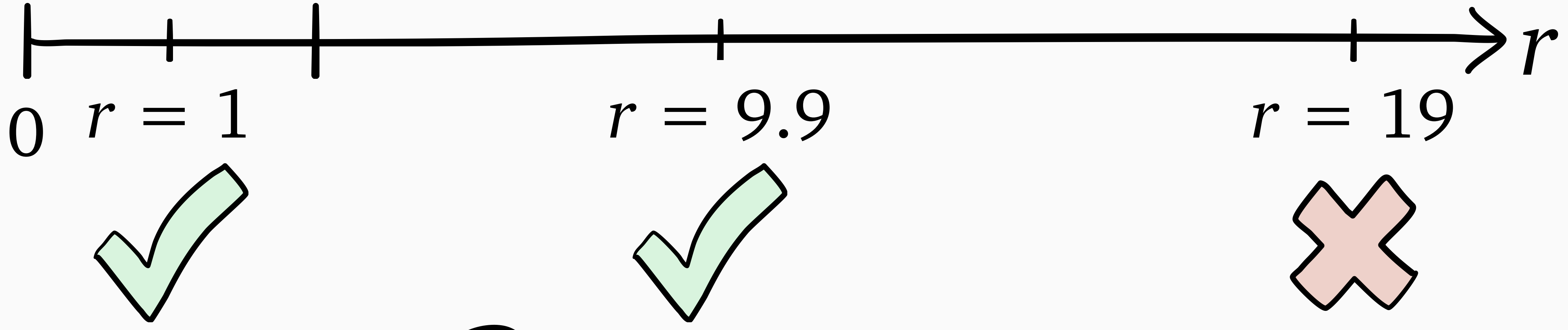
$(S \mid S > \text{age})$

1 (50%)

19 (50%)

Key question: what to do in 1.5-job problem?

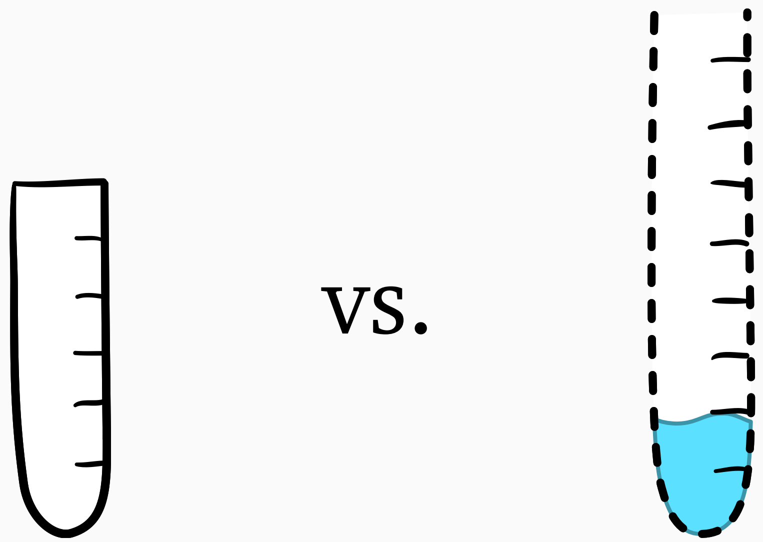
$g = 2$



Serve unknown?

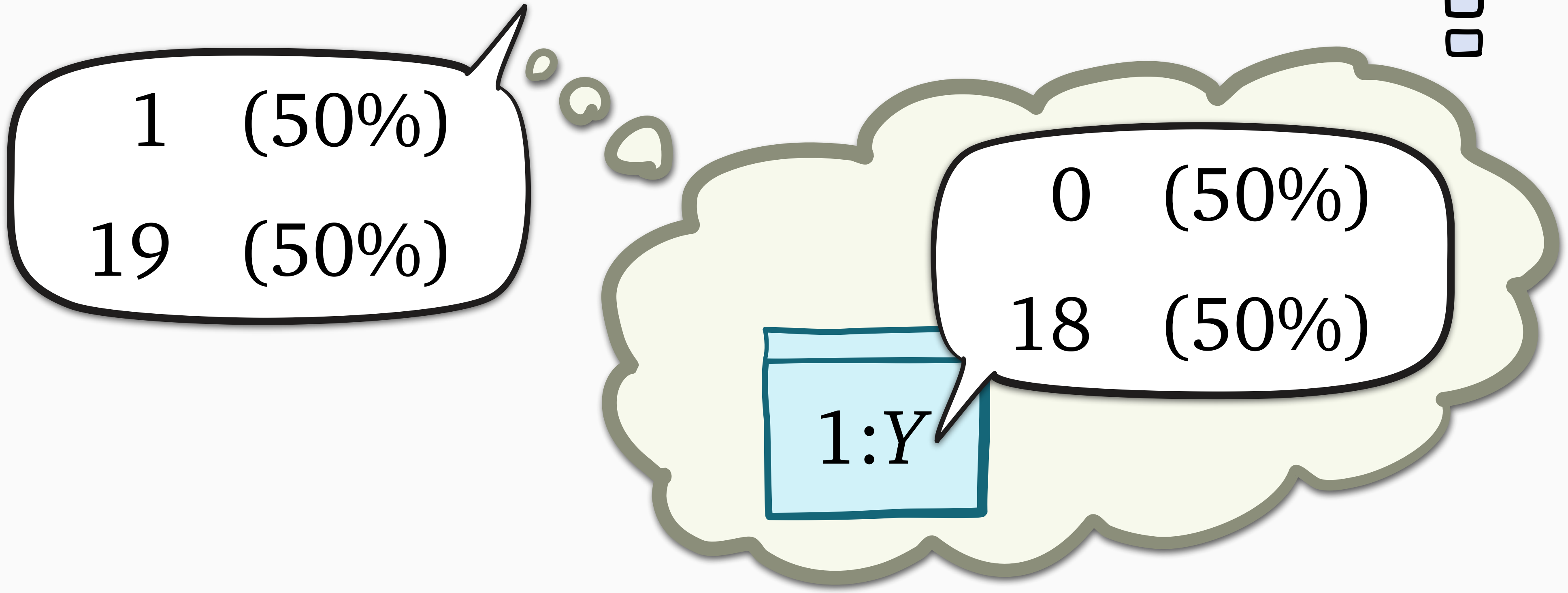
Gittins index of a job

1.5-job problem

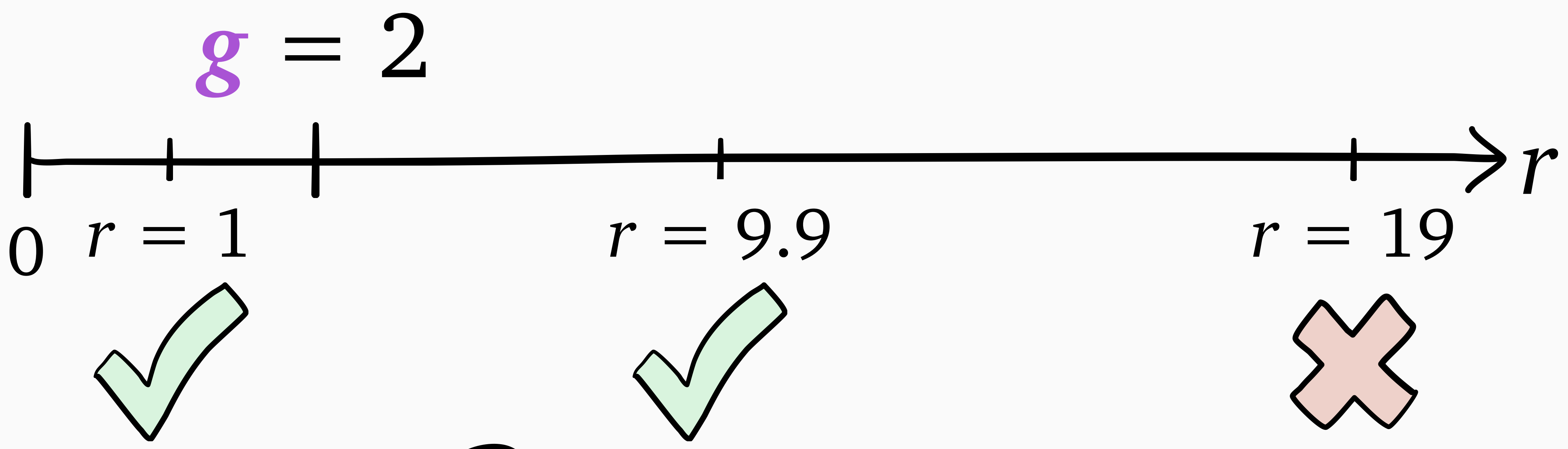


known size r *unknown* size

$(S \mid S > \textbf{age})$



Key question: what to do in 1.5-job problem?



Serve **unknown**?

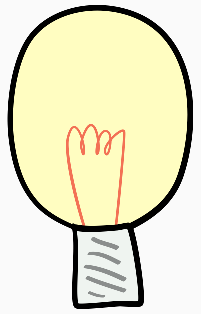
What is the **Gittins index**?

Why is **Gittins** optimal?

What is *(and isn't)* covered by classical **Gittins** theory?

How might we apply **Gittins** *beyond* the classical theory?

What is the **Gittins index**?



The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

What is *(and isn't)* covered by classical **Gittins** theory?

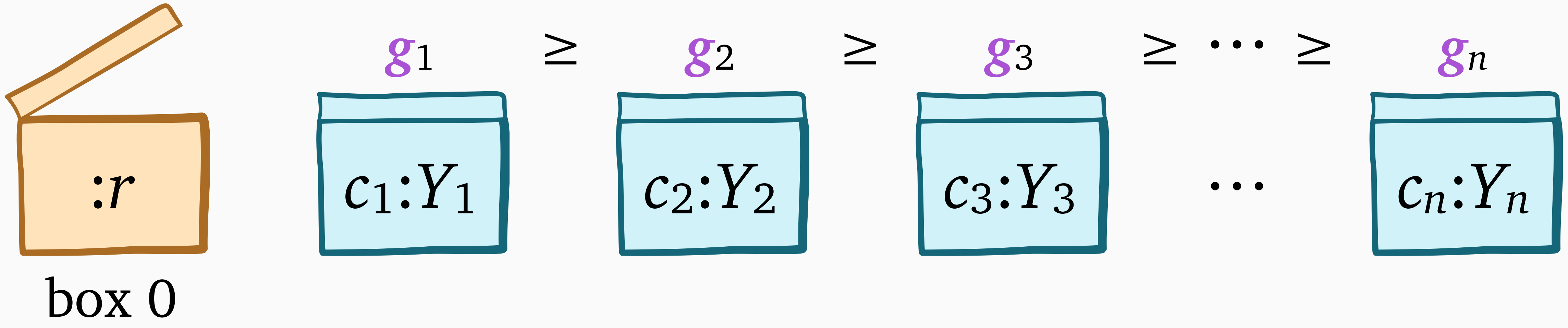
How might we apply **Gittins** *beyond* the classical theory?

Why is the **Gittins index** optimal?

$$g_1 \geq g_2 \geq g_3 \geq \cdots \geq g_n$$

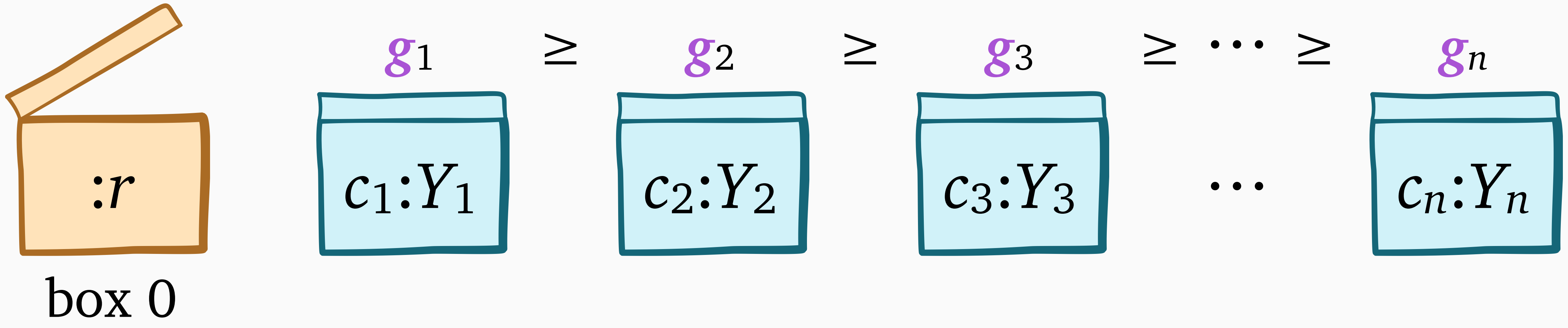
$$c_1{:}Y_1 \quad c_2{:}Y_2 \quad c_3{:}Y_3 \quad \cdots \quad c_n{:}Y_n$$

Why is the **Gittins index** optimal?



Why is the **Gittins index** optimal?

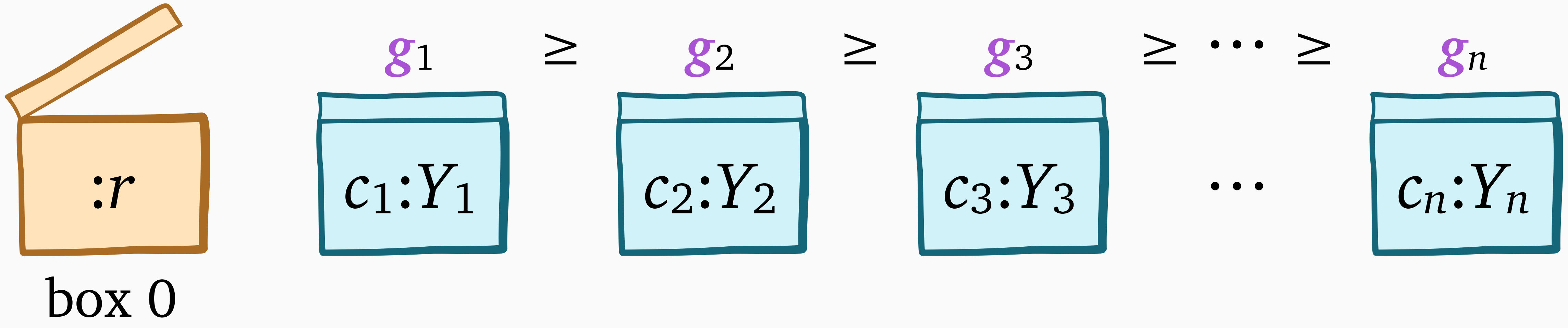
Approach: start at $r = \infty$, then decrease to $r = 0$



Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

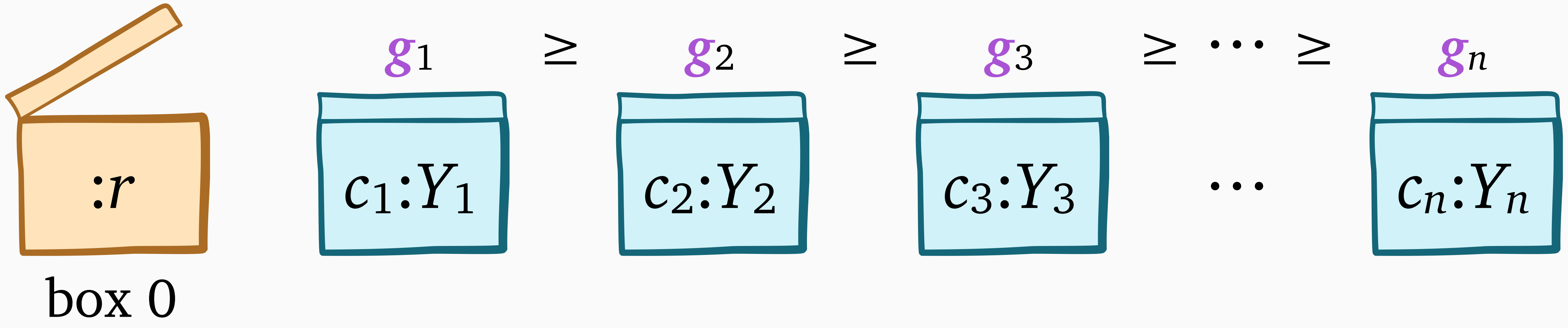
- $r \geq g_1$:



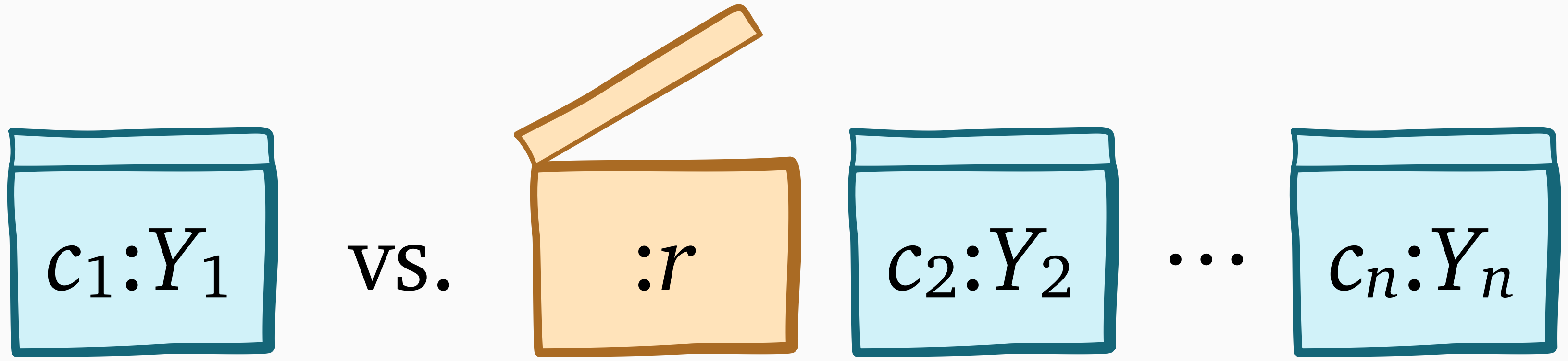
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$:



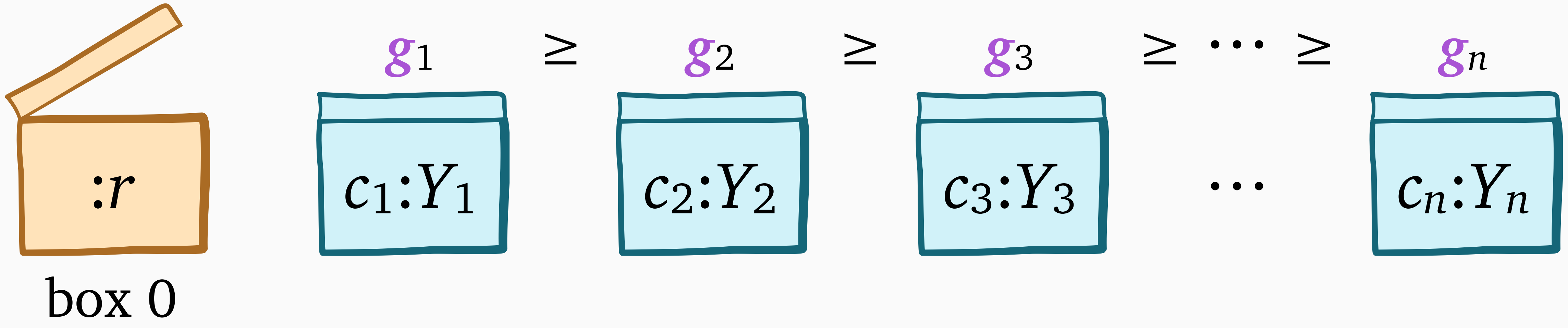
Box 1's perspective



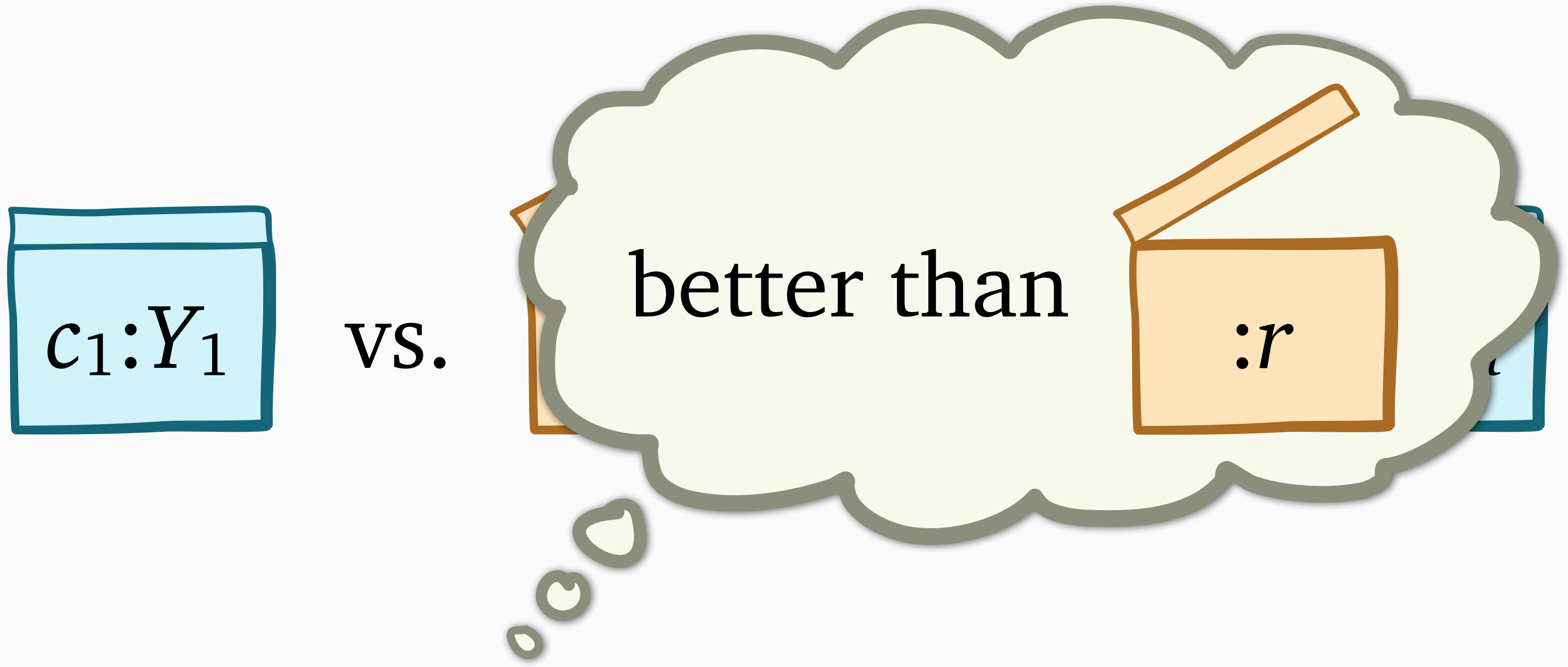
Why is the **Gittins index** optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$:



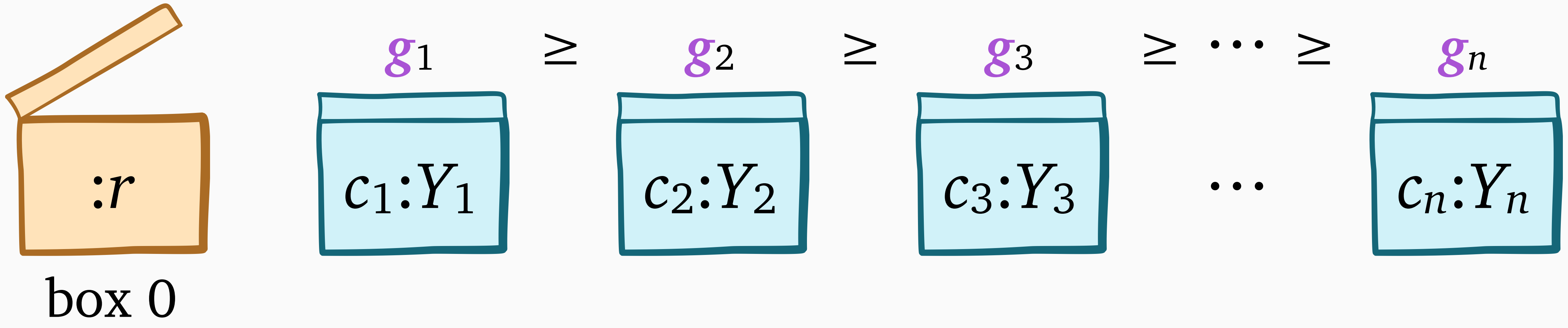
Box 1's perspective



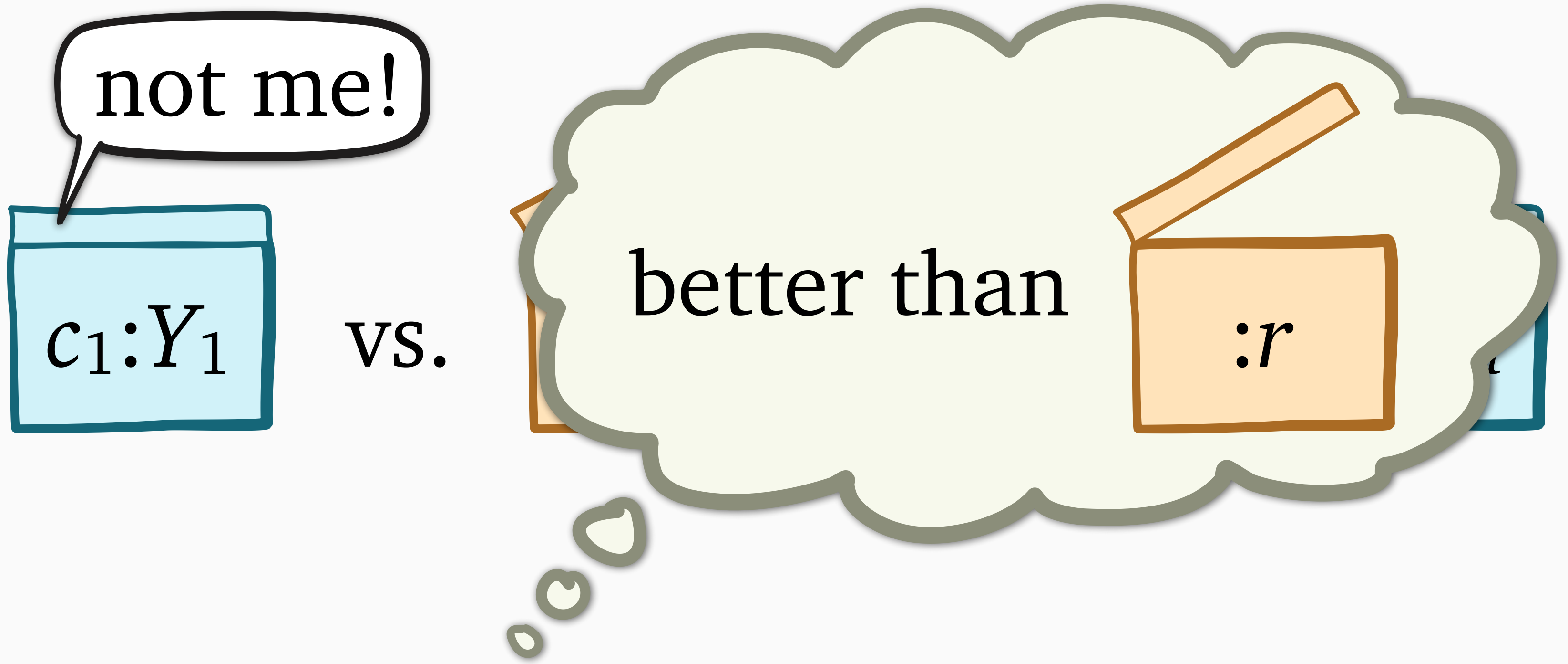
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$:



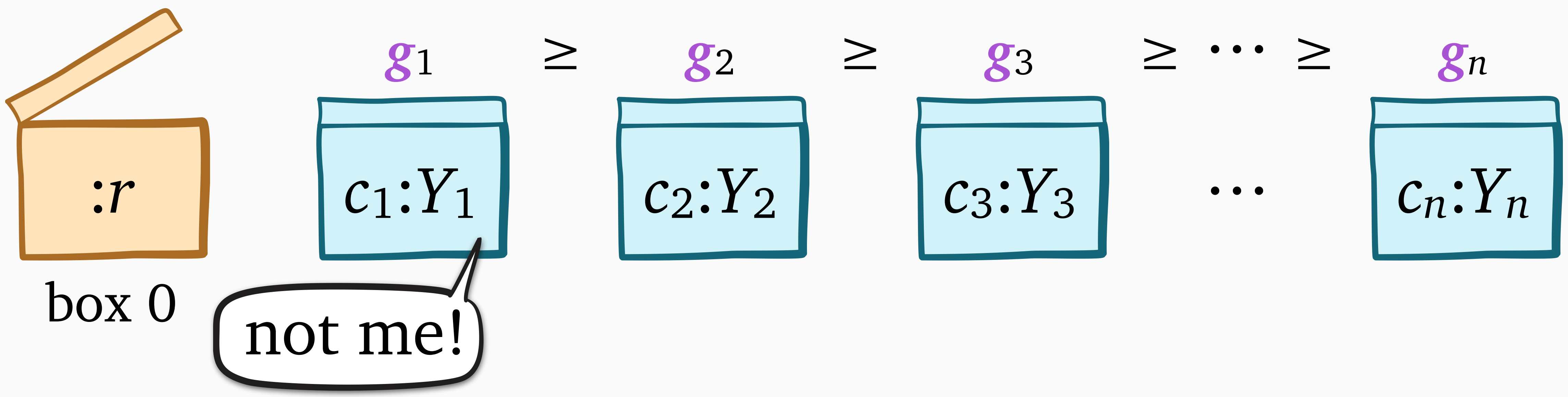
Box 1's perspective



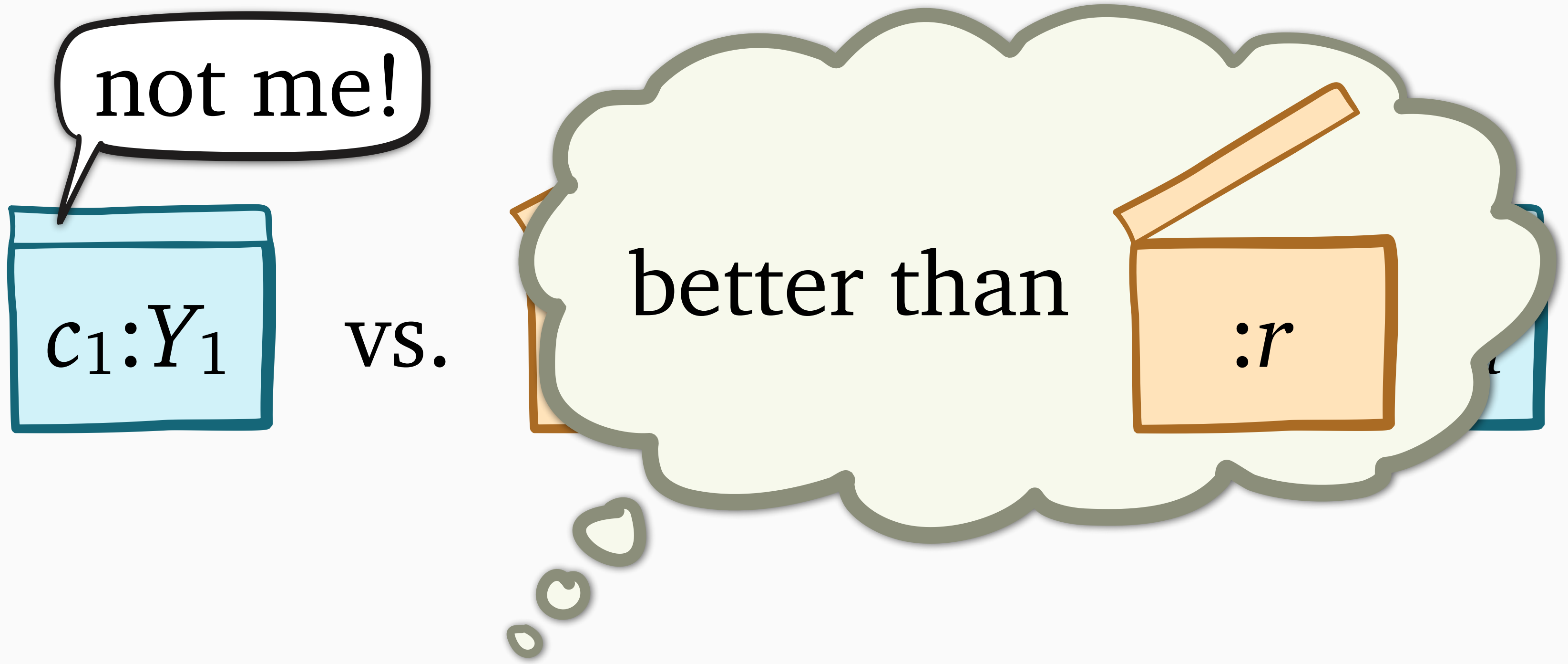
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$:



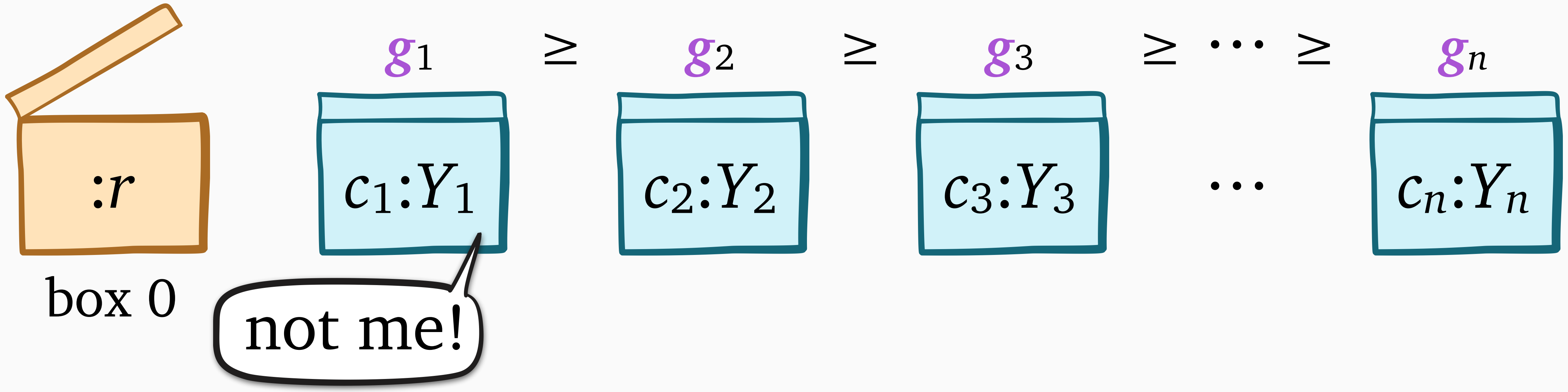
Box 1's perspective



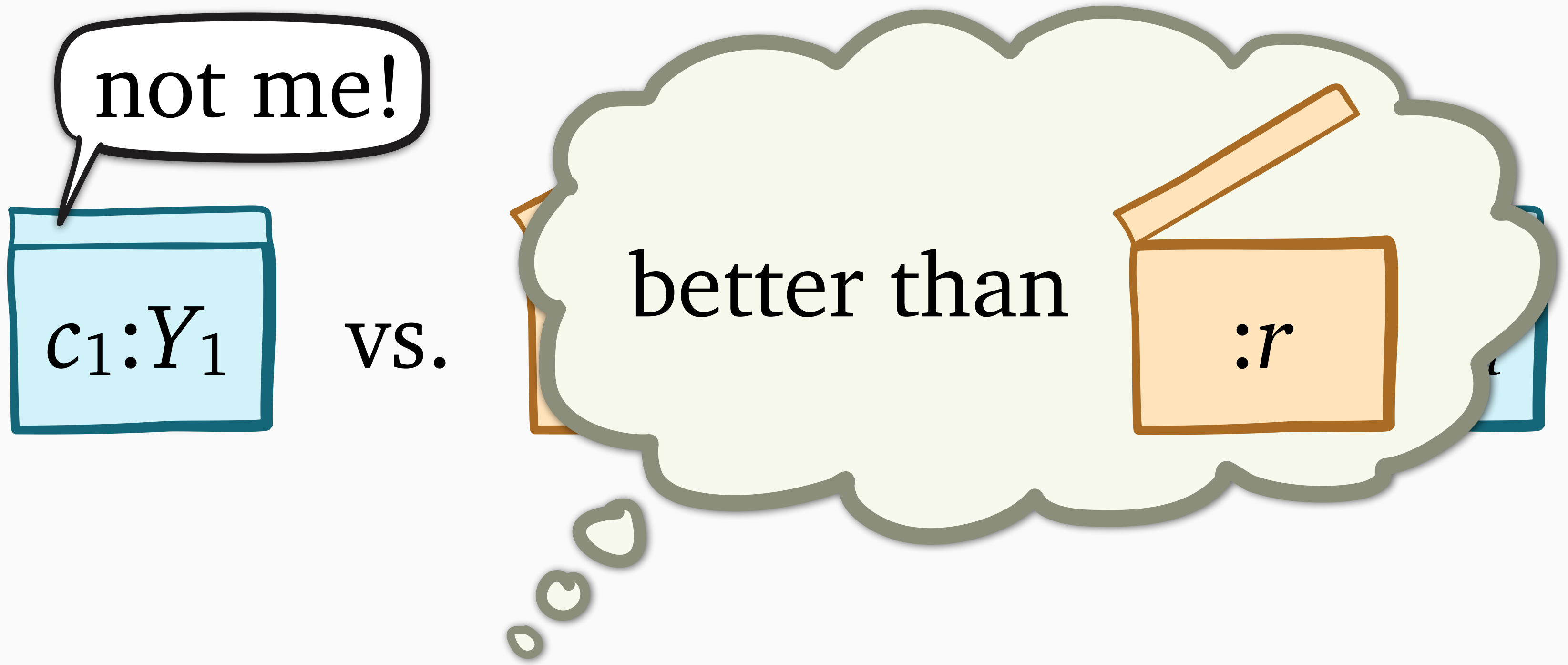
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

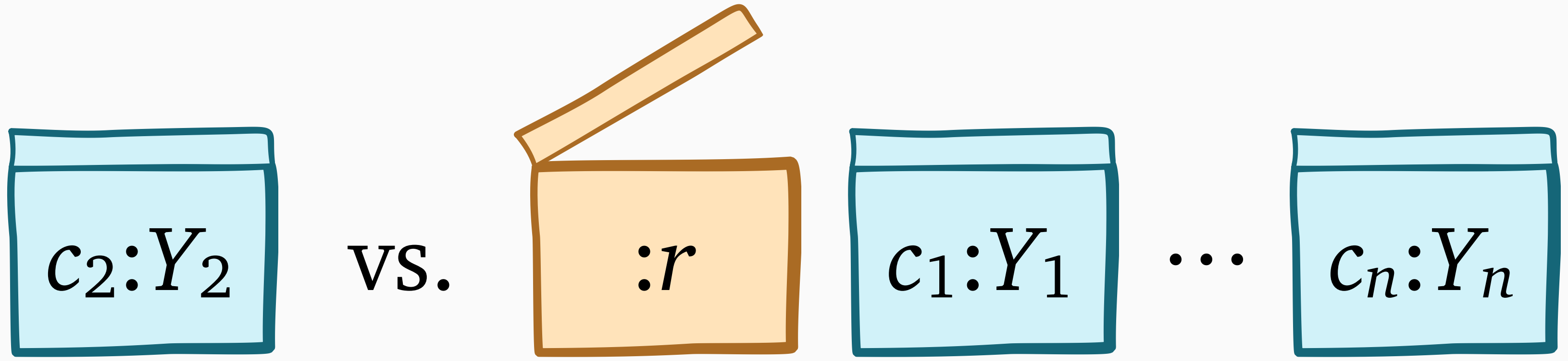
- $r \geq g_1$:



Box 1's perspective



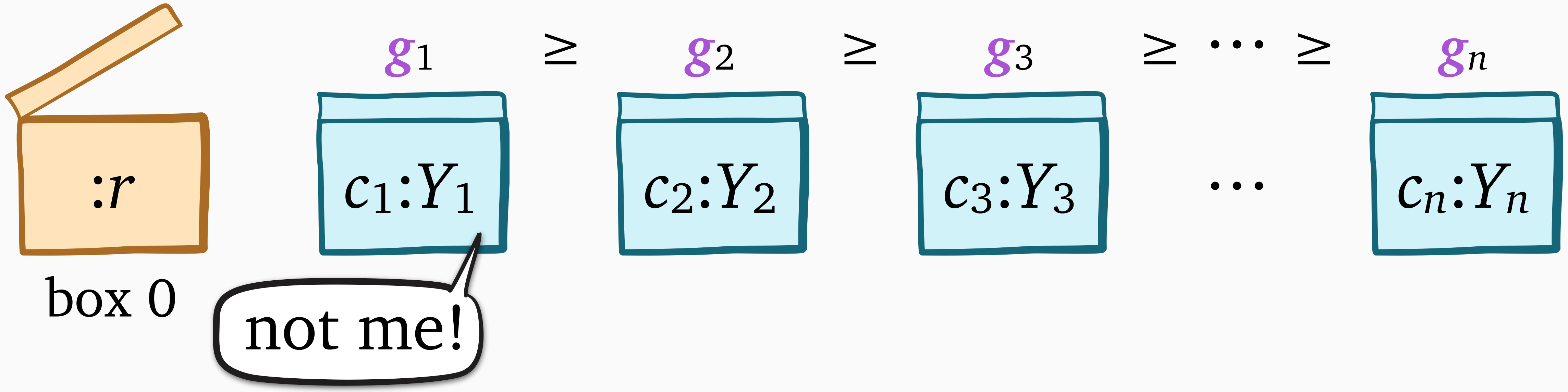
Box 2's perspective



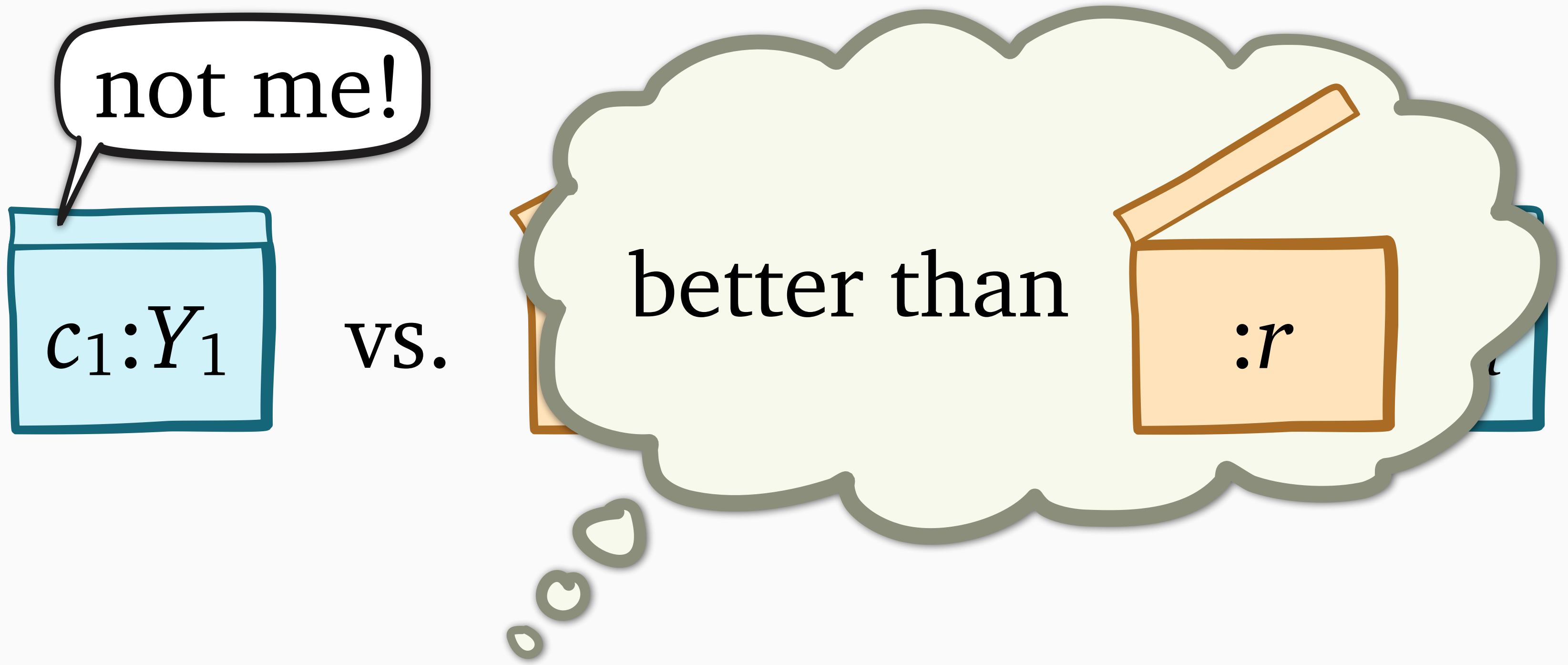
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

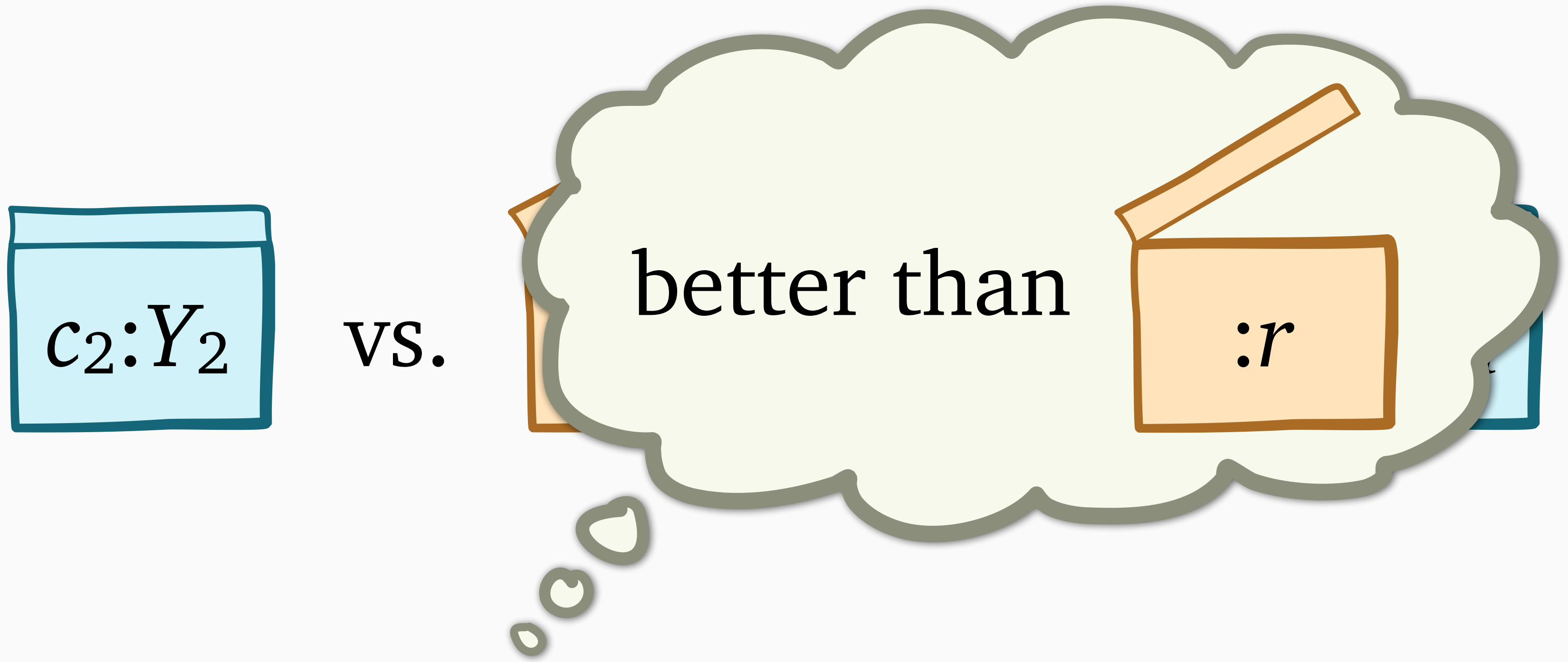
- $r \geq g_1$:



Box 1's perspective



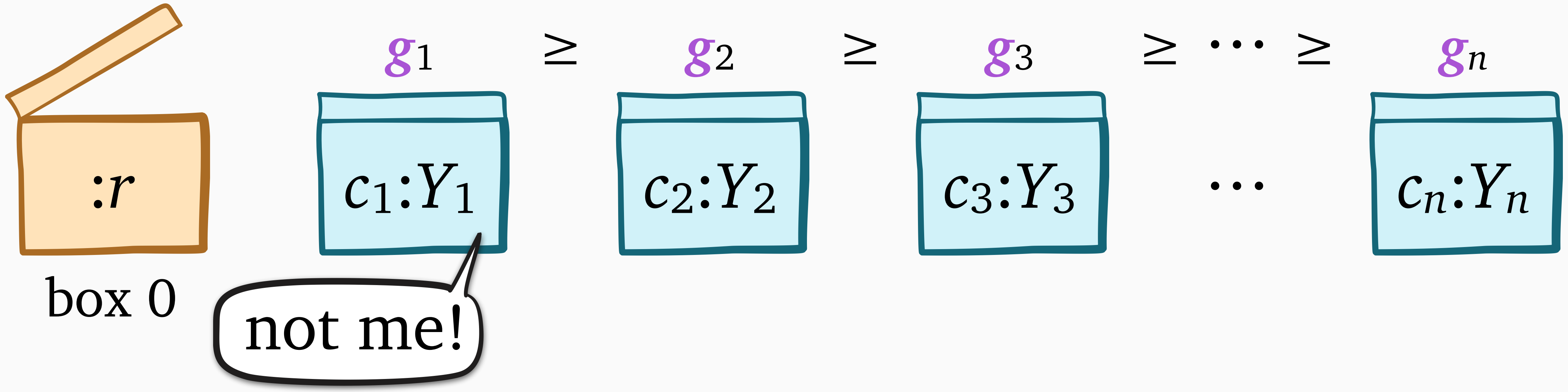
Box 2's perspective



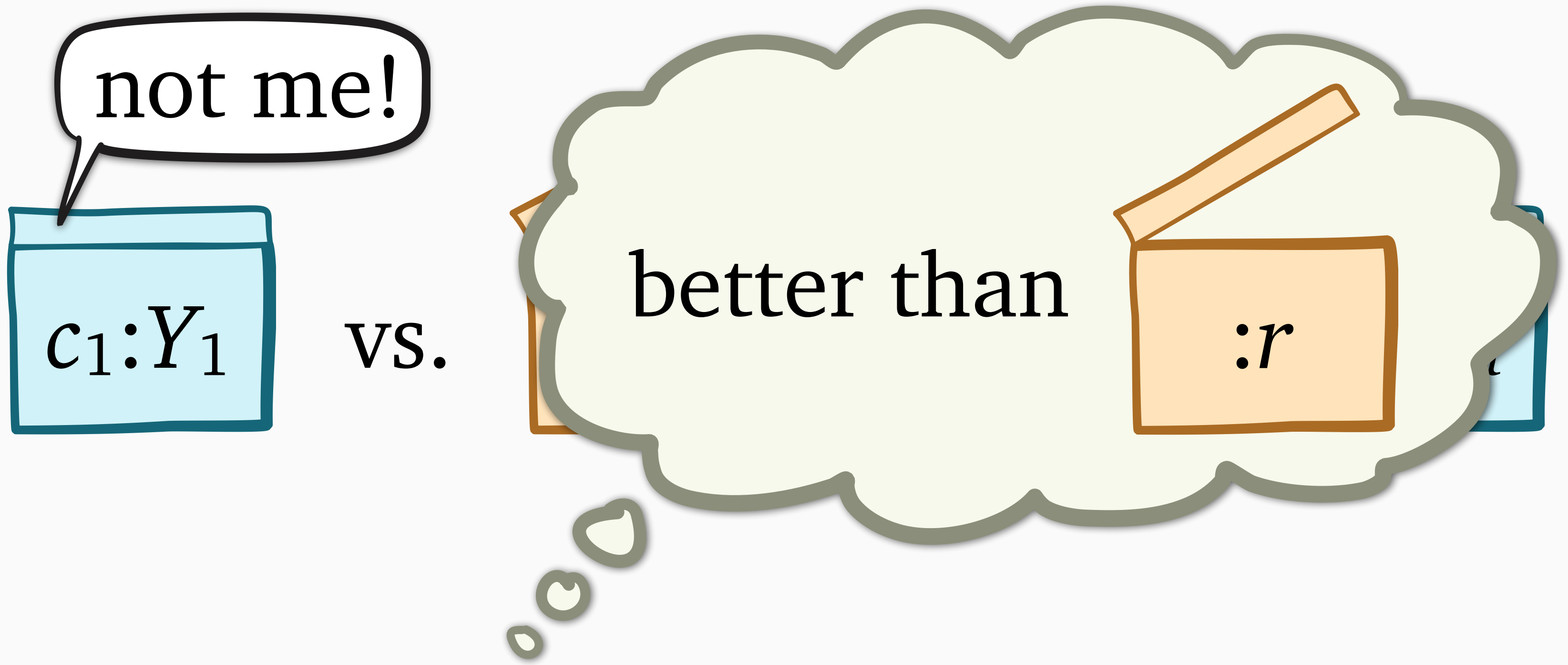
Why is the **Gittins index** optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

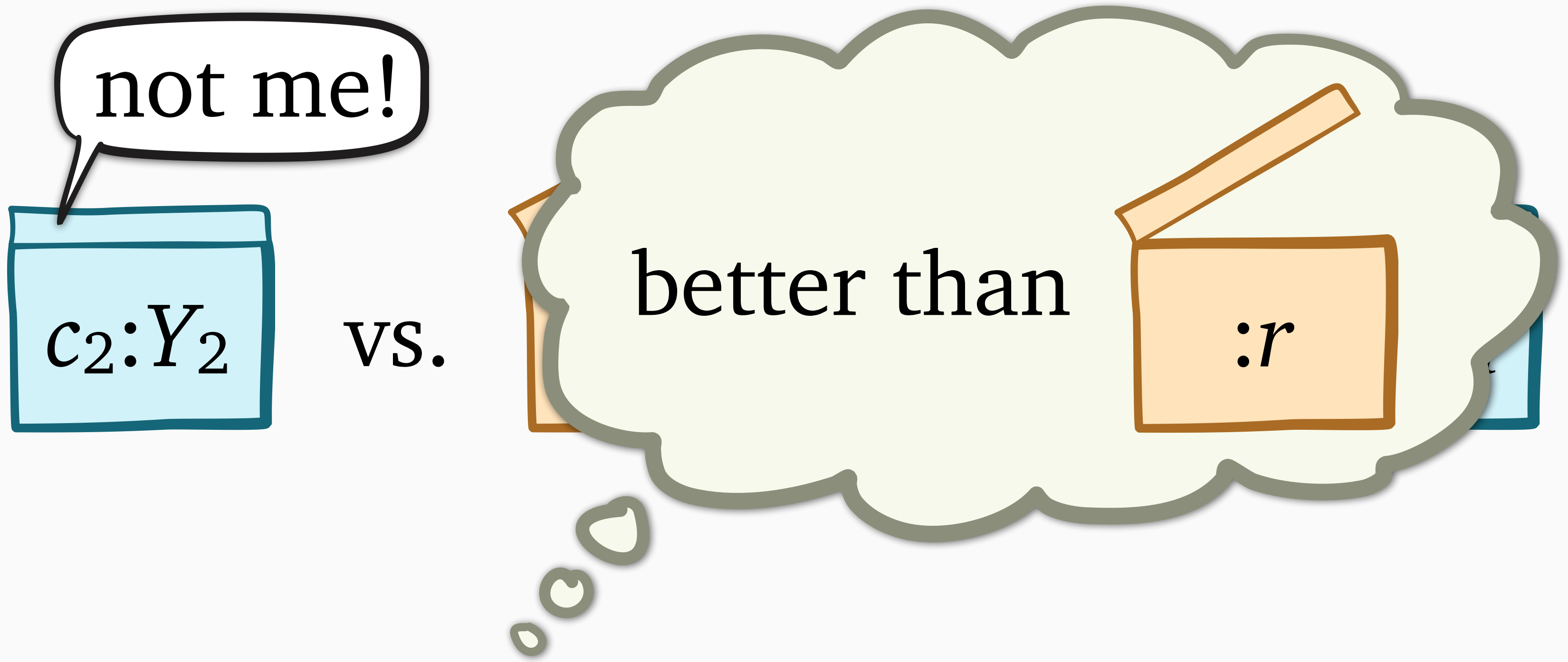
- $r \geq g_1$:



Box 1's perspective



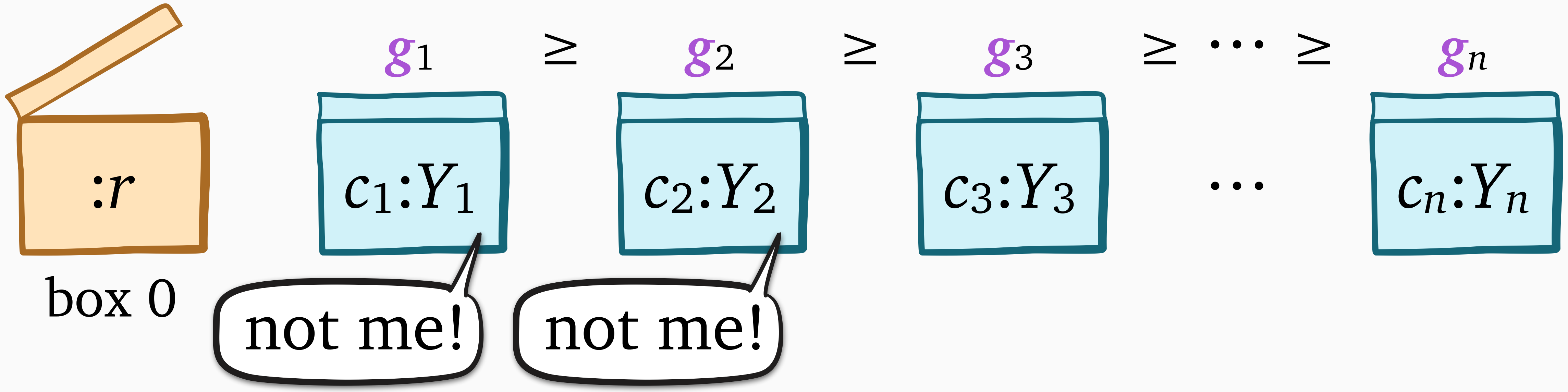
Box 2's perspective



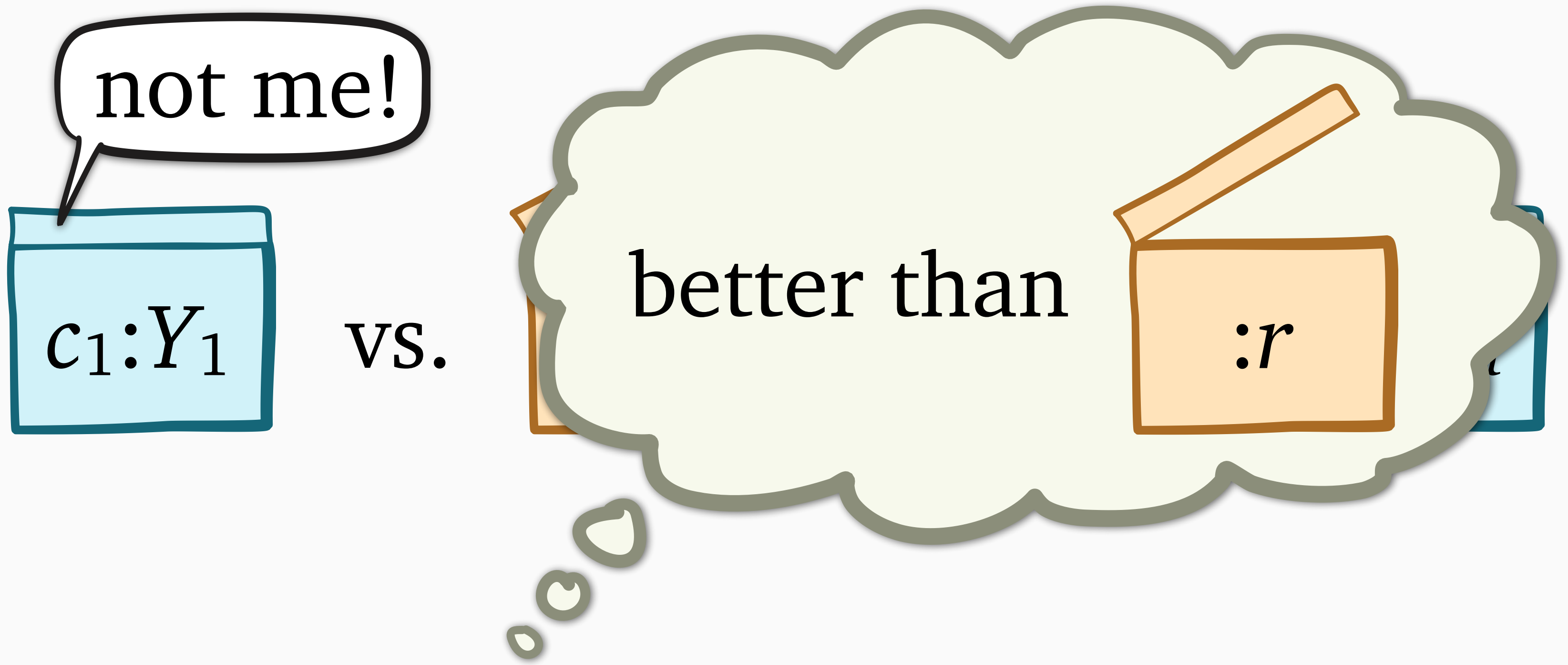
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

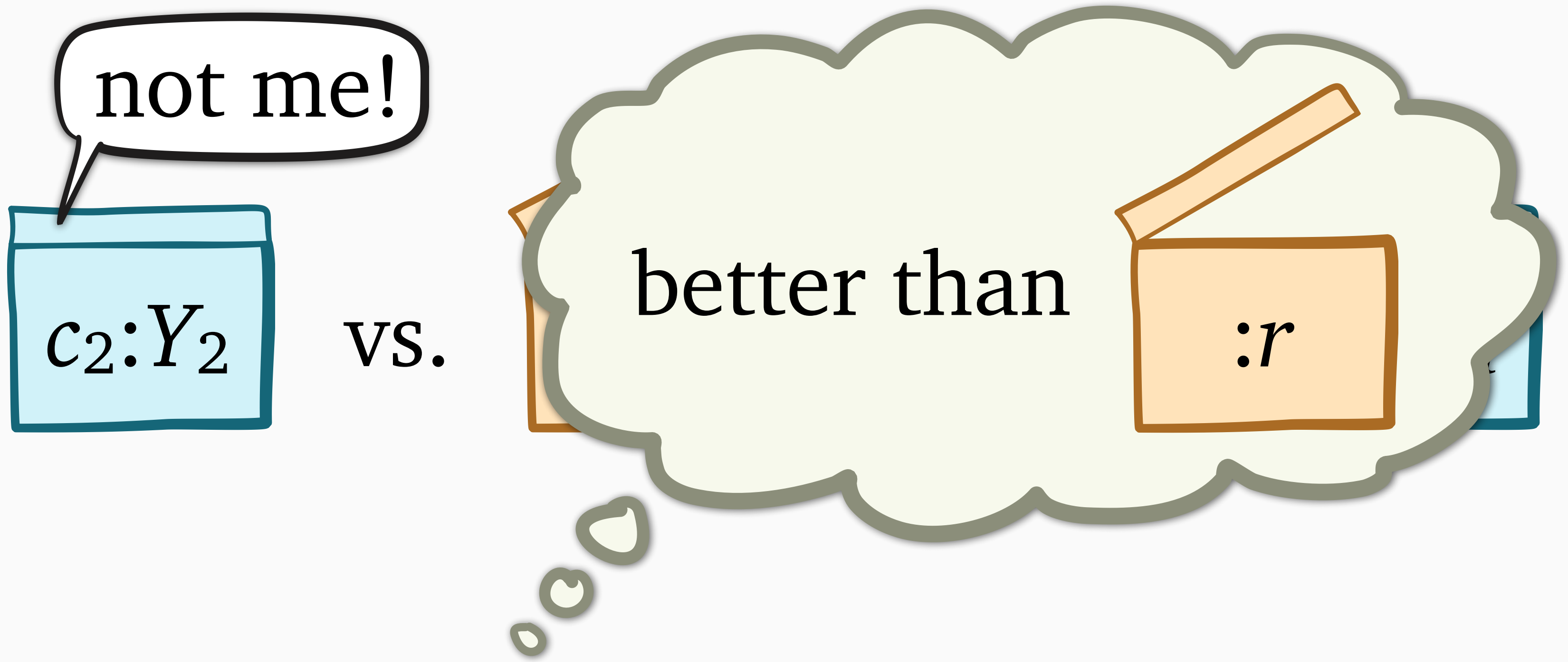
- $r \geq g_1$:



Box 1's perspective



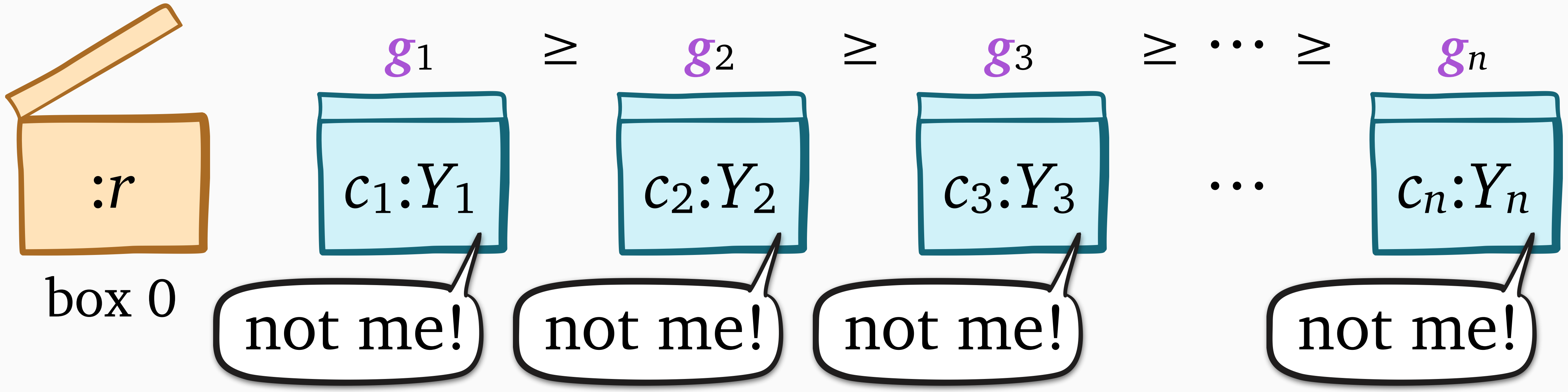
Box 2's perspective



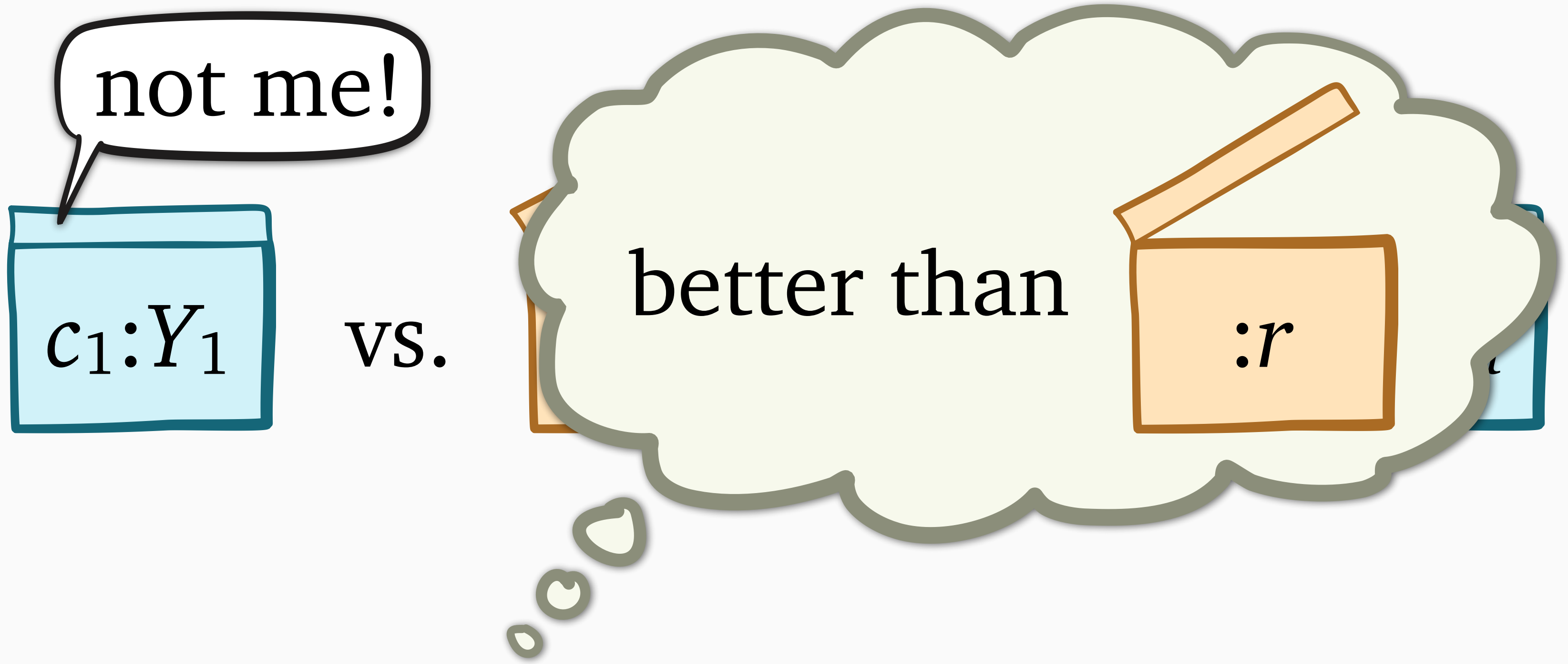
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$:



Box 1's perspective

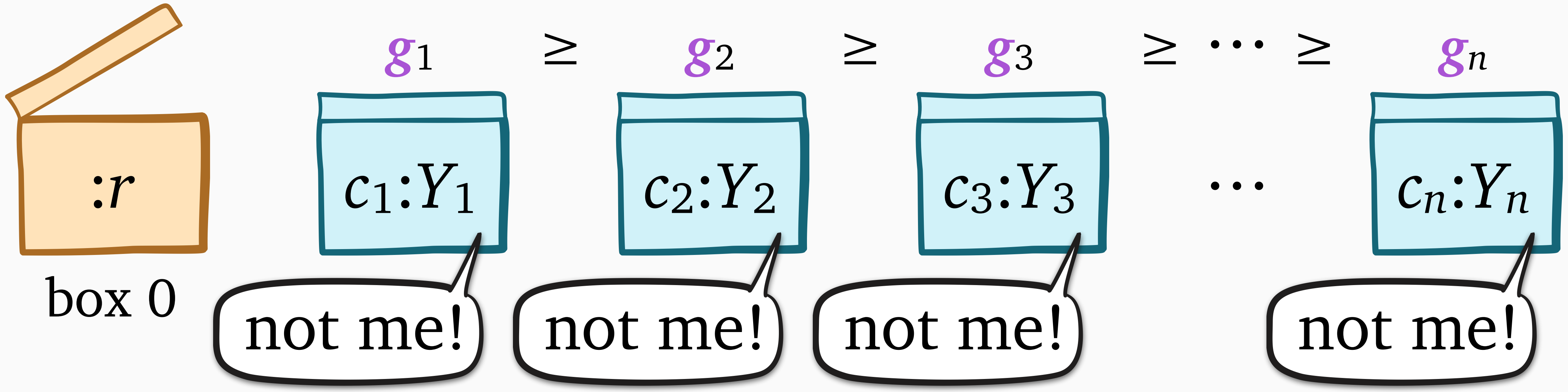


Box 2's perspective

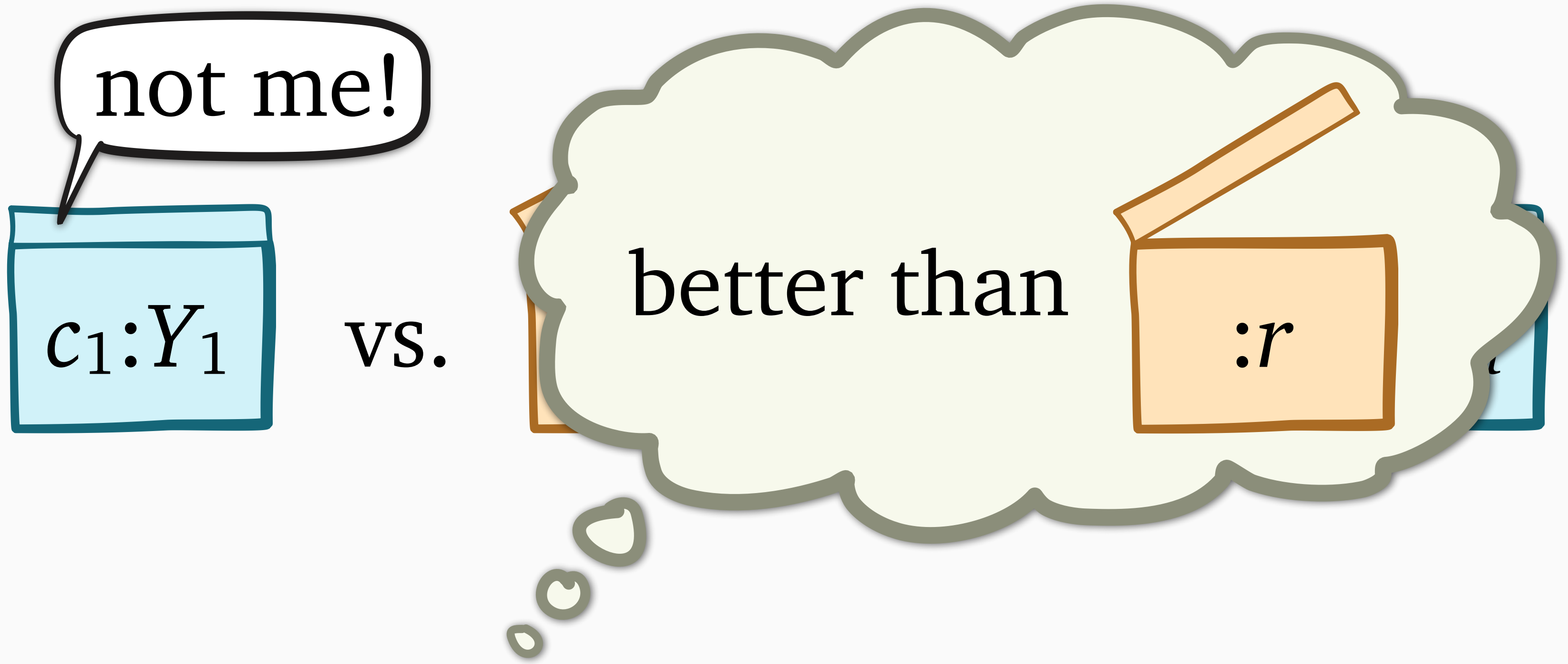
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$: *select box 0*



Box 1's perspective

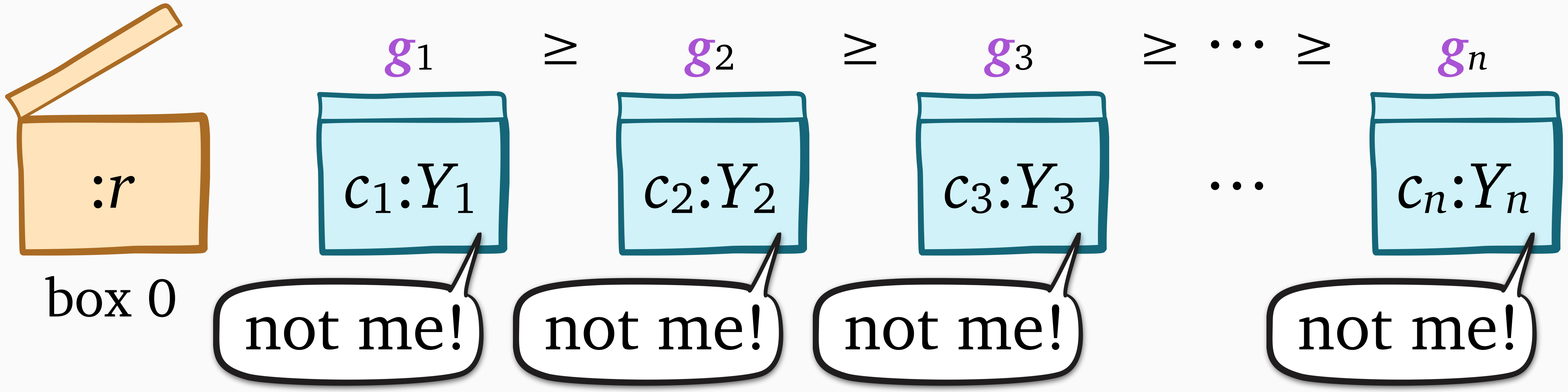


Box 2's perspective

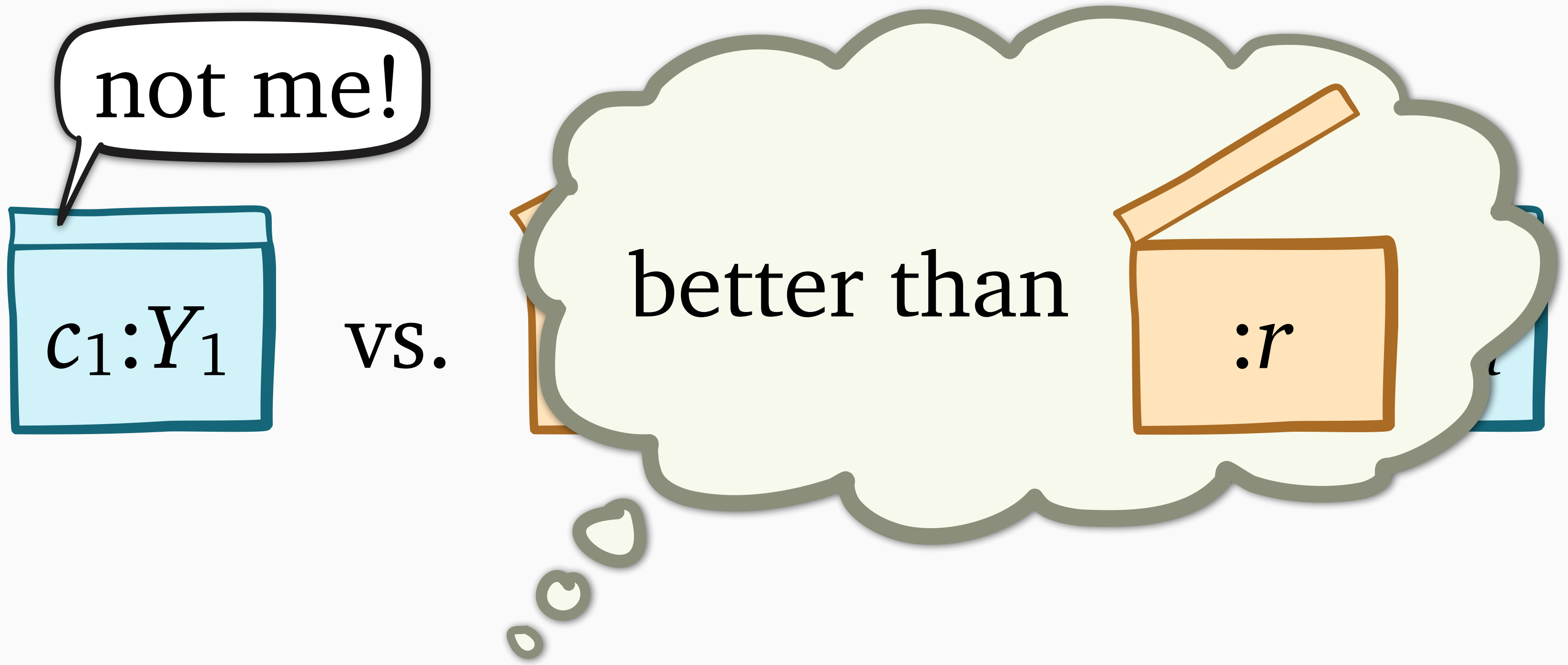
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

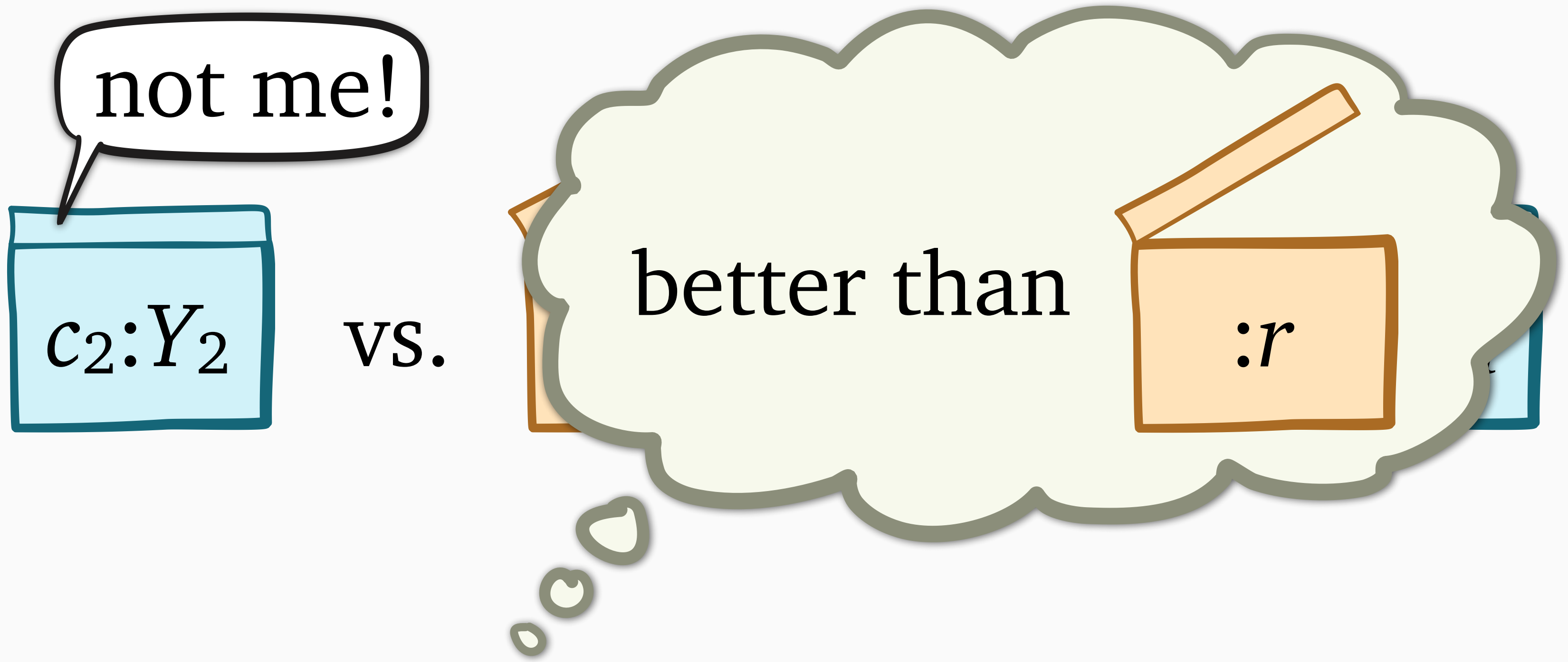
- $r \geq g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective



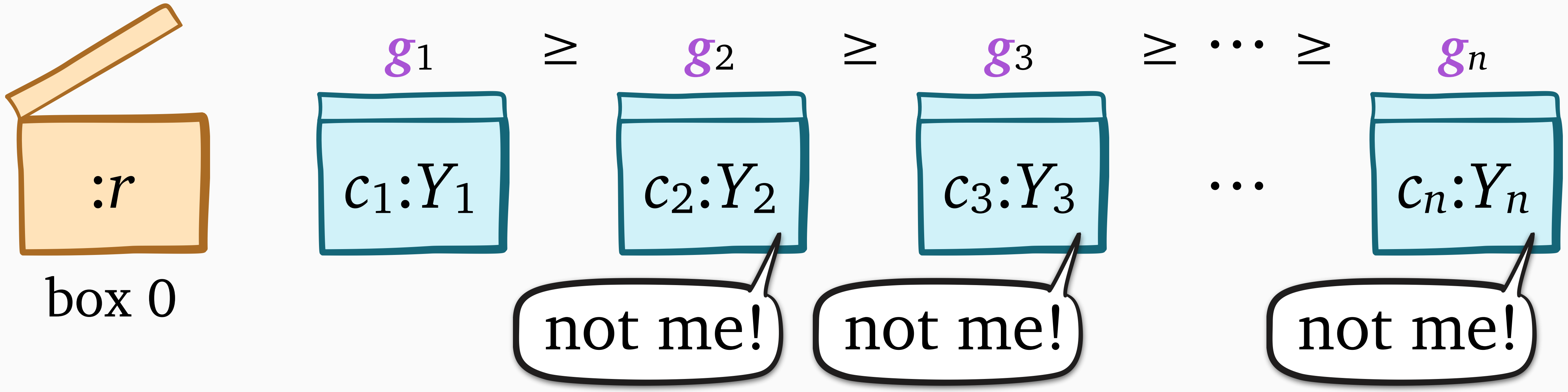
Box 2's perspective



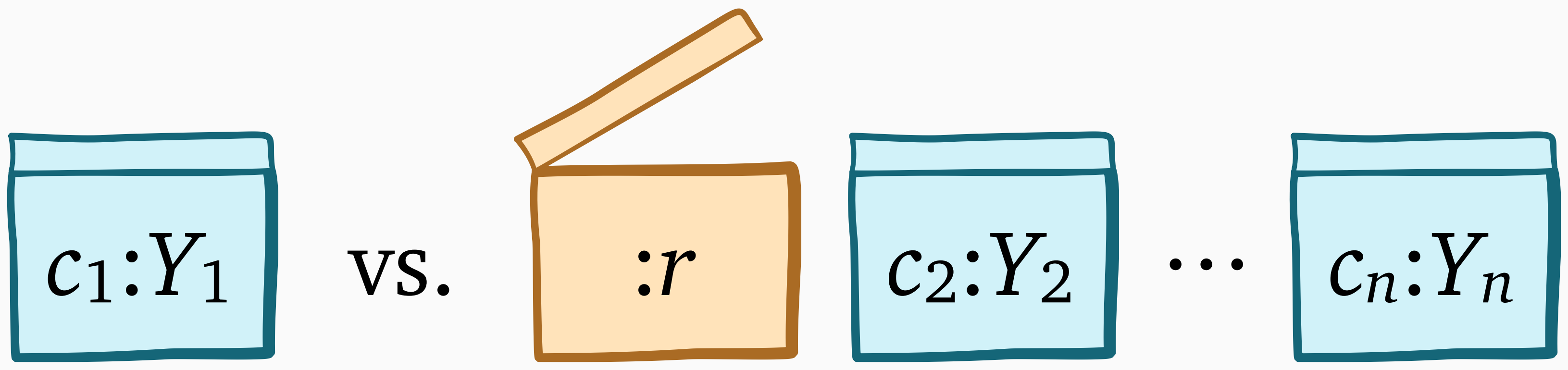
Why is the **Gittins index** optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

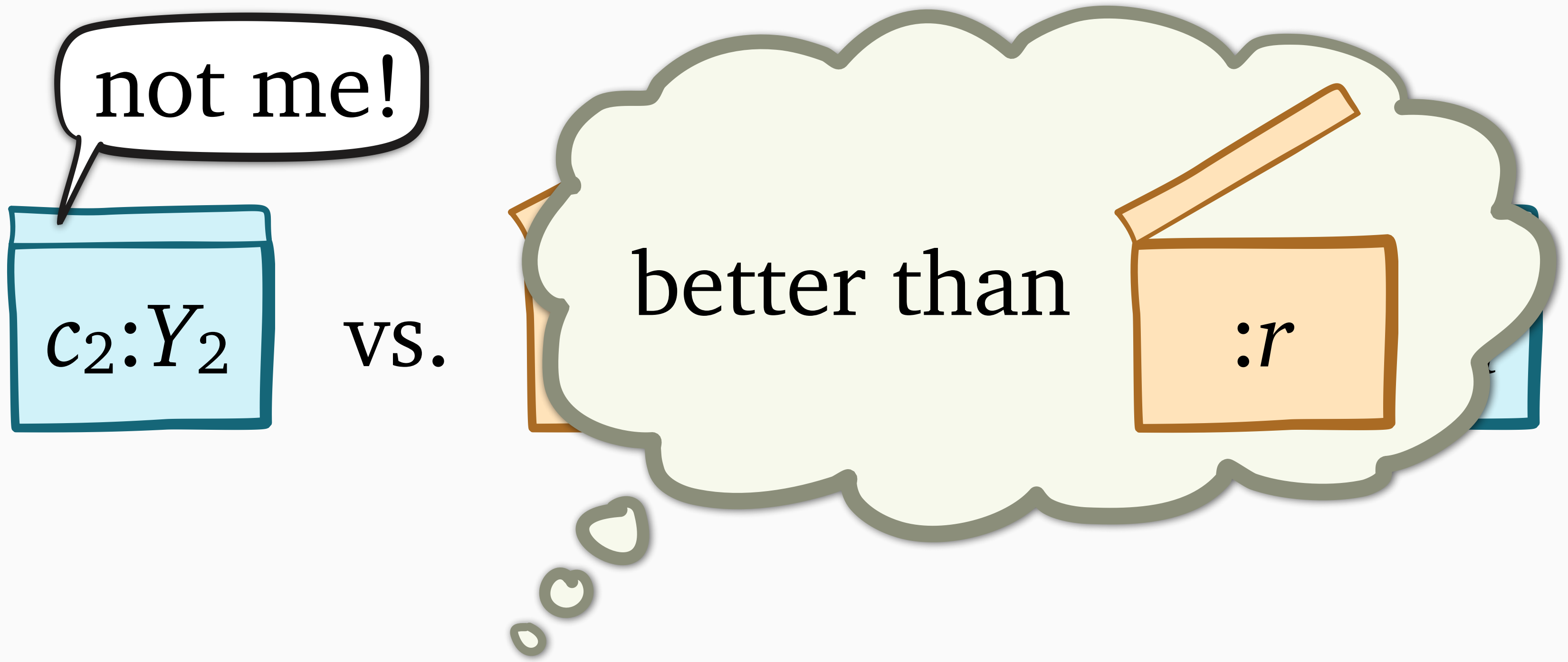
- $r \geq g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective



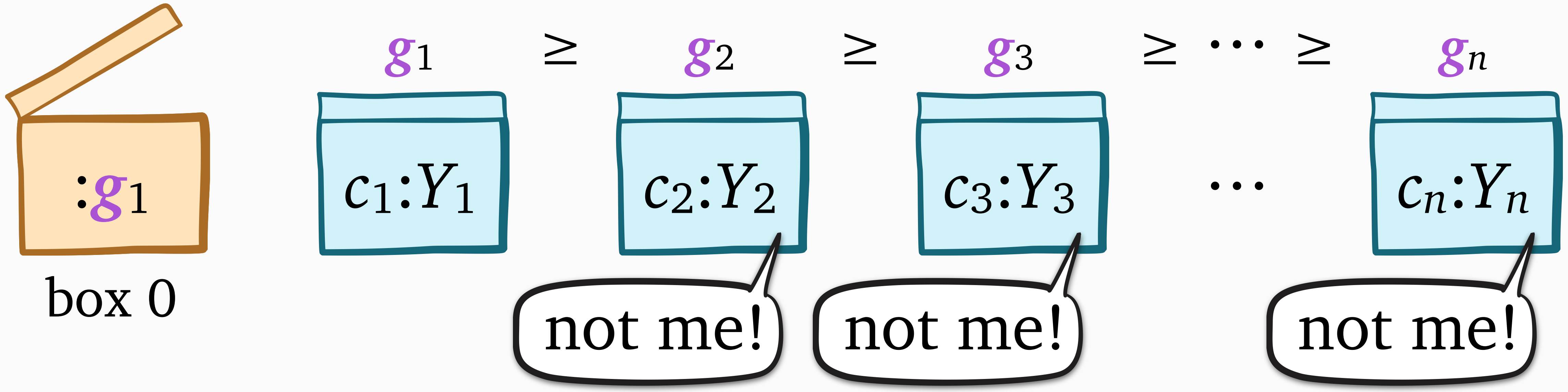
Box 2's perspective



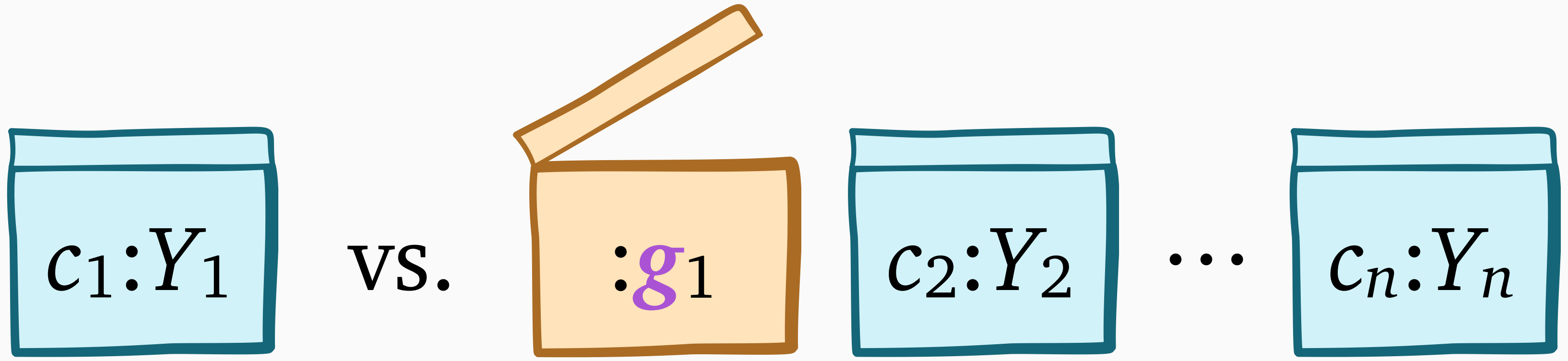
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

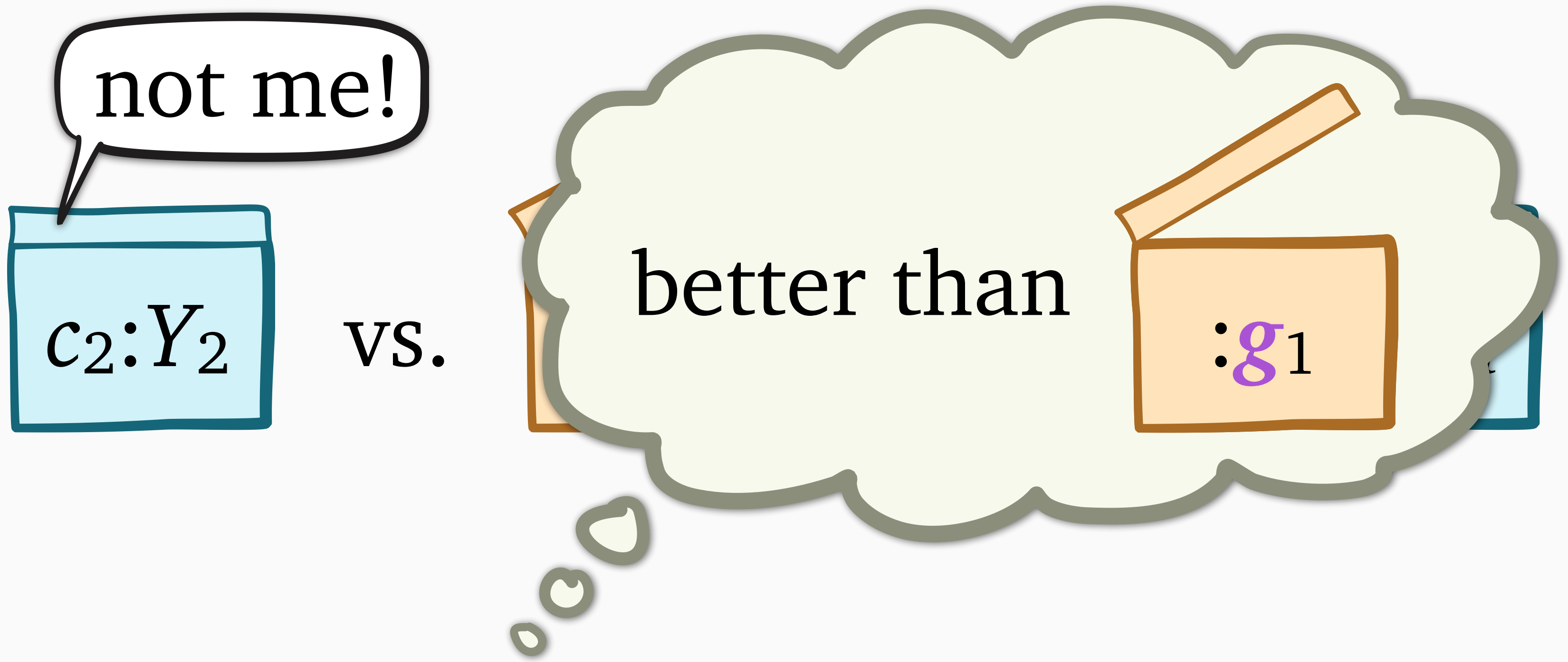
- $r \geq g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective



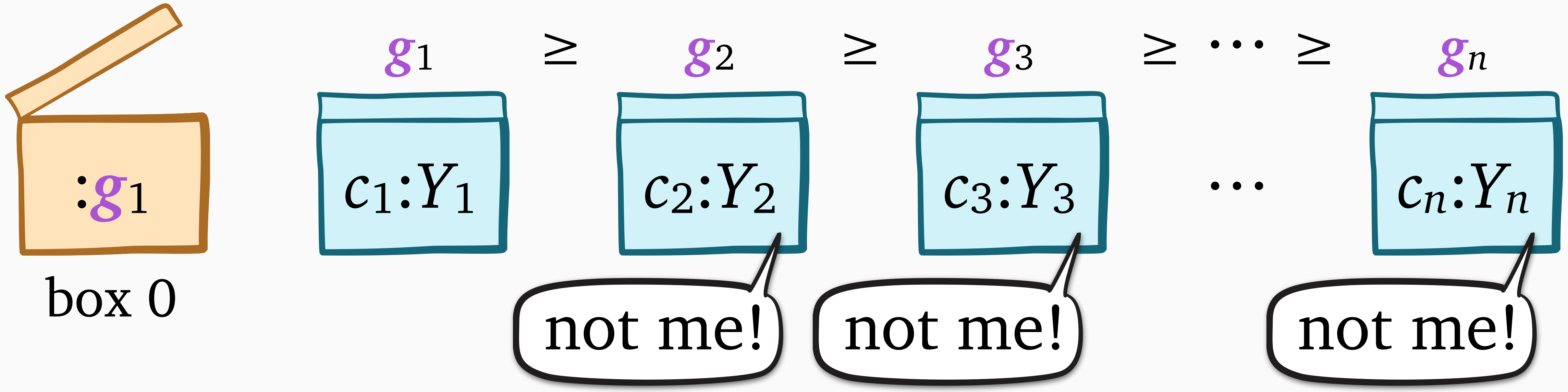
Box 2's perspective



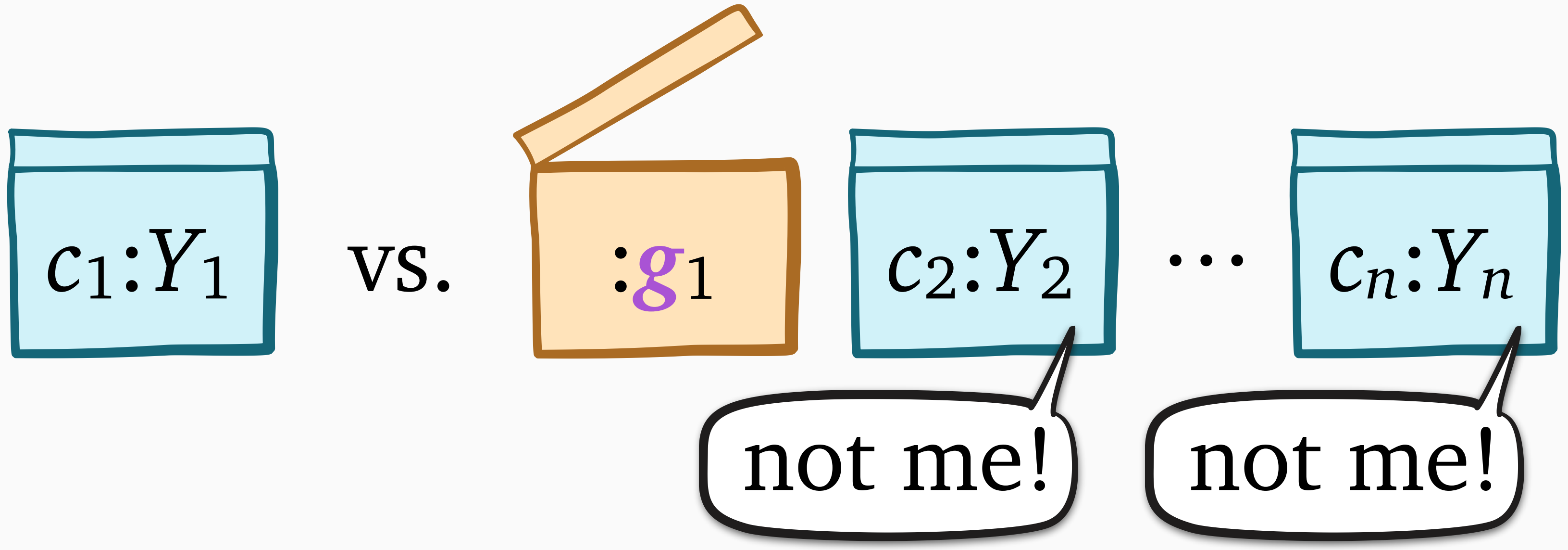
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

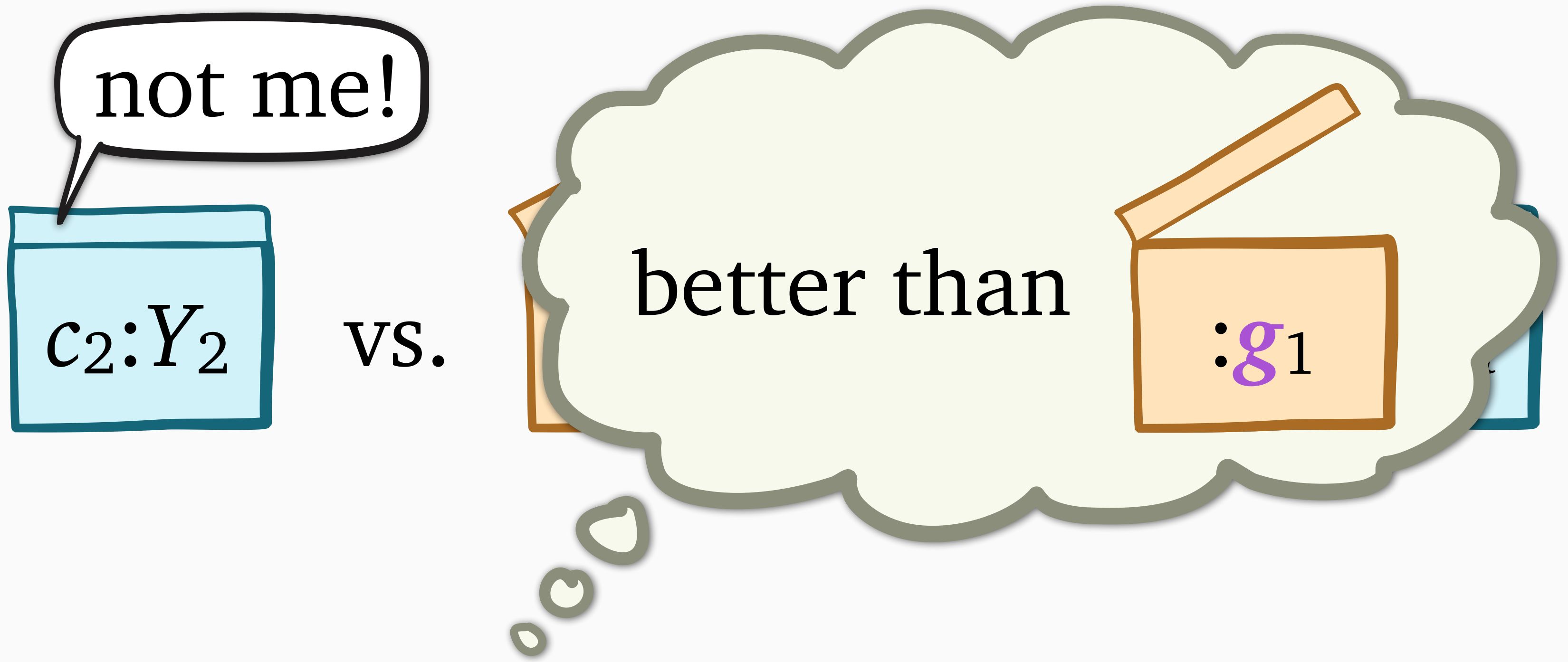
- $r \ge g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective



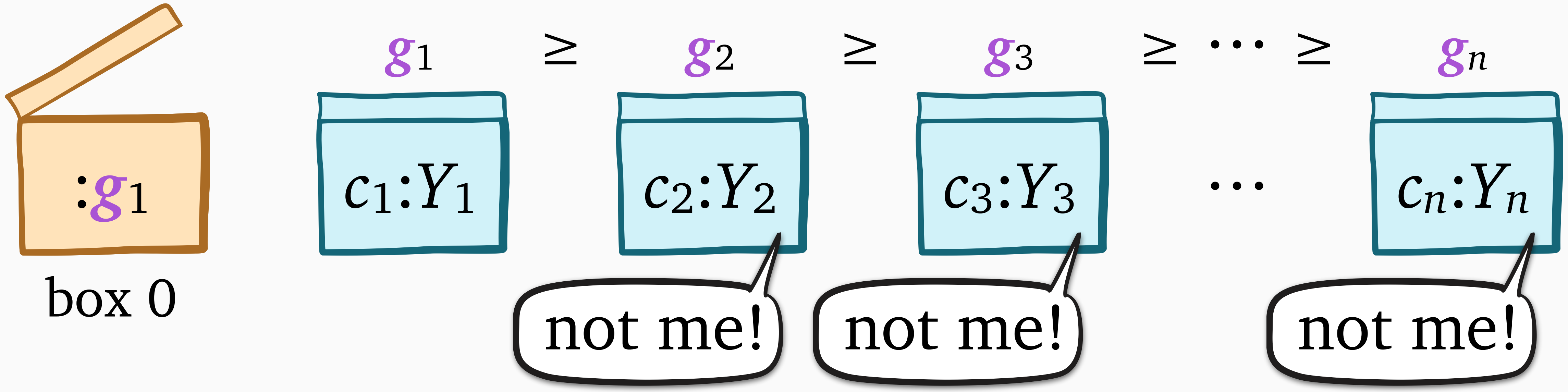
Box 2's perspective



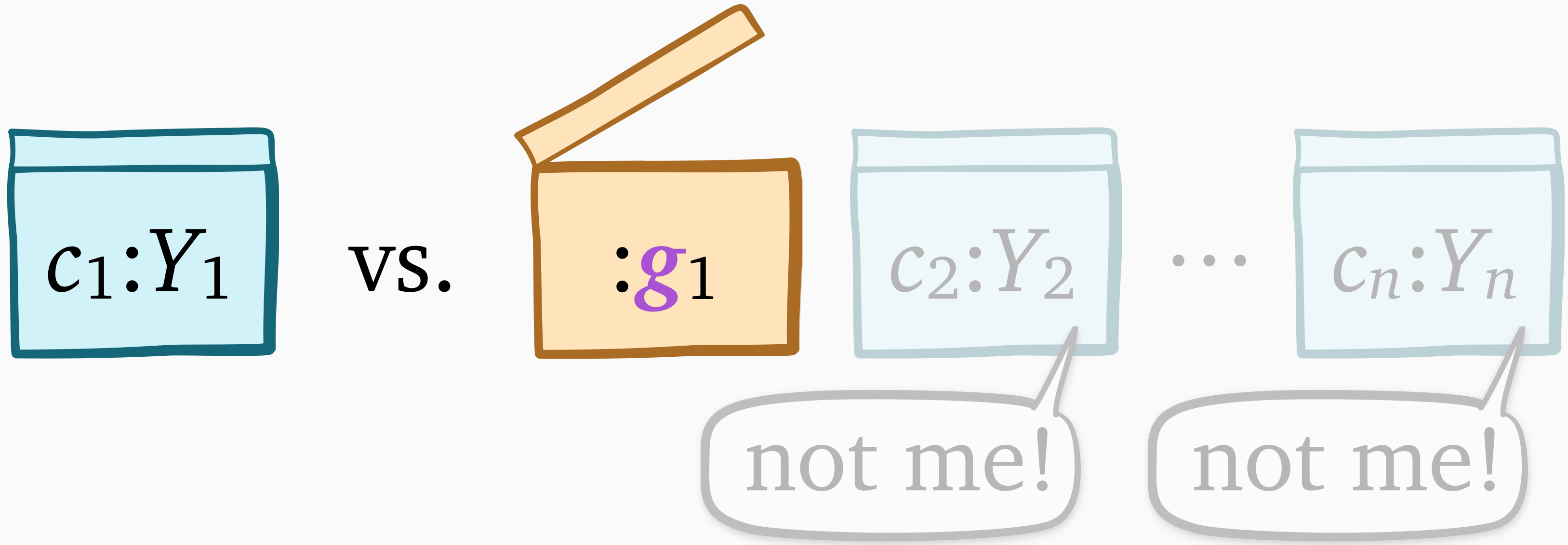
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective

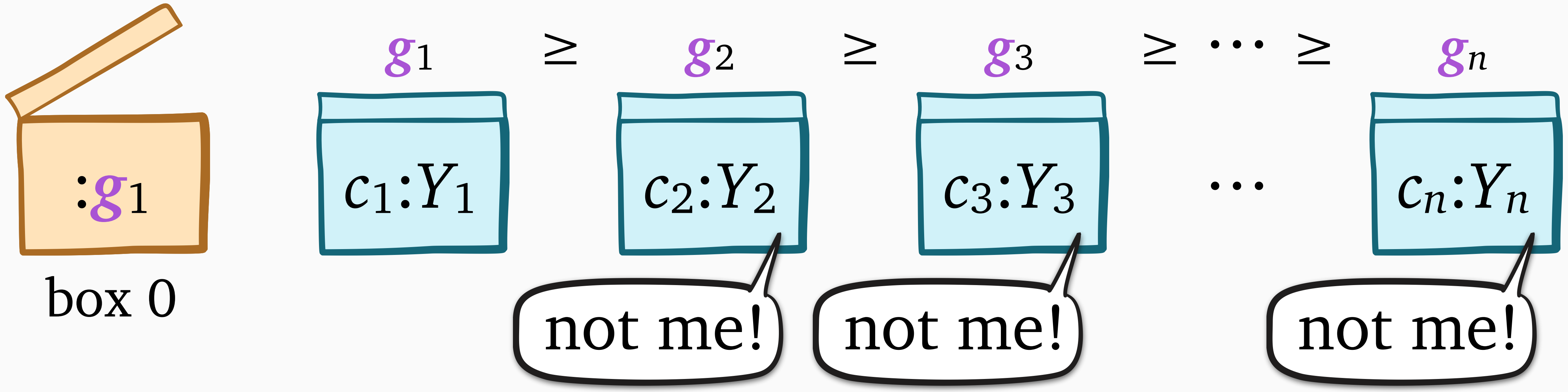


Box 2's perspective

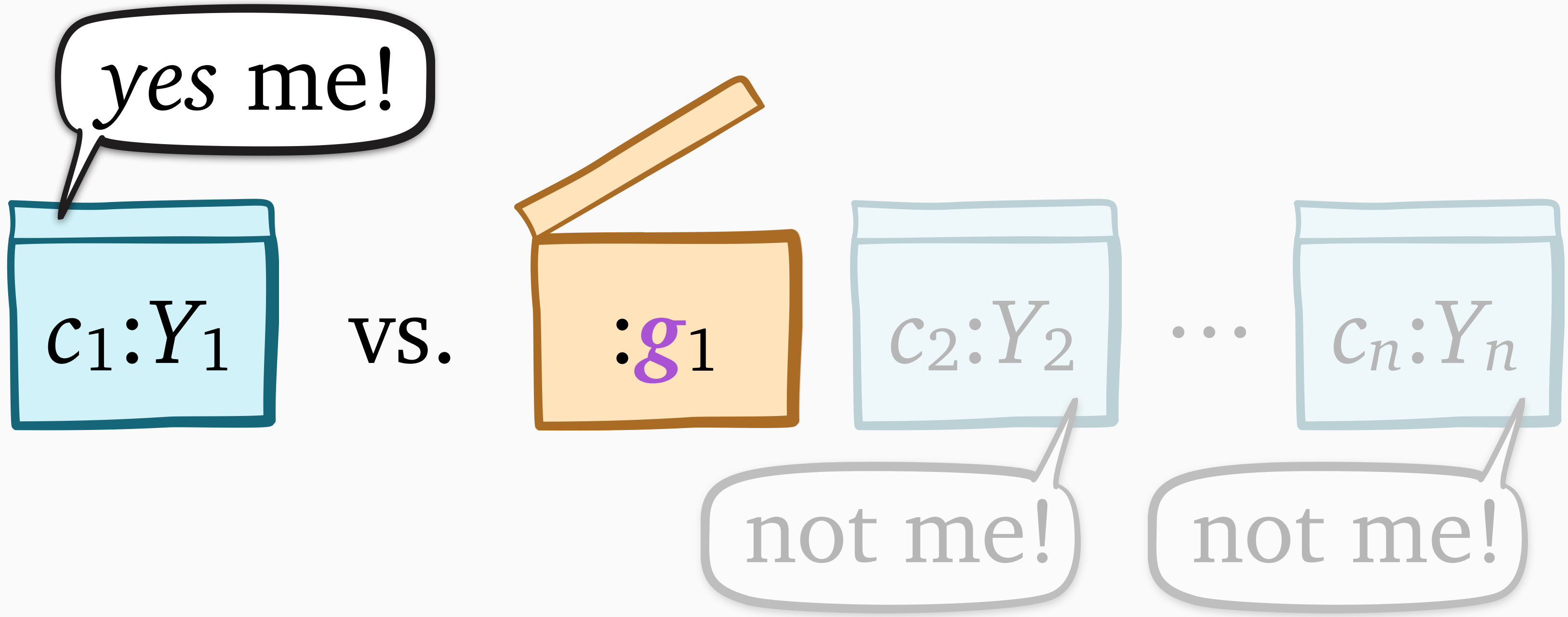
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

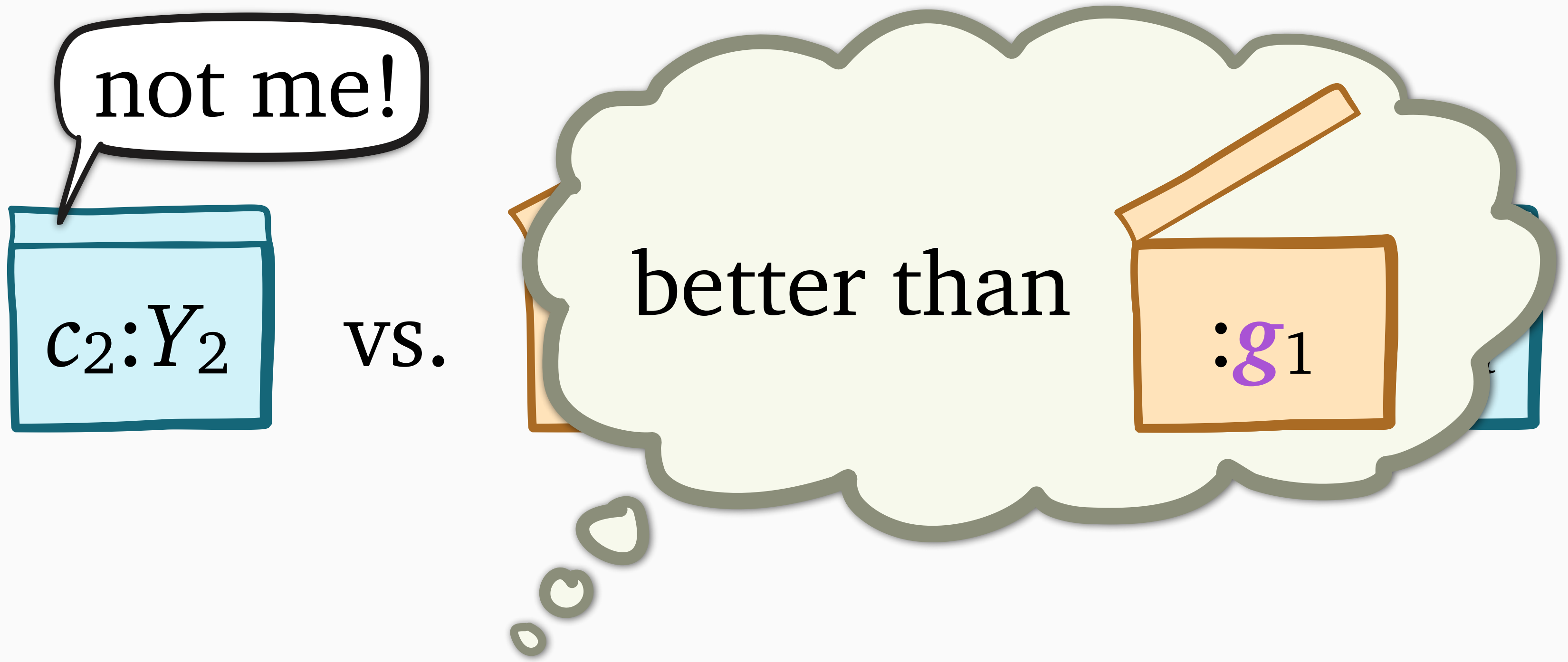
- $r \geq g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective



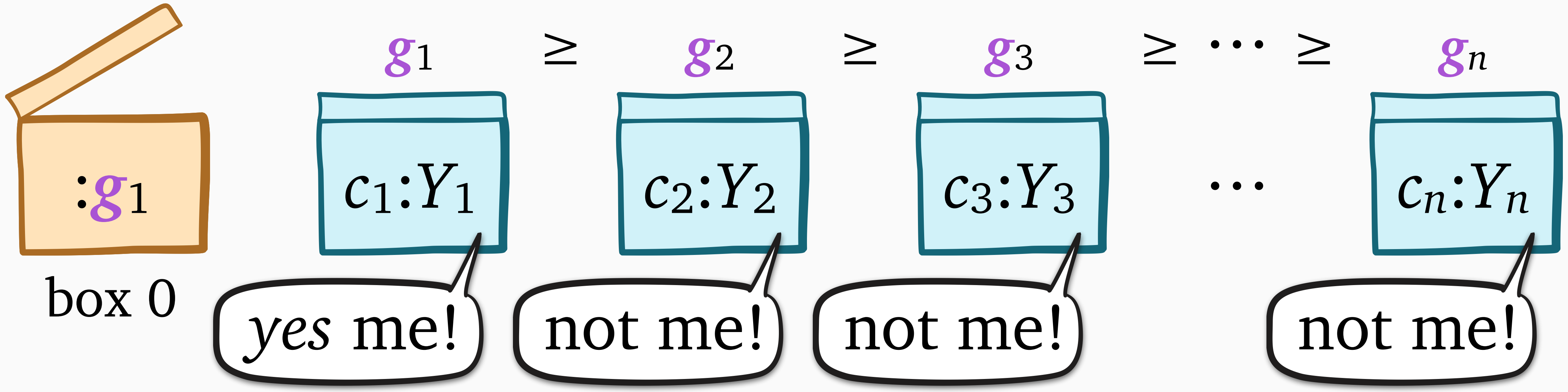
Box 2's perspective



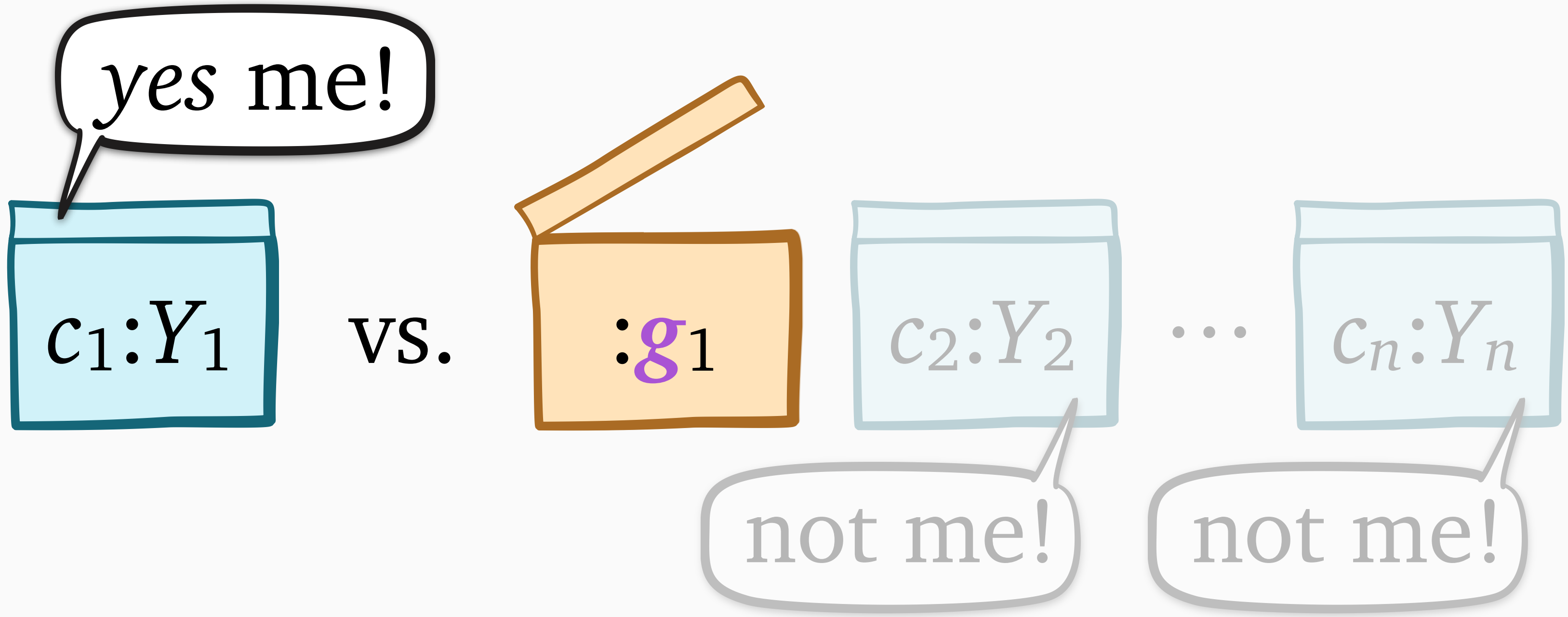
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

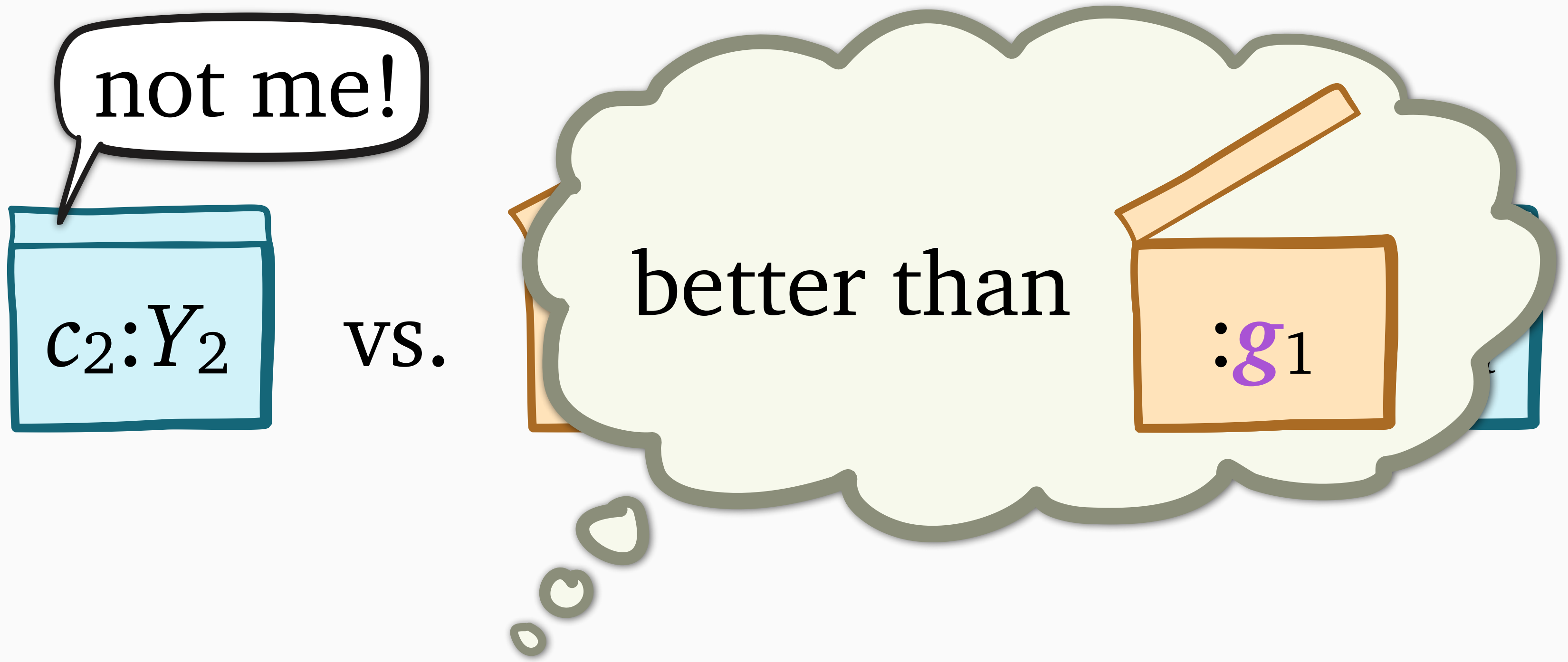
- $r \geq g_1$: *select box 0*
- $r = g_1$:



Box 1's perspective



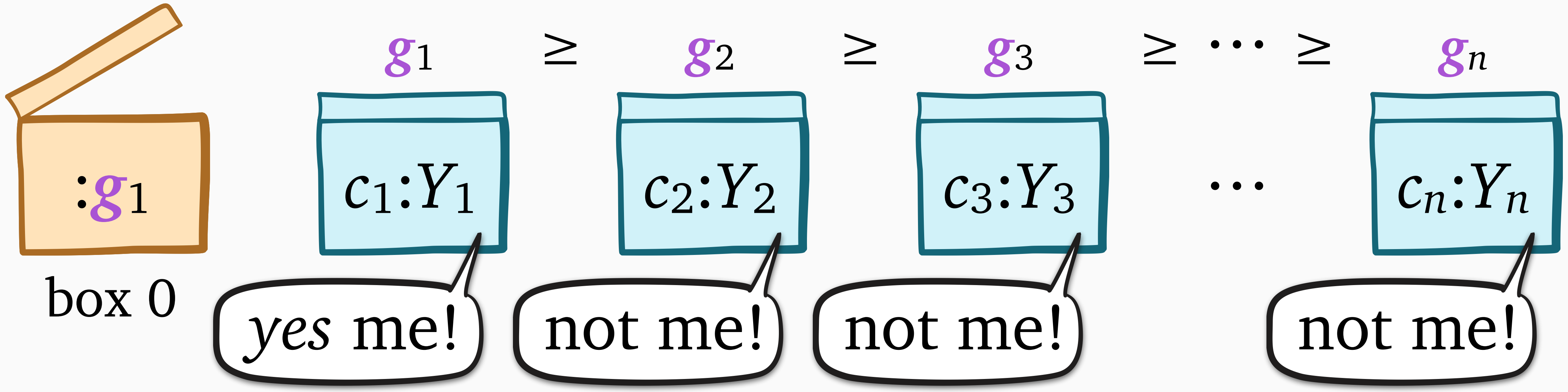
Box 2's perspective



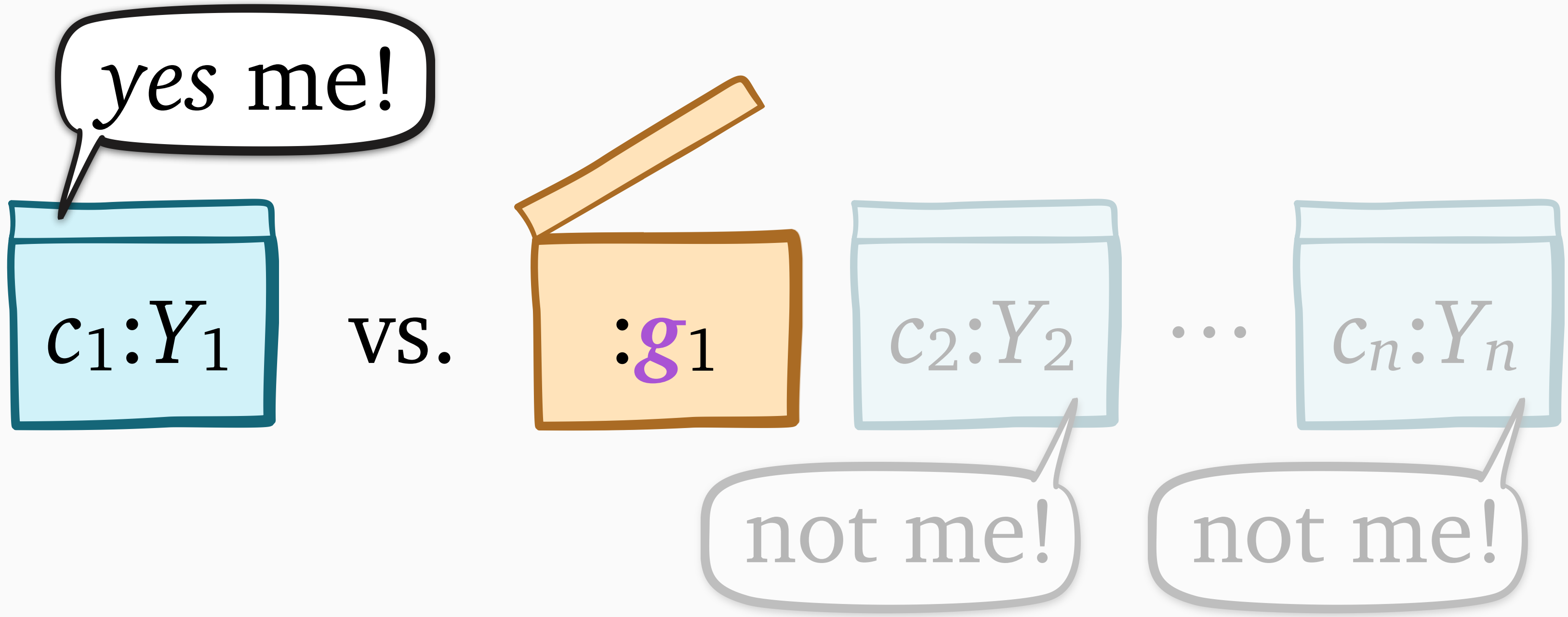
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

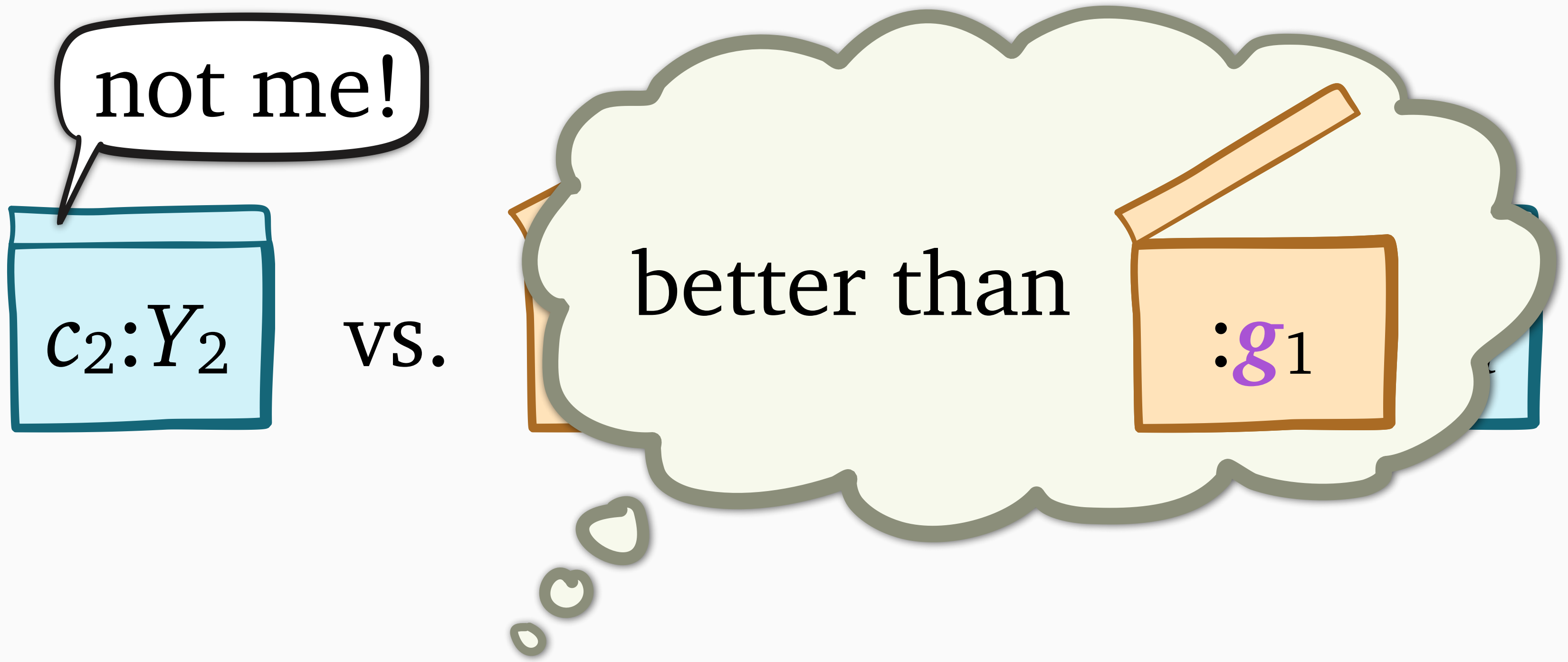
- $r \geq g_1$: *select box 0*
- $r = g_1$: *open box 1*



Box 1's perspective



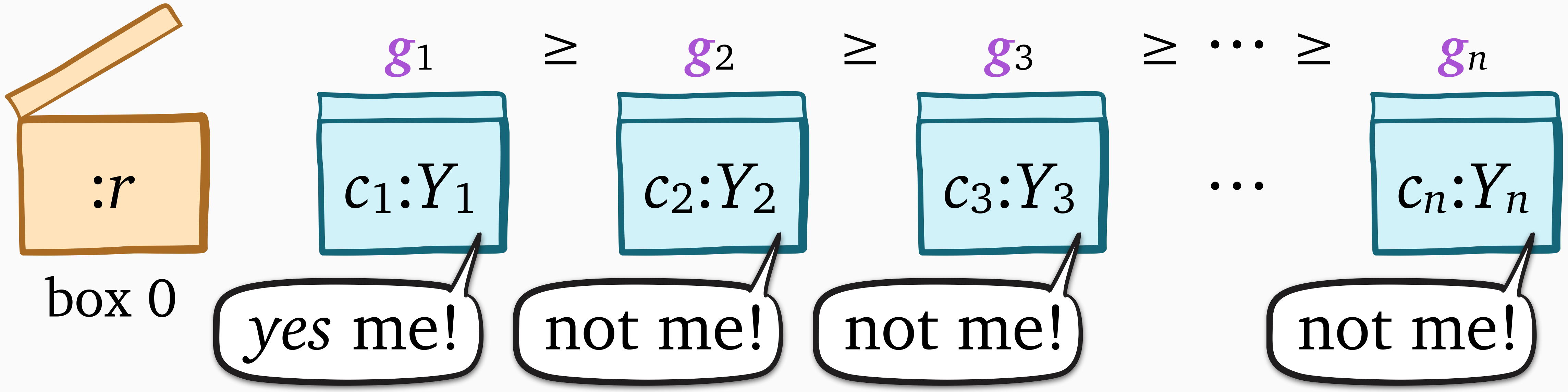
Box 2's perspective



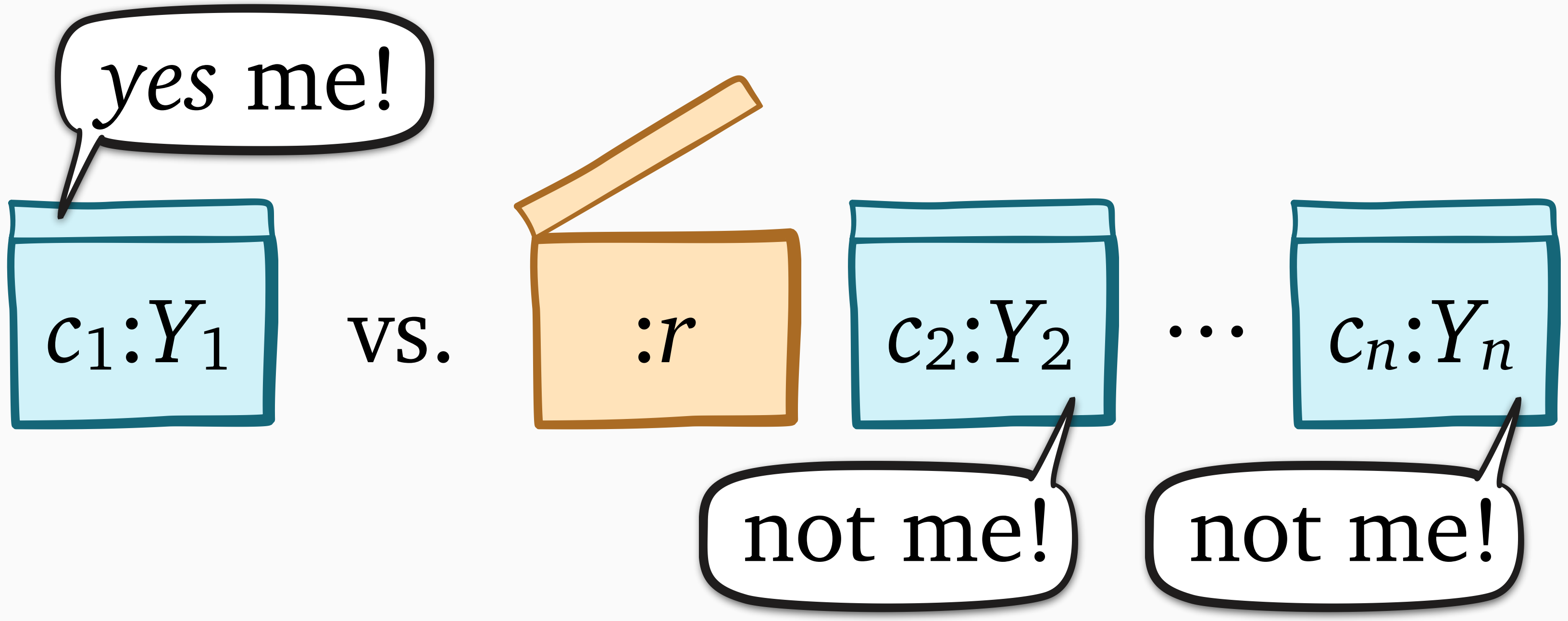
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

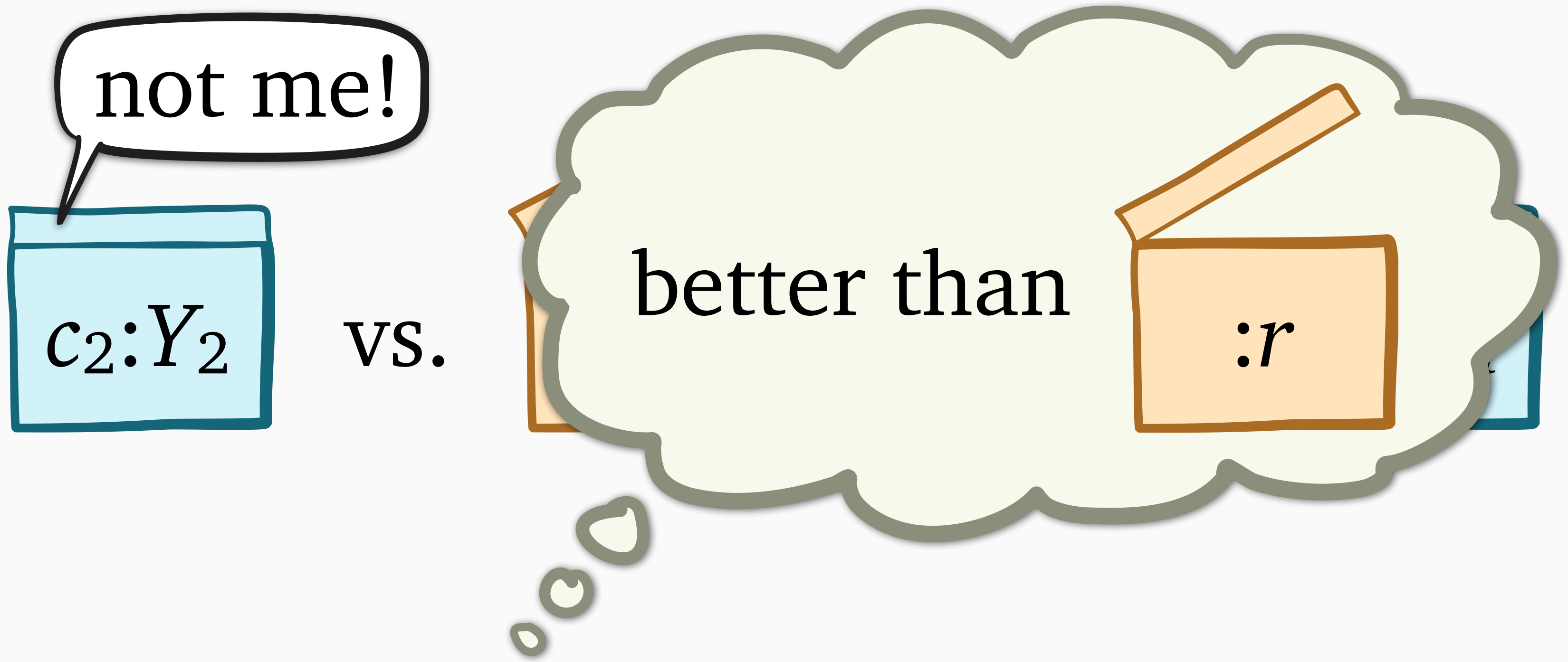
- $r \geq g_1$: *select box 0*
- $r = g_1$: *open box 1*
- $r < g_1$:



Box 1's perspective



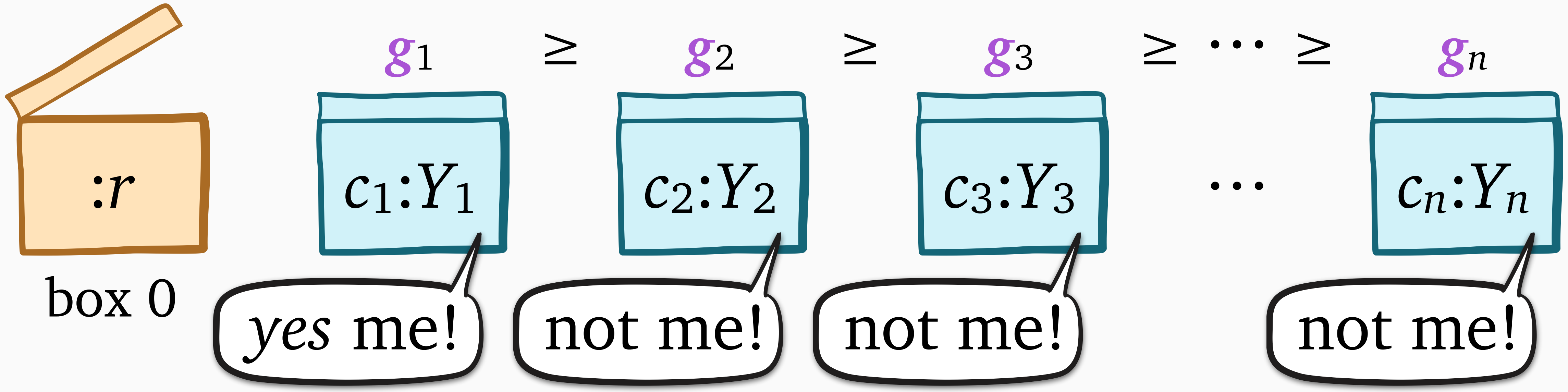
Box 2's perspective



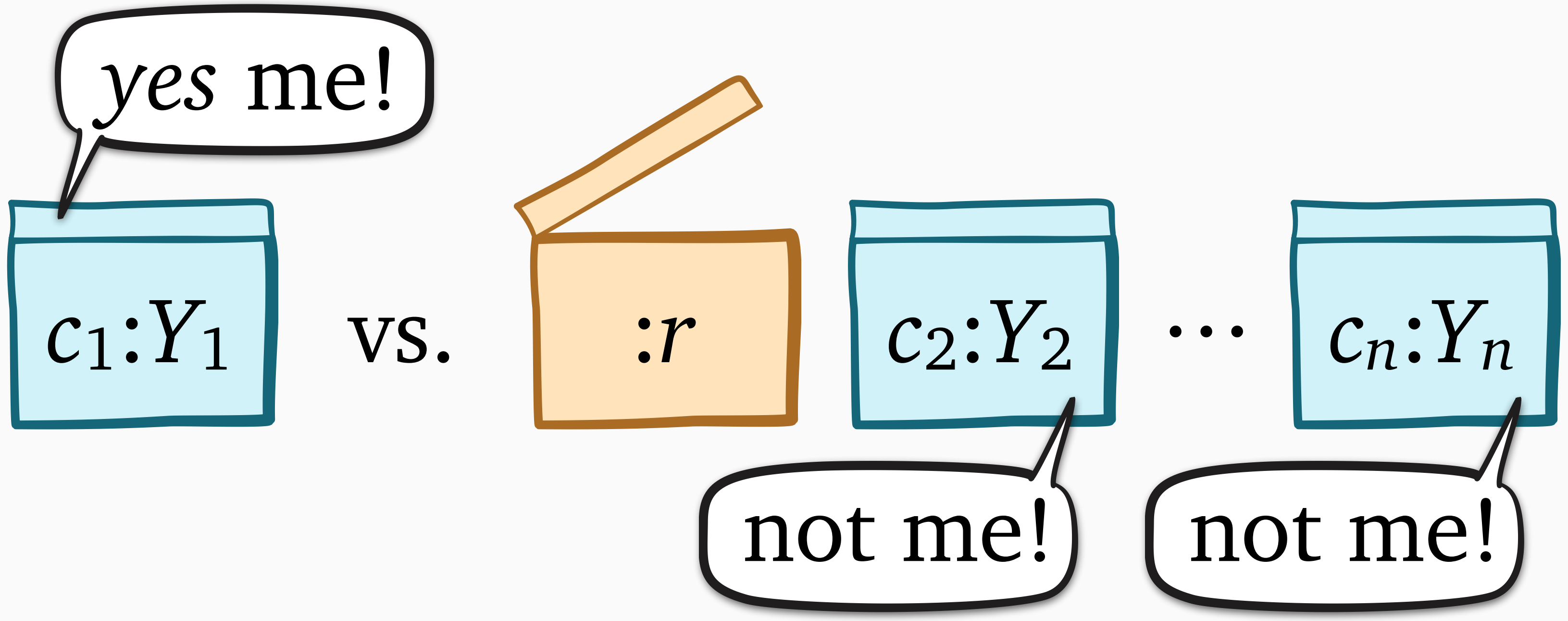
Why is the Gittins index optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

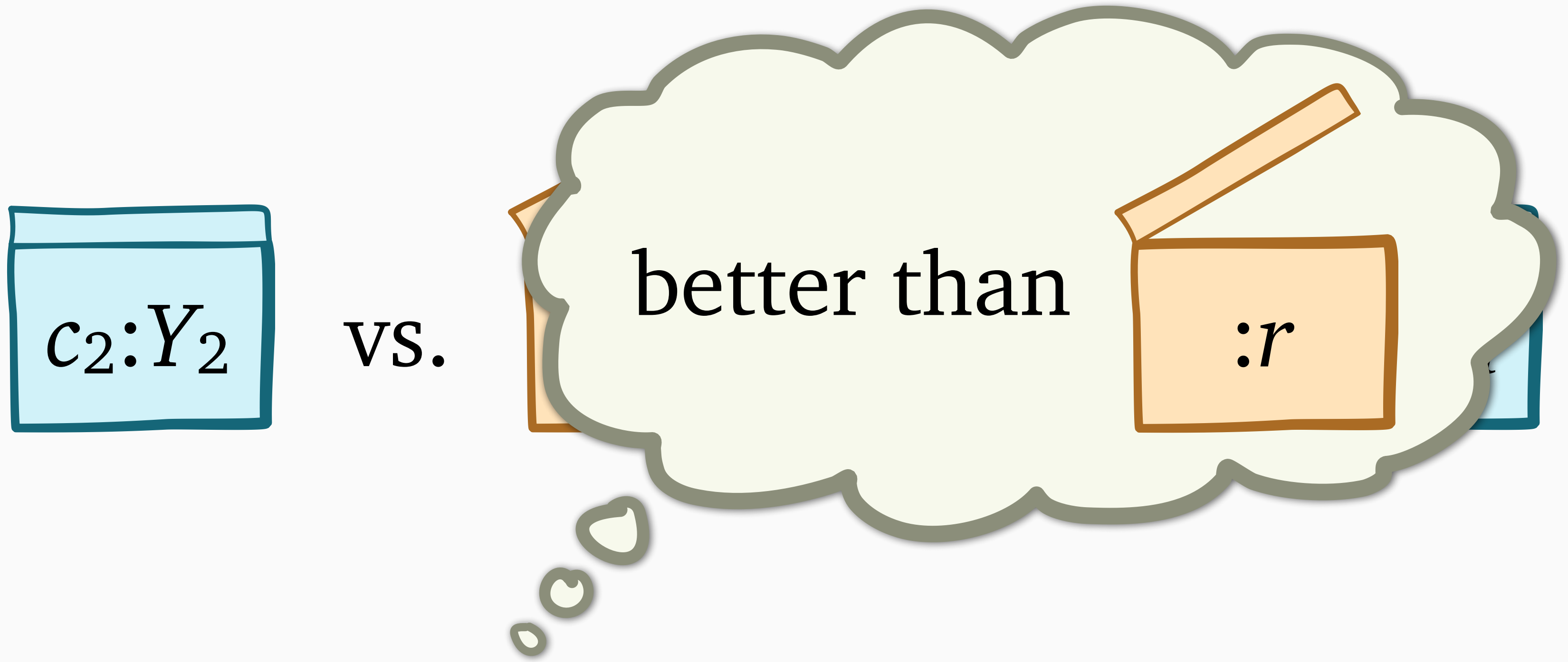
- $r \geq g_1$: *select box 0*
- $r = g_1$: *open box 1*
- $r < g_1$:



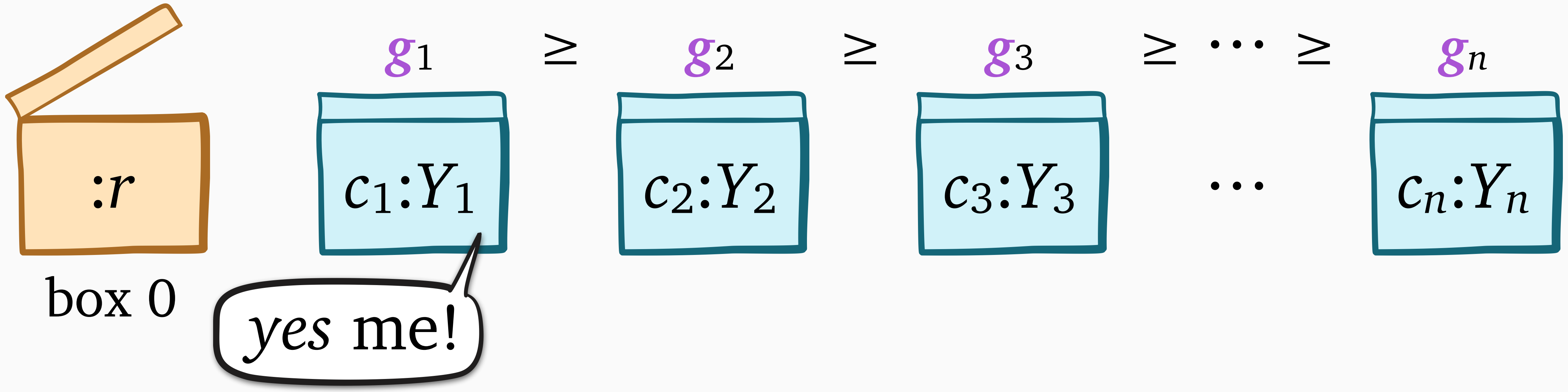
Box 1's perspective



Box 2's perspective



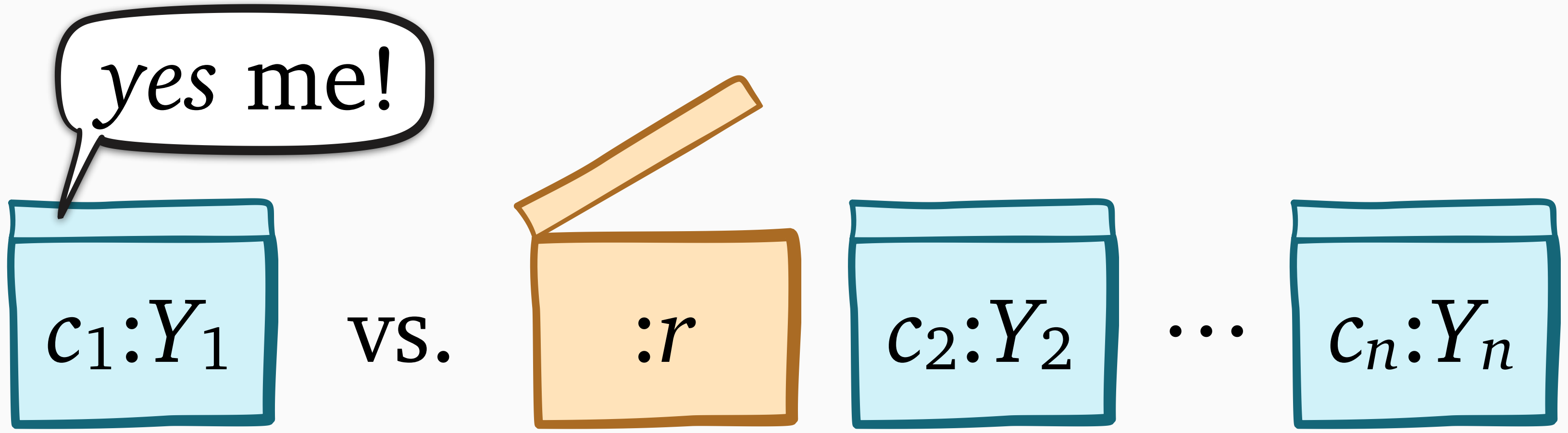
Why is the Gittins index optimal?



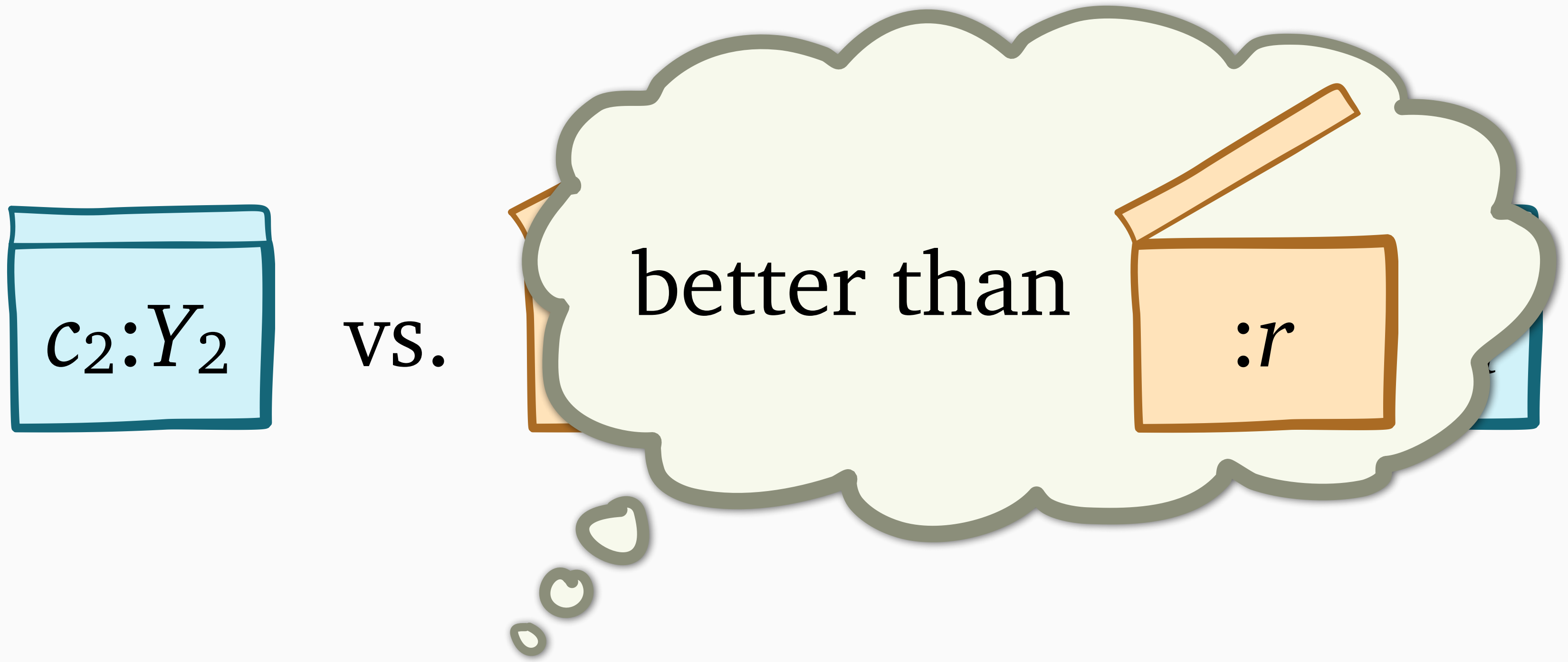
Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$: *select box 0*
- $r = g_1$: *open box 1*
- $r < g_1$:

Box 1's perspective



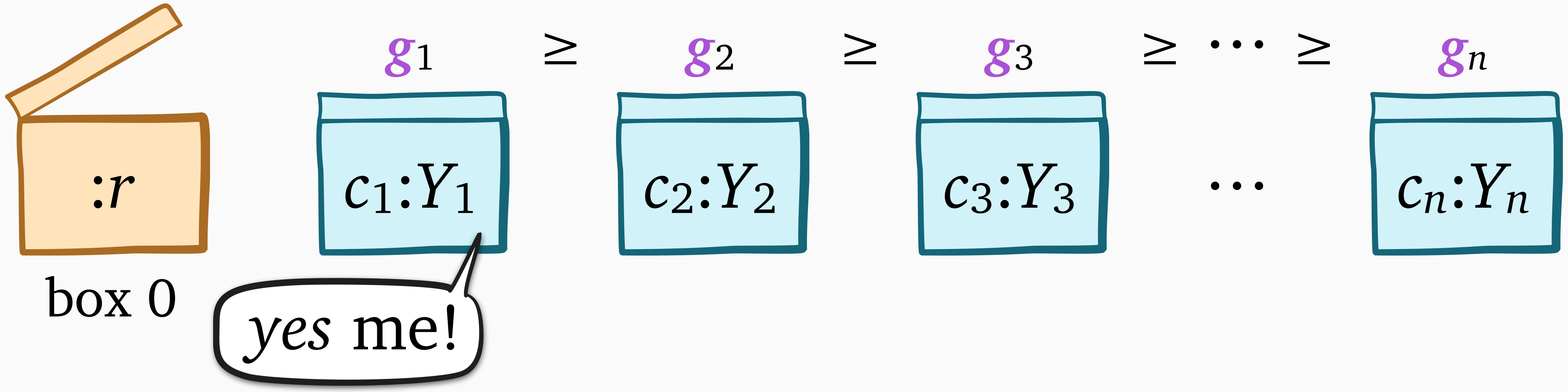
Box 2's perspective



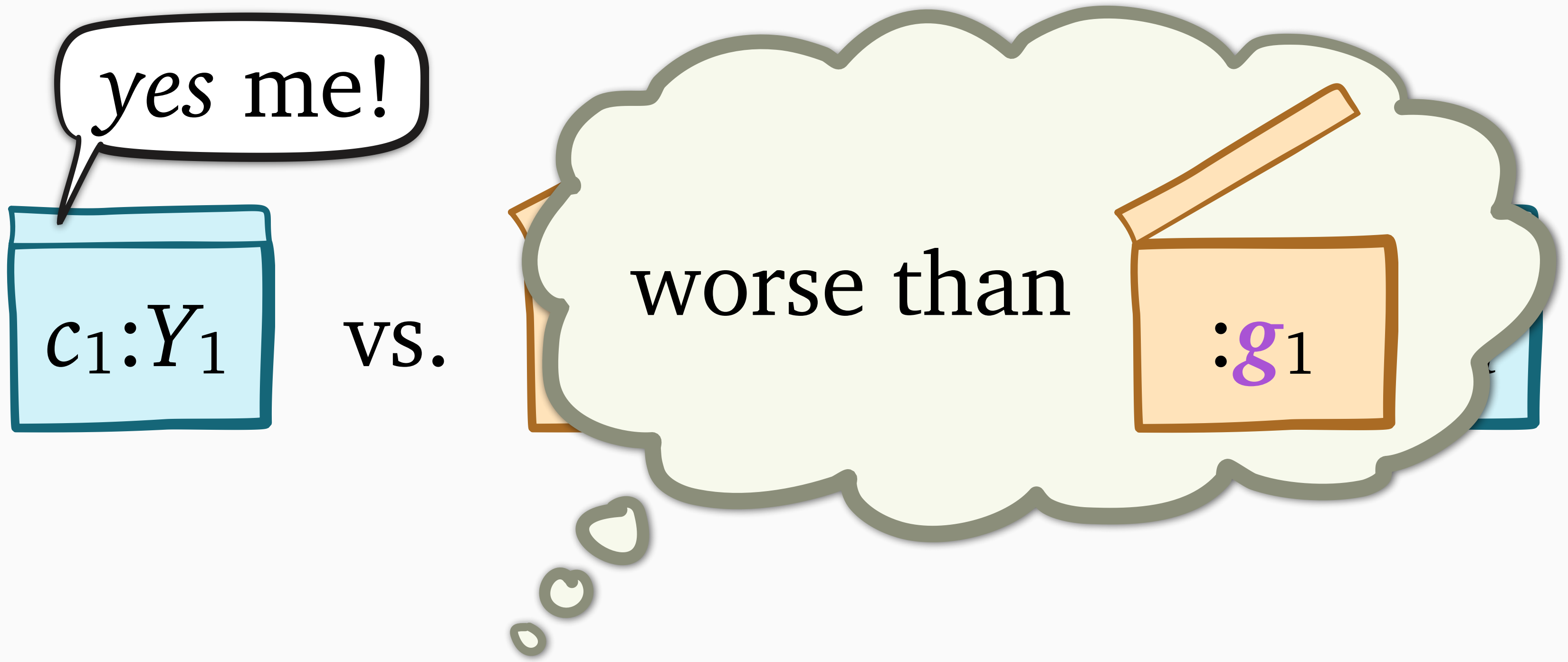
Why is the **Gittins index** optimal?

Approach: start at $r = \infty$, then decrease to $r = 0$

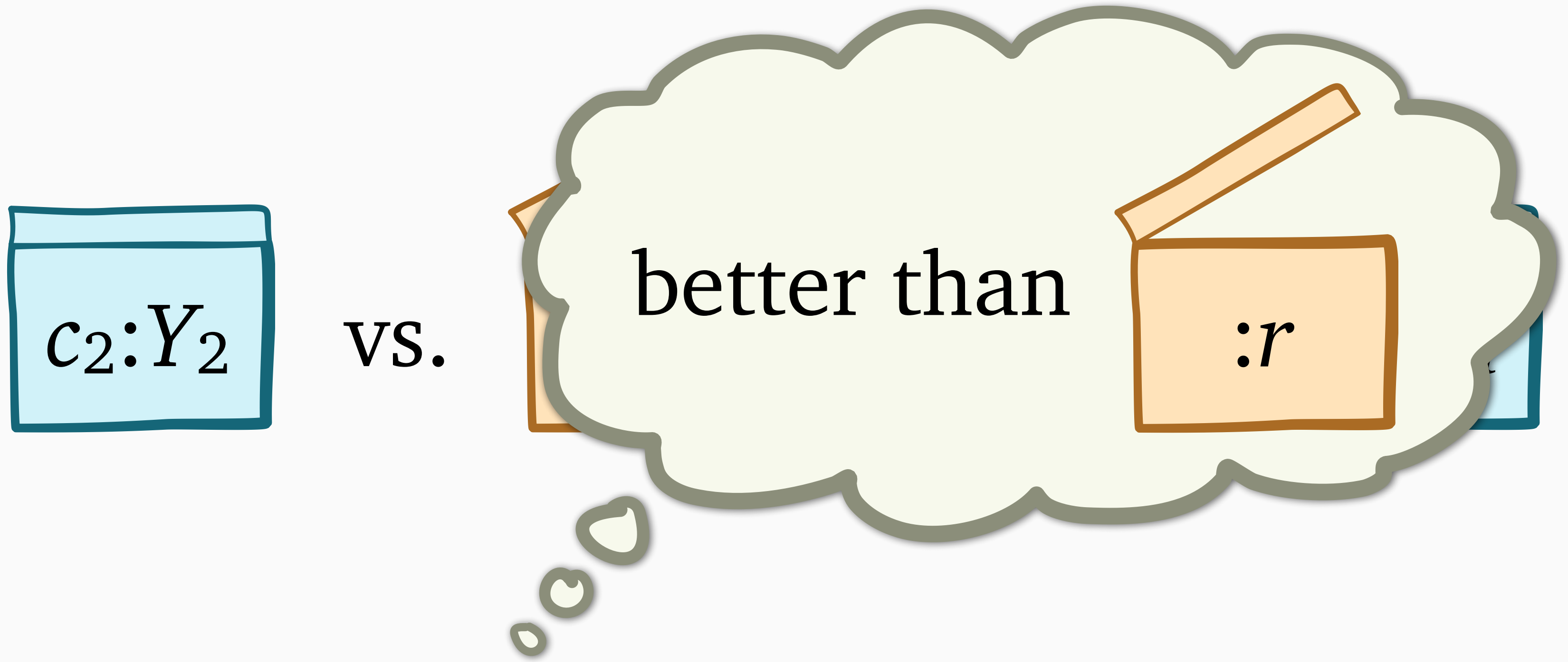
- $r \geq g_1$: *select box 0*
- $r = g_1$: *open box 1*
- $r < g_1$:



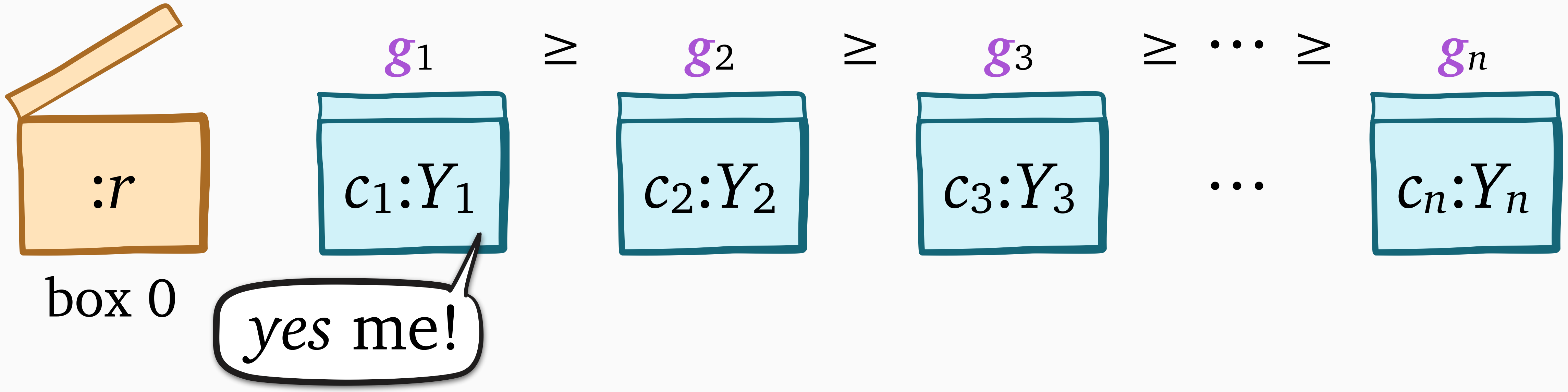
Box 1's perspective



Box 2's perspective



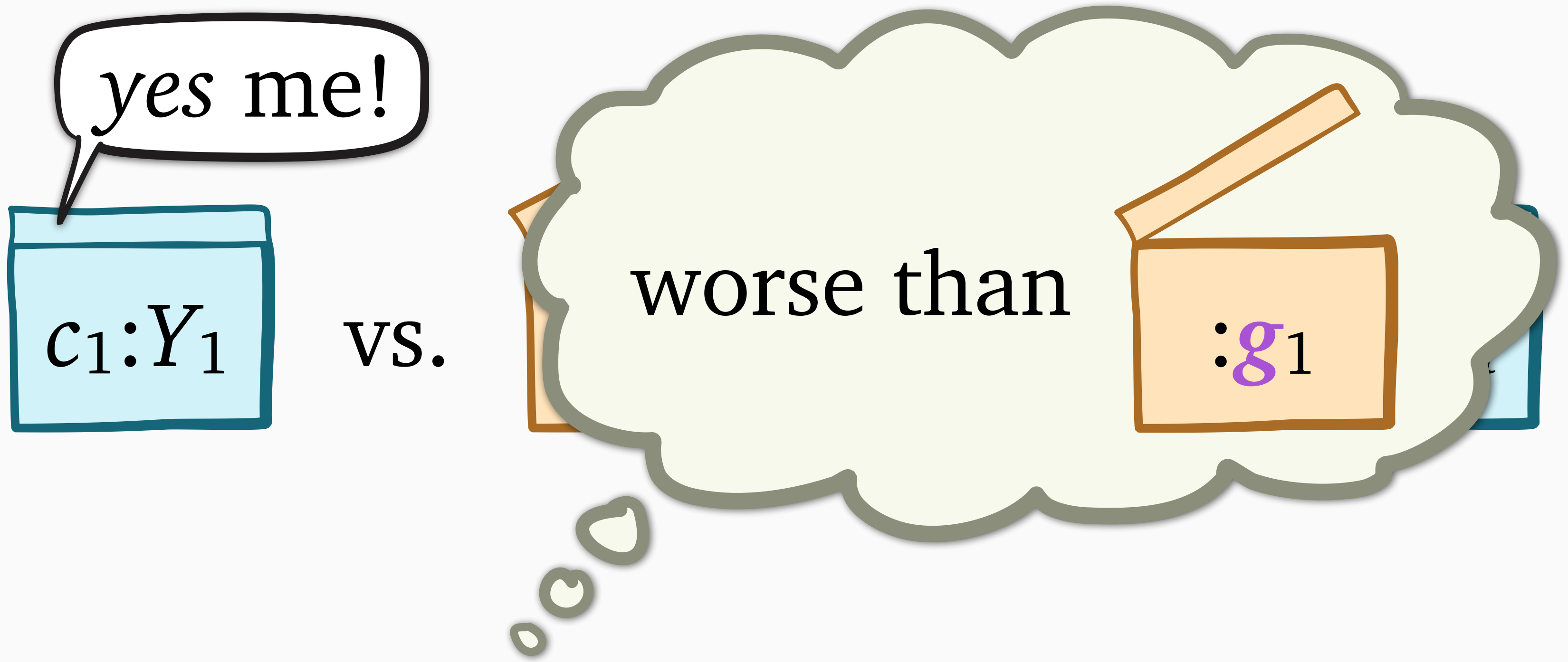
Why is the Gittins index optimal?



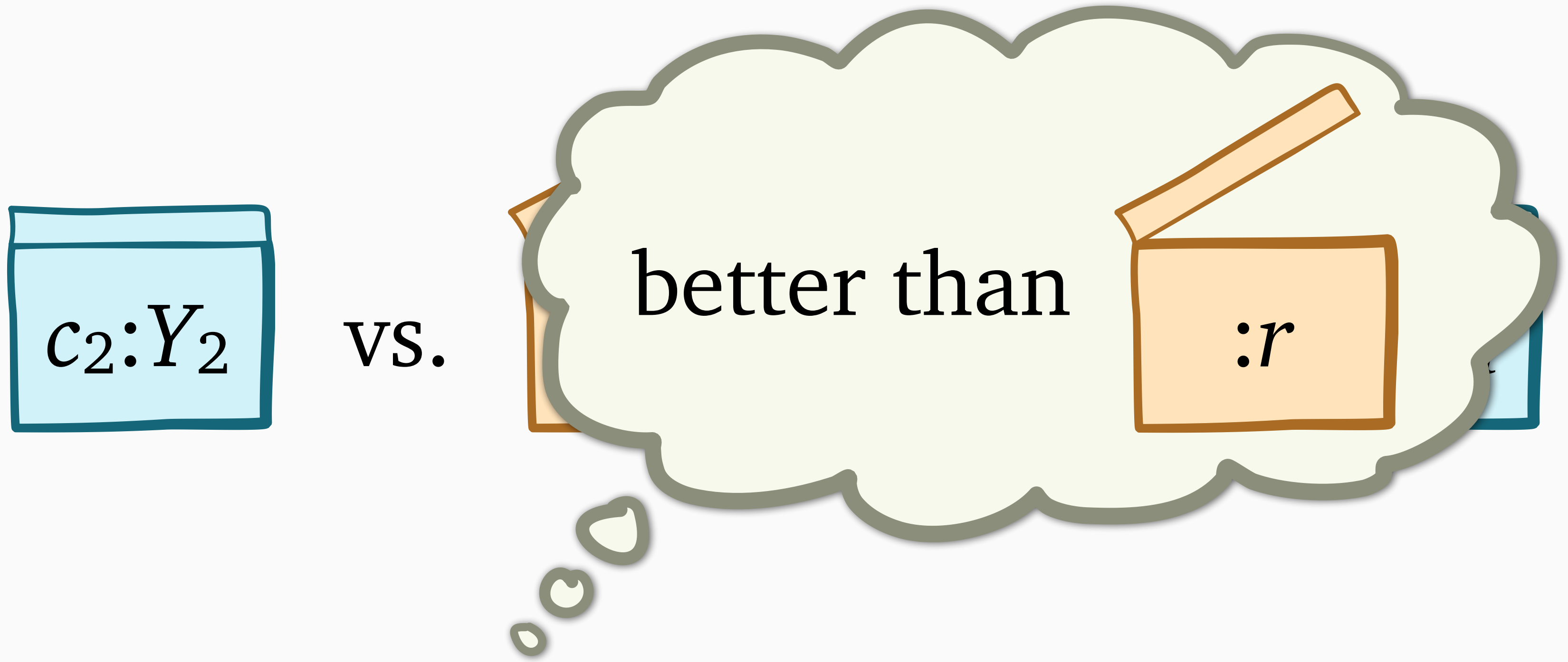
Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq g_1$: *select box 0*
- $r = g_1$: *open box 1*
- $r < g_1$: *open box 1*

Box 1's perspective



Box 2's perspective



What is the **Gittins index**?

The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

What is *(and isn't)* covered by classical **Gittins** theory?

How might we apply **Gittins** *beyond* the classical theory?

What is the **Gittins index**?

The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

1.5-action problem faithfully abstracts full problem

What is *(and isn't)* covered by classical **Gittins** theory?

How might we apply **Gittins** *beyond* the classical theory?

Pandora’s box:

is **Gittins** still optimal if…

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

Pandora's box: is **Gittins** still optimal if…

- … we open k boxes at once?
- … we select k boxes at the end?
- … we select a *spanning tree* of boxes at the end?

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

Pandora's box: is Gittins still optimal if…

✗ … we open k boxes at once?

✓ … we select k boxes at the end?

✓ … we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

Pandora's box:

is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?

Pandora’s box: is Gittins still optimal if…

✗ … we open k boxes at once?

✓ … we select k boxes at the end?

✓ … we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

✗ … we can open at most n boxes?

✓ … we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

… we can select a closed box?

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

Pandora's box: is Gittins still optimal if…

✗ … we open k boxes at once?

✓ … we select k boxes at the end?

✓ … we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

✗ … we can open at most n boxes?

✓ … we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

bandit superprocess

✗ … we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

bandit superprocess

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

… there are correlations between box values?

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

bandit superprocess

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

… there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

bandit superprocess

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

… there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

… there are multiple inspection steps?

Pandora's box: is **Gittins** still optimal if…

… we open *k* boxes at once?

… we select *k* boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most *n* boxes?

… we can open at most *n* boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

bandit superprocess

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

… there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

… there are multiple inspection steps?

Pandora's box: is Gittins still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

bandit superprocess

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

… there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

… there are multiple inspection steps?

… and with discounted costs/rewards

Mean scheduling:

is **Gittins** still optimal if…

Mean scheduling:

is **Gittins** still optimal if…

… we serve k jobs at once?

Mean scheduling: is **Gittins** still optimal if…

… we serve k jobs at once?

Mean scheduling: is **Gittins** still optimal if…

… we serve k jobs at once?

… jobs arrive over time (arbitrary)

Mean scheduling:
is **Gittins** still optimal if…

- ✗ … we serve k jobs at once?
- ✗ … jobs arrive over time (arbitrary)

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once?
- … jobs arrive over time (arbitrary)
- … jobs arrive over time (Poisson)

Mean scheduling: is **Gittins** still optimal if…

- ✗ … we serve k jobs at once?
- ✗ … jobs arrive over time (arbitrary)
- ✓ … jobs arrive over time (Poisson)

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once?
- … jobs arrive over time (arbitrary)
- … jobs arrive over time (Poisson)
- … Poisson arrivals affected by job in service?

Mean scheduling: is Gittins still optimal if…

- ✗ … we serve k jobs at once?
- ✗ … jobs arrive over time (arbitrary)
- ✓ … jobs arrive over time (Poisson)
- ✓ … Poisson arrivals affected by job in service?

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once?
- … jobs arrive over time (arbitrary)
- … jobs arrive over time (Poisson)
- … Poisson arrivals affected by job in service?

branching bandit

Mean scheduling: is **Gittins** still optimal if…

… we serve k jobs at once?

… jobs arrive over time (arbitrary)

… jobs arrive over time (Poisson)

branching bandit

… Poisson arrivals affected by job in service?

… different jobs have different holding costs?

Mean scheduling: is **Gittins** still optimal if…

branching bandit

- ✗ … we serve k jobs at once?
- ✗ … jobs arrive over time (arbitrary)
- ✓ … jobs arrive over time (Poisson)
- ✓ … Poisson arrivals affected by job in service?
- ✓ … different jobs have different holding costs?

Mean scheduling: is **Gittins** still optimal if…

… we serve k jobs at once?

… jobs arrive over time (arbitrary)

… jobs arrive over time (Poisson)

… Poisson arrivals affected by job in service?

… different jobs have different holding costs?

… holding costs change over time (arbitrary)?

branching bandit

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once?
- … jobs arrive over time (arbitrary)
- … jobs arrive over time (Poisson)
- … Poisson arrivals affected by job in service?
- … different jobs have different holding costs?
- … holding costs change over time (arbitrary)?

branching bandit

Mean scheduling:
is **Gittins** still optimal if…

… we serve k jobs at once?

… jobs arrive over time (arbitrary)

… jobs arrive over time (Poisson)

branching bandit

… Poisson arrivals affected by job in service?

… different jobs have different holding costs?

restless bandit

… holding costs change over time (arbitrary)?

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once? ✗
- … jobs arrive over time (arbitrary) ✗
- … jobs arrive over time (Poisson) ✓
- … Poisson arrivals affected by job in service? ✓ (branching bandit)
- … different jobs have different holding costs? ✓
- … holding costs change over time (arbitrary)? ✗ (restless bandit)
- … holding costs change over time (convex)?

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once?
- … jobs arrive over time (arbitrary)
- … jobs arrive over time (Poisson)
- … Poisson arrivals affected by job in service?
- … different jobs have different holding costs?
- … holding costs change over time (arbitrary)?
- … holding costs change over time (convex)?

branching bandit

restless bandit

Mean scheduling: is **Gittins** still optimal if…

branching bandit

restless bandit

… we serve *k* jobs at once?

… jobs arrive over time (arbitrary)

… jobs arrive over time (Poisson)

… Poisson arrivals affected by job in service?

… different jobs have different holding costs?

… holding costs change over time (arbitrary)?

… holding costs change over time (convex)?

… holding costs change *during service*?

Mean scheduling: is **Gittins** still optimal if…

- … we serve k jobs at once? ✗
- … jobs arrive over time (arbitrary) ✗
- … jobs arrive over time (Poisson) ✓
- … Poisson arrivals affected by job in service? ✓
- … different jobs have different holding costs? ✓
- … holding costs change over time (arbitrary)? ✗
- … holding costs change over time (convex)? ✗
- … holding costs change *during service*? ✓

branching bandit

restless bandit

Mean scheduling: is **Gittins** still optimal if…

- ✗ … we serve k jobs at once?
- ✗ … jobs arrive over time (arbitrary)
- ✓ … jobs arrive over time (Poisson)
- ✓ … Poisson arrivals affected by job in service?
- ✓ … different jobs have different holding costs?
- ✗ … holding costs change over time (arbitrary)?
- ✗ … holding costs change over time (convex)?
- ✓ … holding costs change *during service*?

branching bandit

restless bandit

… and with discounted costs/rewards

What is the **Gittins index**?

The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

1.5-action problem faithfully abstracts full problem

What is *(and isn't)* covered by classical **Gittins** theory?

How might we apply **Gittins** *beyond* the classical theory?

What is the **Gittins index**?

The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

1.5-action problem faithfully abstracts full problem

What is *(and isn't)* covered by classical **Gittins** theory?

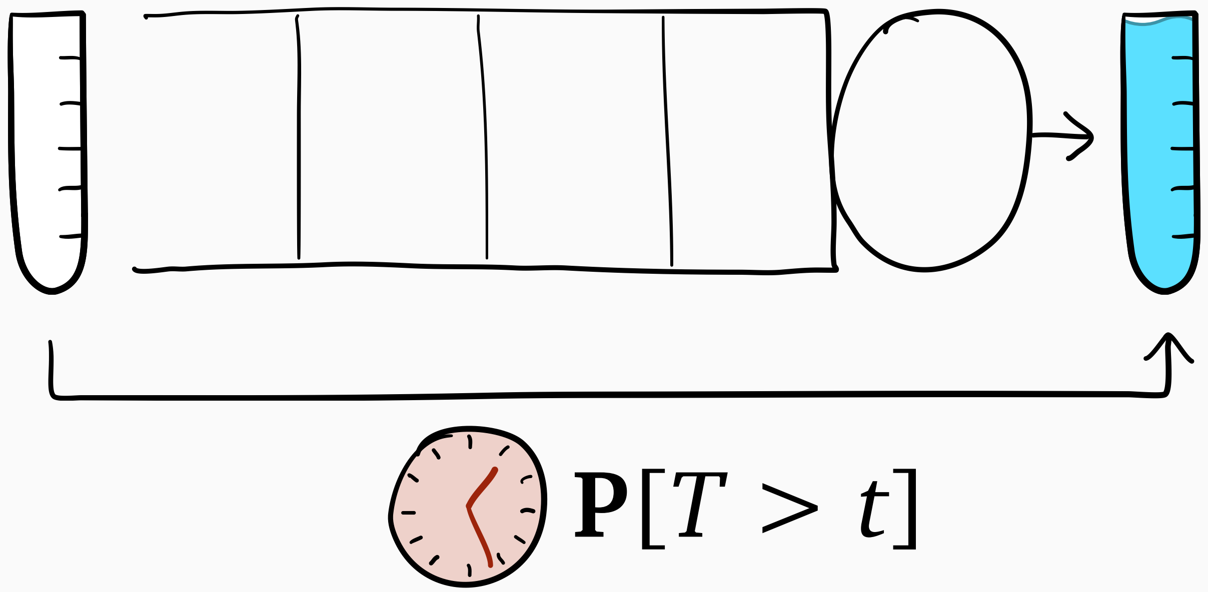
Need independent Markov reward/cost processes (possibly with branching)

How might we apply **Gittins** *beyond* the classical theory?

Two new Gittins applications

Tail scheduling

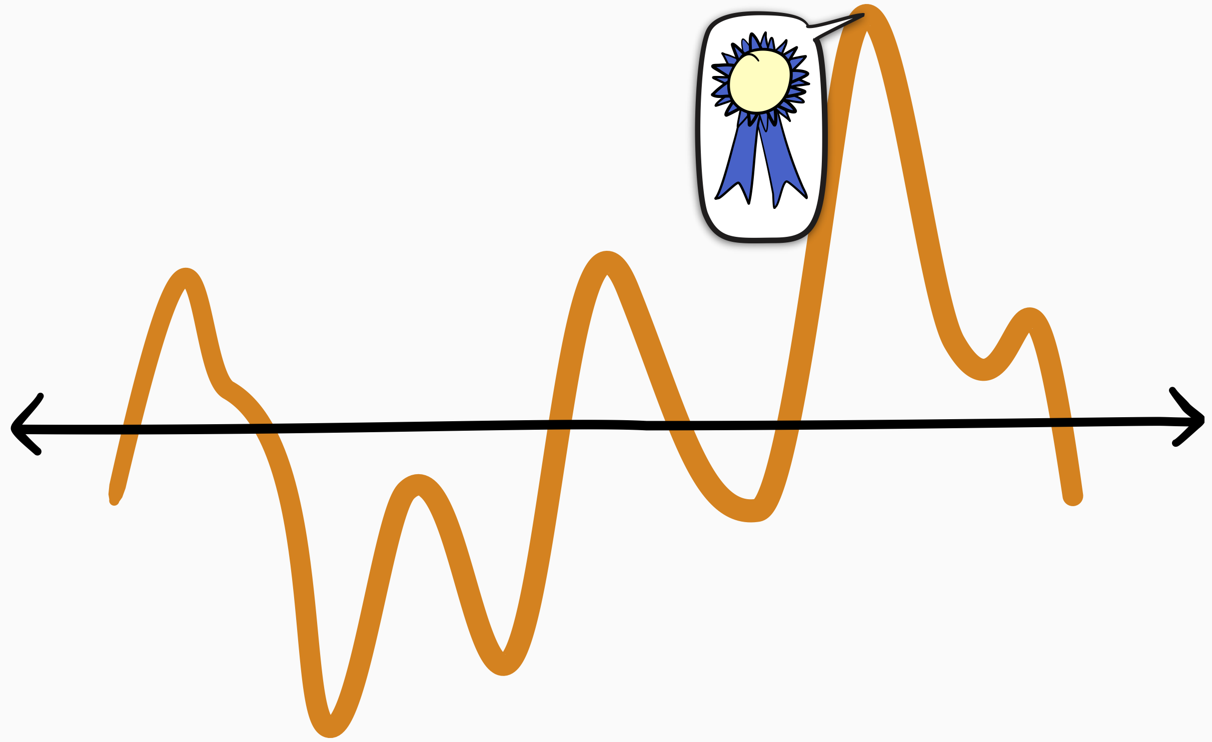
(in single-server queues)



Goal: minimize probability of very long response time

BayesOpt

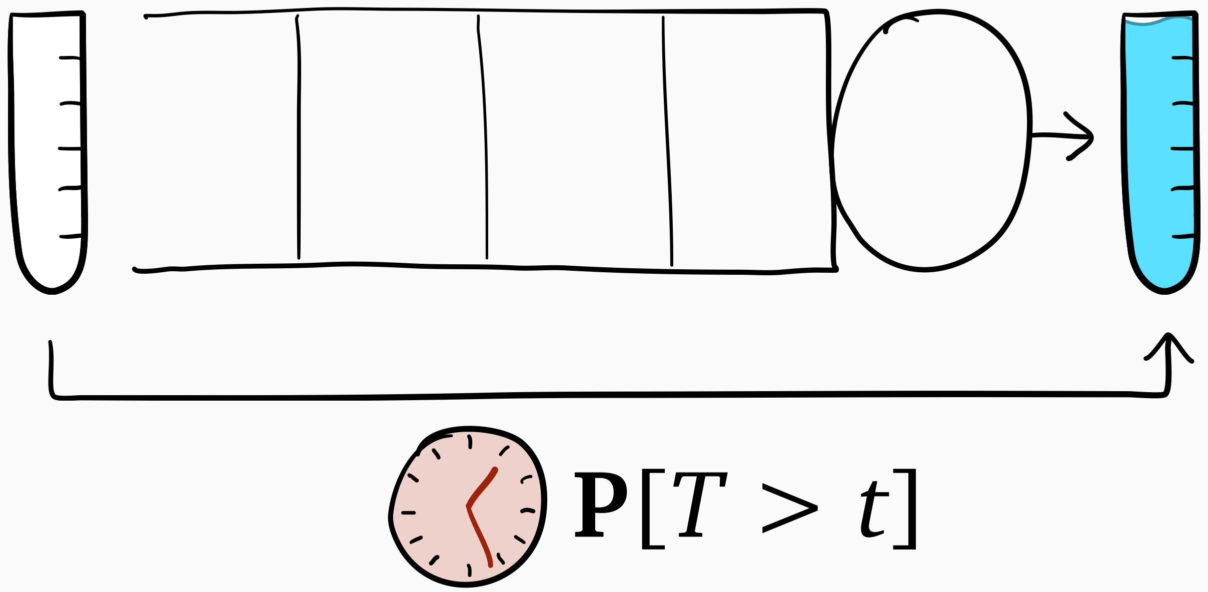
(Bayesian optimization)



Goal: find large function value with few function evaluations

Two new Gittins applications

Tail scheduling
(in single-server queues)



$\mathbf{P}[T > t]$

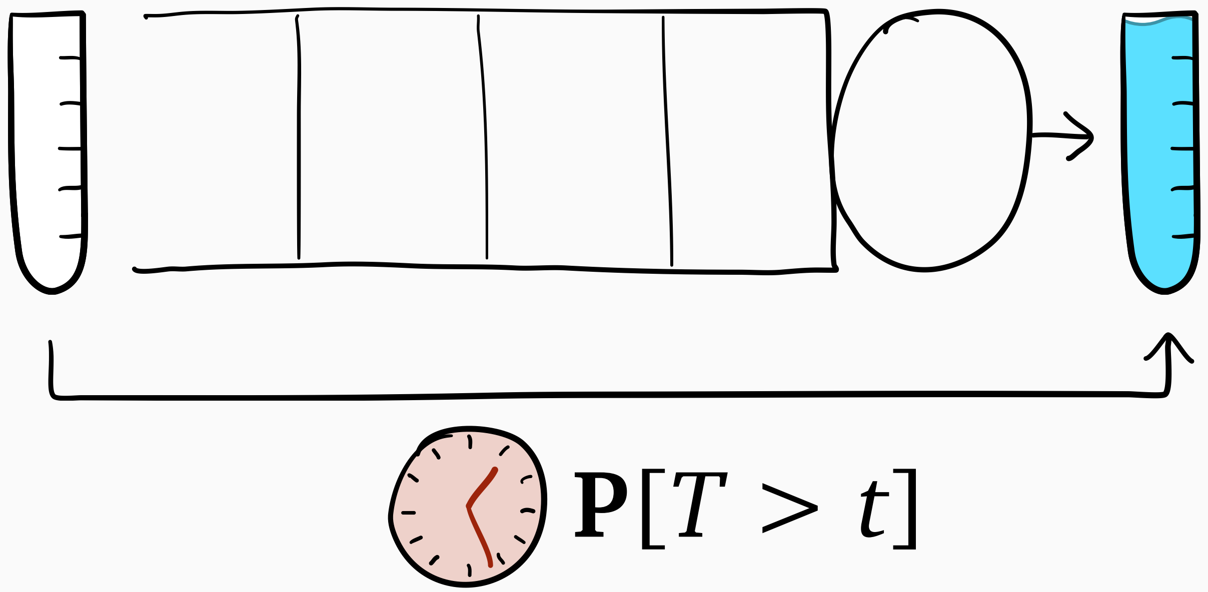
Goal: minimize probability of very long response time

BayesOpt
(Bayesian optimization)

Goal: find large function value with few function evaluations

Two new Gittins applications

Tail scheduling
(in single-server queues)



Goal: minimize probability of very long response time

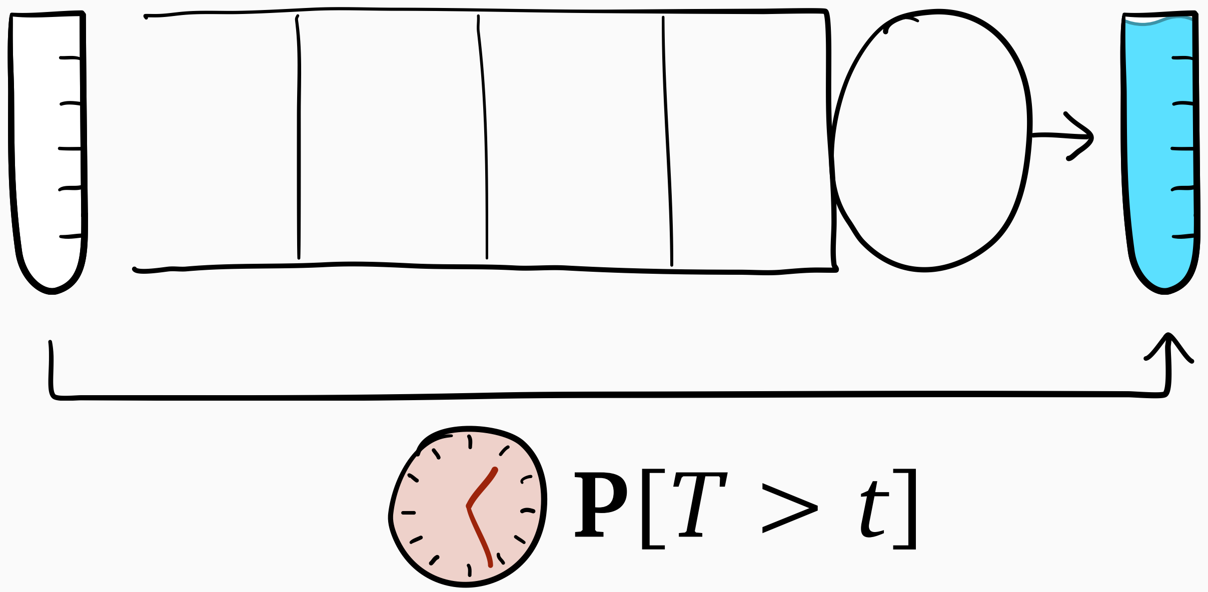
BayesOpt
(Bayesian optimization)

Goal: find large function value with few function evaluations

Two new Gittins applications

Tail scheduling

(in single-server queues)



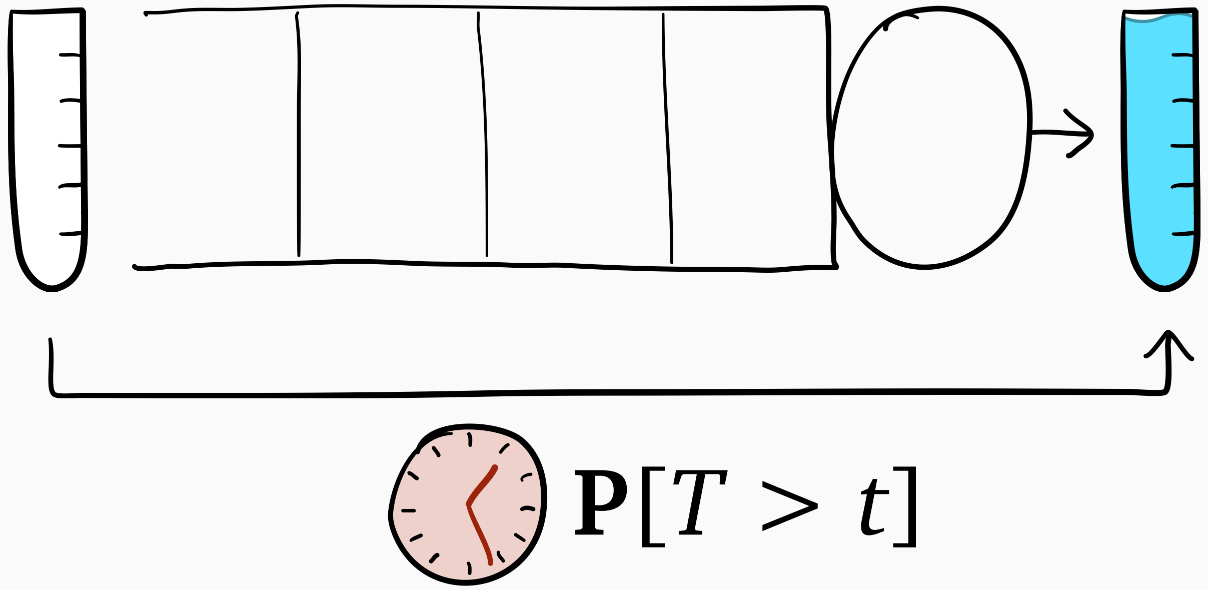
$\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

Two new Gittins applications

Tail scheduling

(in single-server queues)



$\mathbf{P}[T > t]$

Goal: minimize probability of very long response time

Laplace transform result:

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

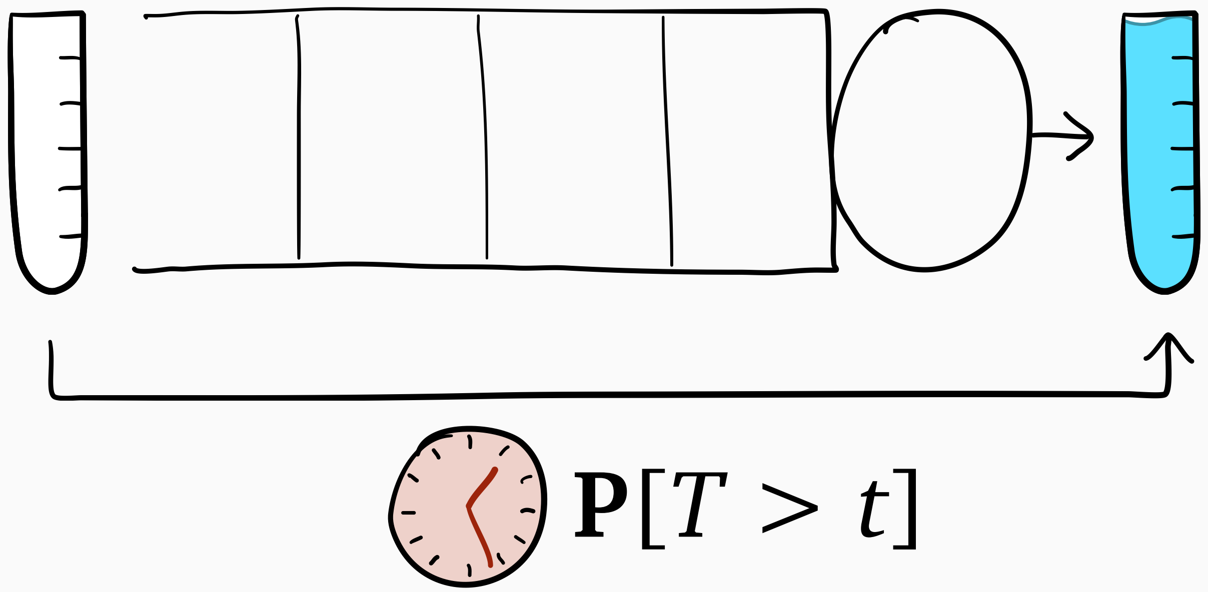
$$\Updownarrow$$

$$\mathbf{E}[e^{(\gamma-\varepsilon)T}] \sim \frac{\gamma C}{\varepsilon}$$

Two new Gittins applications

Tail scheduling

(in single-server queues)



Goal: minimize probability of very long response time

Laplace transform result:

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

$$\Updownarrow$$

$$\mathbf{E}[e^{(\gamma-\varepsilon)T}] \sim \frac{\gamma C}{\varepsilon}$$

exponential holding cost

Mean scheduling:
is **Gittins** still
optimal if…

- ✗ … we serve *k* jobs at once?
- ✗ … jobs arrive over time (arbitrary)
- ✓ … jobs arrive over time (Poisson)
- ✓ … Poisson arrivals affected by job in service?
- ✓ … different jobs have different holding costs?
- ✗ … holding costs change over time (arbitrary)?
- ✗ … holding costs change over time (convex)?
- ✓ … holding costs change *during service*?

… and with discounted
costs/rewards

Mean scheduling: is Gittins still optimal if…

George Yu

Amit Harlev

or inflated!

… and with discounted costs/rewards

- ✗ … we serve *k* jobs at once?
- ✗ … jobs arrive over time (arbitrary)
- ✓ … jobs arrive over time (Poisson)
- ✓ … Poisson arrivals affected by job in service?
- ✓ … different jobs have different holding costs?
- ✗ … holding costs change over time (arbitrary)?
- ✗ … holding costs change over time (convex)?
- ✓ … holding costs change *during service*?

Mean scheduling: is Gittins still optimal if…

… we serve k jobs at once?

… jobs arrive over time (arbitrary)

… jobs arrive over time (Poisson)

… Poisson arrivals affected by job in service?

… different jobs have different holding costs?

… holding costs change over time (arbitrary)?

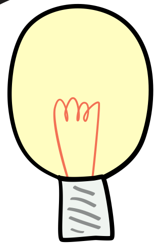
… holding costs change over time (convex)?

… holding costs ch e?

exponential

George Yu

Amit Harlev



or inflated!

… and with discounted costs/rewards

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

… there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

… there are multiple inspection steps?

Pandora's box: is **Gittins** still optimal if…

… we open k boxes at once?

… we select k boxes at the end?

… we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

… we can open at most n boxes?

Choices are difficult

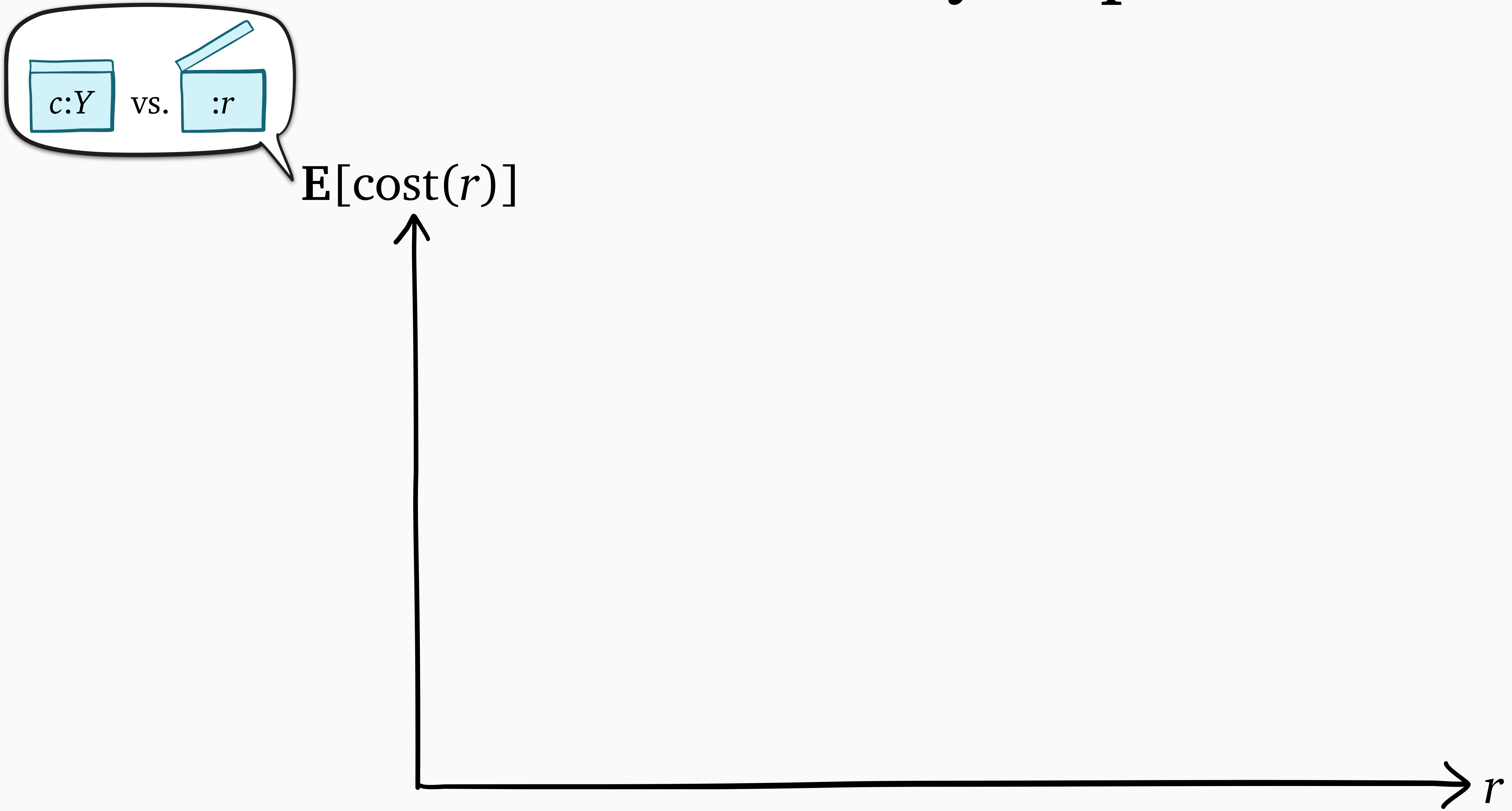
… we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

… we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

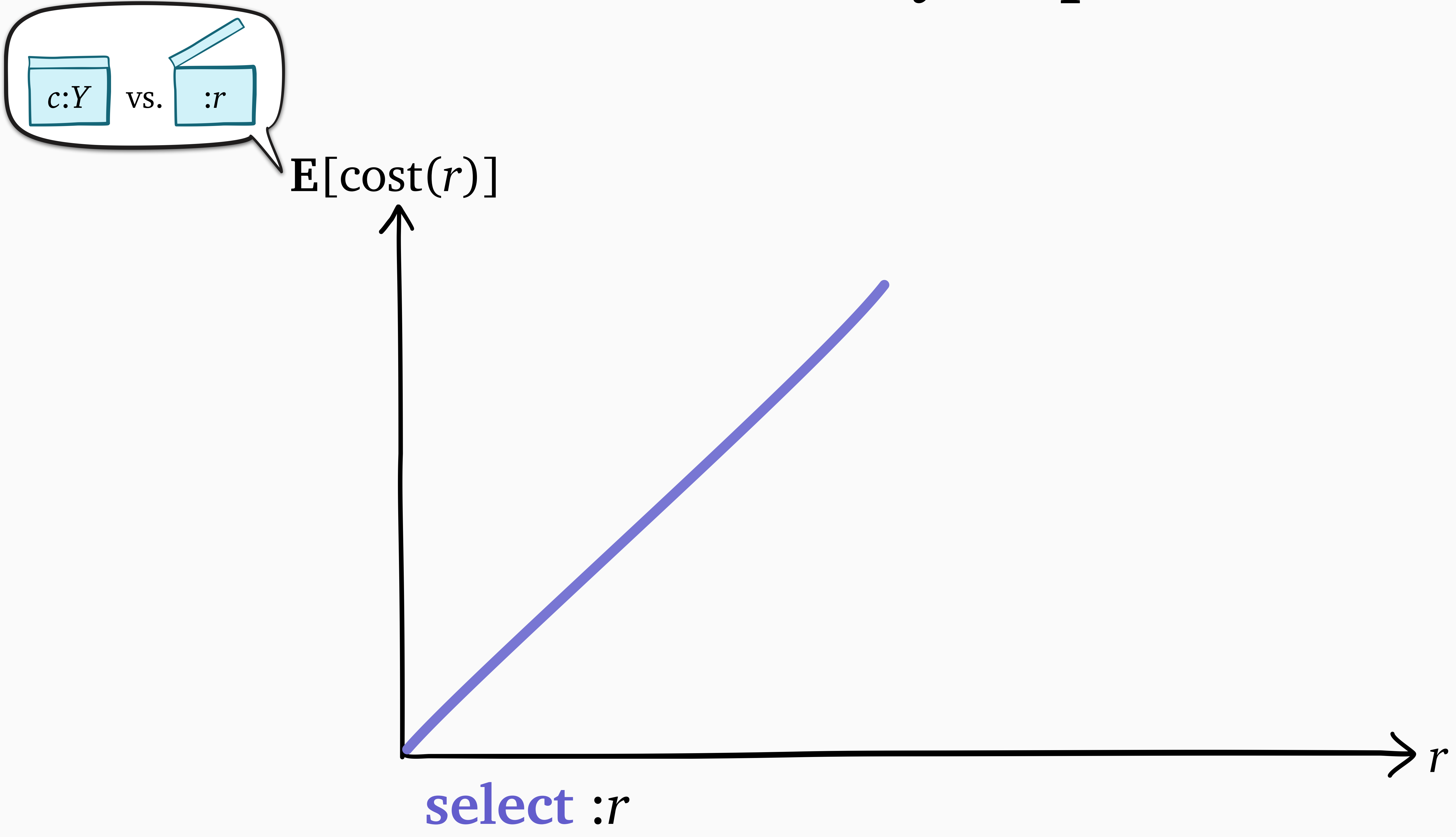
… there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

… there are multiple inspection steps?

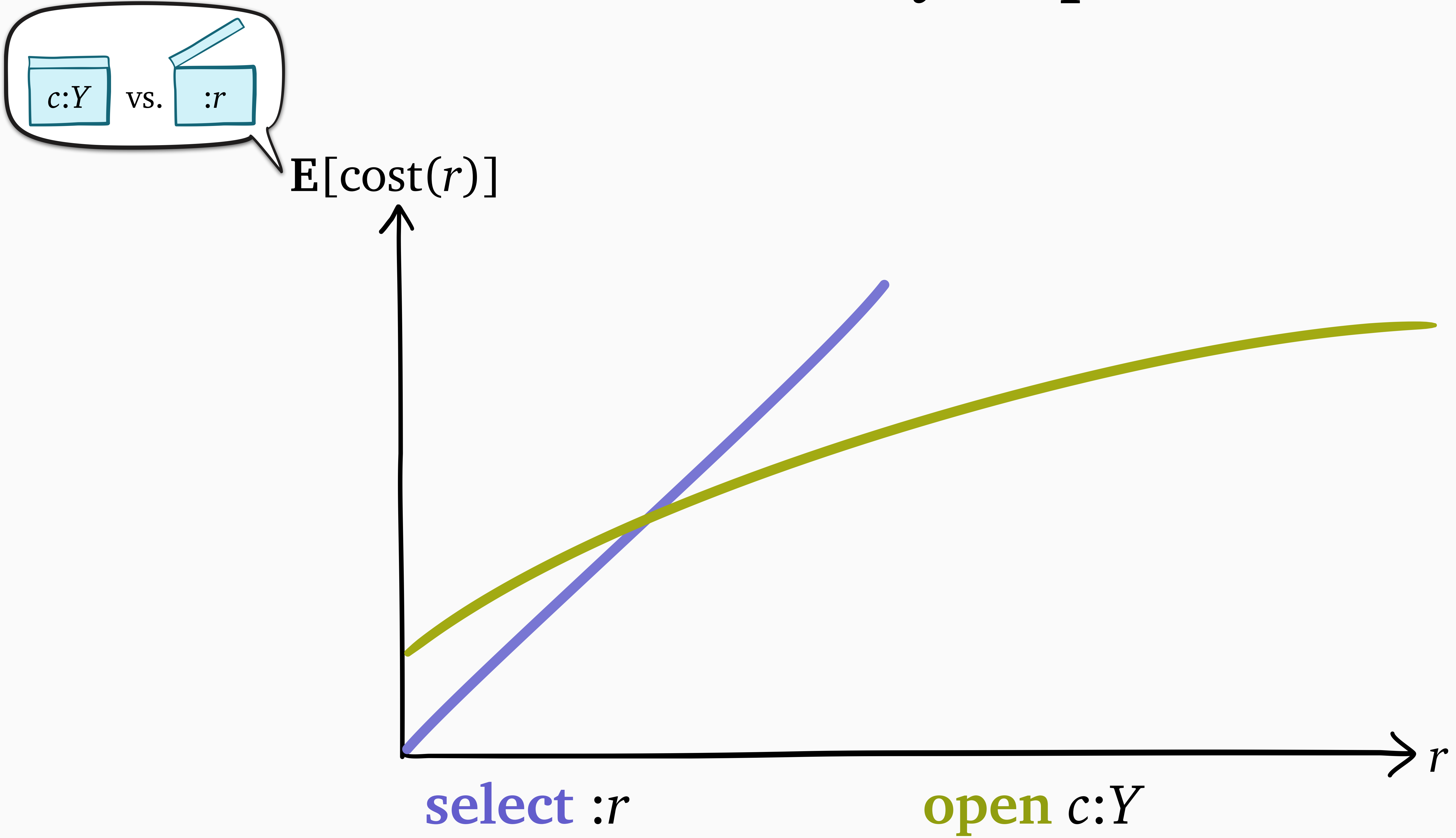
Choices are hard if they depend on context



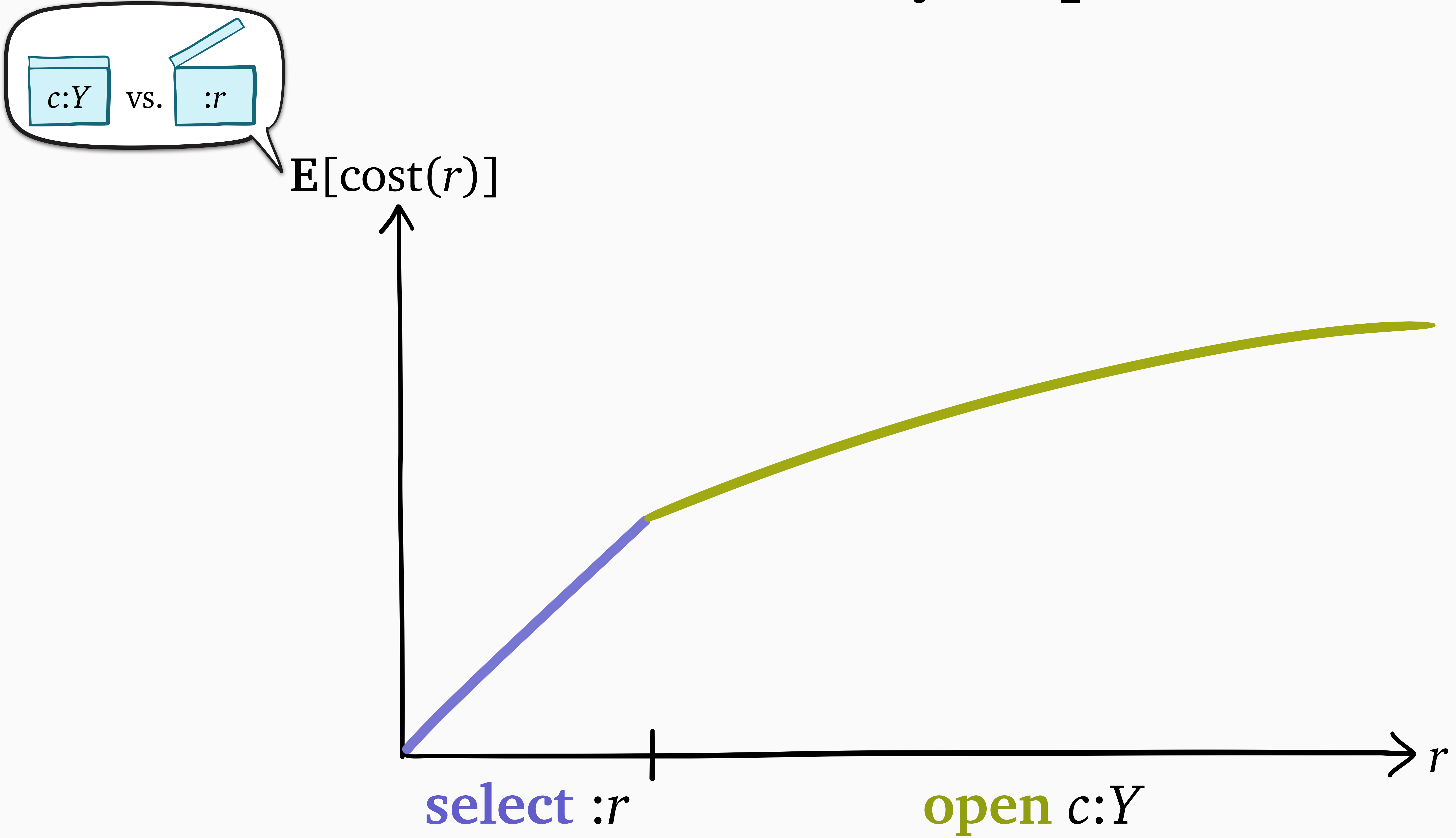
Choices are hard if they depend on context



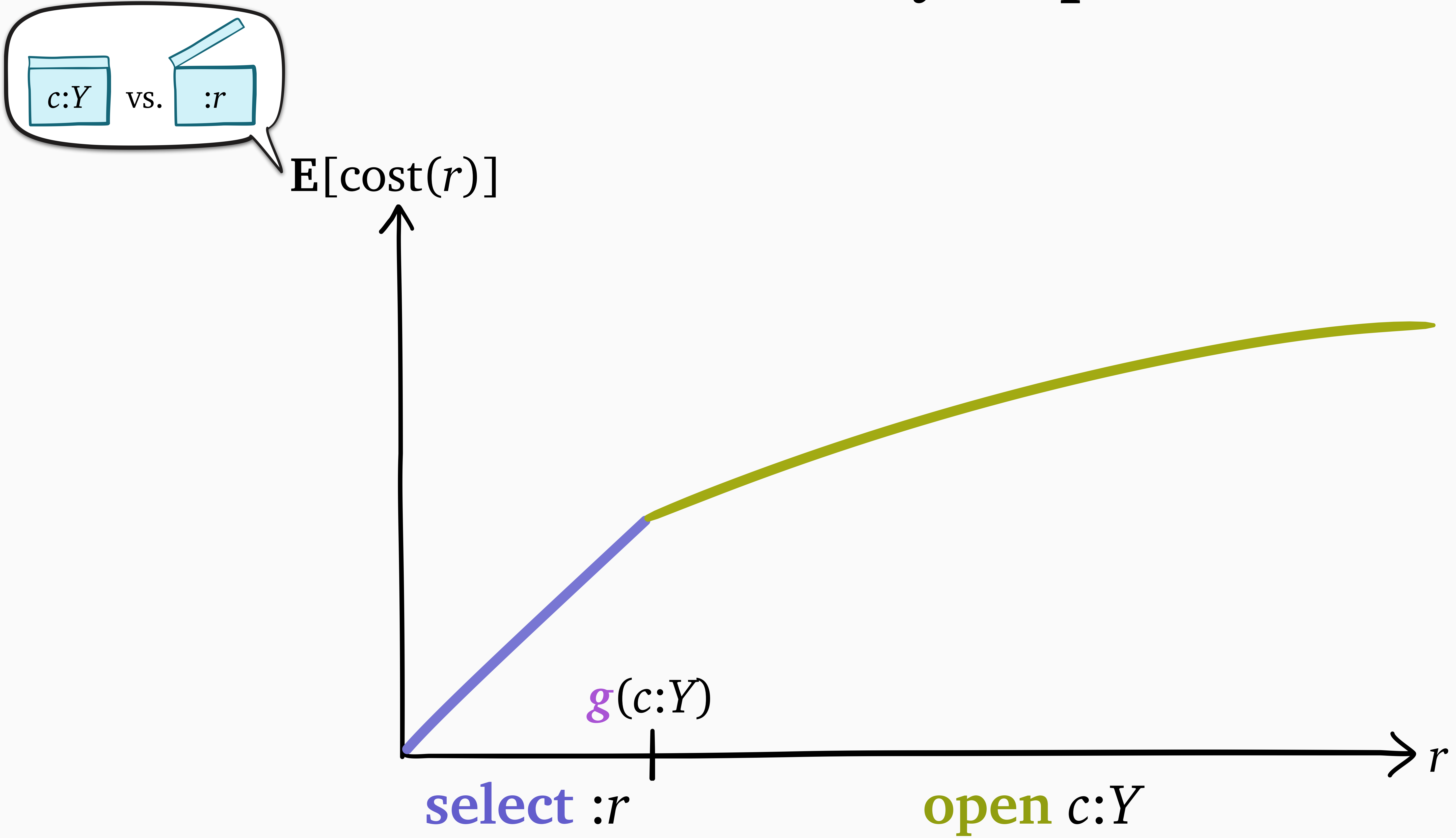
Choices are hard if they depend on context



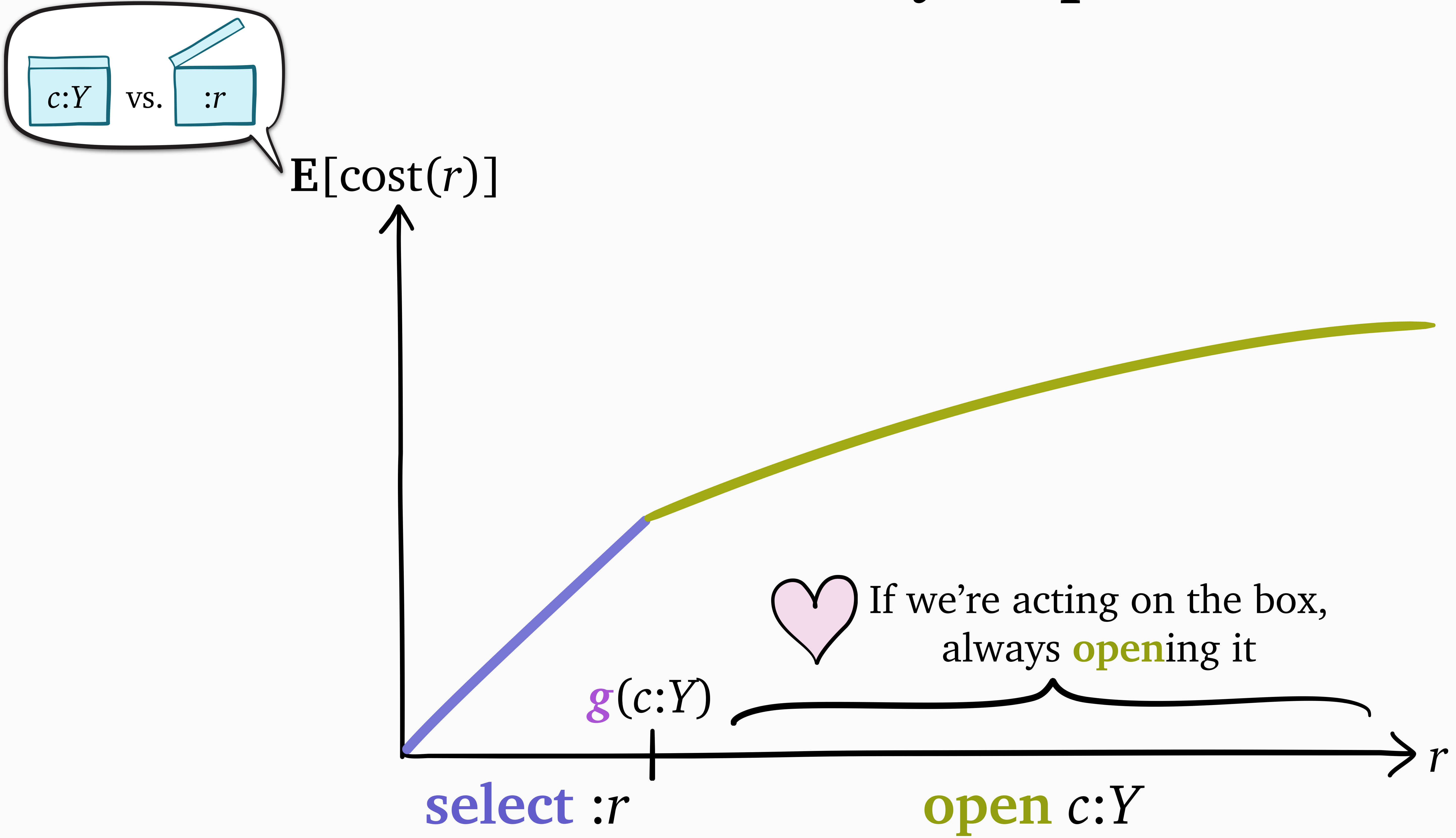
Choices are hard if they depend on context



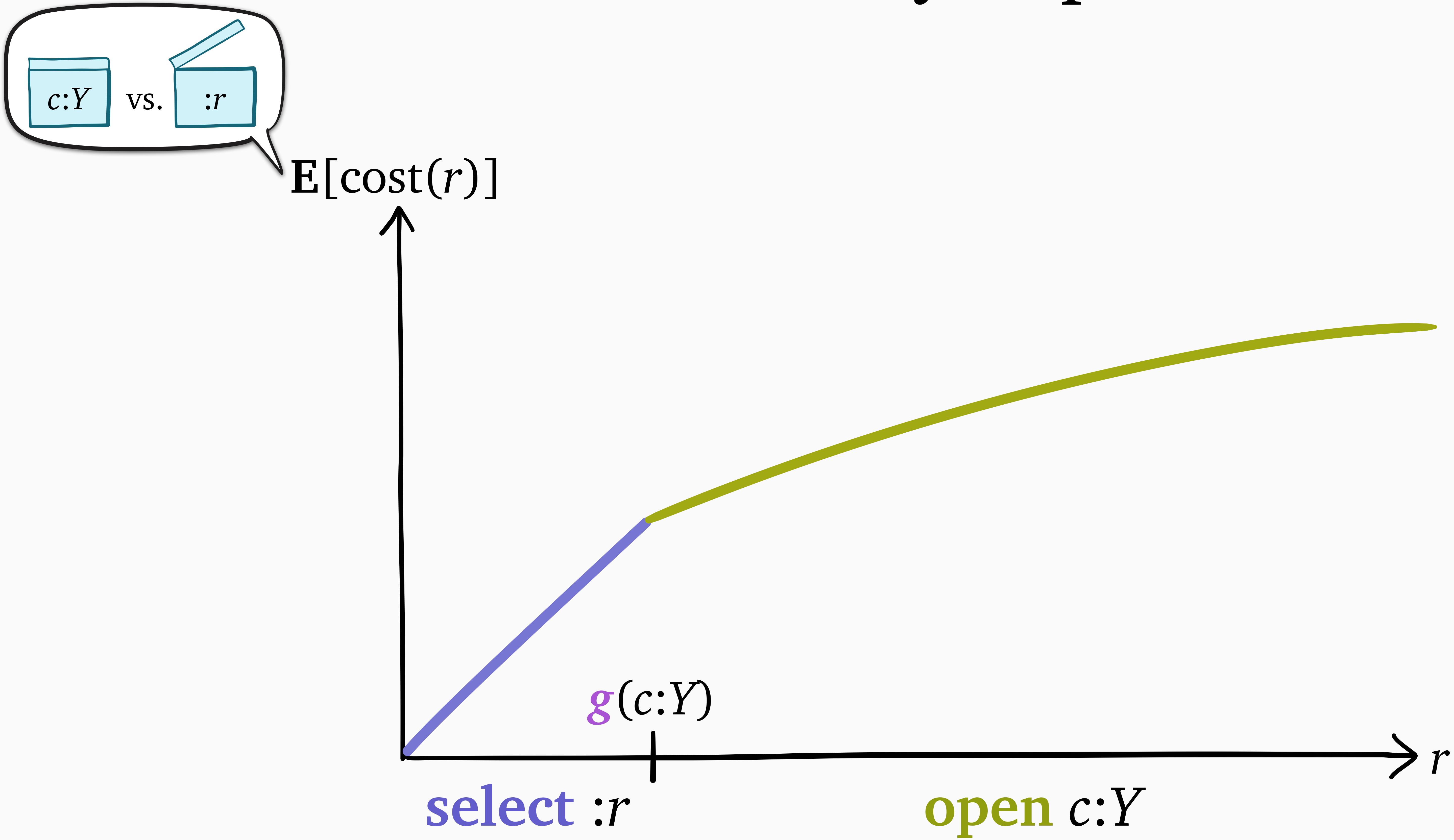
Choices are hard if they depend on context



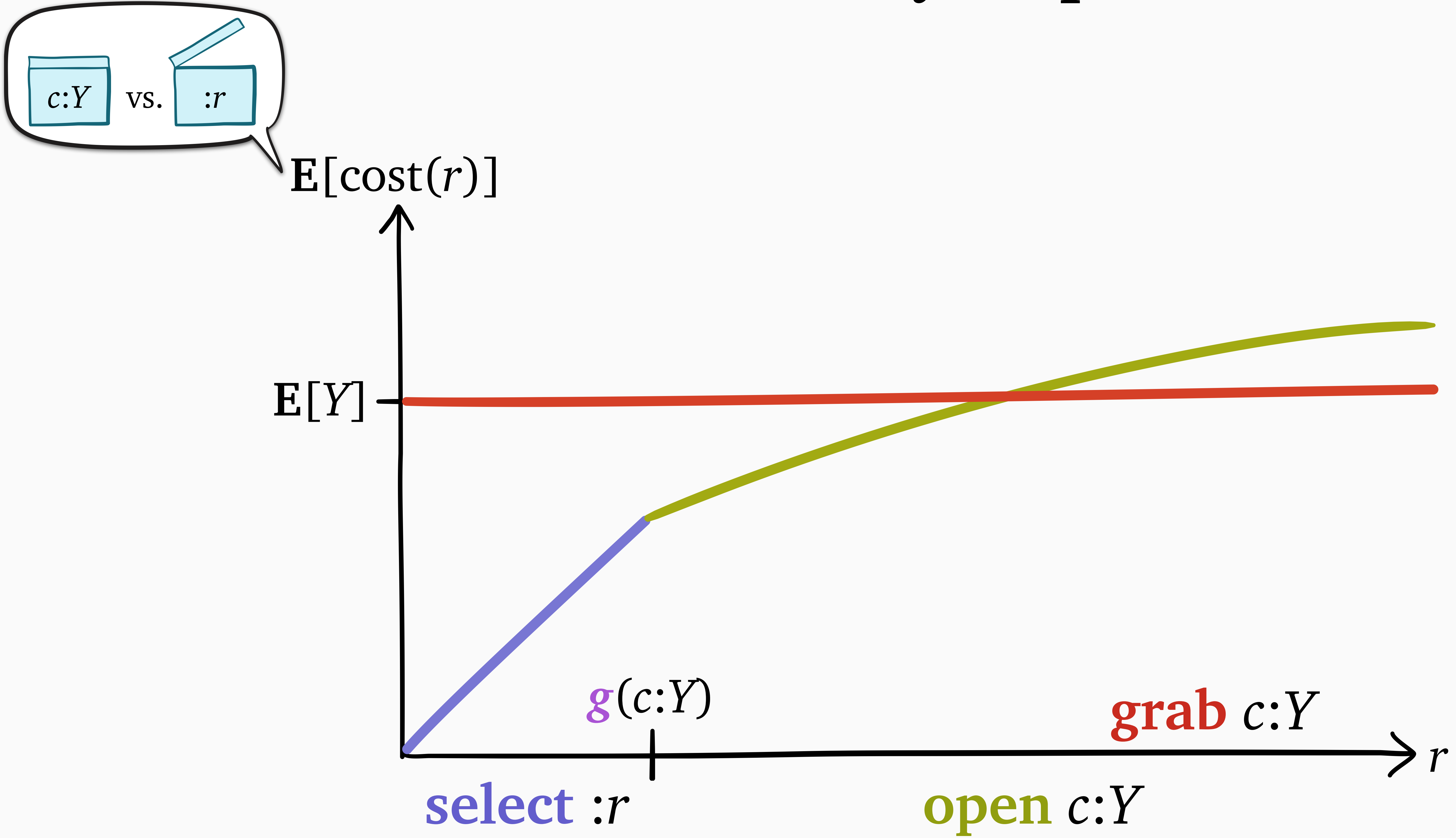
Choices are hard if they depend on context



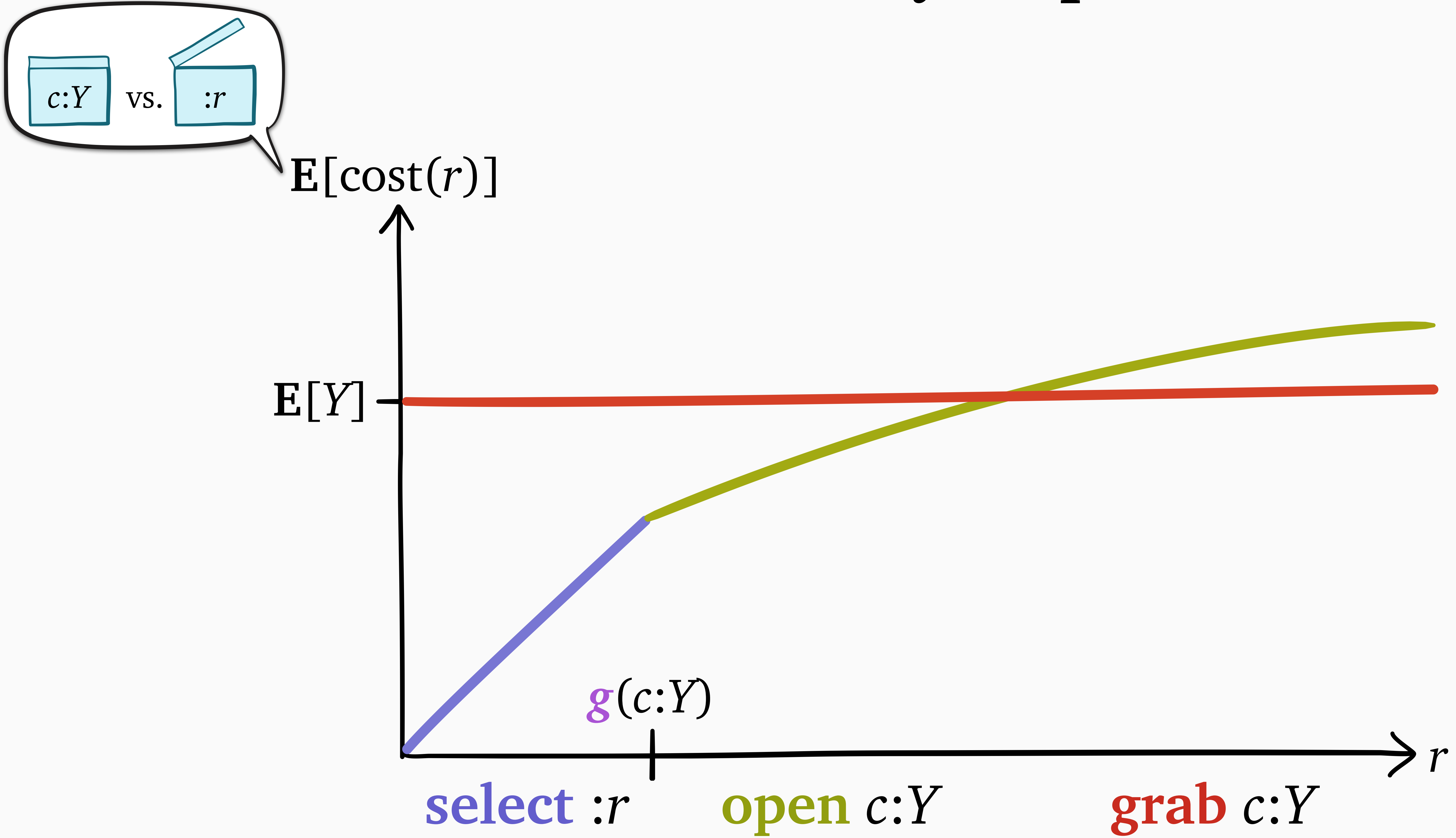
Choices are hard if they depend on context



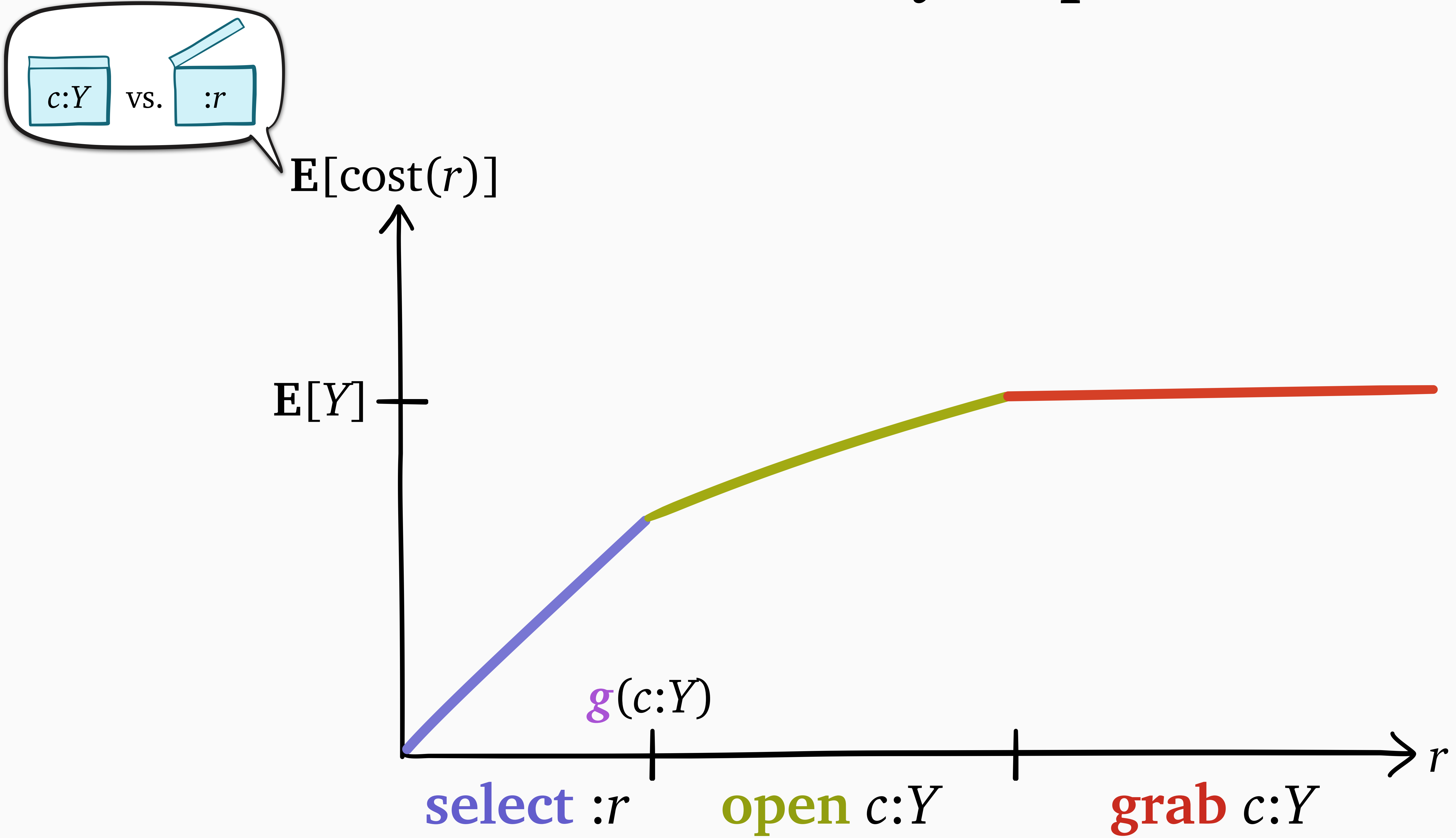
Choices are hard if they depend on context



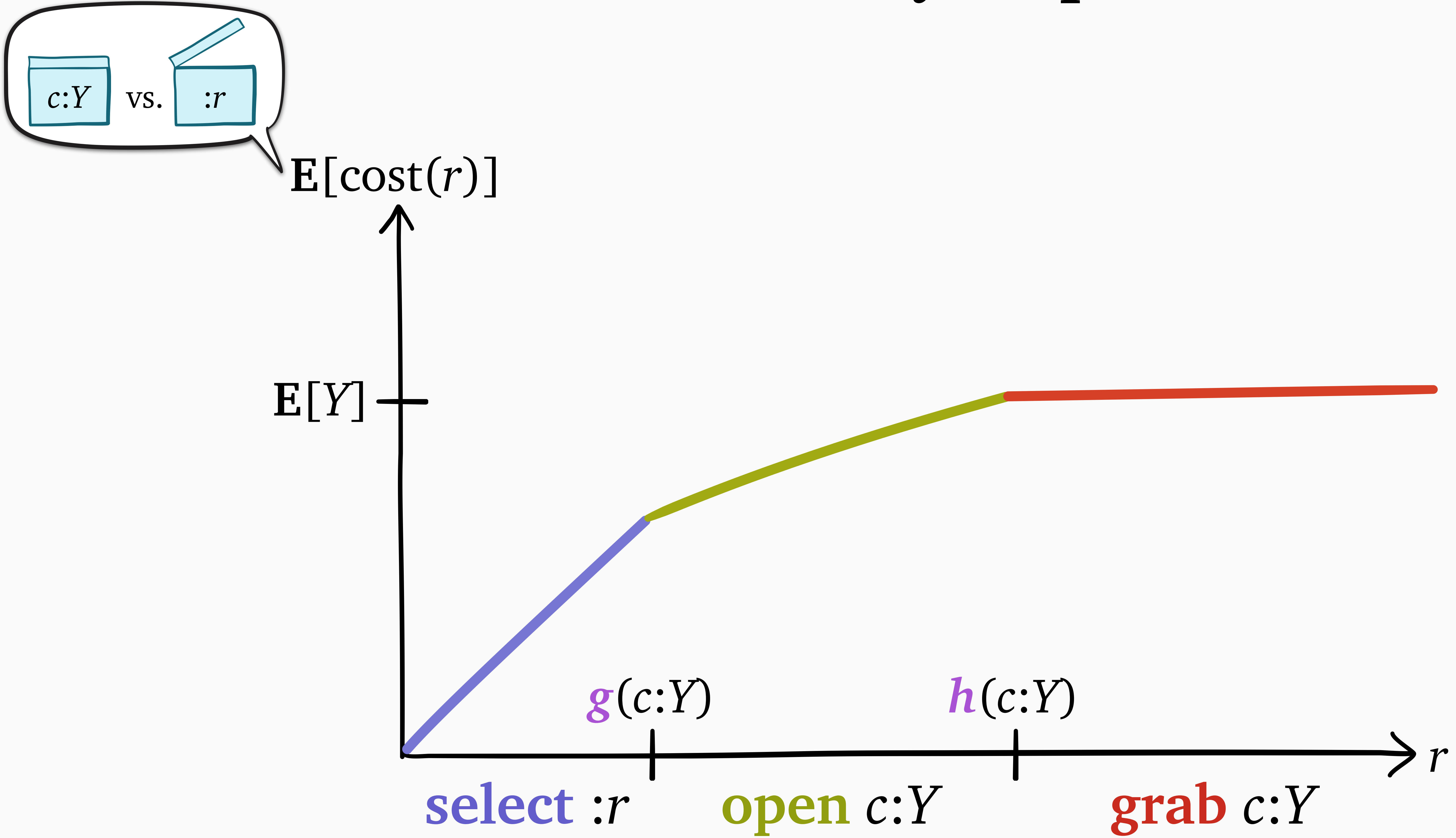
Choices are hard if they depend on context



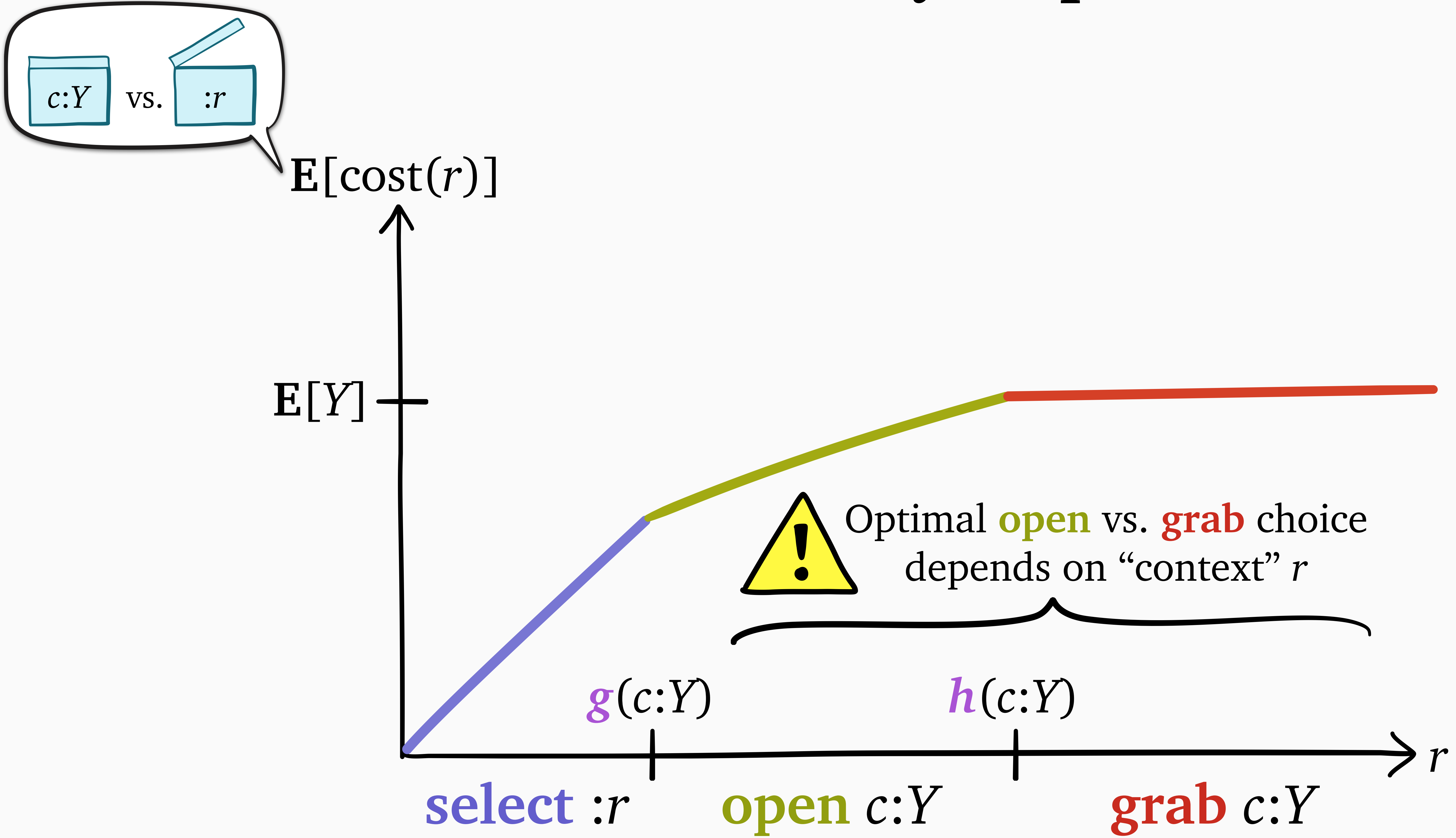
Choices are hard if they depend on context



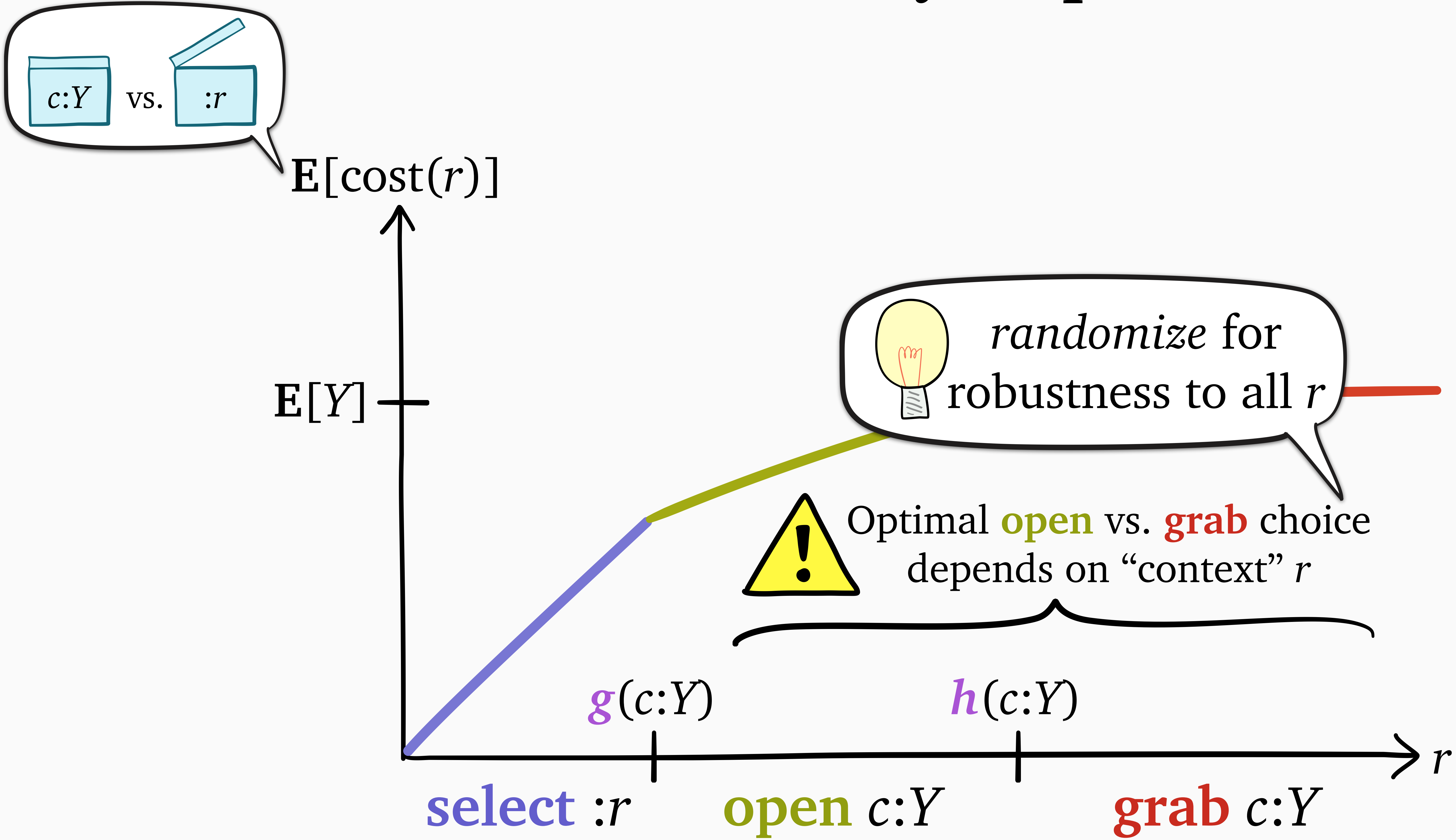
Choices are hard if they depend on context



Choices are hard if they depend on context

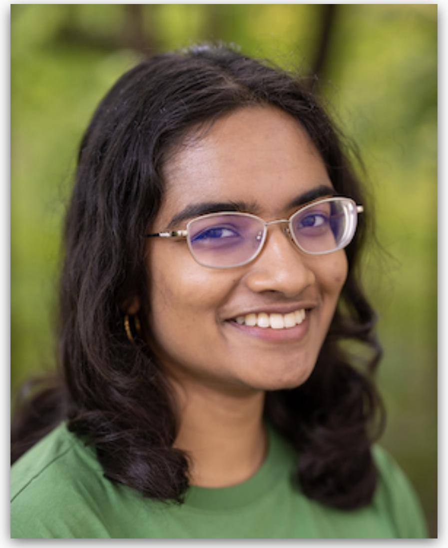


Choices are hard if they depend on context



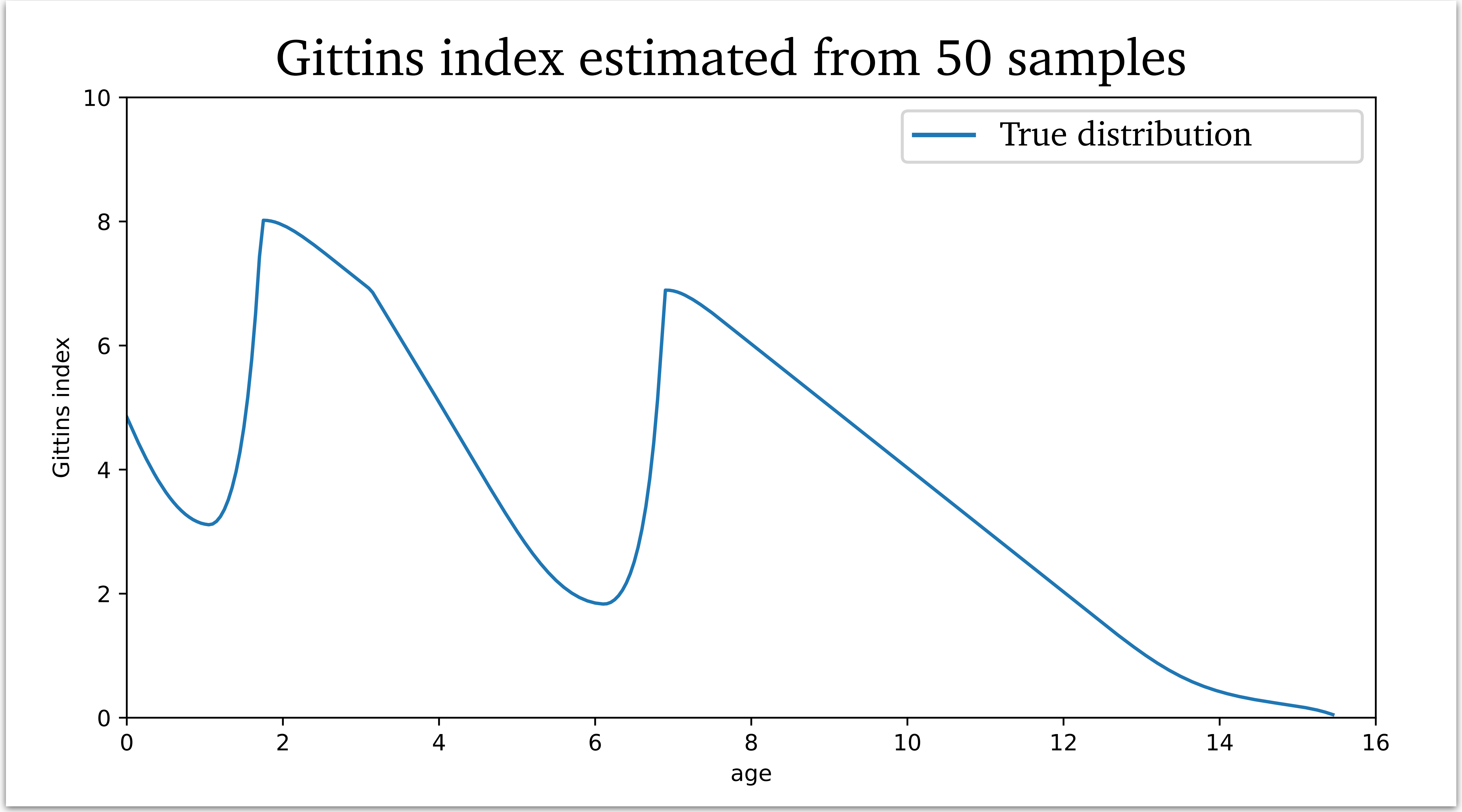
Laura Doval

Open question: **Gittins** from empirical data?

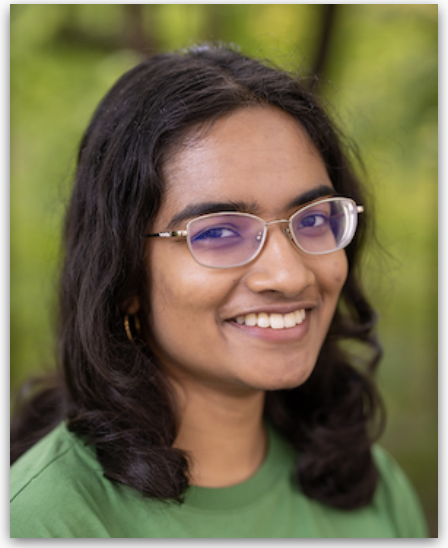


Shefali Ramakrishna

Amit Harlev

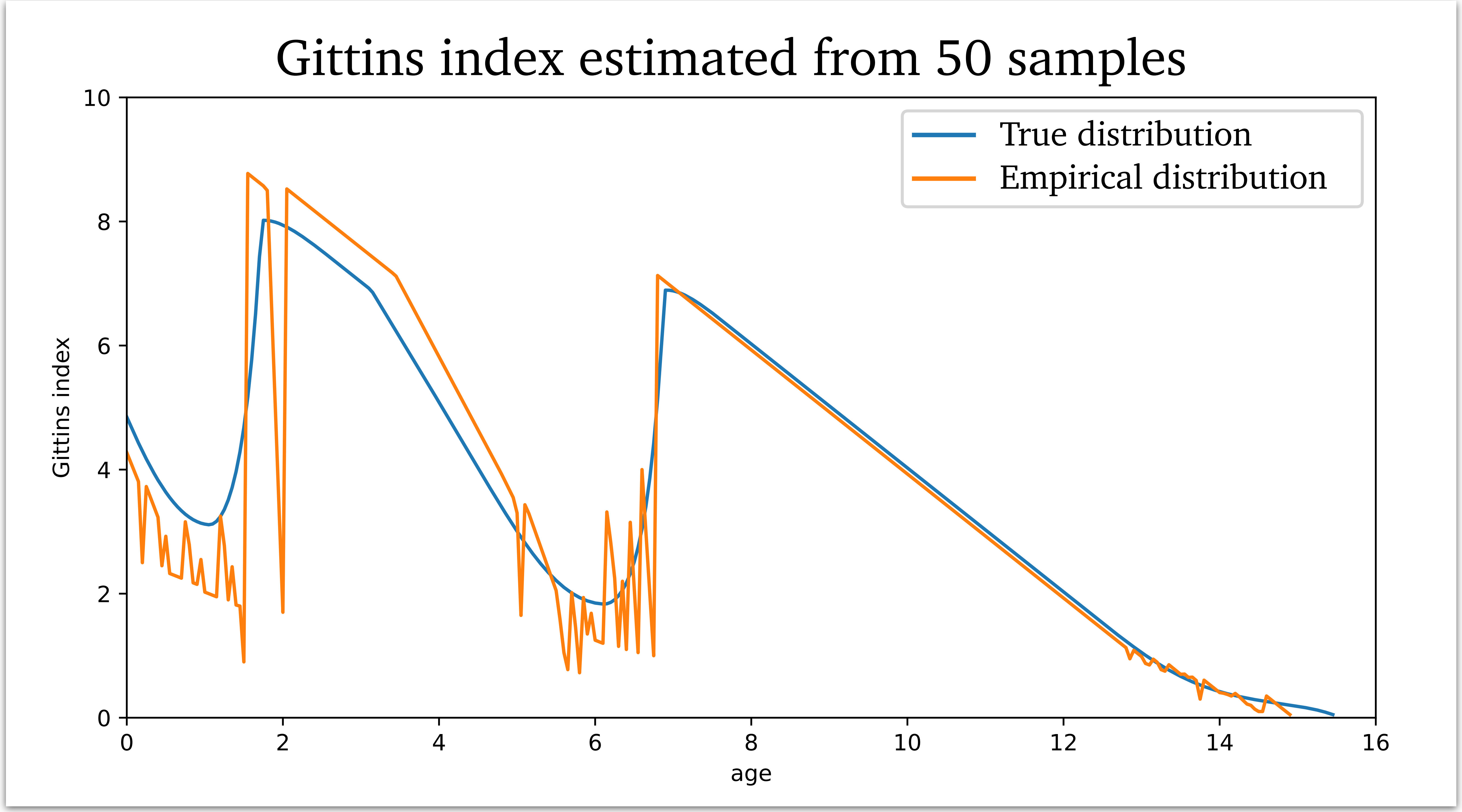


Open question: **Gittins** from empirical data?

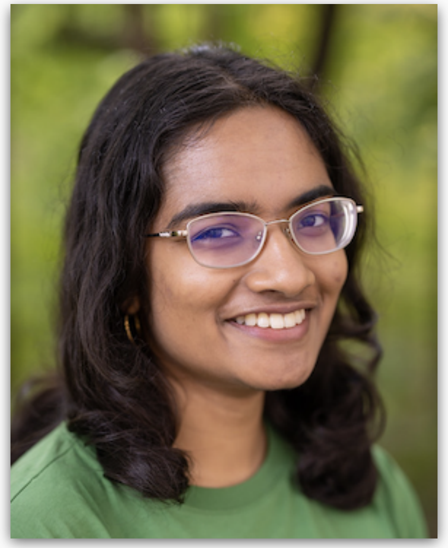


Shefali Ramakrishna

Amit Harlev

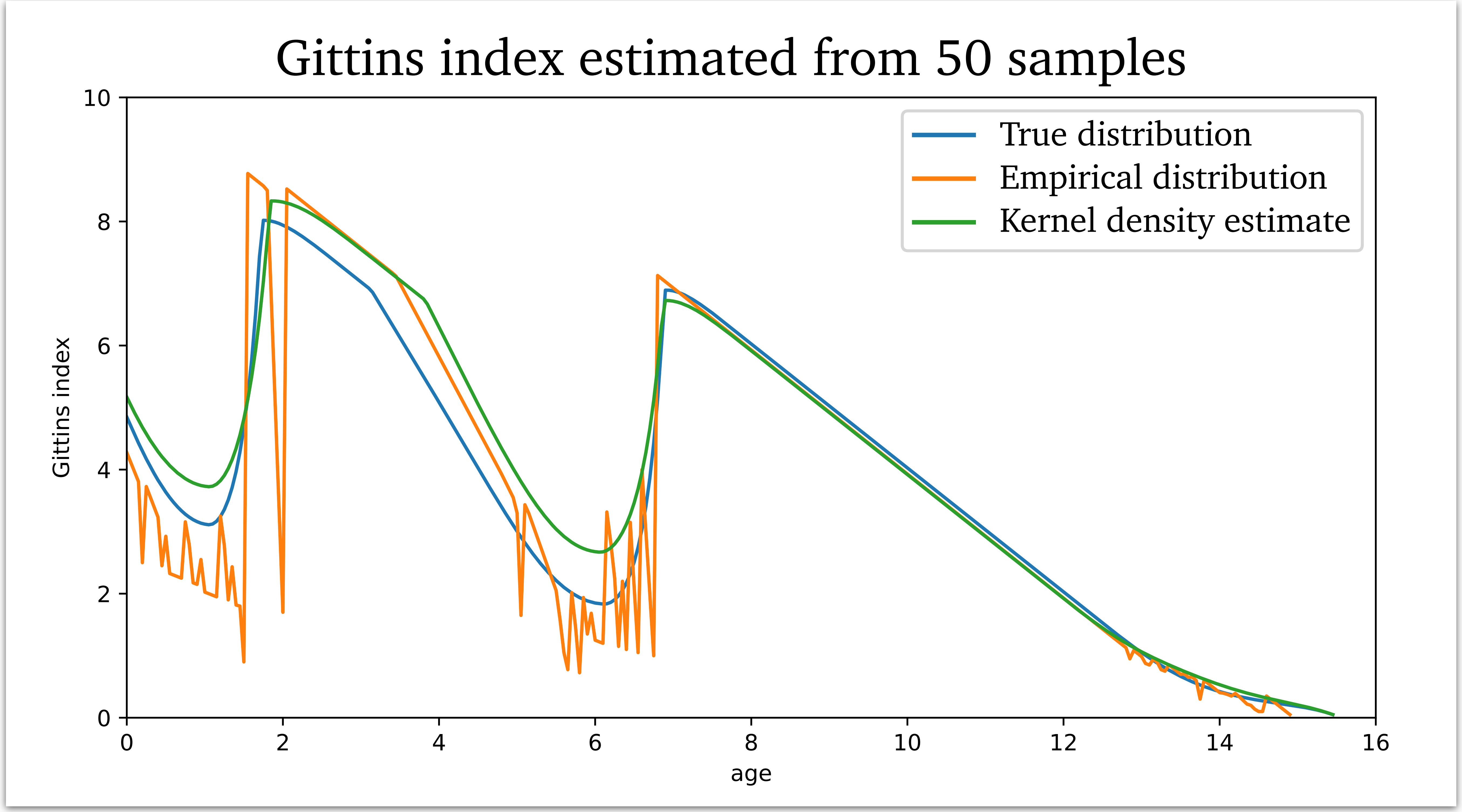


Open question: Gittins from empirical data?



Shefali Ramakrishna

Amit Harlev



What is the **Gittins index**?

The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

1.5-action problem faithfully abstracts full problem

What is *(and isn't)* covered by classical **Gittins** theory?

Need independent Markov reward/cost processes (possibly with branching)

How might we apply **Gittins** *beyond* the classical theory?

What is the **Gittins index**?

The deterministic action that dominates a stochastic action

Why is **Gittins** optimal?

1.5-action problem faithfully abstracts full problem

What is *(and isn't)* covered by classical **Gittins** theory?

Need independent Markov reward/cost processes (possibly with branching)

How might we apply **Gittins** *beyond* the classical theory?

Lots of approaches, but also: just try it!