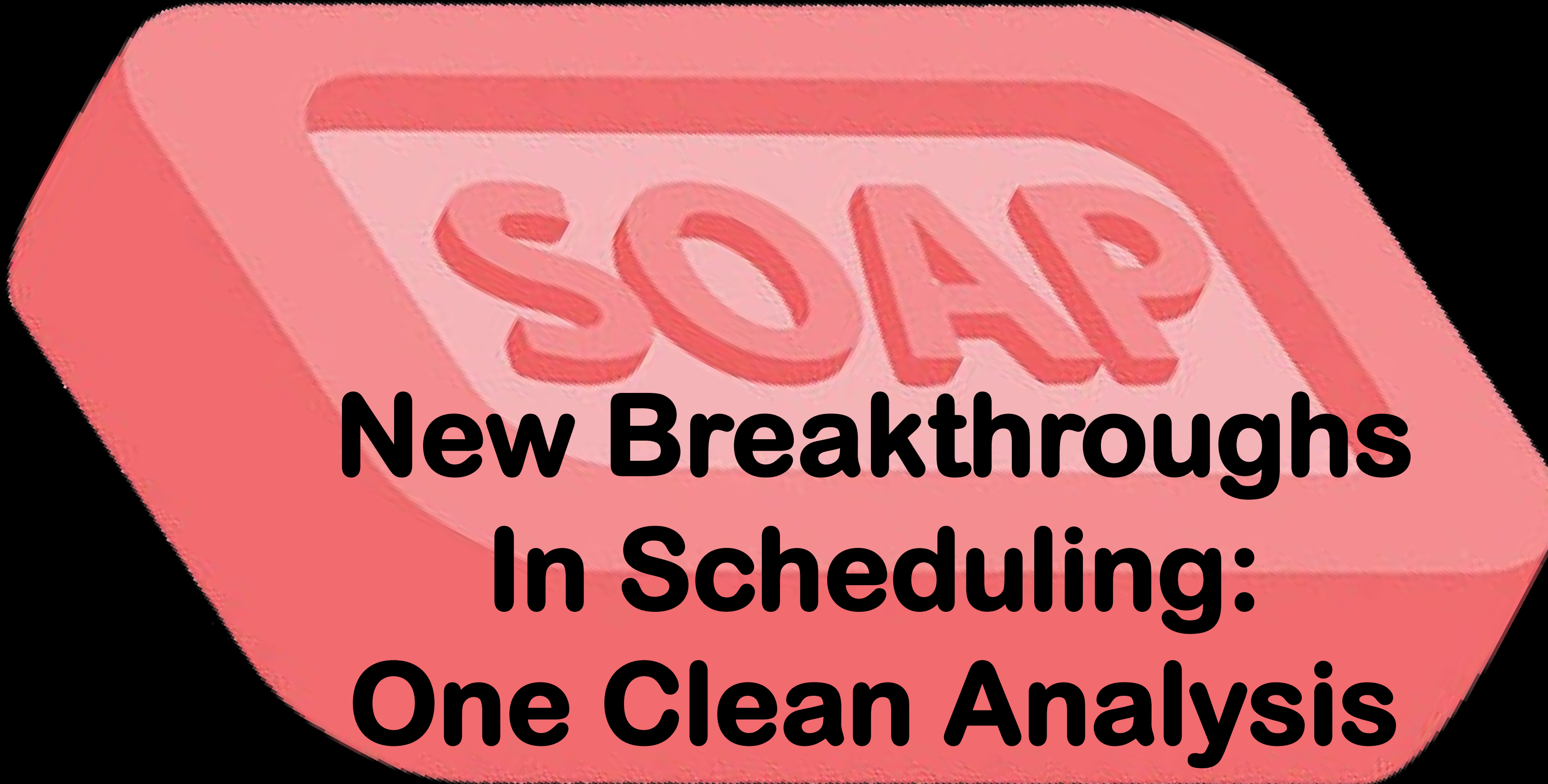


New Breakthroughs In Scheduling: One Clean Analysis

Mor Harchol-Balter, CMU

Based on SIGMETRICS '18 SOAP Paper with:

Ziv Scully  INFORMS APS Best Student Paper Finalist

Alan Scheller-Wolf 

SRPT & Fairness

Size Interval Scheduling

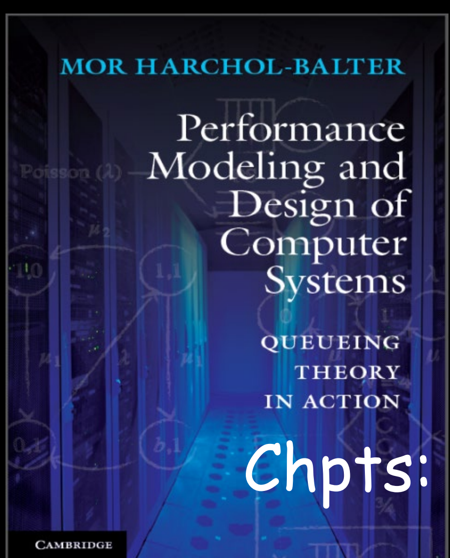
The SMART class

FB for Heavy-tails

AutoScale Power Scheduling

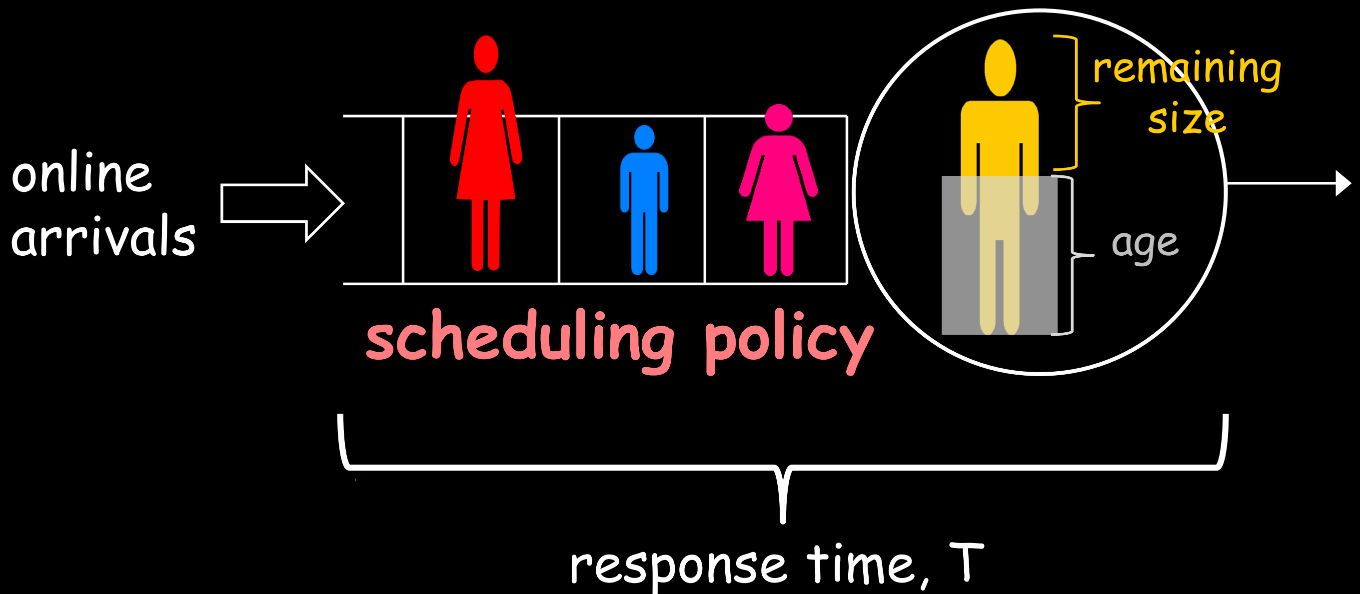
Mor LOVES Scheduling

Scheduling with redundancy



Chpts: 28, 29, 30, 31, 32, 33 … All Scheduling!

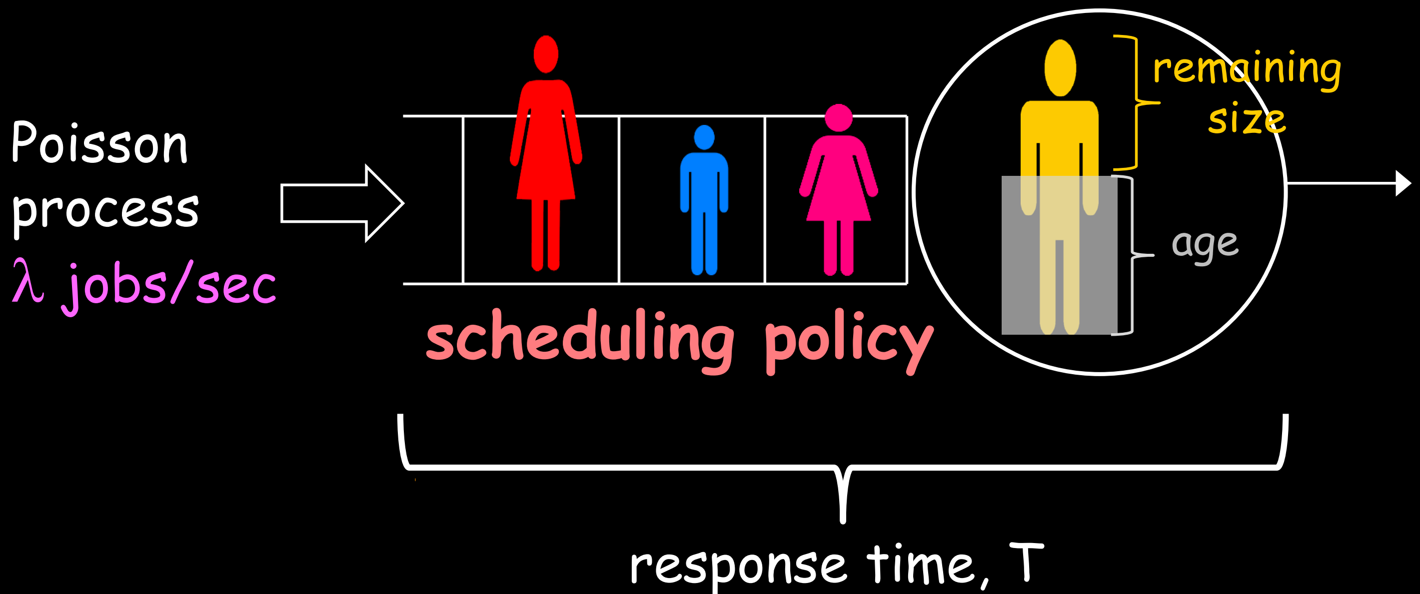
Scheduling Tutorial

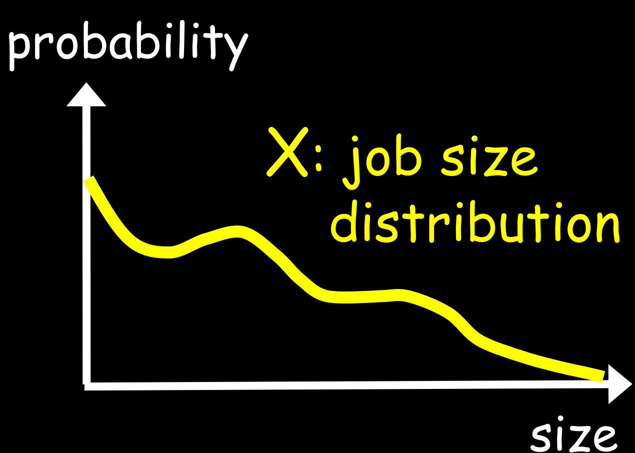


Example:

- CPU run queue
- Router buffer queue
- Disk queue
- Memory bank queue
- Database lock queue

Assume M/G/1

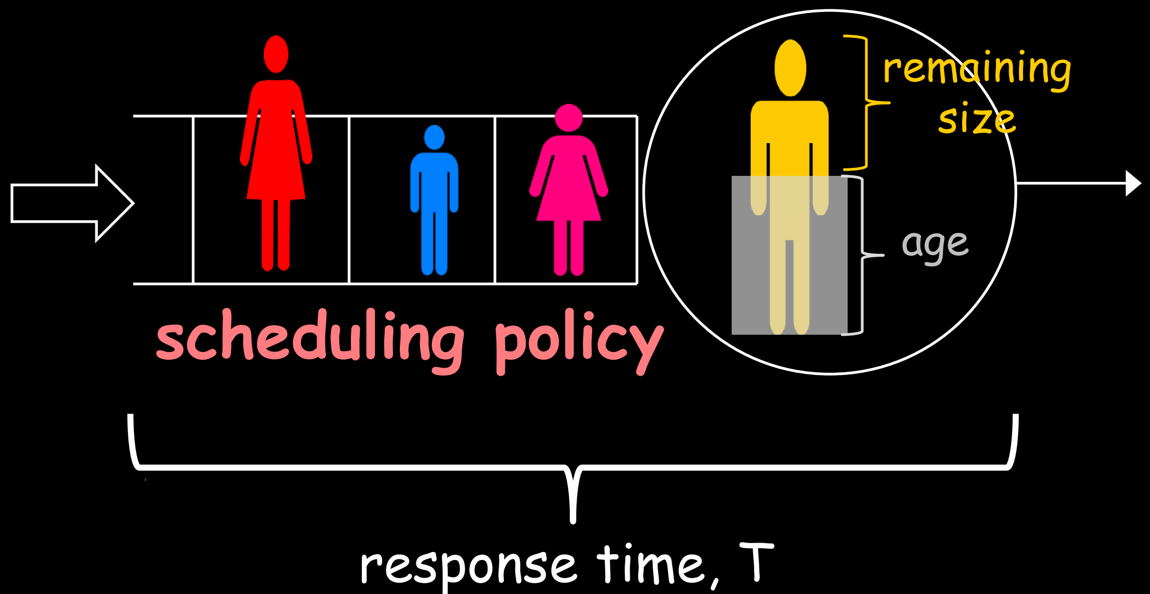




"Load" = fraction time server busy

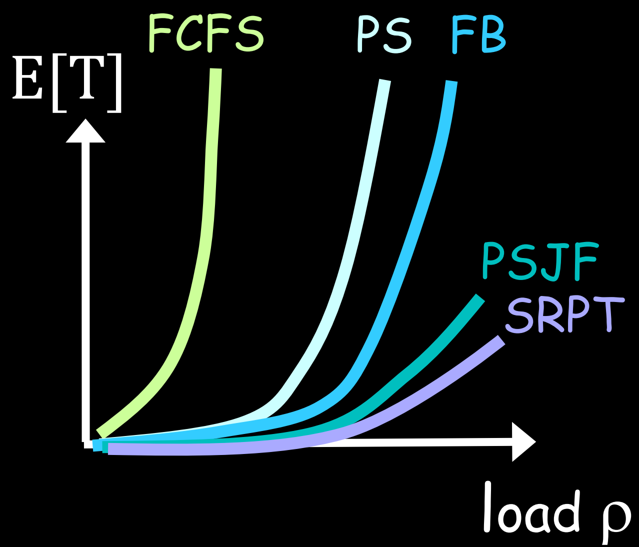
$$\rho = \lambda \cdot E[X] < 1$$

Q: If know job sizes, what scheduling policy minimizes E[T]?



A: SRPT – Shortest Remaining Processing Time [Schrage 1968]

Analysis known since 1966



FCFS = First-Come-First-Served

PS = Processor-Sharing

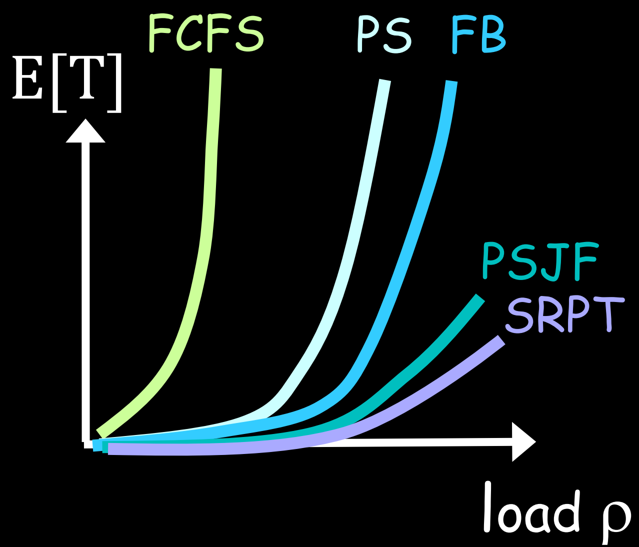
FB = Foreground-Background

PSJF = Preemptive-Shortest-Job-First

SRPT = Shortest-Remaining-Process-Time

(Results shown for high-variability job size distribution)

Analysis known since 1968



FCFS = First-Come-First-Served

PS = Processor-Sharing

FB = Foreground-Background

PSJF = Preemptive-Shortest-Job-First

SRPT = Shortest-Remaining-Process-Time

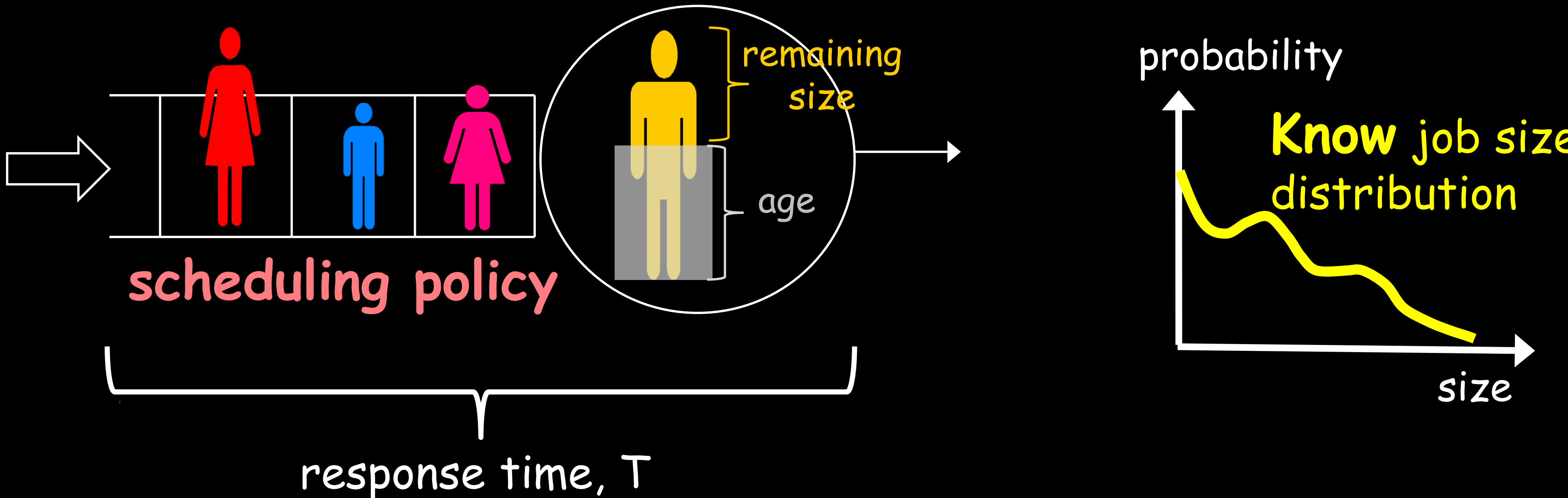
Examples of SRPT in computer systems:

- 2003 [Harchol-Balter et al.] – SYNC project – Web static content
- 2018 [Benham et al.] – HOMA project – Data center message queues

Q: If SRPT is so awesome, why isn't it used all the time?

A: Needs size

Q: But what if don't know job size?

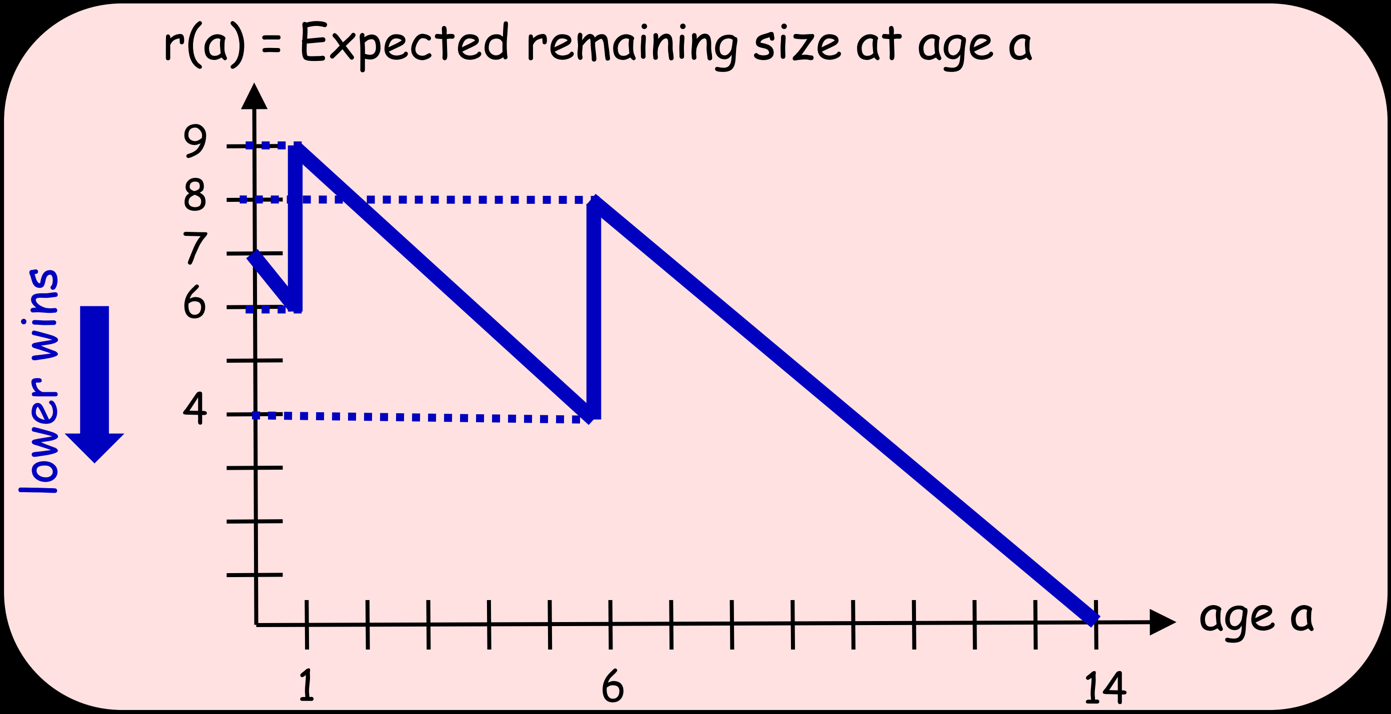


A: SERPT ? -- Shortest Expected Remaining Processing Time

SERPT Example

Always run job with lowest rank

$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$



Q: Is SERPT optimal for E[T], when don't know size?

A: No!

Q: What is?

Optimal policy: Gittins Index

$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

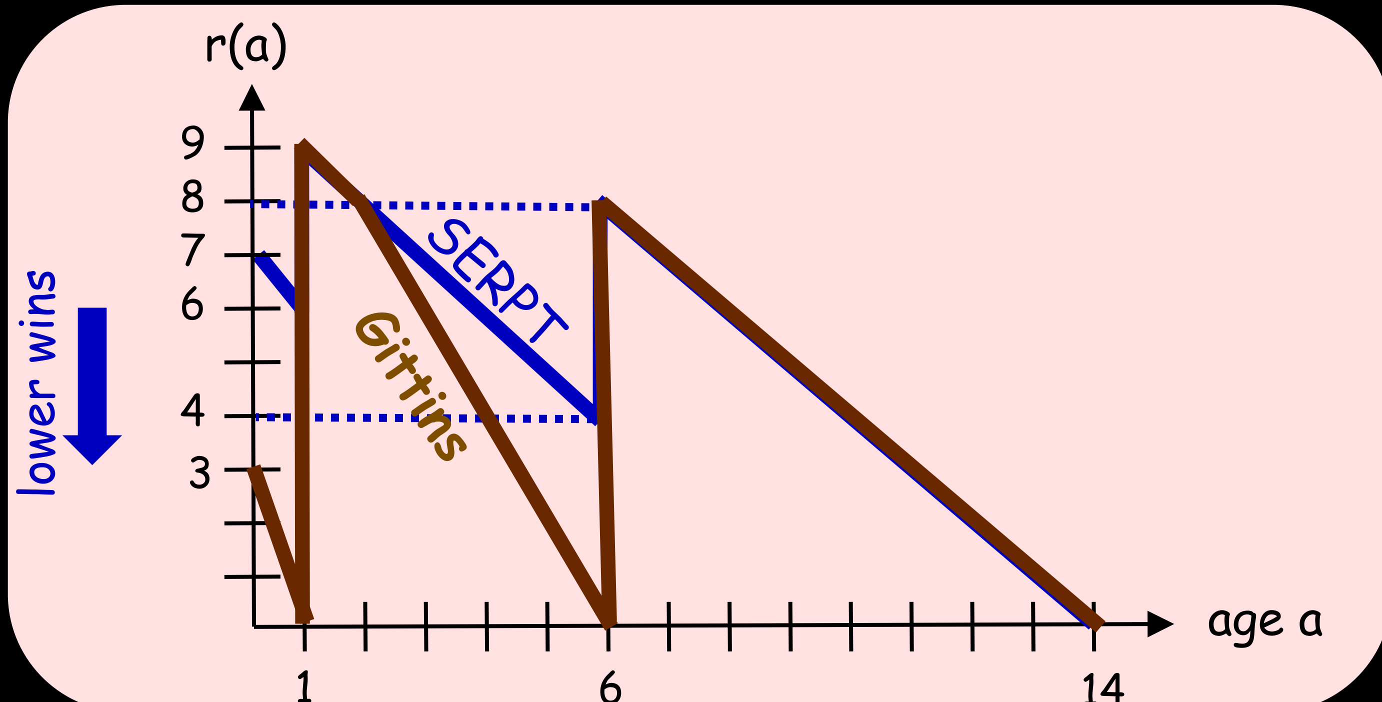
Optimal policy: Gittins Index

Always run job with lowest rank

$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

$$r(a) = \inf_{\Delta} \frac{E[\min\{X - a\,,\ \Delta\} \mid X > a]}{P\{X - a \le \Delta \mid X > a\}}$$

$$r(a) = E[X - a \mid X > a]$$



Q: Gittins policy is optimal for unknown job sizes. But what is its E[T]?

A: E[T] under Gittins **open until 2018.**

E[T] under SERPT **open until 2018.**

E[T] under all policies w/non-monotonic rank fcn **open until 2018.**

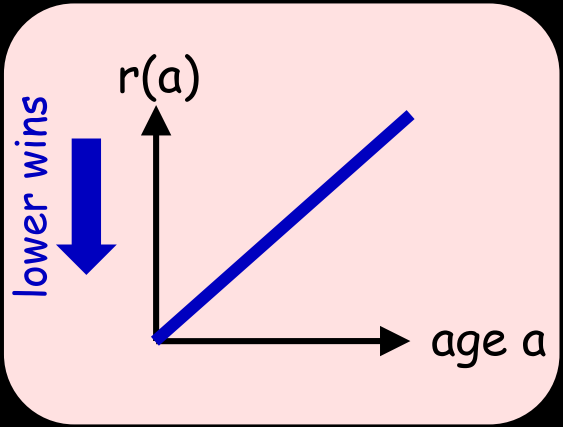
SOAP

SOAP Policies: all policies expressible via a rank function.

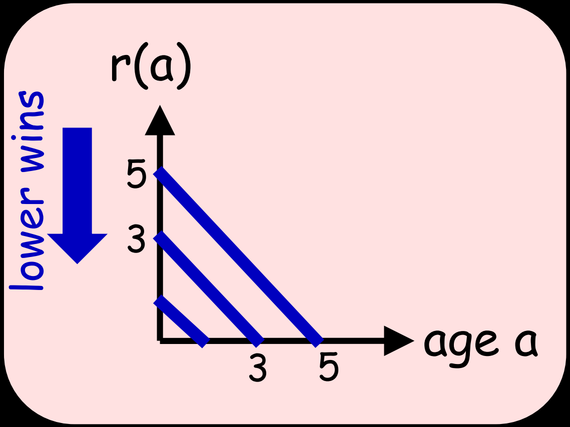
- Rank is a function of age
  (and the job's size or class)
- Always serve job of lowest rank
- FCFS tie-breaking

Examples of known SOAP Policies

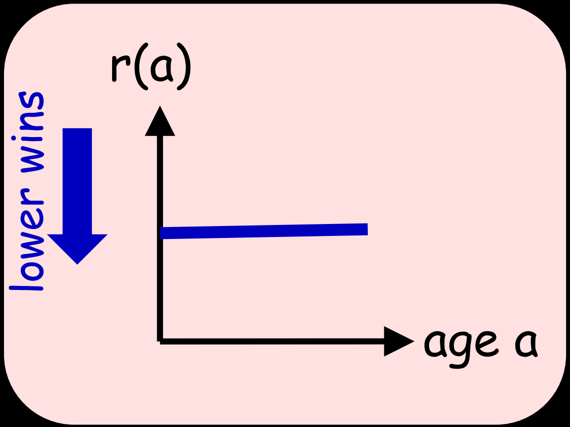
FB



SRPT

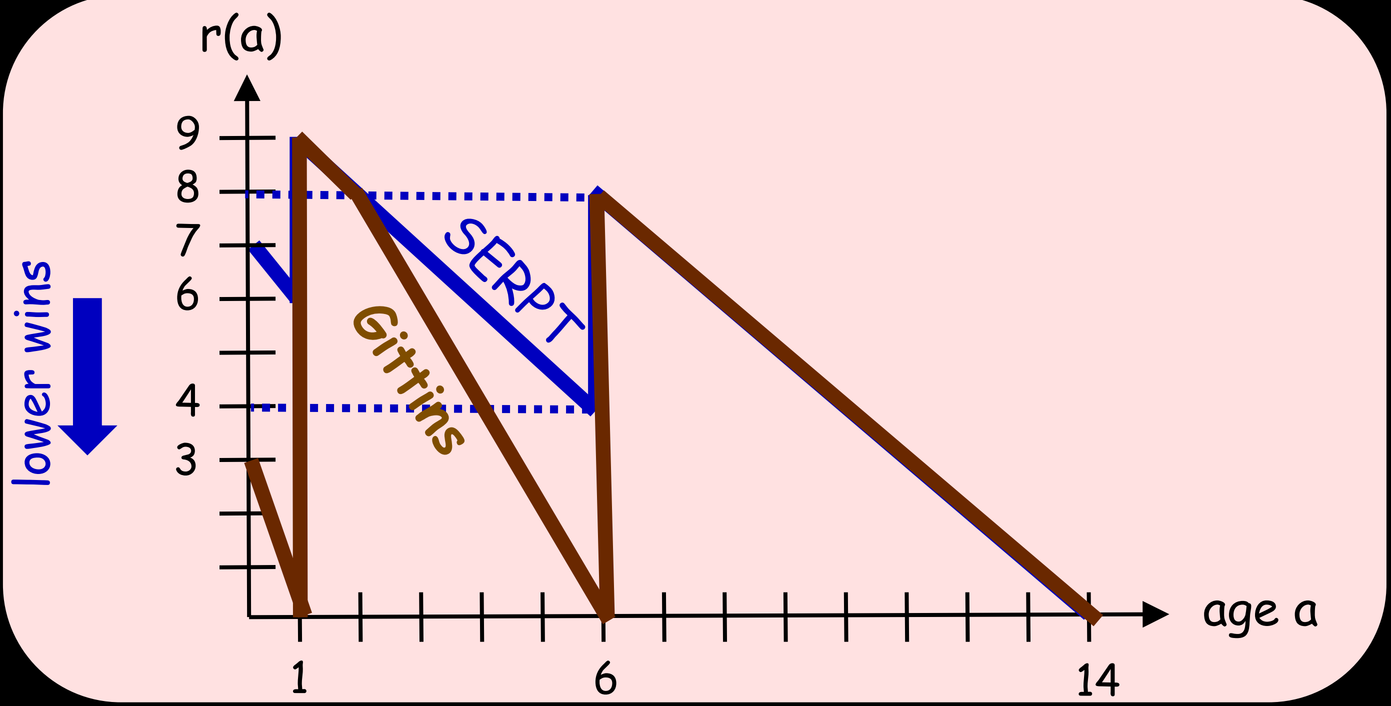


FCFS



SOAP policies with previously UNKNOWN analyses:

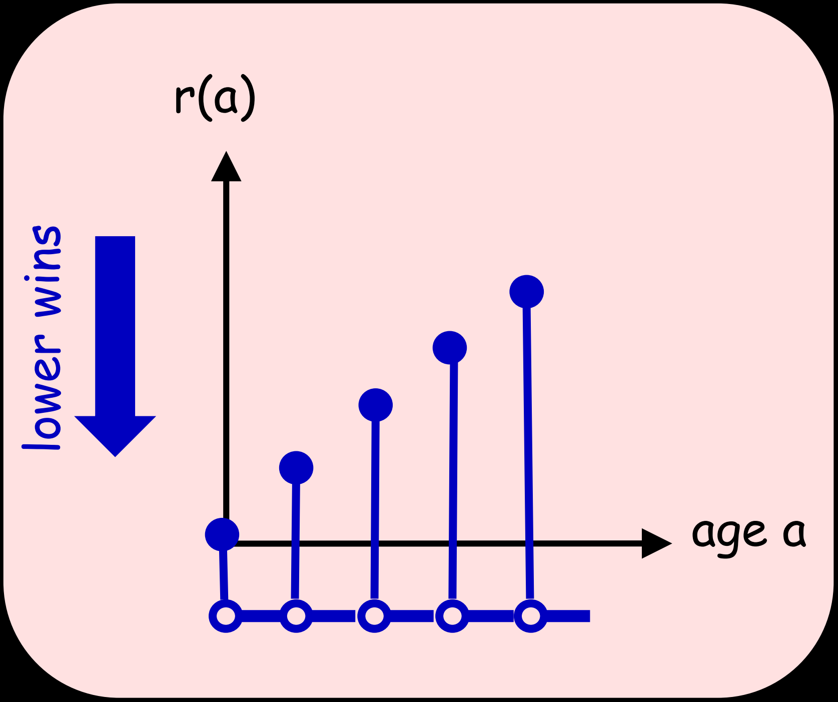
SOAP SERPT SOAP Gittins



rank NOT monotonic

SOAP policies with previously UNKNOWN analyses:

Discretized FB



rank NOT monotonic

SOAP policies with previously UNKNOWN analyses:

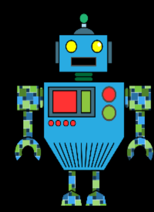
Mixed Classes

Humans (prio 1)

- Non-Preempt
- Unknown Size
- FCFS

vs.

Robots (prio 2)



- Preempt
- Known Size
- SRPT

Twist:
If $remsize(robot) < x_H$ *then robot has priority over un-started human.*

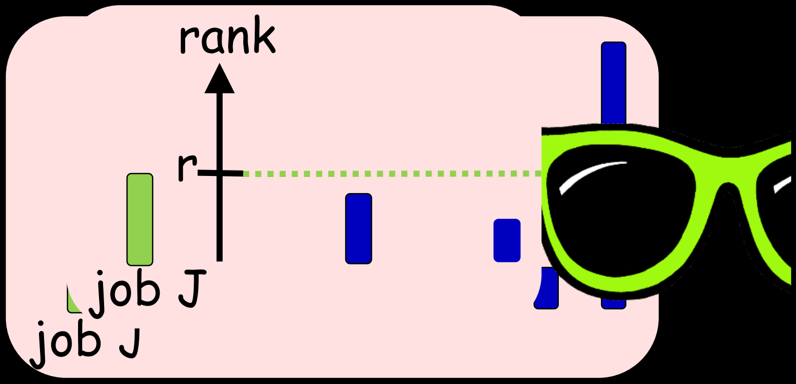
Our Result:

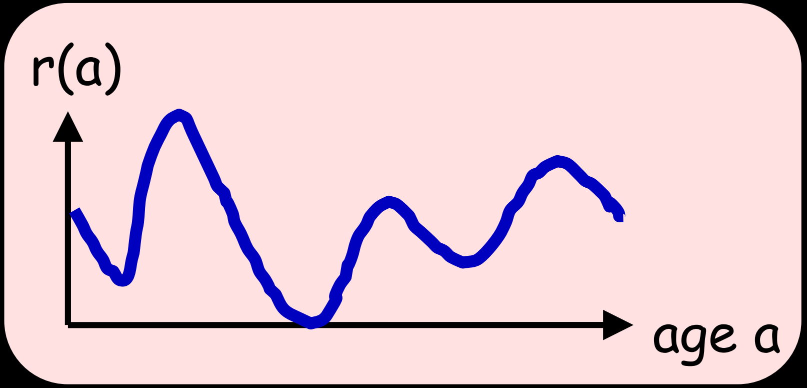
First analysis of all **SOAP** policies.

Given: any rank function	→	Closed-form expression of response time transform

Non-Monotonicity makes analysis hard

"Relevant Jobs"





- My rank goes up & down as I age!
- Everyone else's ranks go up & down too!

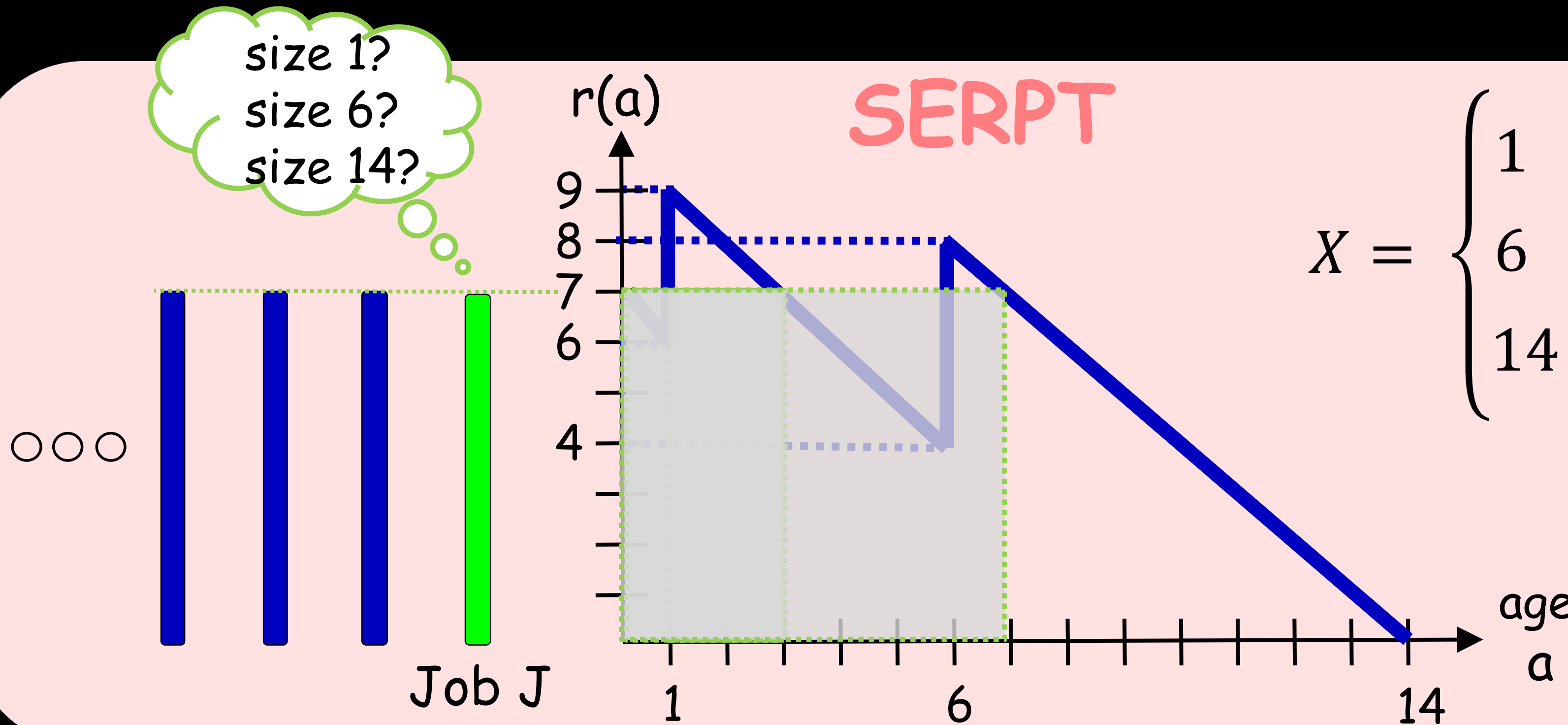
Remainder of talk:

Show how to derive mean response time for **SERPT**.

In general, can handle much more complex examples.

SOAP Warmup: J arrives to EMPTY system

Tagged job



$$X = \begin{cases} 1 & w.p.\ \frac{1}{3} \\ 6 & w.p.\ \frac{1}{3} \\ 14 & w.p.\ \frac{1}{3} \end{cases}$$

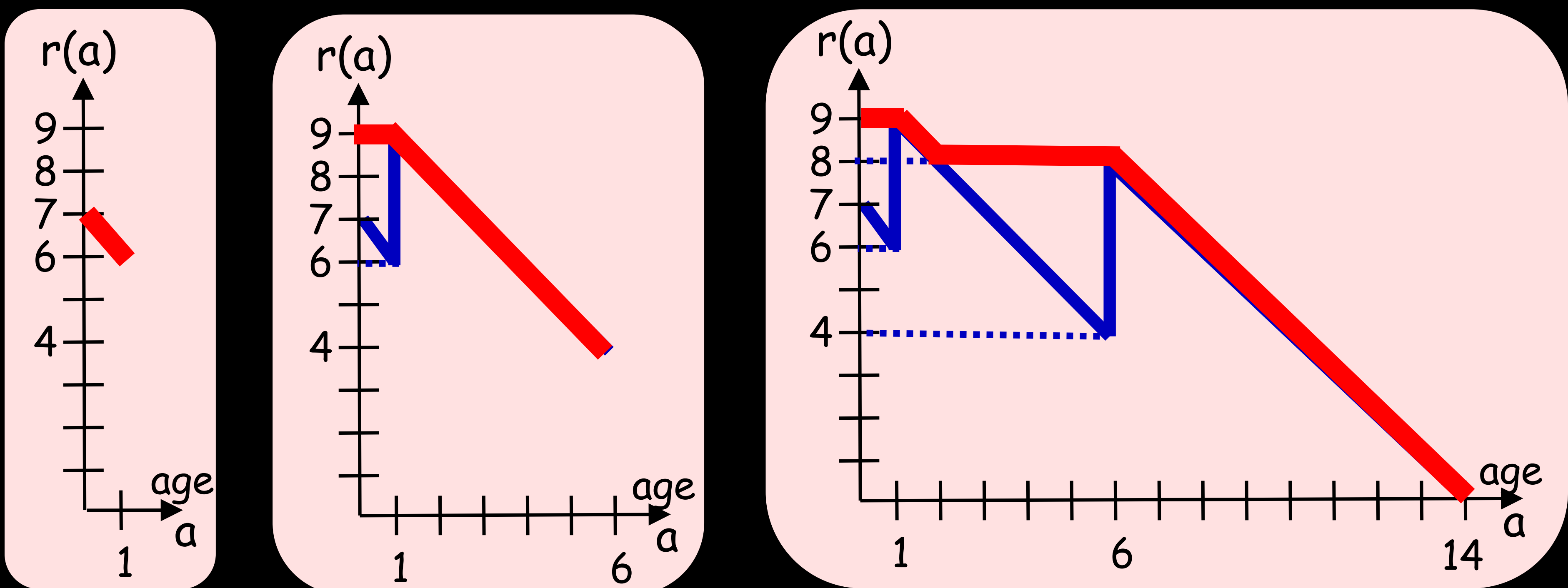
	If size(J)=1	If size(J)=6	If size(J)=14
Which arrivals contribute to J's response time?	No one	Arrivals when 0 < age(J) < 3	Arrivals when 0 < age(J) < 7
How much contribution?	∅	1	1

SOAP Warmup: J arrives to EMPTY system

Tagged job

Key Idea: PESSIMISM PRINCIPLE

"At every age, use J's Worst Future Rank"



size(J)=1 size(J)=6 size(J)=14

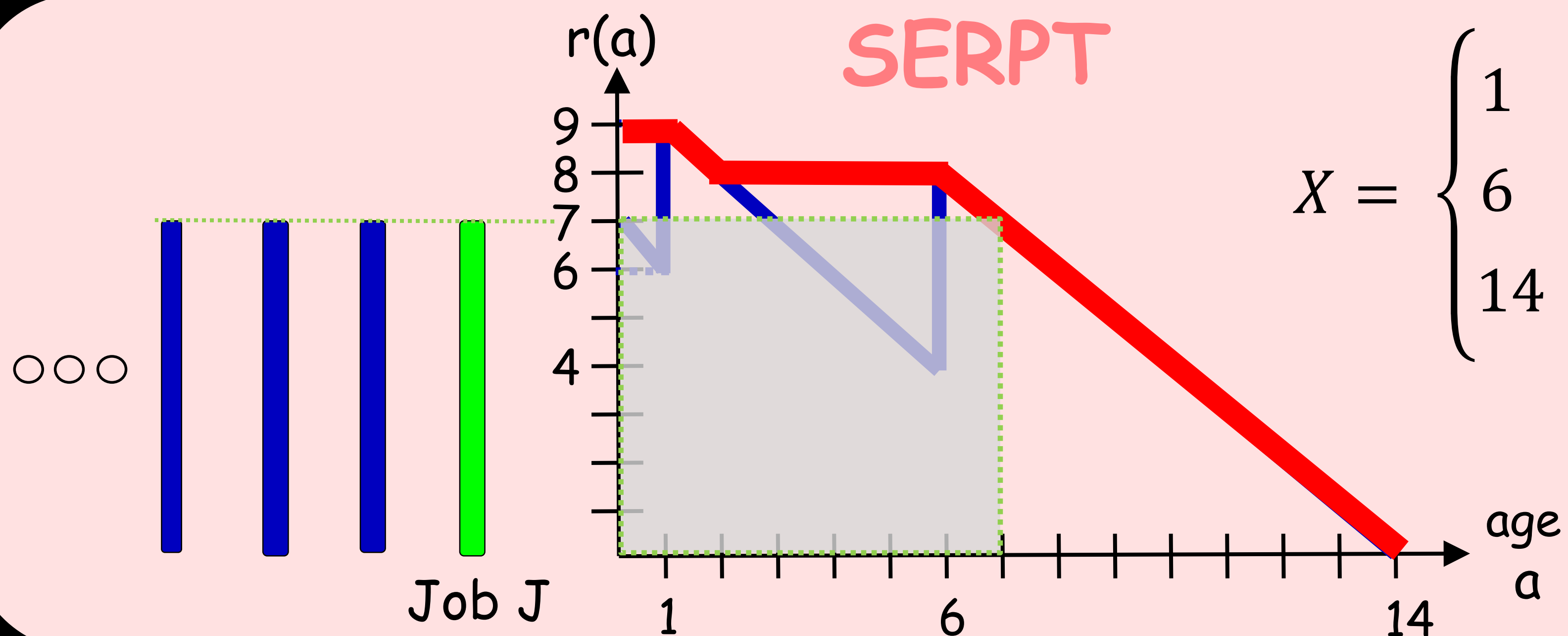
SOAP Warmup: J arrives to EMPTY system

Tagged job

Suppose size(J)=14

$$E[T(J)] = \int_{a=0}^{a=14} \frac{da}{1-\rho(a)}$$

$$\rho(a) = \begin{cases} \lambda \cdot 1 \text{ if } 0 < a < 7 \\ \lambda \cdot 0 \text{ if } 7 < a < 14 \end{cases}$$

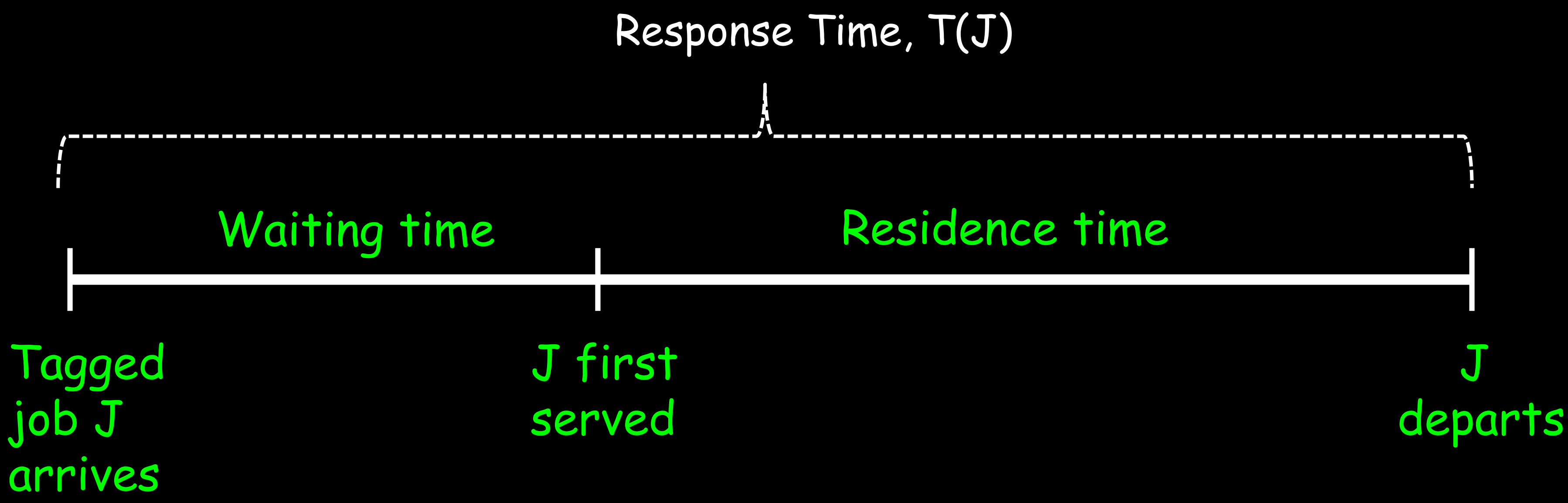


$$X = \begin{cases} 1 & w.p. \ \frac{1}{3} \\ 6 & w.p. \ \frac{1}{3} \\ 14 & w.p. \ \frac{1}{3} \end{cases}$$

Everyone gets in, with contribution 1

No one gets in

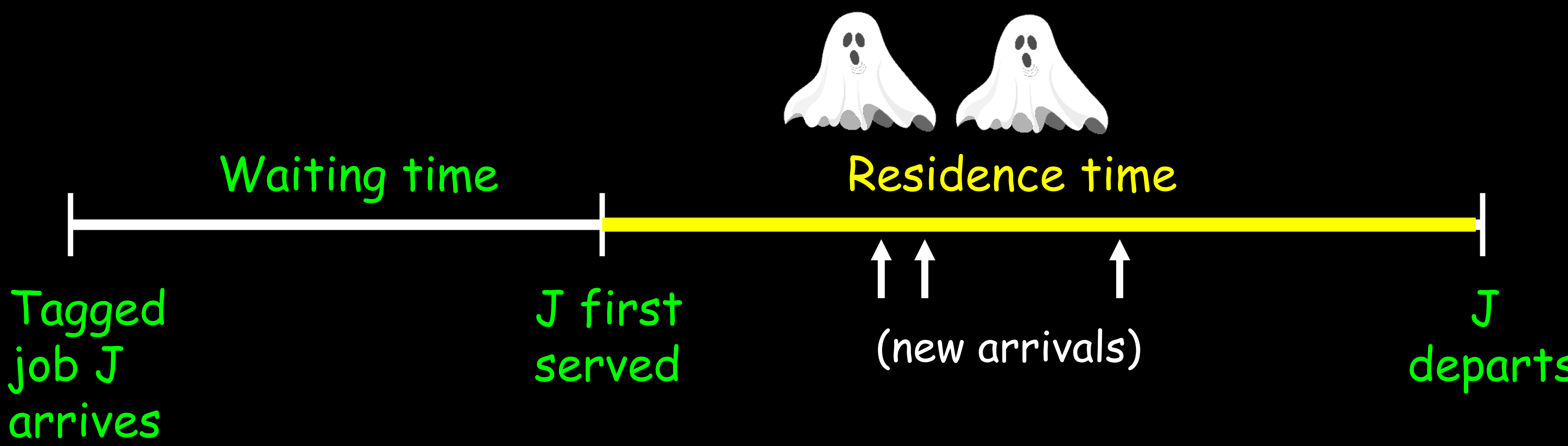
SOAP Analysis



Residence time Analysis

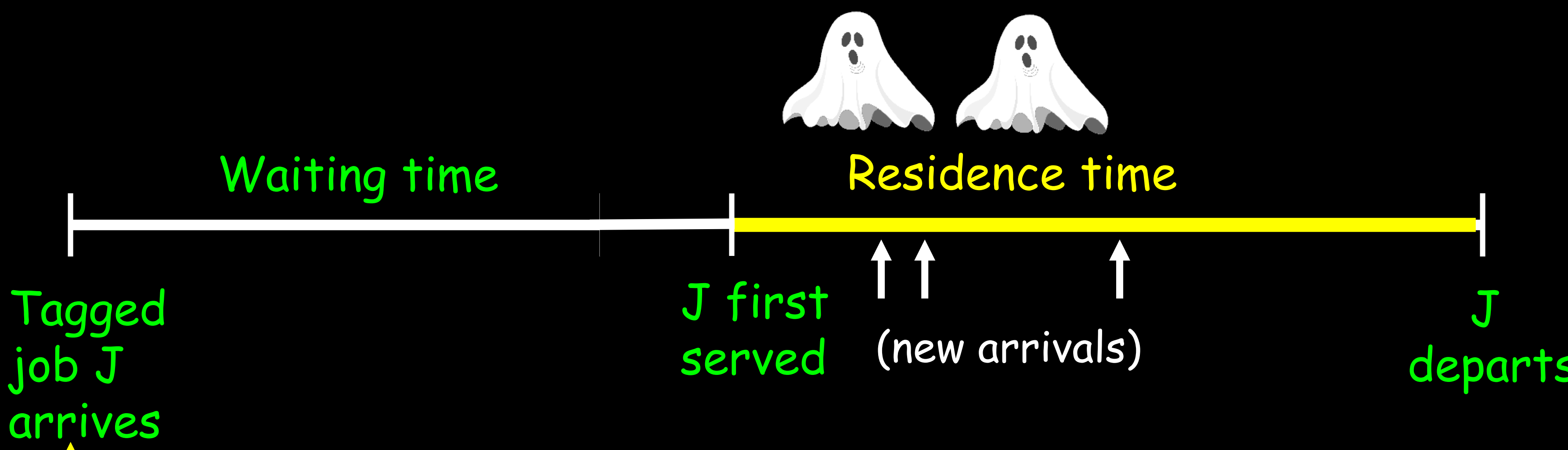
Q: Is residence time just J's service time?

Residence time Analysis



Q: Is residence time equivalent to J arriving into system with 0 work?

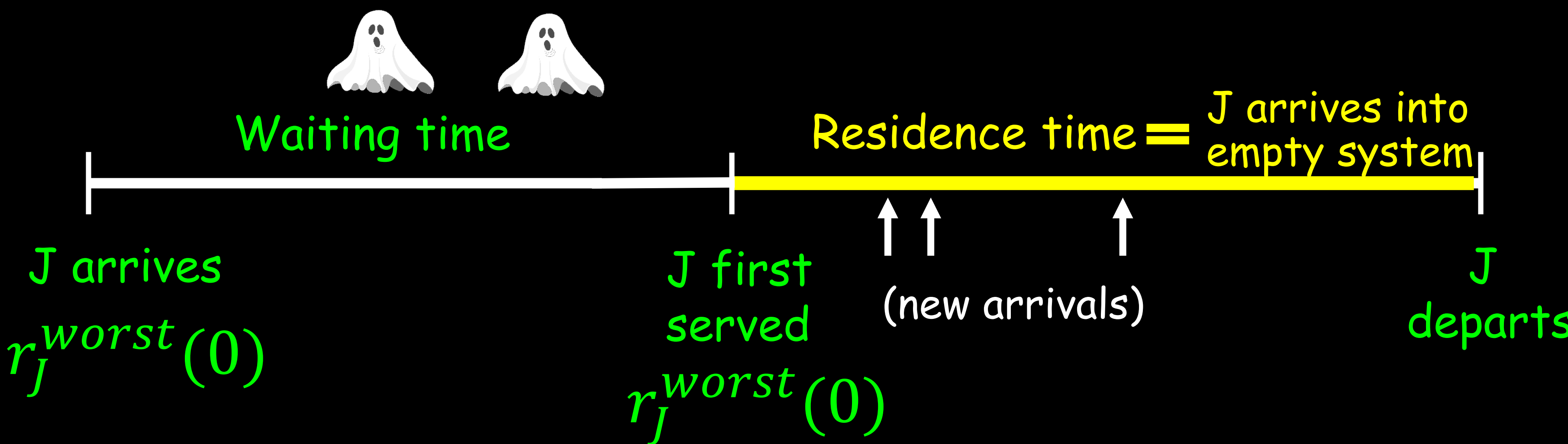
Residence time Analysis



set $rank(J) = r_J^{worst}(0)$

SOLUTION: Pessimism Principle Again!
"Always assume J has its worst future rank"

Residence time Analysis



This trick does NOT change J's total response time!

Waiting time Analysis

Waiting time

J arrives

$r_J^{worst}(0)$

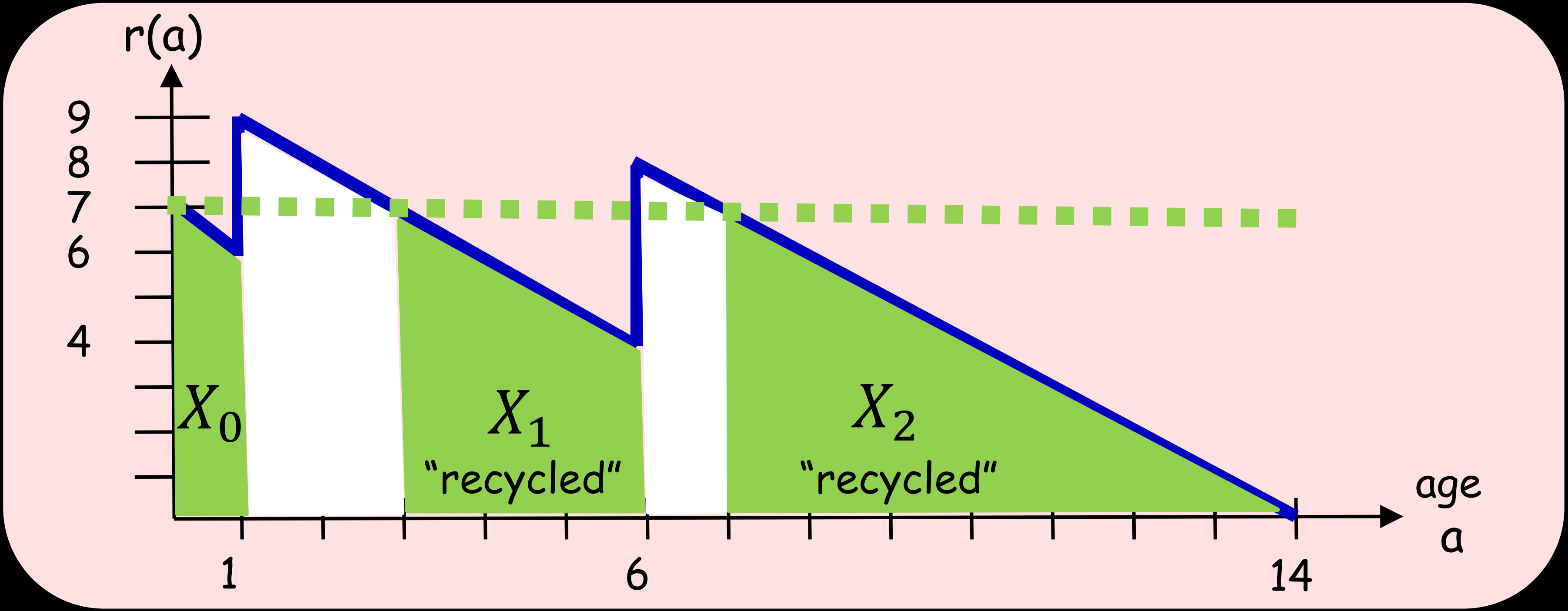
J sees "relevant work" W^{rel}

J first served

$r_J^{worst}(0)$

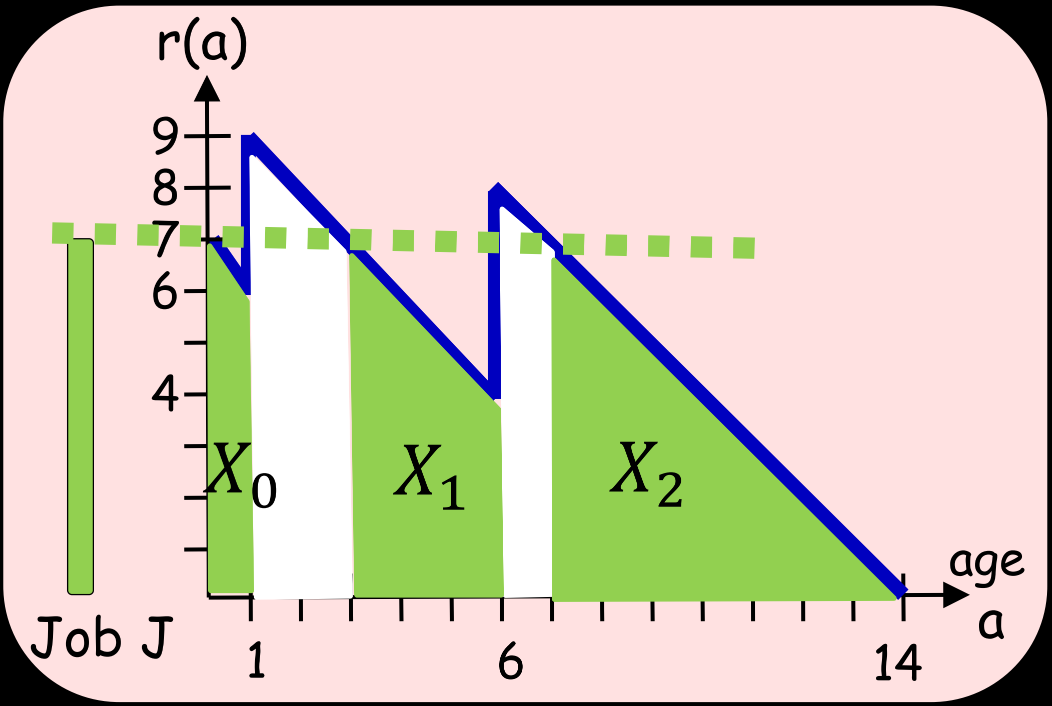
Waiting time is busy period started by W^{rel}

Understanding W^{rel}... wrt rank 7



Q: How do we add up all these X_i parts?

What is W^{rel}: relevant work seen by tagged job J



Vacation Transformation

- There may be many X_0's, but at most one "recycled job" at a time.
- If ∃ recycled job, it must start relevant busy period.
- $W^{rel} = \mathrm{W}^{\mathrm{M/G/1/Vacation}}$

X_0 $\quad X_1, X_2, \ldots$

Final Formula (Exact)

Waiting time | Residence time

J arrives — J first served — J departs

$r \equiv r_J^{worst}(0)$

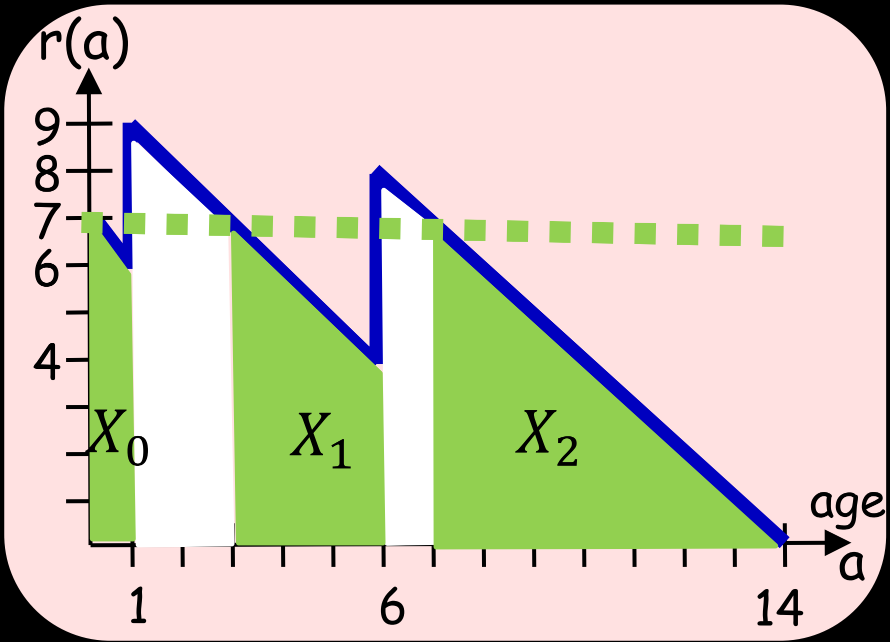
$r(a) \equiv r_J^{worst}(a)$

$$E[T(J)] = \begin{matrix} Relevant\ busy\ period \\ started\ by\ W^{rel}\ work \end{matrix} + \begin{matrix} Time\ if\ J\ sees \\ empty\ system \end{matrix}$$

$$= \frac{1}{1-\rho^{new}[r]} \cdot \frac{\lambda \sum_{i=0}^{\infty} E[X_i[r]^2]}{2(1-\lambda E[X_0][r])} + \int_0^{size(J)} \frac{da}{1-\rho^{new}[r(a)]}$$

1 minute analysis of SERPT | size(J) = 1

$$E[T(J)] = \frac{1}{1-\rho^{new}[r]} \cdot \frac{\lambda \sum_{i=0}^{\infty} E[X_i[r]^2]}{2(1-\lambda E[X_0][r])} + \int_0^{size(J)} \frac{da}{1-\rho^{new}[r(a)]}$$



$$r \equiv r_J^{worst}(0) = 7$$

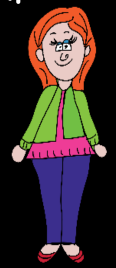
Generalizations

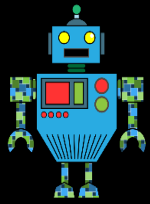
1. Can have multiple classes, each with its own rank function.

2. Can have multi-dimensional rank functions, w/ lexicographic ordering.

Example:

Humans (prio 1)		Robots (prio 2)
○ Non-Preempt ○ Unknown Size ○ FCFS	vs.	○ Preempt ○ Known Size ○ SRPT



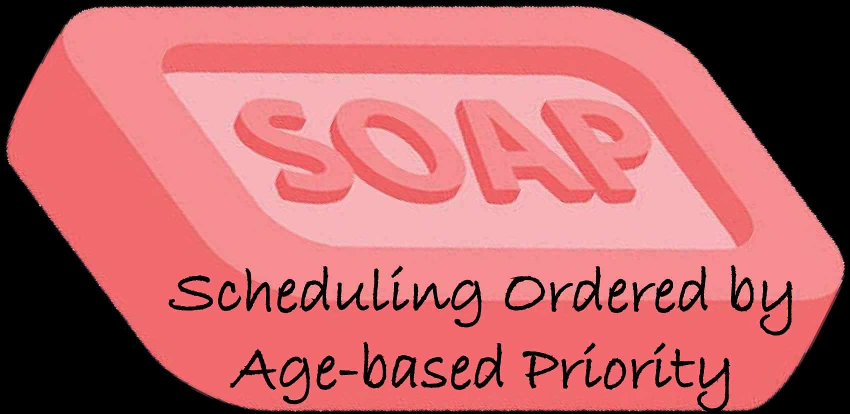


Twist:
If $remsize(robot) < x_H$ *then robot has priority over un-started human.*

$$r_{Human}(a) = (-a\,,\ x_H\,)$$

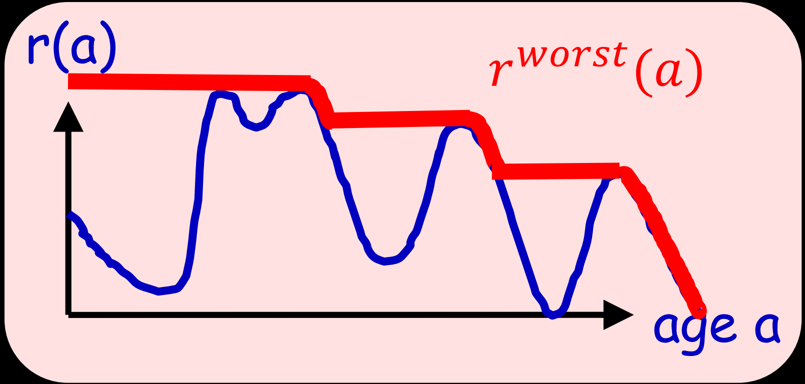
$$r_{Robot(x)}(a) = (0\,,\ x - a)$$

Conclusion

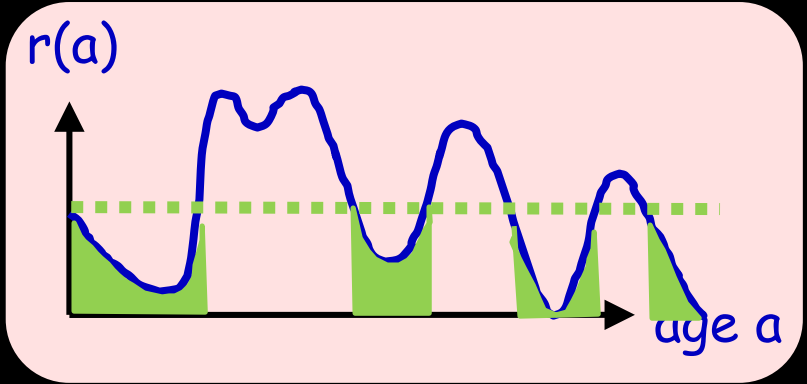


- SERPT
- Gittins Index
- Discretized Policies
- Mixed Classes

Pessimism Principle



Vacation Transformation

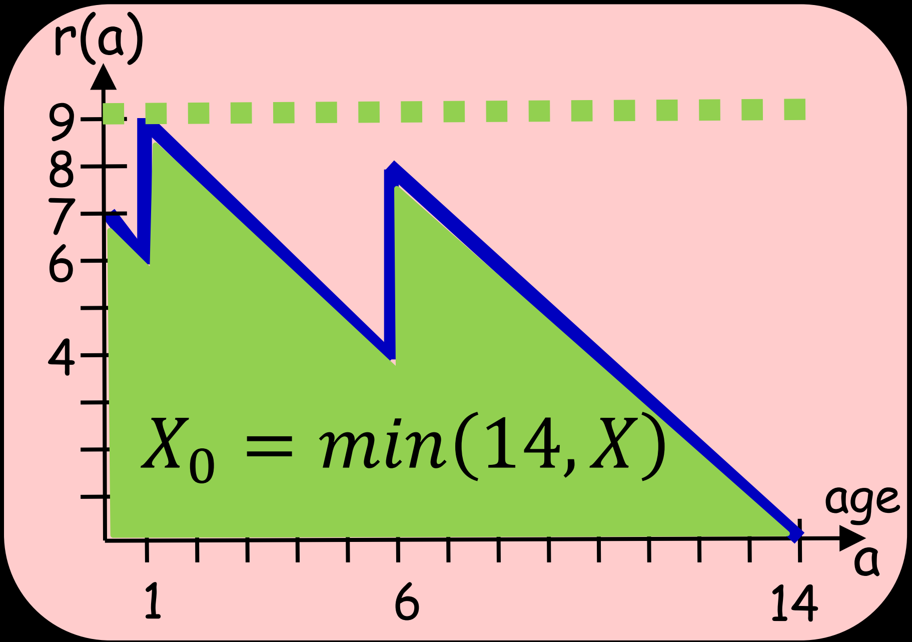


BACKUP

1 minute analysis of SERPT, $E[T(\mathbf{14})]$

$$E[T(J)] = \frac{1}{1-\rho^{new}[r]} \cdot \frac{\lambda \sum_{i=0}^{\infty} E[X_i[r]^2]}{2(1-\lambda E[X_0][r])} + \int_0^{size(J)} \frac{da}{1-\rho^{new}[r(a)]}$$

$$E[T(14)] = \frac{1}{1-\lambda\cdot 1} \cdot \frac{\lambda(1^2+6^2+14^2)\cdot\frac{1}{3}}{2\left(1-\lambda\cdot(1+6+14)\cdot\frac{1}{3}\right)} + \int_0^7 \frac{da}{1-\lambda\cdot 1} + \int_7^{14} \frac{da}{1-\lambda\cdot 0}$$



$$r \equiv r_J^{worst}(0) = 9$$