The Power of SOAP Scheduling

Mor Harchol-Balter
Ziv Scully
Carnegie Mellon University
M/G/1 Queue
M/G/1 Queue

queue

server
M/G/1 Queue
M/G/1 Queue

queue

server

job

size
M/G/1 Queue

queue

server

job

size
M/G/1 Queue
M/G/1 Queue

queue

server

job

size
M/G/1 Queue

random arrivals

size

remaining size

age

queue

server

job
M/G/1 Queue

- Random arrivals
- Queue
- Server
- Job

- Size
- Remaining size
- Age
M/G/1 Queue

\[ X = \text{size distribution} \]
\[ \lambda = \text{arrival rate} \]
\[ \rho = \lambda E[X] < 1 \]
M/G/1 Queue

$X = \text{size distribution}$

$\lambda = \text{arrival rate}$

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

Scheduling policy: picks which job to serve
M/G/1 Queue

$X = \text{size distribution}$

$\lambda = \text{arrival rate}$

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

random arrivals

queue

Scheduling policy: picks which job to serve

size

remaining size

age

server

job
M/G/1 Queue

\[ X = \text{size distribution} \]
\[ \lambda = \text{arrival rate} \]
\[ \rho = \lambda E[X] < 1 \]

Scheduling policy: picks which job to serve
$X = \text{size distribution}$

$\lambda = \text{arrival rate}$

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

**Scheduling policy:** picks which job to serve
M/G/1 Queue

- $X = \text{size distribution}$
- $\lambda = \text{arrival rate}$
- $\rho = \lambda E[X] < 1$

Scheduling policy:
- picks which job to serve

Random arrivals

Queue

Size
Age

Remaining size

Server

Job
Response Time
Response Time
Response Time

\[ T = \text{response time} \]
Response Time

\[ T = \text{response time} \]

Goal: analyze mean response time \( \mathbb{E}[T] \)
Response Time

Goal: analyze mean response time $E[T]$

Depends on scheduling policy
Impact of Scheduling

What scheduling policy minimizes $E[T]$?
Impact of Scheduling

What scheduling policy minimizes $E[T]$?

Shortest remaining processing time (SRPT)
Impact of Scheduling

What scheduling policy minimizes $E[T]$?

Shortest remaining processing time (SRPT)

![Graph showing the impact of different scheduling policies on $E[T]$](graph.png)
Impact of Scheduling

What scheduling policy minimizes $E[T]$?

Shortest remaining processing time (SRPT)

... but nobody uses SRPT!
Why Not SRPT?
Why Not SRPT?

Unknown job sizes
Why Not SRPT?

- FCFS (first come, first served)
- Unknown job sizes
Why Not SRPT?

Unknown job sizes

\[
\begin{align*}
\text{FCFS} & \quad \text{(first come, first served)} \\
\text{FB} & \quad \text{(foreground-background: least age)}
\end{align*}
\]
Why Not SRPT?

Unknown job sizes

- **FCFS** (first come, first served)
- **FB** (foreground-background: least age)
- **SERPT** (least expected remaining size)
Why Not SRPT?

Unknown job sizes

\[
\begin{align*}
\text{FCFS} & \text{ (first come, first served)} \\
\text{FB} & \text{ (foreground-background: least age)} \\
\text{SERPT} & \text{ (least } expected \text{ remaining size)} \\
\text{Gittins} & \text{ (optimal!)}
\end{align*}
\]
Why Not SRPT?

Unknown job sizes

FCFS (first come, first served)
FB (foreground-background: least age)
SERPT (least expected remaining size)
Gittins (optimal!)

Hardware constraints
Why Not SRPT?

Unknown job sizes

- **FCFS** (first come, first served)
- **FB** (foreground-background: least age)
- **SERPT** (least expected remaining size)
- **Gittins** (optimal!)

Hardware constraints

- “Discrete” SRPT
  (preempt only at checkpoints)
Why Not SRPT?

Unknown job sizes

- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least expected remaining size)
- Gittins (optimal!)

Hardware constraints

- “Discrete” SRPT (preempt only at checkpoints)
- “Bucketed” SRPT (limited number of priority levels)
Why Not SRPT?

Unknown job sizes

- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least expected remaining size)
- Gittins (optimal!)

Hardware constraints

- “Discrete” SRPT, FB, etc. (preempt only at checkpoints)
- “Bucketed” SRPT, FB, etc. (limited number of priority levels)
Why Not SRPT?

Unknown job sizes

- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least expected remaining size)
- Gittins (optimal!)

Hardware constraints

- “Discrete” SRPT, FB, etc.
  (preempt only at checkpoints)
- “Bucketed” SRPT, FB, etc.
  (limited number of priority levels)

Metric other than $E[T]$
Why Not SRPT?

Unknown job sizes

- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least expected remaining size)
- Gittins (optimal!)

Hardware constraints

- “Discrete” SRPT, FB, etc.
  (preempt only at checkpoints)
- “Bucketed” SRPT, FB, etc.
  (limited number of priority levels)

Metric other than $E[T]$

- Priority classes
Why Not SRPT?

Unknown job sizes
- FCFS (first come, first served)
- FB (foreground-background: least age)
- SERPT (least expected remaining size)
- Gittins (optimal!)

Hardware constraints
- “Discrete” SRPT, FB, etc.
  (preempt only at checkpoints)
- “Bucketed” SRPT, FB, etc.
  (limited number of priority levels)

Metric other than $E[T]$
- Priority classes
- RS (optimal for mean slowdown)
Many Scheduling Policies
Many Scheduling Policies

$E[T]$ known
Many Scheduling Policies

\[
E[T] \text{ known}
\]

SRPT
Many Scheduling Policies

\[ E[T] \text{ known} \]

SRPT
FCFS
Many Scheduling Policies

$E[T]$ known

SRPT
FCFS
FB
Many Scheduling Policies

\[ E[T] \text{ known} \]

SRPT
FCFS
FB
Simple priority classes
Many Scheduling Policies

E[T] known

SRPT
FCFS
FB
Simple priority classes

E[T] unknown!
Many Scheduling Policies

- **E[T] known**
  - SRPT
  - FCFS
  - FB
  - Simple priority classes

- **E[T] unknown!**
  - SERPT
  - Gittins
  - Discrete SRPT
  - Discrete FB
  - Bucketed SRPT
  - Bucketed FB
  - RS*
  - Complex priority classes
  - ... and more!
Many Scheduling Policies

**E[\(T\)] known**
- SRPT
- FCFS
- FB
- Simple priority classes

**E[\(T\)] unknown!**
- SERPT
- Gittins
- Discrete SRPT
- Discrete FB
- Bucketed SRPT
- Bucketed FB
- RS*
- Complex priority classes
  - ... and more!
SOAP

Broad class of scheduling policies…
SOAP

Broad class of scheduling policies…

… with universal response time analysis
SOAP
Schedule Ordered by Age-based Priority

Broad class of scheduling policies…

… with *universal* response time analysis
Outline
Outline

Part 1: defining **SOAP** policies
Outline

Part 1: defining **SOAP** policies

Part 2: analyzing **SOAP** policies
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP
Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Part 1: defining SOAP policies
Scheduling with Ranks
Scheduling with Ranks

FB
serve by least age
Scheduling with Ranks

FB
serve by least age
Scheduling with Ranks

FB
serve by least age

SRPT
serve by least remaining size
Scheduling with **Ranks**

**FB**
serve by least age

**SRPT**
serve by least remaining size
Scheduling with **Ranks**

**FB**
serve by least age

**SRPT**
serve by least remaining size

Common theme: a job’s rank (priority) depends on its age
Scheduling with **Ranks**

- **FB**
  - serve by least age

- **SRPT**
  - serve by least remaining size

**Common theme**: a job’s **rank** (priority) depends on its **age**

(lower is better)
Scheduling with **Ranks**

**FB**
serve by least age

**SRPT**
serve by least remaining size

*Common theme*: a job’s rank (priority) depends on its age

**rank**

lower is better

**age**

**rank**

**age**
Scheduling with **Ranks**

**FB**
serve by least age

**SRPT**
serve by least remaining size

**Common theme:** a job’s **rank** (priority) depends on its **age**
A **SOAP** policy is a **rank** function with one rule:
A SOAP policy is a rank function with one rule:

always serve the job of minimum rank
A *SOAP* policy is a *rank* function with one rule:

always serve the job of *minimum rank*  
(break ties FCFS)
Classic SOAP Policies

FB

SRPT

rank

age

rank

large medium small

age
Classic SOAP Policies

FB

SRPT

FCFS
Classic SOAP Policies

**FB**

```
rank --> age
```

**SRPT**

```
rank

medium
small
large

age
```

**FCFS**

```
rank --> age
```

**Preemptive Priority**

```
rank

normal
urgent

age
```
Classic SOAP Policies

- FB
  - rank
  - age
  - E\[T\] known
  - normal
  - urgent

- SRPT
  - rank
  - large
  - medium
  - age

Preemptive Priority
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3}
\end{cases} \]
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
SOAP Policy: SERPT

Job size distribution:

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

Rank nonmonotonic in age
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
& \text{w.p. } \frac{1}{3}
\end{cases} \]

\( E[T] \) unknown!

Rank nonmonotonic in age
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
\end{cases} \]

\( E[T] \) unknown!

Rank nonmonotonic in age
SOAP Policy: SERPT

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
SOAP Policy: Gittins

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
SOAP Policy: Gittins

Job size distribution:

\[ X = \begin{cases} 
1 & \text{w.p.} \frac{1}{3} \\
6 & \text{w.p.} \frac{1}{3} \end{cases} \]

\[ \mathbb{E}[T] \text{ unknown!} \]
SOAP Policy: Discrete FB
SOAP Policy: Discrete FB

FB, but preempt only at age checkpoints
SOAP Policy: Discrete FB

FB, but preempt only at age checkpoints
SOAP Policy: Discrete FB

FB, but preempt only at age checkpoints
SOAP Policy: Discrete FB

FB, but preempt only at age checkpoints

\[E[T] \text{ unknown!} \]
SOAP Policy: Bucketed SRPT
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:
• Small: [0, 2), rank = 1
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

• Small: $[0, 2)$, \textbf{rank} $= 1$
• Medium: $[2, 7)$, \textbf{rank} $= 2$
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: \([0, 2)\), \(\text{rank} = 1\)
- Medium: \([2, 7)\), \(\text{rank} = 2\)
- Large: \([7, \infty)\), \(\text{rank} = 3\)
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: $[0, 2)$, rank = 1
- Medium: $[2, 7)$, rank = 2
- Large: $[7, \infty)$, rank = 3
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: $[0, 2)$, rank = 1
- Medium: $[2, 7)$, rank = 2
- Large: $[7, \infty)$, rank = 3
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: $[0, 2)$, rank $= 1$
- Medium: $[2, 7)$, rank $= 2$
- Large: $[7, \infty)$, rank $= 3$

2 remaining
7 remaining
SOAPE Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: $[0, 2)$, \textbf{rank} = 1
- Medium: $[2, 7)$, \textbf{rank} = 2
- Large: $[7, \infty)$, \textbf{rank} = 3

\begin{itemize}
  \item 7 remaining
  \item 2 remaining
\end{itemize}
SOAP Policy: Bucketed SRPT

SRPT with three size buckets:

- Small: \([0, 2)\), \(\text{rank} = 1\)
- Medium: \([2, 7)\), \(\text{rank} = 2\)
- Large: \([7, \infty)\), \(\text{rank} = 3\)

\(E[T]\) unknown!
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

**Humans**

- unknown size
- nonpreemptible
- FCFS
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

**Humans**
- unknown size
- nonpreemptible
- FCFS

**Robots**
- known size
- preemptible
- SRPT
Two customer classes: *humans* and *robots*

**Humans**
- unknown size
- nonpreemptible
- FCFS
- priority over robots

**Robots**
- known size
- preemptible
- SRPT
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

**Humans**
- unknown size
- nonpreemptible
- FCFS
- priority over robots

**Robots**
- known size
- preemptible
- SRPT

**Twist**: small robots outrank humans
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

**Humans**
- unknown size
- nonpreemptible
- FCFS
- priority over robots

**Robots**
- known size
- preemptible
- SRPT

**Twist:** small robots outrank humans

\[ \text{size} < x_H \]
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

**Humans**
- unknown size
- **nonpreemptible**
- FCFS
- priority over robots

**Robots**
- known size
- preemptible
- SRPT

**Twist:** small robots outrank humans

size \(< x_H\)
SOAP Policy: Mixture

Two customer classes: *humans* and *robots*

**Humans**
- unknown size
- nonpreemptible
- FCFS
- priority over robots

**Robots**
- known size
- preemptible
- SRPT

$E[T]$ unknown!

$size < x_H$

**Twist:** small robots outrank humans
A **SOAP** policy is any policy expressible by a **rank** function of the form:
Full SOAP Definition

A SOAP policy is any policy expressible by a rank function of the form:

\[ \text{descriptor} \times \text{age} \rightarrow \text{rank} \]
A SOAP policy is any policy expressible by a rank function of the form:

\[ \text{descriptor } \times \text{ age } \rightarrow \text{ rank} \]
Full SOAP Definition

A SOAP policy is any policy expressible by a rank function of the form:

\[ \text{descriptor} \times \text{age} \rightarrow \text{rank} \]

\[ r_{\emptyset}(a) = a \]
A **SOAP policy** is any policy expressible by a **rank function** of the form:

\[
\text{descriptor} \times \text{age} \rightarrow \text{rank}
\]

**FB**

\[
r_\emptyset(a) = a
\]

**SRPT**

\[
r_x(a) = x - a
\]
A **SOAP policy** is any policy expressible by a **rank function** of the form:

$$\text{descriptor} \times \text{age} \rightarrow \text{rank}$$

**Descriptor** can be anything that:
- does not change while a job is in the system
- is i.i.d. for each job

**FB**

$$r_\emptyset(a) = a$$

**SRPT**

$$r_x(a) = x - a$$
FAQ:
What isn’t a SOAP policy?
FAQ:
What isn’t a SOAP policy?

• Rank changes when not in service
FAQ: What isn’t a SOAP policy?

• Rank changes when not in service
• Rank depends on system-wide state
FAQ:
What isn’t a SOAP policy?

• Rank changes when not in service
• Rank depends on system-wide state
• Non-FCFS tiebreaking
FAQ:
What isn’t a SOAP policy?

• **Rank** changes when not in service
• **Rank** depends on system-wide state
• Non-FCFS tiebreaking

**Excludes:** EDF, accumulating priority, PS
Part 1: defining SOAP policies
Part 1: defining SOAP policies
Part 1: *defining* SOAP policies

Part 2: *analyzing* SOAP policies

Part 3: *policy design* with SOAP

Part 4: *optimality proofs* with SOAP
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Part 2: analyzing SOAP policies
Tagged Job Analysis

“me” → random system state
Tagged Job Analysis

"me"

random system state

\[ = \text{rank} \]
Tagged Job Analysis

“me” → random system state

I = rank
Tagged Job Analysis

"me"

random system state

\[ I = \text{rank} \]
Tagged Job Analysis

random system state

“me”

Nomonotonic rank function

rank

rank

age
Tagged Job Analysis

Two obstacles:

- Random system state
- Nonmonotonic rank function

⚠️ Two obstacles:
Tagged Job Analysis

Two obstacles:

• My rank goes up and down

random system state
Two obstacles:

- *My rank* goes up and down
- *Others’ ranks* go up and down too
Running example:

**SERPT**

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Warmup: Empty System

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Warmup: Empty System

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$
Warmup: Empty System

later arrivals  me

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|---|---
1 |  |  
6 |  |  
14 |  |  

Later arrivals | Me
---|---

\[
X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3}
\end{cases}
\]
Warmup: Empty System

My size
Which arrivals delay me?  By how much?

1
6
14
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|---|---
1 | none | 
6 | 
14 | 

Later arrivals | me
---|---

\[ X = \{ \begin{array}{c} 1 \text{ w.p. } \frac{1}{3} \\ 6 \text{ w.p. } \frac{1}{3} \\ 14 \text{ w.p. } \frac{1}{3} \end{array} \]
Warmup: Empty System

later arrivals  me

My size  Which arrivals delay me?  By how much?

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>none</td>
<td>n/a</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|---|---
1 | none | n/a
6 | none | n/a
14 | none | n/a
Warmup: Empty System

<table>
<thead>
<tr>
<th>My size</th>
<th>Which arrivals delay me?</th>
<th>By how much?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>none</td>
<td>n/a</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|---|---
1 | none | n/a
6 | when $0 \leq \text{my age} < 3$ | 
14 | | 

$X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3}
\end{cases}$
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|---|---
1 | none | n/a
6 | when $0 \leq \text{my age} < 3$ | 1
14 |  | 

$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$
My size | Which arrivals delay me? | By how much?
---|---|---
1 | none | n/a
6 | when 0 \leq \text{my age} < 3 | 1
14 | | |
Warmup: Empty System

![Diagram of later arrivals and me]

<table>
<thead>
<tr>
<th>My size</th>
<th>Which arrivals delay me?</th>
<th>By how much?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>none</td>
<td>n/a</td>
</tr>
<tr>
<td>6</td>
<td>when $0 \leq$ my age $&lt; 3$</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|-----------------|---------
1  | none            | n/a     
6  | when $0 \leq \text{my age} < 3$ | 1       
14 | when $0 \leq \text{my age} < 7$ |
Warmup: Empty System

My size | Which arrivals delay me? | By how much?
---|---|---
1 | none | n/a
6 | when $0 \leq \text{my age} < 3$ | 1
14 | when $0 \leq \text{my age} < 7$ | 1

$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$
SOAP Insight #1: Pessimism Principle

Replace my *rank* with my *worst* future rank
Pessimism Principle

Replace my rank with my worst future rank
Pessimism Principle

Replace my rank with my worst future rank

my size = 1
Pessimism Principle

Replace my rank with my worst future rank

my size = 1
Pessimism Principle

Replace my rank with my worst future rank

my size = 1         my size = 6
Pessimism Principle

Replace my **rank** with my **worst** future rank

my size = 1  
my size = 6
Pessimism Principle

Replace my rank with my worst future rank

my size = 1  my size = 6  my size = 14
Pessimism Principle

Replace my **rank** with my **worst** future rank

my size = 1  
my size = 6  
my size = 14
Pessimism Principle

Replace my **rank** with my **worst** future rank

my size = 14
Pessimism Principle

Replace my **rank** with my **worst** future rank

my size = 14

Arrivals delay me by 1
Pessimism Principle

Replace my rank with my worst future rank

my size = 14

Arrivals delay me by 1

Arrivals don’t delay me
Pessimism Principle

Replace my **rank** with my **worst** future rank

\[
\rho_{\text{new}}(a) = \begin{cases} 
\lambda \cdot 1 & 0 \leq a < 7 \\
\lambda \cdot 0 & 7 \leq a < 14
\end{cases}
\]

my size = 14

Arrivals delay me by 1

Arrivals don’t delay me
Pessimism Principle

Replace my rank with my worst future rank

\[
\rho_{\text{new}}(a) = \begin{cases} 
  \lambda \cdot 1 & 0 \leq a < 7 \\
  \lambda \cdot 0 & 7 \leq a < 14 
\end{cases}
\]

\[
E[T_{14} | \text{empty}] = \int_0^{14} \frac{da}{1 - \rho_{\text{new}}(a)}
\]

Arrivals delay me by 1

Arrivals don’t delay me
Response Time Analysis

arrival

response time

departure
Response Time Analysis

arrival

first service

response time

departure
Response Time Analysis

arrival

waiting time

first service

response time

residence time

departure
Residence Time

arrival

first service

residence time

departure
Residence Time

Question: is residence time...
Residence Time

Question: is residence time...

• my size?
Question: is residence time...
• my size?
Question: is residence time...
• my size? ✗
**Residence Time**

**Question:** is residence time...

- my size? $\times$
- $E[T | \text{empty}]$?
Question: is residence time...
• my size? \( \times \)
• \( E[T \mid \text{empty}] \)?
Residence Time

Question: is residence time...
- my size? ✗
- $E[T \mid \text{empty}]$?
Residence Time

**Question**: is residence time...

- my size? \( \times \)
- \( E[T \mid \text{empty}] \)?
Residence Time

Question: is residence time...
• my size? $\times$
• $E[T \mid \text{empty}]$?
Question: is residence time...
- my size? √
- $E[T \mid \text{empty}]$?

Pessimism Principle: replace my rank with my worst future rank
Residence Time

arrival

first service

departure

my rank jumps up

**Question:** is residence time...
  - my size? ✗
  - $\mathbb{E}[T \mid \text{empty}]$?

**Pessimism Principle:** replace my rank with my worst future rank
Residence Time

**Question**: is residence time...
- my size? \(\times\)
- \(E[T | \text{empty}]\)?

**Pessimism Principle**: replace my rank with my worst future rank
**Residence Time**

**Question:** is residence time...
- my size? \( X \)
- \( \mathbb{E}[T \mid \text{empty}] \)?

**Pessimism Principle:** replace my rank with my worst future rank.
Residence Time

**Question**: is residence time...
- my size? $\times$
- $E[T \mid \text{empty}]$? $\checkmark$

**Pessimism Principle**: replace my rank with my worst future rank
**Residence Time**

**Question**: is residence time...
- my size? \( \xmark \)
- \( \mathbb{E}[T \mid \text{empty}]? \) \( \checkmark \)

**Pessimism Principle**: replace my rank with my worst future rank

\[
\mathbb{E}[R_{14}] = \mathbb{E}[T_{14} \mid \text{empty}] = \int_0^{14} \frac{da}{1 - \rho_{\text{new}}(a)}
\]
Waiting Time

arrival

waiting time

first service

departure
Waiting Time

arrival

waiting time

worst future rank = w

first service

departure
Waiting Time

See relevant work with rank \( \leq w \)

worst future rank = \( w \)
Waiting Time

arrival

waiting time

worst future rank = $w$

See relevant work with rank $\leq w$

$U[w] = \text{relevant work}$
Waiting Time

arrival

waiting time

worst future rank = \( w \)

See \textit{relevant work} with rank \( \leq w \)

 Relevant jobs gone

first service

departure

\[ U[w] = \text{relevant work} \]
Waiting Time

\[ U[w] = \text{relevant work} \]

Waiting time is *busy period* started by \( U[w] \)

See relevant work with \( \text{rank} \leq w \)

**worst** future rank = \( w \)

relevant jobs gone

first service

arrival

departure

waiting time
Response Time: Size 14
Response Time: Size 14
Response Time: Size 14
Response Time: Size 14

Relevant work ($w = 9$):
Response Time: Size 14

Relevant work ($w = 9$):

$$E[U[9]] = \frac{\lambda}{2} \cdot \frac{E[X^2]}{1 - \rho}$$
Response Time: Size 14

Relevant work ($w = 9$):

$$E[U[9]] = \frac{\lambda}{2} \cdot \frac{E[X^2]}{1 - \rho}$$

Waiting time:

$$E[Q_{14}] = \frac{E[U[9]]}{1 - \rho_{new}(0)}$$
Response Time: Size 14

Relevant work ($w = 9$):

$$E[U[9]] = \frac{\lambda}{2} \cdot \frac{E[X^2]}{1-\rho}$$

Waiting time:

$$E[Q_{14}] = \frac{E[U[9]]}{1-\rho_{\text{new}}(0)}$$

Residence time:

$$E[R_{14}] = \int_0^{14} \frac{da}{1-\rho_{\text{new}}(a)}$$
Response Time: Size 14

Relevant work \((w = 9)\):
\[
E[U[9]] = \frac{\lambda}{2} \cdot \frac{E[X^2]}{1 - \rho}
\]

Waiting time:
\[
E[Q_{14}] = \frac{E[U[9]]}{1 - \rho_{\text{new}}(0)}
\]

Residence time:
\[
E[R_{14}] = \int_{0}^{14} \frac{da}{1 - \rho_{\text{new}}(a)}
\]

\[
E[T_{14}] = E[Q_{14}] + E[R_{14}]
\]
Response Time: Size 14

Relevant work ($w = 9$):

$$E[U[9]] = \frac{\lambda}{2} \cdot \frac{E[X^2]}{1 - \rho}$$

Waiting time:

$$E[Q_{14}] = \frac{E[U[9]]}{1 - \rho_{\text{new}}(0)}$$

Residence time:

$$E[R_{14}] = \int_0^{14} \frac{\text{da}}{1 - \rho_{\text{new}}(a)}$$

$$E[T_{14}] = E[Q_{14}] + E[R_{14}]$$

$$\rho_{\text{new}}(a) = \begin{cases} 
\lambda \cdot 1 & 0 \leq a < 7 \\
\lambda \cdot 0 & 7 \leq a < 14 
\end{cases}$$
Response Time: Size 1
Response Time: Size 1
Response Time: Size 1
Response Time: Size 1

Relevant work ($w = 7$):

![Graph showing response time vs age]

- Rank
- Age
Response Time: Size 1

Relevant work ($w = 7$):

$\mathbb{E}[U[7]] = ???$
Relevant Work
Relevant Work

Suppose my size = 1
Relevant Work

Suppose my size = 1

\( w = 7 \)
Relevant Work

Suppose my size = 1

$w = 7$

$I_0$

age

rank

1 6 14
Relevant Work

Suppose my size = 1

\[ w = 7 \]

\[ I_0 \]

\[ I_1 \]
Suppose my size = 1

\( w = 7 \)
Relevant Work

Two causes of relevant work:

Suppose my size = 1

$w = 7$

$I_0$, $I_1$, $I_2$
Relevant Work

Suppose my size = 1

Two causes of relevant work:
• $I_0$: arrivals
Relevant Work

Two causes of relevant work:
- $I_0$: arrivals
- $I_1, I_2$: recyclings
Relevant Work

Two causes of relevant work:

- $I_0$: arrivals
- $I_1, I_2$: recyclings

Suppose my size = 1

Go from $\text{rank} > w$ to $\text{rank} \leq w$
Relevant Work

Suppose my size = 1

Observations:

$w = 7$
Relevant Work

Suppose my size = 1

Observations:
- at most one **recycled** job at a time
relevant work

$$\text{Suppose my size} = 1$$

Observations:
- at most one *recycled* job at a time
Relevant Work

Suppose my size = 1

Observations:
- at most one recycled job at a time
Relevant Work

Suppose my size = 1

Observations:
- at most one **recycled** job at a time
Relevant Work

Suppose my size = 1

Observations:
- at most one recycled job at a time
- recyclings occur only when no relevant work
SOAP Insight #2: Vacation Transformation

Replace recycled jobs with server vacations
Vacation Transformation

\[ w = 7 \]

\[ I_0 \quad I_1 \quad I_2 \]

age

rank

1 \quad 6 \quad 14
$w = 7$

$\text{Vacation Transformation}$
Vacation Transformation

\[ w = 7 \]

\[ \mathbf{E}[\mathbf{U}[7]] = \frac{\lambda}{2} \cdot \frac{\mathbf{E}[X_0^2] + \mathbf{E}[X_1^2] + \mathbf{E}[X_2^2]}{1 - \lambda \mathbf{E}[X_0]} \]

(Fuhrmann and Cooper, 1985)
Vacation Transformation

$X_i = \text{service a job receives in } I_i$

$w = 7$

$I_0$, $I_1$, $I_2$

$E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}$

(Fuhrmann and Cooper, 1985)
$X_i = \text{service a job receives in } I_i$

$X_0 =$

$X_1 =$

$X_2 =$

$E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}$

(Fuhrmann and Cooper, 1985)
Vacation Transformation

\[ X_i = \text{service a job receives in } I_i \]

\[ w = 7 \]

\[ X_0 = 1 \]

\[ X_1 = \]

\[ X_2 = \]

\[ \mathbb{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbb{E}[X_0^2] + \mathbb{E}[X_1^2] + \mathbb{E}[X_2^2]}{1 - \lambda \mathbb{E}[X_0]} \]

(Fuhrmann and Cooper, 1985)
Vacation Transformation

\[ X_i = \text{service a job receives in } I_i \]

\[ \mathbb{E}[U[7]] = \frac{\lambda}{2} \cdot \frac{\mathbb{E}[X_0^2] + \mathbb{E}[X_1^2] + \mathbb{E}[X_2^2]}{1 - \lambda \mathbb{E}[X_0]} \]

(Fuhrmann and Cooper, 1985)
Vacation Transformation

\[ X_i = \text{service a job receives in } I_i \]

\[ w = 7 \]

\[ X_0 = 1 \]

\[ X_1 = \begin{cases} 
0 & \text{w.p. } \frac{1}{3} \\
3 & \text{w.p. } \frac{2}{3} 
\end{cases} \]

\[ X_2 = \begin{cases} 
0 & \text{w.p. } \frac{2}{3} \\
7 & \text{w.p. } \frac{1}{3} 
\end{cases} \]

\[
E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}
\]

(Fuhrmann and Cooper, 1985)
Response Time: Size 1

Relevant work ($w = 7$):

$$E[U[7]] = ???$$
Response Time: Size 1

Relevant work ($w = 7$):

$$E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}$$
Response Time: Size 1

Relevant work \((w = 7)\):

\[
E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}
\]

Waiting time:

\[
E[Q_1] = \frac{E[U[7]]}{1 - \rho_{\text{new}}(0)}
\]
Response Time: Size 1

Relevant work ($w = 7$):

$$E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}$$

Waiting time:

$$E[Q_1] = \frac{E[U[7]]}{1 - \rho_{new}(0)}$$

Residence time:

$$E[R_1] = \int_0^1 \frac{da}{1 - \rho_{new}(a)}$$
Response Time: Size 1

Relevant work ($w = 7$):

$$E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}$$

Waiting time:

$$E[Q_1] = \frac{E[U[7]]}{1 - \rho_{new}(0)}$$

Residence time:

$$E[R_1] = \int_0^1 \frac{da}{1 - \rho_{new}(a)}$$

$$\rho_{new}(a) = \lambda \cdot 0$$
Response Time: Size 1

Relevant work ($w = 7$):

$$E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}$$

Waiting time:

$$E[Q_1] = \frac{E[U[7]]}{1 - \rho_{\text{new}}(0)} = E[U[7]]$$

Residence time:

$$E[R_1] = \int_0^1 \frac{\rho_{\text{new}}(a)}{1 - \rho_{\text{new}}(a)} \, da = 1$$

$$\rho_{\text{new}}(a) = \lambda \cdot 0$$
Response Time: Size 1

Relevant work \((w = 7)\):

\[
E[U[7]] = \frac{\lambda}{2} \cdot \frac{E[X_0^2] + E[X_1^2] + E[X_2^2]}{1 - \lambda E[X_0]}
\]

Waiting time:

\[
E[Q_1] = \frac{E[U[7]]}{1 - \rho_{\text{new}}(0)} = E[U[7]]
\]

Residence time:

\[
E[R_1] = \int_{0}^{1} \frac{\rho_{\text{new}}(a) = \lambda \cdot 0}{1 - \rho_{\text{new}}(a)} \, da = 1
\]

Response time:

\[
E[T_1] = E[Q_1] + E[R_1]
\]
Running example: SERPT

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Running example: SERPT

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3}
\end{cases} \]
$E[T]$ of any SOAP Policy
Worst Future Rank
Worst Future Rank

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]
Worst Future Rank

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]
Relevant Intervals
Relevant Intervals

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]
Relevant Intervals

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]
Relevant Intervals

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]

**Detail:** start with \( i = 0 \) iff first interval contains age 0, else start with \( i = 1 \)
Relevant Intervals

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]

**Detail:** start with \( i = 0 \) iff first interval contains age 0, else start with \( i = 1 \)

**Detail:** interval can be empty
SOAP Analysis: One Descriptor
**SOAP** Analysis: One **Descriptor**

**Worst Future Rank**

\[ w_x(a) = \sup_{a\leq b < x} r(b) \]

**Relevant Intervals**

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]
**SOAP Analysis: One Descriptor**

**Worst Future Rank**

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]

\[
E[T_x] = \frac{\lambda}{2} \sum_{i=0}^{\infty} E[X_i[w_x]^2] \frac{(1 - \rho_0[w_x])(1 - \rho_{\text{new}}[w_x])}{(1 - \rho_{\text{new}}[w_x])} + \int_0^x \frac{da}{1 - \rho_{\text{new}}[w_x(a)]}
\]

**Relevant Intervals**

\[ I_i[w] = i\text{th interval when } r(a) \leq w \]
SOAP Analysis: One Descriptor

Worst Future Rank
\[ w_x(a) = \sup_{a \leq b < x} r(b) \]
\[ w_x = w_x(0) \]

Relevant Intervals
\[ I_i[w] = i\text{th interval when } r(a) \leq w \]

\[
E[T_x] = \frac{\lambda}{2} \sum_{i=0}^{\infty} E[X_i[w_x]^2] \frac{(1 - \rho_0[w_x])(1 - \rho_{\text{new}}[w_x])}{(1 - \rho_{\text{new}}[w_x])(1 - \rho_{\text{new}}[w_x])} + \int_{0}^{x} \frac{da}{1 - \rho_{\text{new}}[w_x(a)]}
\]
SOAP Analysis: One Descriptor

**Worst Future Rank**

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]
\[ w_x = w_x(0) \]

**Relevant Intervals**

\[ I_i[w] = i\text{th interval when } r(a) \leq w \]
\[ X_i[w] = \text{service a job receives in } I_i[w] \]

\[
E[T_x] = \frac{\lambda}{2} \sum_{i=0}^{\infty} E[X_i[w_x]^2] \cdot \frac{(1 - \rho_0[w_x])(1 - \rho_{\text{new}}[w_x])}{x} + \int_0^x \frac{da}{1 - \rho_{\text{new}}[w_x(a)]}
\]
**SOAP Analysis: One Descriptor**

**Worst Future Rank**

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]

\[ w_x = w_x(0) \]

**Relevant Intervals**

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]

\[ X_i[w] = \text{service a job receives in } I_i[w] \]

\[ \rho_0[w] = \lambda \mathbf{E}[X_0[w]] \]

\[
\mathbf{E}[T_x] = \frac{\lambda}{2} \sum_{i=0}^{\infty} \mathbf{E}[X_i[w_x]^2]
\]

\[
= \frac{X}{(1 - \rho_0[w_x])(1 - \rho_{\text{new}}[w_x])}
\]

\[
+ \int_0^x \frac{da}{1 - \rho_{\text{new}}[w_x(a)]}
\]
**SOAP Analysis: One Descriptor**

**Worst Future Rank**

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]

\[ w_x = w_x(0) \]

**Relevant Intervals**

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]

\[ X_i[w] = \text{service a job receives in } I_i[w] \]

\[ \rho_0[w] = \lambda \mathbb{E}[X_0[w]] \]

\[ \rho_{\text{new}}[w] = \lambda \mathbb{E}[X_0[w-]] \]

\[ \mathbb{E}[T_x] = \frac{\frac{\lambda}{2} \sum_{i=0}^{\infty} \mathbb{E}[X_i[w_x]^2]}{(1 - \rho_0[w_x])(1 - \rho_{\text{new}}[w_x])} \]

\[ + \int_0^x \frac{da}{1 - \rho_{\text{new}}[w_x(a)]} \]
**Worst Future Rank**

\[ w_x(a) = \sup_{a \leq b < x} r(b) \]

**Relevant Intervals**

\[ I_i[w] = \text{ith interval when } r(a) \leq w \]
**Worst** Future Rank

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]

**Relevant** Intervals

\[ I_{i,d}[w] = \text{ith interval when } r_d(a) \leq w \]
**SOAP Analysis: Complete**

**Worst Future Rank**

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]

**Relevant Intervals**

\[ I_{i,d}[w] = \text{ith interval when } r_d(a) \leq w \]

\[ X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w] \]
**Worst Future Rank**

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]

**Relevant Intervals**

\[ I_{i,d}[w] = \text{ith interval when } r_d(a) \leq w \]

\[ X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w] \]

\[ X_d = \text{size distribution for descriptor } d \]
SOAP Analysis: Complete

**Worst Future Rank**

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]

**Relevant Intervals**

\[ I_{i,d}[w] = \text{ith interval when } r_d(a) \leq w \]

\[ X_{d}[w] = \text{size distribution for descriptor } d \]

\[ X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w] \]

\[ X_{i}[w] = X_{i,D}[w] \]
**Worst Future Rank**

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]

**Relevant Intervals**

\[ I_{i,d}[w] = i\text{th interval when } r_d(a) \leq w \]
\[ X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w] \]
\[ X_i[w] = X_{i,D}[w] \]

\( X_d = \text{size distribution for descriptor } d \)

\( D = \text{descriptor distribution} \)
Worst Future Rank

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]
\[ w_{d,x} = w_{d,x}(0) \]

Relevant Intervals

\[ I_{i,d}[w] = i\text{th interval when } r_d(a) \leq w \]
\[ X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w] \]
\[ X_i[w] = X_{i,D}[w] \]
\[ \rho_0[w] = \lambda \mathbb{E}[X_0[w]] \]
\[ \rho_{\text{new}}[w] = \lambda \mathbb{E}[X_0[w-]] \]
**Worst Future Rank**

\[ w_{d,x}(a) = \sup_{a \leq b < x} r_d(b) \]

\[ w_{d,x} = w_{d,x}(0) \]

**Relevant Intervals**

\[ I_{i,d}[w] = \text{ith interval when } r_d(a) \leq w \]

\[ X_{i,d}[w] = \text{service a job of descriptor } d \text{ receives in } I_{i,d}[w] \]

\[ X_i[w] = X_{i,D}[w] \]

\[ \rho_0[w] = \lambda E[X_0[w]] \]

\[ \rho_{\text{new}}[w] = \lambda E[X_0[w-]] \]

\[ X_d = \text{size distribution for descriptor } d \]

\[ E[T_{d,x}] = \frac{\frac{\lambda}{2} \sum_{i=0}^{\infty} E[X_i[w_{d,x}]^2]}{(1 - \rho_0[w_{d,x}])(1 - \rho_{\text{new}}[w_{d,x}])} + \int_0^x \frac{da}{1 - \rho_{\text{new}}[w_{d,x}(a)]} \]
Example: Preemptive Priority

- Normal: 2
- Urgent: 1
Example: Preemptive Priority

Urgent ($d = U, r = 1$)

Normal ($d = N, r = 2$)
Example: Preemptive Priority

Urgent \((d = U, r = 1)\)

• 1/4 of all jobs

Normal \((d = N, r = 2)\)
Example: Preemptive Priority

**Urgent** \((d = U, \ r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, \ r = 2)\)
Example: Preemptive Priority

Urgent ($d = U, r = 1$)
- 1/4 of all jobs
- Size distribution $X_U$

Normal ($d = N, r = 2$)
- 3/4 of all jobs
Example: Preemptive Priority

**Urgent** \( (d = U, \ r = 1) \)
- 1/4 of all jobs
- Size distribution \( X_U \)

**Normal** \( (d = N, \ r = 2) \)
- 3/4 of all jobs
- Size distribution \( X_N \)
Example: Preemptive Priority

**Urgent** ($d = U, \ r = 1$)
- 1/4 of all jobs
- Size distribution $X_U$

**Normal** ($d = N, \ r = 2$)
- 3/4 of all jobs
- Size distribution $X_N$

$I_{0,U}[1-] = \quad I_{0,N}[1-] =
I_{0,U}[1] = \quad I_{0,N}[1] =
I_{0,U}[2-] = \quad I_{0,N}[2-] =
I_{0,U}[2] = \quad I_{0,N}[2] =
Example: Preemptive Priority

Urgent ($d = U$, $r = 1$)
• 1/4 of all jobs
• Size distribution $X_U$

Normal ($d = N$, $r = 2$)
• 3/4 of all jobs
• Size distribution $X_N$

$I_{0,U}[1-] =$

$I_{0,U}[1] =$

$I_{0,U}[2-] =$

$I_{0,U}[2] =$

$I_{0,N}[1-] =$

$I_{0,N}[1] =$

$I_{0,N}[2-] =$

$I_{0,N}[2] =$
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
I_{0,U}[1] = \emptyset \\
I_{0,U}[1] = \\
I_{0,U}[2] = \\
I_{0,U}[2] =
\]

\[
I_{0,N}[1] = \\
I_{0,N}[1] = \\
I_{0,N}[2] = \\
I_{0,N}[2] =
\]
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
I_{0,U}[1-] = \emptyset
\]
\[
I_{0,U}[1] =
\]
\[
I_{0,U}[2-] =
\]
\[
I_{0,U}[2] =
\]
\[
I_{0,N}[1-] = \emptyset
\]
\[
I_{0,N}[1] =
\]
\[
I_{0,N}[2-] =
\]
\[
I_{0,N}[2] =
\]
Example: Preemptive Priority

**Urgent** \( (d = U, r = 1) \)
- 1/4 of all jobs
- Size distribution \( X_U \)

**Normal** \( (d = N, r = 2) \)
- 3/4 of all jobs
- Size distribution \( X_N \)

\[
\begin{align*}
I_{0,U}[1-] &= \emptyset \\
I_{0,U}[1] &= \\
I_{0,U}[2-] &= \\
I_{0,U}[2] &= \\
I_{0,N}[1-] &= \emptyset \\
I_{0,N}[1] &= \\
I_{0,N}[2-] &= \\
I_{0,N}[2] &= 
\end{align*}
\]
Example: Preemptive Priority

Urgent ($d = U, r = 1$)
- 1/4 of all jobs
- Size distribution $X_U$

Normal ($d = N, r = 2$)
- 3/4 of all jobs
- Size distribution $X_N$

$I_{0,U}[1] = [0, \infty)$
$I_{0,U}[2] = I_{0,U}[1] = [0, \infty)$

$I_{0,N}[1] = I_{0,N}[2] = [0, 1)$

$I_{0,N}[1] = \emptyset$
$I_{0,N}[2] = \emptyset$

$I_{0,U}[1] = \emptyset$
$I_{0,U}[2] = \emptyset$
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
I_{0, U}[1] = [0, \infty)
\]
\[
I_{0, U}[2] =
\]
\[
I_{0, N}[1] = \emptyset
\]
\[
I_{0, N}[2] =
\]
Example: Preemptive Priority

**Urgent** ($d = U$, $r = 1$)
- 1/4 of all jobs
- Size distribution $X_U$

**Normal** ($d = N$, $r = 2$)
- 3/4 of all jobs
- Size distribution $X_N$

\[
I_{0,U}[1^+] = \emptyset
\]
\[
I_{0,U}[1] = [0, \infty)
\]
\[
I_{0,U}[2^+] = \]
\[
I_{0,U}[2] = \]

\[
I_{0,N}[1^+] = \emptyset
\]
\[
I_{0,N}[1] = \emptyset
\]
\[
I_{0,N}[2^+] = \]
\[
I_{0,N}[2] = \]
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
I_{0,U}[1-] = \emptyset \\
I_{0,U}[1] = [0, \infty) \\
I_{0,U}[2-] = [0, \infty) \\
I_{0,U}[2] =
\]

\[
I_{0,N}[1-] = \emptyset \\
I_{0,N}[1] = \emptyset \\
I_{0,N}[2-] = \\
I_{0,N}[2] =
\]
Example: Preemptive Priority

**Urgent** ($d = U$, $r = 1$)
- 1/4 of all jobs
- Size distribution $X_U$

**Normal** ($d = N$, $r = 2$)
- 3/4 of all jobs
- Size distribution $X_N$

$I_{0,U}[1-] = \emptyset$
$I_{0,U}[1] = [0, \infty)$
$I_{0,U}[2-] = [0, \infty)$
$I_{0,U}[2] = \emptyset$

$I_{0,N}[1-] = \emptyset$
$I_{0,N}[1] = \emptyset$
$I_{0,N}[2-] = \emptyset$
$I_{0,N}[2] = \emptyset$
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
\begin{align*}
I_{0,U}[1-] &= \emptyset \\
I_{0,U}[1] &= [0, \infty) \\
I_{0,U}[2-] &= [0, \infty) \\
I_{0,U}[2] &= \\
I_{0,N}[1-] &= \emptyset \\
I_{0,N}[1] &= \emptyset \\
I_{0,N}[2-] &= \emptyset \\
I_{0,N}[2] &= \\
\end{align*}
\]
Example: Preemptive Priority

**Urgent** \((d = U, \ r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, \ r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
I_{0,U}[1-] = \emptyset \\
I_{0,U}[1] = [0, \infty) \\
I_{0,U}[2-] = [0, \infty) \\
I_{0,U}[2] = [0, \infty)
\]

\[
I_{0,N}[1-] = \emptyset \\
I_{0,N}[1] = \emptyset \\
I_{0,N}[2-] = \emptyset \\
I_{0,N}[2] = \emptyset
\]
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
I_{0,U}[1-] = \emptyset \\
I_{0,U}[1] = [0, \infty) \\
I_{0,U}[2-] = [0, \infty) \\
I_{0,U}[2] = [0, \infty)
\]

\[
I_{0,N}[1-] = \emptyset \\
I_{0,N}[1] = \emptyset \\
I_{0,N}[2-] = \emptyset \\
I_{0,N}[2] = [0, \infty)
\]
Example: Preemptive Priority

**Urgent** \((d = U, r = 1)\)
- 1/4 of all jobs
- Size distribution \(X_U\)

**Normal** \((d = N, r = 2)\)
- 3/4 of all jobs
- Size distribution \(X_N\)

\[
\begin{align*}
X_{0,U}[1^-] &= 0 \\
X_{0,U}[1] &= X_U \\
X_{0,U}[2^-] &= X_U \\
X_{0,U}[2] &= X_U \\
\end{align*}
\]

\[
\begin{align*}
X_{0,N}[1^-] &= 0 \\
X_{0,N}[1] &= 0 \\
X_{0,N}[2^-] &= 0 \\
X_{0,N}[2] &= X_N \\
\end{align*}
\]
Example: Preemptive Priority

<table>
<thead>
<tr>
<th>Priority</th>
<th>Normal</th>
<th>Urgent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jobs</td>
<td>$\frac{3}{4}$ of all jobs</td>
<td>$\frac{1}{4}$ of all jobs</td>
</tr>
<tr>
<td>Size</td>
<td>$X_N$</td>
<td>$X_U$</td>
</tr>
</tbody>
</table>

Normal:
- $X_0[2-] = \begin{cases} X_U & \text{w.p.} \frac{1}{4} \\ 0 & \text{w.p.} \frac{3}{4} \end{cases}$

Urgent:
- $X_0[1-] = 0$
- $X_0[1] = \begin{cases} X_U & \text{w.p.} \frac{1}{4} \\ 0 & \text{w.p.} \frac{3}{4} \end{cases}$

$X_0[2-] = \begin{cases} X_U & \text{w.p.} \frac{1}{4} \\ 0 & \text{w.p.} \frac{3}{4} \end{cases}$

$X_{0,U}[2-] = X_U$  
$X_{0,U}[2] = X_U$  
$X_{0,N}[1-] = 0$  
$X_{0,N}[1] = 0$  
$X_{0,N}[2-] = 0$  
$X_{0,N}[2] = X_N$
Part 2: analyzing SOAP policies
Part 2: analyzing SOAP policies
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Part 3: policy design with SOAP
Design Problems

Bucketed SRPT

Noisy Systems
Question: given number of priority levels, which job sizes go in which size buckets?
Two Buckets

\[ X = \text{bounded Pareto on } [1, 10^6] \text{ with } \alpha = 1 \]
Two Buckets

\[ X = \text{bounded Pareto on } [1, 10^6] \text{ with } \alpha = 1 \]

\[ t = \text{threshold between buckets} \]
Two Buckets

$X = \text{bounded Pareto on } [1, 10^6] \text{ with } \alpha = 1$

$t = \text{threshold between buckets}$

Bucketed SRPT
Two Buckets

\[ X = \text{bounded Pareto on } [1, 10^6] \text{ with } \alpha = 1 \]

\[ t = \text{threshold between buckets} \]

**Bucketed SRPT**

\[ E[T] \]

- \( t = 10^1 \)
- \( t = 10^2 \)
- \( t = 10^3 \)
- \( t = 10^4 \)
- \( t = 10^5 \)
Two Buckets

\[ X = \text{bounded Pareto on } [1, 10^6] \text{ with } \alpha = 1 \]
\[ t = \text{threshold between buckets} \]

**Bucketed PSJF**

![Graph showing the expected waiting time for different thresholds](chart.png)
Noisy System
Noisy System

Gittins minimizes $E[T]$

$$X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases}$$
Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?
Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?
Noisy System

Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?
Noisy System

Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?
Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?

A: Each age has rank range
Noisy System

Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?
A: Each age has rank range
Gittins minimizes $E[T]$

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

Q: What if we have noisy age information?
A: Each age has rank range

Q: How do we analyze resulting scheduling policy?
Noisy System

Gittins minimizes $\mathbb{E}[T]$

Q: What if we have noisy age information?
A: Each age has rank range

Q: How do we analyze resulting scheduling policy?
A: SOAP Bubble analysis

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

Idea: do tagged job analysis, but...
Idea: do tagged job analysis, but...

• I get *worst* possible rank
Idea: do tagged job analysis, but...

- I get worst possible rank
- Everyone else gets best possible rank
**Idea:** do tagged job analysis, but...

- I get *worst* possible rank
- Everyone else gets *best* possible rank

**Theorem:** this *always* gives an upper bound on $E[T]$
**SOAP Bubble Analysis**

**Idea:** do tagged job analysis, but...
- I get *worst* possible rank
- Everyone else gets *best* possible rank

Noise could be adversarial!

**Theorem:** this *always* gives an upper bound on $E[T]$
Designing for Noisy Systems

Gittins

\[
X = \begin{cases} 
1 \text{ w.p. } \frac{1}{3} \\
6 \text{ w.p. } \frac{1}{3} \\
14 \text{ w.p. } \frac{1}{3}
\end{cases}
\]
Designing for Noisy Systems

Gittins

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Problem:
I can jump up to rank 9 before age 1

Gittins

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3}
\end{cases} \]
Designing for Noisy Systems

Problem:
I can jump up to rank 9 before age 1

Solution: shift

Gittins

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Designing for Noisy Systems

Problem:
I can jump up to rank 9 before age 1

Solution: \textit{shift}

\[ X = \begin{cases} 1 & \text{w.p. } \frac{1}{3} \\ 6 & \text{w.p. } \frac{1}{3} \\ 14 & \text{w.p. } \frac{1}{3} \end{cases} \]
Designing for Noisy Systems

Problem:
I can jump up to rank 9 before age 1

Solution: shift

Problem:
other jobs might not reach rank 9

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Designing for Noisy Systems

Problem:
I can jump up to rank 9 before age 1
Solution: shift

Problem:
other jobs might not reach rank 9
Solution: flatten

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Designing for Noisy Systems

Problem:
I can jump up to rank 9 before age 1
Solution: *shift*

Problem:
other jobs might not reach rank 9
Solution: *flatten*
Designing for Noisy Systems

Problem:
I can jump up to rank 9 before age 1
Solution: shift

Problem:
other jobs might not reach rank 9
Solution: flatten

Theorem:
\[ E[T \text{ of Shift-Flat Gittins with noise } \Delta] = E[T \text{ of Gittins without noise}] + O(\Delta) \]
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Part 4: optimality proofs with SOAP
Gittins vs. SERPT
Gittins vs. SERPT

Gittins

\[ r(a) = \sup_{\Delta > 0} \frac{\mathbb{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbb{P}[X - a \leq \Delta \mid X > a]} \]
Gittins vs. SERPT

Gittins

\[ r(a) = \sup_{\Delta > 0} \frac{E[\min\{X - a, \Delta\} | X > a]}{P[X - a \leq \Delta | X > a]} \]

SERPT

\[ r(a) = E[X - a | X > a] \]
Gittins vs. SERPT

Gittins

\[ r(a) = \sup_{\Delta > 0} \frac{\mathbb{E}[\min\{X - a, \Delta\} | X > a]}{\mathbb{P}[X - a \leq \Delta | X > a]} \]

⚠️ Minimizes \( \mathbb{E}[T] \), but can be intractable

SERPT

\[ r(a) = \mathbb{E}[X - a | X > a] \]
Gittins vs. SERPT

Gittins

\[ r(a) = \sup_{\Delta > 0} \frac{\mathbb{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbb{P}[X - a \leq \Delta \mid X > a]} \]

⚠️ Minimizes \( \mathbb{E}[T] \), but can be intractable

SERPT

\[ r(a) = \mathbb{E}[X - a \mid X > a] \]

⚠️ Simple, but no \( \mathbb{E}[T] \) guarantee
Gittins vs. SERPT

Gittins

\[ r(a) = \sup_{\Delta > 0} \frac{\mathbb{E}[\min\{X - a, \Delta\} \mid X > a]}{\mathbb{P}[X - a \leq \Delta \mid X > a]} \]

⚠️ Minimizes \( \mathbb{E}[T] \), but can be intractable

SERPT

\[ r(a) = \mathbb{E}[X - a \mid X > a] \]

⚠️ Simple, but no \( \mathbb{E}[T] \) guarantee

**Question**: is there a simple policy with near-optimal \( \mathbb{E}[T] \)?
Monotonic SERPT

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3}
\end{cases} \]
**Monotonic SERPT**

M-SERPT is like SERPT, but *rank* never goes down

\[ X = \begin{cases} 
1 & \text{w.p. } \frac{1}{3} \\
6 & \text{w.p. } \frac{1}{3} \\
14 & \text{w.p. } \frac{1}{3} 
\end{cases} \]
Monotonic SERPT

M-SERPT is like SERPT, but *rank* never goes down

\[ X = \begin{pmatrix} 1 & \text{w.p.} \frac{1}{3} \\ 6 & \text{w.p.} \frac{1}{3} \\ 14 & \text{w.p.} \frac{1}{3} \end{pmatrix} \]
Monotonic SERPT

M-SERPT is like SERPT, but rank never goes down

Theorem:

\[
\frac{E[T \text{ of M-SERPT}]}{E[T \text{ of Gittins}]} \leq 5
\]
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP
Outline

Part 1: defining SOAP policies

Part 2: analyzing SOAP policies

Part 3: policy design with SOAP

Part 4: optimality proofs with SOAP


References: Possible Applications


