Optimal Scheduling and Exact Response Time Analysis for Multistage Jobs

Ziv Scully
Mor Harchol-Balter
Alan Scheller-Wolf
Carnegie Mellon University
Response Time
Response Time
Response Time

$= T = \text{response time}$
Response Time

Goal: schedule to minimize mean response time $E[T]$
Perfect Information

Known job sizes
Perfect Information

Known job sizes

6 12 4 9
Perfect Information

Known job sizes

Optimal policy: *SRPT* (serve job of smallest remaining size)
Perfect Information

Known job sizes

Optimal policy: *SRPT*
(serve job of smallest *remaining size*)

Zero Information

Unknown job sizes
Perfect Information

Known job sizes
Optimal policy: *SRPT* (serve job of smallest remaining size)

Zero Information

Unknown job sizes
Optimal policy: *Gittins policy* (serve job of smallest Gittins rank)
Perfect Information

Known job sizes
Optimal policy: **SRPT**
(serve job of smallest *remaining* size)

Zero Information

Unknown job sizes
Optimal policy: **Gittins policy**
(serve job of smallest *Gittins rank*)

6 12 4 9

\[G = 5\]
\[G = 3\]
\[G = 8\]
\[G = 11\]
Perfect Information

Known job sizes
Optimal policy: SRPT
(serve job of smallest remaining size)

Zero Information

Unknown job sizes
Optimal policy: Gittins policy
(serve job of smallest Gittins rank)

Open problem: partial information
Partial Information: Multistage Jobs
Partial Information: Multistage Jobs

Jobs have *multiple stages*
Partial Information: Multistage Jobs

Jobs have *multiple stages*

- unknown stage sizes
Partial Information: Multistage Jobs

Jobs have *multiple stages*
- unknown stage sizes
- unknown stage sequence
Partial Information: Multistage Jobs

Jobs have *multiple stages*
- unknown stage sizes
- unknown stage sequence
- … but know *which stage* we’re on
Partial Information: Multistage Jobs

Jobs have multiple stages
• unknown stage sizes
• unknown stage sequence
• ... but know which stage we’re on

For each job, scheduler knows:
Partial Information: Multistage Jobs

Jobs have *multiple stages*
- unknown stage sizes
- unknown stage sequence
- … but know *which stage* we’re on

For each job, scheduler knows:
- *stage* in progress
Partial Information: Multistage Jobs

Jobs have *multiple stages*
- unknown stage sizes
- unknown stage sequence
  - ... but know *which stage* we’re on

For each job, scheduler knows:
- *stage* in progress
- *age* of that stage
Multistage Job Examples
Multistage Job Examples

Job R: Repairing an item
Multistage Job Examples

Job R: Repairing an item
Multistage Job Examples

Job R: Repairing an item

Diagnosis

\[ p \]

Repair
(easy)
Multistage Job Examples

Job R: Repairing an item

Diagnosis

\[ p \quad 1-p \]

Repair (easy)  Repair (hard)
Multistage Job Examples

Job R: Repairing an item

- Diagnosis
  - $p$
  - $1 - p$
  - Repair (easy)
  - Repair (hard)

Job G: Google ad placement
Multistage Job Examples

Job R: Repairing an item

Diagnosis

\[ p \quad 1 - p \]

Repair (easy) \quad Repair (hard)

Job G: Google ad placement

Preprocessing (uniform)
**Multistage Job Examples**

Job R: Repairing an item

- **Diagnosis**
  - $p$
  - $1 - p$
  - Repair (easy)
  - Repair (hard)

Job G: Google ad placement

- **Preprocessing** (uniform)
- **Targeting** (Pareto)
Multistage Job Examples

Job R: Repairing an item

- Diagnosis
  - $p$
  - $1 - p$
- Repair (easy)
- Repair (hard)

Job G: Google ad placement

- Preprocessing (uniform)
- Targeting (Pareto)
- Selection (Pareto)
Scheduling Multistage Jobs
Scheduling Multistage Jobs

Job J

3

1 or 9
Scheduling Multistage Jobs

Job J

Job K

1 or 9

1 or 9

3

3
Scheduling Multistage Jobs

Job J

1 or 9

Job K

1 or 9

Which should we serve first?
Which should we serve first? Job K first
Which should we serve first? Job K first
What if we shorten J’s first stage?
Scheduling Multistage Jobs

Job J

1 or 9

2

1 or 9

Job K

3

1 or 9

Which should we serve first? Job K first

What if we shorten J’s first stage?

• Shorten to 2:
Scheduling Multistage Jobs

Job J
- 2
- 1 or 9

Job K
- 1 or 9
- 3

Which should we serve first? Job K first

What if we shorten J’s first stage?
- Shorten to 2: still K first
Which should we serve first? Job K first
What if we shorten J’s first stage?
• Shorten to 2: still K first
• Shorten to 1:
Which should we serve first? Job K first

What if we shorten J’s first stage?
• Shorten to 2: still K first
• Shorten to 1: now J first
Prioritizing "short" jobs

Prioritizing "informative" jobs

Tradeoff
Tradeoff

Prioritizing “short” jobs

Job J
2
1 or 9

Prioritizing “informative” jobs

Job K
1 or 9
3

Job K
Tradeoff

Prioritizing “short” jobs

Prioritizing “informative” jobs
How do we balance the tradeoff?

Prioritizing “short” jobs

Prioritizing “informative” jobs

Job J

Job K

How do we balance the tradeoff?
How do we balance the tradeoff?

**Multistage Gittins policy**
For multistage jobs:
For multistage jobs:

Gittins rank is defined...
For multistage jobs:

Gittins rank is defined...

Gittins policy is optimal for minimizing $E[T]$...
Good News

For multistage jobs:

Gittins rank is *defined*...

Gittins policy is *optimal* for minimizing $E[T]...$
Good News

Gittins rank is *defined* ...

Bad News

Gittins policy is *optimal* for minimizing $\mathbb{E}[T]$ ...

For multistage jobs:
Good News

Gittins rank is defined...

Gittins policy is optimal for minimizing $E[T]$...

Bad News

... but is intractable to compute

For multistage jobs:
Good News

Gittins rank is defined…

Gittins policy is optimal for minimizing $E[T]$…

Bad News

… but is intractable to compute

… but unknown how to analyze $E[T]$
Good News

Gittins rank is defined...

Gittins policy is optimal for minimizing $E[T]$...

Bad News

... but is intractable to compute

... but unknown how to analyze $E[T]$

For multistage jobs:

Need new version of Gittins policy
Good News

Gittins rank is defined...

Gittins policy is optimal for minimizing $E[T]$...

Bad News

... but is intractable to compute

... but unknown how to analyze $E[T]$

For multistage jobs:

Need new version of Gittins policy with no bad news
Our contribution: a new approach to the Gittins policy that naturally scales to multistage jobs
Our contribution: a new approach to the Gittins policy that naturally scales to multistage jobs

New approach: single-job profit (SJP)
Our contribution:
a new approach to the Gittins policy that naturally scales to multistage jobs

New approach: single-job profit (SJP)
- Helps *compute* Gittins rank of multistage jobs
Our contribution: a new approach to the Gittins policy that naturally scales to multistage jobs

New approach: single-job profit (SJP)
• Helps compute Gittins rank of multistage jobs
• Yields exact formula for $E[T]$
Wanted: Composition Law

Sequential Composition

J ▷ K

Diagram:

1. Sequential Composition
2. The notation J ▷ K represents sequential composition.
3. The diagram shows a flow from J to K, indicating the composition of functions J and K.
Wanted: Composition Law

Sequential Composition

\[ J \triangleright K \]

\[ \begin{array}{c}
J \\
G_J \\
\downarrow \\
K \\
G_K \\
\end{array} \]
Wanted: Composition Law

Sequential Composition

\[ G_{J \triangleleft K} = ??? \]
Single-Job Profit

Game with a job and potential reward
Single-Job Profit

Game with a job and potential reward
• Get reward if we complete the job
Single-Job Profit

Game with a job and potential reward
- Get reward if we complete the job
- Pay for time spent serving the job
Single-Job Profit

Game with a job and potential **reward**
- Get **reward** if we complete the job
- Pay for **time** spent serving the job
- Can **give up** at any time
Single-Job Profit

Game with a job and potential reward
• Get reward if we complete the job
• Pay for time spent serving the job
• Can give up at any time

Goal: maximize profit:
\[ E[\text{reward received} - \text{time spent}] \]
Single-Job Profit Example

1 (50%)
9 (50%)
Single-Job Profit Example

\[ V_J(r) = \text{profit} \]

\[ r \text{ = reward} \]

1 (50%)
9 (50%)
Single-Job Profit Example

\[ V_J(r) = \text{profit} \]

Definition: \( V_J \) is the SJP function of \( J \)
Definition: $V_J$ is the SJP function of $J$
Single-Job Profit Example

\[ V_J(r) = \text{profit} \]

\[ r = \text{reward} \]

**Definition:** \( V_J \) is the SJP function of \( J \)

1 (50%)
9 (50%)
Single-Job Profit Example

\[ V_J(r) = \text{profit} \]

Definition: \( V_J \) is the SJP function of \( J \)
Single-Job Profit Example

\[ V_J(r) = \text{profit} \]

Definition: \( V_J \) is the SJP function of \( J \)
Single-Job Profit Example

**Definition**: \( V_J(r) = \) profit

**Theorem**: \( G_J = V_J^{-1}(0) \)

**Definition**: \( V_J \) is the SJP function of \( J \)
SJP Composition Law

Theorem: \( V_{J \searrow K}(r) = V_J(V_K(r)) \)
SJP Composition Law

Theorem: $V_{J>K}(r) = V_J(V_K(r))$

Proof:
SJP Composition Law

Theorem: $V_{J \triangleright K}(r) = V_J(V_K(r))$

Proof:

\[ \begin{array}{c}
J \\
\downarrow \\
K \\
\downarrow \\
\text{\$} \\
\end{array} \quad \begin{array}{c}
J \\
\downarrow \\
K \\
\downarrow \\
\text{\$} \\
\end{array} \]
Response Time Impact

Job R
Response Time Impact

Job R

1

2/3 1/3

4 12
Response Time Impact

Compare three policies:

Job R

Compare three policies:
Response Time Impact

Compare three policies:
• *First-come, first-served* (FCFS)
Response Time Impact

Compare three policies:

- *First-come, first-served* (FCFS)
- *Blind Gittins policy* (BGP): ignores stage information
Compare three policies:

- *First-come, first-served* (FCFS)
- *Blind Gittins policy* (BGP): ignores stage information
- *Multistage Gittins policy* (MGP): exploits stage information
Compare three policies:

- First-come, first-served (FCFS)
- Blind Gittins policy (BGP): ignores stage information
- Multistage Gittins policy (MGP): exploits stage information

<table>
<thead>
<tr>
<th></th>
<th>FCFS</th>
<th>BGP</th>
<th>MGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rho = 0.75 )</td>
<td>21.95</td>
<td>21.53</td>
<td>17.87</td>
</tr>
<tr>
<td>( \rho = 0.937 )</td>
<td>79.08</td>
<td>76.4</td>
<td>54.86</td>
</tr>
</tbody>
</table>

**Job R**

- 1
- 2/3
- 1/3
- 4
- 12

**Response Time Impact**

- 40% reduction

**Graph**

- Bar chart comparing E[T] for FCFS, BGP, and MGP under different \( \rho \) values.
- FCFS and BGP show higher E[T] compared to MGP.
- MGP consistently shows the lowest E[T] across both \( \rho \) values.
Compare three policies:

• First-come, first-served (FCFS)

• Blind Gittins policy (BGP): ignores stage information

• Multistage Gittins policy (MGP): exploits stage information

\[
\begin{array}{ccc}
\text{Job R} & \text{1} & \text{2/3} & \text{1/3} \\
\text{4} & \text{12} & & \\
\end{array}
\]

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

\[\text{E}[X]\]

\[
\begin{array}{cccc}
\text{FCFS} & \text{BGP} & \text{MGP} \\
\rho = 0.75 & & & \\
21.95 & 21.53 & 17.87 \\
\rho = 0.937 & & & \\
79.08 & 76.4 & & 54.86 \\
\end{array}
\]

\[28\% \text{ reduction}\]
Problem: Gittins policy for multistage jobs
Problem: Gittins policy for multistage jobs

Solution: new single-job profit (SJP) approach
Problem: Gittins policy for multistage jobs

Solution: new single-job profit (SJP) approach
  • SJP composition law
Problem: Gittins policy for multistage jobs

Solution: new single-job profit (SJP) approach

• SJP composition law

Impact: significantly reduces $E[T]$
Two Building Blocks
Two Building Blocks

Sequential Composition

\[ J \xrightarrow{\bowtie} K \]

J \rightarrow K
Two Building Blocks

Sequential Composition

\[ V_{J \searrow K}(r) = V_J(V_K(r)) \]
Two Building Blocks

Sequential Composition

$J \triangleright K$

$V_{J \triangleright K}(r) = V_J(V_K(r))$

Mixture Composition

$J \mid K$

$J \mid K = pJ + (1-p)K$

Two Building Blocks

Sequential Composition

\[ V_{J \triangleright K}(r) = V_J(V_K(r)) \]

Mixture Composition

\[ V_{J|K}(r) = pV_J(r) + (1 - p)V_K(r) \]
Two Building Blocks

Sequential Composition

\[ V_{J \triangleright K}(r) = V_J(V_K(r)) \]

Mixture Composition

\[ V_{J | K}(r) = pV_J(r) + (1 - p)V_K(r) \]

Every multistage job can be built from these