Greedy Scheduling

https://cs.pomona.edu/classes/cs140/
Outline

Topics and Learning Objectives
• Introduce greedy algorithms
• Discuss the greedy scheduling algorithm
• Discuss exchange argument proofs

Exercise
• Greedy scheduling
Extra Resources

• Introduction to Algorithms, 3rd, chapter 16
• Algorithms Illuminated Part 3: Chapter 13

• Lots of examples: https://www.geeksforgeeks.org/greedy-algorithms/
Greedy Algorithms

• Iteratively make myopic (short-sighted) decisions and hope it works
• Never go back and recheck/reevaluate that you were correct

Contrasting with Divide and Conquer
• It is generally easier to create greedy algorithms (good and bad to this)
• It is typically easier to analyze greedy algorithms (no master theorem)
• It is often harder to prove/understand the correctness of greedy algorithms
• It is common for greedy algorithms to be incorrect
Greedy Algorithms

Proofs of correctness

• It can sometimes feel like more of an art than a science

1. Proof by induction on the greedy decision
2. Proof by induction on an exchange argument
   1. Either by contraction
   2. Or by exchanging with the optimal solution
3. Whatever works...
Example of a greedy algorithm

• We’ve seen one greedy algorithm before. What was it?

• What path length does Dijkstra’s output for $S \rightarrow W$?
• What is the correct shortest path length for $S \rightarrow W$?
Scheduling (ignoring concurrency)

You have a shared resource
For example, a processor
You have many jobs that need to use the resource

Each job $j$ has:
• A Priority $P_j$ that stands for the job’s importance
• A Duration $D_j$ that stands for the length of time to run the job

In what sequence should we complete the jobs?
Scheduling (ignoring concurrency)

In what sequence should we complete the jobs?

• What is our criteria? What do we want to optimize?
• Let’s start by looking at job j’s **completion time** $C_j$
• Given three jobs: $D_1 = 1$, $D_2 = 2$, $D_3 = 3$
• What is the completion time for each if they are scheduled in order?
What is the completion time of Job 5?
Scheduling

Optimization objective: \( \textit{minimize the weighted sum of completion times} \)

\[
S_{\text{cost}} = \min \left[ \sum_{j=1}^{n} P_j C_j \right]
\]

What is the weighted sum of completion times if we schedule the following jobs in order?

<table>
<thead>
<tr>
<th>Job</th>
<th>( J_1 )</th>
<th>( J_2 )</th>
<th>( J_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>( D_1 = 1 )</td>
<td>( D_2 = 2 )</td>
<td>( D_3 = 3 )</td>
</tr>
<tr>
<td>Priority</td>
<td>( P_1 = 3 )</td>
<td>( P_2 = 2 )</td>
<td>( P_3 = 1 )</td>
</tr>
</tbody>
</table>
Exercise Question 1, 2, and 3
Scheduling

Calculate the weighted sum of completion times for the following jobs if they are scheduled in the order: 1, 2, 3.

<table>
<thead>
<tr>
<th>Job</th>
<th>( J_1 )</th>
<th>( J_2 )</th>
<th>( J_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration</td>
<td>( D_1 = 1 )</td>
<td>( D_2 = 2 )</td>
<td>( D_3 = 3 )</td>
</tr>
<tr>
<td>Priority</td>
<td>( P_1 = 3 )</td>
<td>( P_2 = 2 )</td>
<td>( P_3 = 1 )</td>
</tr>
<tr>
<td>Completion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Weighted sum of completion times: ?
Greedy Scheduling

Our process for creating a greedy scheduling algorithm

1. Look at some special cases for our problem
2. Describe some possible greedy criteria
3. Compare our greedy criteria
4. Select the “best” one
5. Prove correctness if possible
Greedy Scheduling

Goal: devise a greedy algorithm to minimize the weighted sum of completion times

Why are we approaching this problem with a greedy algorithm?
• It’s a pretty easy way to start.
• Compare the approach we go through in these slides with a Divide and Conquer approach
1. What are some special cases to consider?

Consider **two** jobs with **equal durations** (\(D\))

- These jobs have different **priorities** (\(P_H\) and \(P_L\))
- **Do we schedule the lower or higher priority job first?**

![Diagram](image.png)
1. What are some special cases to consider?

Consider two jobs with equal durations (D)
• These jobs have different priorities ($P_H$ and $P_L$)
• Do we schedule the lower or higher priority job first?
1. What are some special cases to consider?

Consider **two** jobs with **equal durations** \((D)\)

- These jobs have different **priorities** \((P_H \text{ and } P_L)\)
- **Do we schedule the lower or higher priority job first?**
Schedule with Lower Priority First

Schedule with Higher Priority First
1. What are some special cases to consider?

Consider two jobs with equal durations (D)

- These jobs have different priorities ($P_H$ and $P_L$)
- Do we schedule the lower or higher priority job first?

Schedule higher priority jobs first so that they have the quickest completion times.
1. What are some special cases to consider?

Consider two jobs with equal priorities (P)
- These jobs have different durations ($D_E$ and $D_S$)
- Do we schedule the shorter or longer (Extended) job first?

<table>
<thead>
<tr>
<th>Job</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_{E}$</td>
<td>Longer/Extended</td>
</tr>
<tr>
<td>$J_{S}$</td>
<td></td>
</tr>
</tbody>
</table>
1. What are some special cases to consider?

Consider two jobs with equal priorities (P)

- These jobs have different durations ($D_E$ and $D_S$)
- Do we schedule the shorter or longer (Extended) job first?

![Diagram showing two jobs with different durations](image-url)
1. What are some special cases to consider?

Consider two jobs with equal priorities (P)

- These jobs have different durations ($D_E$ and $D_S$)
- **Do we schedule the shorter or longer (Extended) job first?**
Schedule with Shorter Job First

Schedule with Longer Job First
1. What are some special cases to consider?

Consider **two** jobs with **equal priorities** (P)

- These jobs have different **durations** ($D_E$ and $D_S$)
- **Do we schedule the shorter or longer (Extended) job first?**

<table>
<thead>
<tr>
<th>Job</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$J_{SS}$</td>
<td>(Shorter)</td>
</tr>
<tr>
<td>$J_{EE}$</td>
<td>(Longer/Extended)</td>
</tr>
</tbody>
</table>

Schedule **shortest jobs** first to minimize average completion times.
2. Describe some possible greedy criteria

What do we do when in the more general case:

\[ P_i > P_j \text{ and } D_i > D_j \] (job i has higher priority and longer duration)

What are some simple scoring functions that aggregate our criteria?

We want a function for which jobs with a bigger score are scheduled first:
• Score increases for higher priorities
• Score increases for shorter times

1. Greedy Criterion 1: \( P_i - D_i \) (take the difference)
2. Greedy Criterion 2: \( \frac{P_i}{D_i} \) (take the ratio)

1. Schedule highest priority first
2. Schedule shortest duration first
3. Compare our greedy criteria

• Jobs will be ordered from biggest to smallest value

<table>
<thead>
<tr>
<th>Job with same duration</th>
<th>Difference Metric ($P_i - D_i$)</th>
<th>Ratio Metric ($P_i/D_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: P=2, D=1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job 2: P=5, D=1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which job should be scheduled first?
3. Compare our greedy criteria

- Jobs will be ordered from biggest to smallest value

<table>
<thead>
<tr>
<th>Job with same duration</th>
<th>Difference Metric ( (P_i - D_i) )</th>
<th>Ratio Metric ( (P_i/D_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: ( P=2, D=1 )</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Job 2: ( P=5, D=1 )</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Highest priority

Total weighted sum
3. Compare our greedy criteria

• Jobs will be ordered from biggest to smallest value

<table>
<thead>
<tr>
<th>Job with same duration</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i/D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=2, D=1)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Job 2: (P=5, D=1)</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

Total weighted sum: 5*1 + 2*2 = 9  
5*1 + 2*2 = 9

Job with same priority

<table>
<thead>
<tr>
<th>Job with same priority</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i/D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=1, D=3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job 2: (P=1, D=4)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total weighted sum

Which job should be scheduled first?

Same Result
3. Compare our greedy criteria

- Jobs will be ordered from biggest to smallest value

<table>
<thead>
<tr>
<th>Job with same duration</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i/D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=2, D=1)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Job 2: (P=5, D=1)</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td>(5<em>1 + 2</em>2 = 9)</td>
<td>(5<em>1 + 2</em>2 = 9)</td>
</tr>
</tbody>
</table>

Highest priority

<table>
<thead>
<tr>
<th>Job with same priority</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i/D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=1, D=3)</td>
<td>-2</td>
<td>1/3</td>
</tr>
<tr>
<td>Job 2: (P=1, D=4)</td>
<td>-3</td>
<td>1/4</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Shortest time

Which job should be scheduled first?
3. Compare our greedy criteria

• Jobs will be ordered from biggest to smallest value

<table>
<thead>
<tr>
<th>Job with same duration</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i / D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=2, D=1)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Job 2: (P=5, D=1)</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td>(5<em>1 + 2</em>2 = 9)</td>
<td>(5<em>1 + 2</em>2 = 9)</td>
</tr>
</tbody>
</table>

Highest priority

<table>
<thead>
<tr>
<th>Job with same priority</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i / D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=1, D=3)</td>
<td>-2</td>
<td>(1/3)</td>
</tr>
<tr>
<td>Job 2: (P=1, D=4)</td>
<td>-3</td>
<td>(1/4)</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td>(1<em>3 + 1</em>7 = 10)</td>
<td>(1<em>3 + 1</em>7 = 10)</td>
</tr>
</tbody>
</table>

Shortest time

Which job should be scheduled first? Same Result
3. Compare our greedy criteria

• Let’s try to get them to disagree.
• Why does it matter if they don’t produce the same result?
• One scoring metric must be better than the other

• Apply the two greedy algorithms and calculate their weighted sum of completion times
3. Compare our greedy criteria

• Jobs will be ordered from biggest to smallest metric value

<table>
<thead>
<tr>
<th>Job</th>
<th>Difference Metric ($P_i - D_i$)</th>
<th>Ratio Metric ($P_i/D_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: P=3, D=5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job 2: P=1, D=2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total weighted sum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exercise Question 4
3. Compare our greedy criteria

• Jobs will be ordered from biggest to smallest metric value

<table>
<thead>
<tr>
<th>Job</th>
<th>Difference Metric ($P_i - D_i$)</th>
<th>Ratio Metric ($P_i/D_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: $P=3$, $D=5$</td>
<td>-2</td>
<td>3/5</td>
</tr>
<tr>
<td>Job 2: $P=1$, $D=2$</td>
<td>-1</td>
<td>1/2</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Which job goes first?
3. Compare our greedy criteria

• Jobs will be ordered from biggest to smallest metric value

<table>
<thead>
<tr>
<th>Job</th>
<th>Difference Metric ((P_i - D_i))</th>
<th>Ratio Metric ((P_i / D_i))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: (P=3), (D=5)</td>
<td>-2</td>
<td>3/5</td>
</tr>
<tr>
<td>Job 2: (P=1), (D=2)</td>
<td>-1</td>
<td>1/2</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which job goes first?
What is the priority sum?
4. Select the “best” one

- Jobs will be ordered from biggest to smallest metric value

<table>
<thead>
<tr>
<th>Job</th>
<th>Difference Metric ($P_i - D_i$)</th>
<th>Ratio Metric ($P_i/D_i$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job 1: $P=3$, $D=5$</td>
<td>-2</td>
<td>3/5</td>
</tr>
<tr>
<td>Job 2: $P=1$, $D=2$</td>
<td>-1</td>
<td>1/2</td>
</tr>
<tr>
<td>Total weighted sum</td>
<td>$1<em>2 + 3</em>7 = 23$</td>
<td>$3<em>5 + 1</em>7 = 22$</td>
</tr>
</tbody>
</table>

Which job goes first?
What is the priority sum?
Which criteria is better?
5. Prove correctness if possible

Is criteria 2 optimal?
• We don’t know yet.

Claim: Criteria 2 is optimal for minimizing the weighted sum of completion times.

• We’re going to prove this using an exchange argument!
Exchange Arguments

• Consider your greedy solution, G

• Consider an alternative solution, A
  • A can be anything that is not G
  • Create A by changing G in some way

• Compare these solutions
  • Show that turning A into G makes A get better
Proof

• Assume that we have no ties (all $P_i/D_i$ are distinct numbers)
• Fix an arbitrary input with $n$ jobs
• Let’s perform a proof using an exchange argument contradiction

Let $G =$ the greedy schedule and $A =$ the (alternative) optimal schedule
• Let's assume that $A$ must be better than $G$ (assume greedy is not optimal)
• To perform the contradiction, we must show that $G$ is better than $A$, thus contradicting the purported optimality of $A$
Proof

Let $G = \text{the greedy schedule}$ and $A = \text{the optimal schedule}$

• Assume that: $P_1/D_1 > P_2/D_2 > \ldots > P_n/D_n$
• We can just rename all jobs after we calculate their scores...
• Thus, $G$ is just job 1 followed by job 2 etc. (1, 2, \ldots, n)

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Weight</th>
<th>Length</th>
<th>Ratio</th>
<th>Reorder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0.3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1.3</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
<td>6.0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>0.2</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>9</td>
<td>0.1</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>3</td>
<td>2.3</td>
<td>2</td>
</tr>
</tbody>
</table>
Proof

Let $G =$ the greedy schedule and $A =$ the optimal schedule

- Assume that: $P_1/D_1 > P_2/D_2 > \ldots > P_n/D_n$
- We can just rename all jobs after we calculate their scores...
- Thus, $G$ is just job 1 followed by job 2 etc. (1, 2, …, n)
- For $A$ there must be at least two jobs that are “out of order”
  - Specifically, jobs $i$ and $j$ where $i$ is scheduled after $j$, but $S_i > S_j$ (for example, Job$_5$ after Job$_6$)
- The greedy schedule is the only schedule where the jobs are in order
(jobs i and j where i is scheduled after j, but $P_i/D_i > P_j/D_j$)

Job i has a larger greedy score

For example, i=7 and j=8
(jobs i and j where i is scheduled after j, but $P_i/D_i > P_j/D_j$)

Job i has a larger greedy score

How does the exchange affect the completion time for:
1. Jobs other than i and j?
2. Job i
3. Job j

For example, i=7 and j=8
G vs A

(jobs i and j where i is scheduled after j, but $P_i/D_i > P_j/D_j$)

Job i has a larger greedy score

A Schedule

Before j i After

G Schedule

Before i j After

Ordered based on greedy scores

What is the weighted sum of completion times for each schedule?

For example, i=7 and j=8
\[
\text{Cost}(A) = \text{Cost(Before)} + P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) + \text{Cost(After)}
\]
\[
\text{Cost}(G) = \text{Cost(Before)} + P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) + \text{Cost(After)}
\]
\[
\text{Cost}(A) < \text{Cost}(G) \quad \text{Implied by optimality of } A
\]
\[
\text{Cost(Before)} + P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) + \text{Cost(After)}
\]
\[
< \text{Cost(Before)} + P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) + \text{Cost(After)}
\]
\[ \text{Cost}(A) = \text{Cost(Before)} + P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) + \text{Cost(After)} \]
\[ \text{Cost}(G) = \text{Cost(Before)} + P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) + \text{Cost(After)} \]

\[ \text{Cost}(A) < \text{Cost}(G) \quad \text{Implied by optimality of } A \]

\[ \text{Cost(Before)} + P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) + \text{Cost(After)} < \text{Cost(Before)} + P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) + \text{Cost(After)} \]

\[ P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) < P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) \]

\[ P_j \times T_b + P_j \times D_j + P_i \times T_b + P_i \times D_j + P_i \times D_i < P_i \times T_b + P_i \times D_i + P_j \times T_b + P_j \times D_i + P_j \times D_j \]
\[
\text{Cost}(A) = \text{Cost(Before)} + P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) + \text{Cost(After)}
\]
\[
\text{Cost}(G) = \text{Cost(Before)} + P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) + \text{Cost(After)}
\]

\[
\text{Cost}(A) < \text{Cost}(G) \quad \text{Implied by optimality of A}
\]

\[
\text{Cost(Before)} + P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) + \text{Cost(After)} < \text{Cost(Before)} + P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j) + \text{Cost(After)}
\]

\[
P_j \times (T_b + D_j) + P_i \times (T_b + D_j + D_i) < P_i \times (T_b + D_i) + P_j \times (T_b + D_i + D_j)
\]

\[
P_j \times T_b + P_j \times D_j + P_i \times T_b + P_i \times D_j + P_i \times D_i < P_i \times T_b + P_i \times D_i + P_j \times T_b + P_j \times D_i + P_j \times D_j
\]

\[
P_i \times D_j < P_j \times D_i
\]

\[
P_i / D_i < P_j / D_j \quad \text{Contradiction to how they were ordered by our greedy criteria}
\]
Multiple Re-orderings

Our proof doesn’t account for this
Multiple Re-orderings
Example with Randomly Generated Jobs

<table>
<thead>
<tr>
<th>Job ID</th>
<th>Weight</th>
<th>Length</th>
<th>Ratio</th>
<th>Reorder</th>
<th>Greedy Time</th>
<th>Greedy Weighted</th>
<th>Unoptimized Time</th>
<th>Unoptimized Weighted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>0.3</td>
<td>4</td>
<td>14</td>
<td>14</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>6</td>
<td>1.3</td>
<td>3</td>
<td>10</td>
<td>80</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>1</td>
<td>6.0</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>11</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>0.2</td>
<td>5</td>
<td>19</td>
<td>19</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>9</td>
<td>0.1</td>
<td>6</td>
<td>28</td>
<td>28</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>3</td>
<td>2.3</td>
<td>2</td>
<td>4</td>
<td>28</td>
<td>28</td>
<td>196</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>175</strong></td>
<td></td>
<td></td>
<td><strong>387</strong></td>
</tr>
</tbody>
</table>
Summary of Greedy Scheduling

- Given $n$ jobs, each with a priority and a duration
- Give each job a score based on their ratio of priority to duration
- Schedule jobs in decreasing order of their score
- This gives us an optimal schedule

- What do we do if we’re given more jobs while these are running?
- Any issues with this scheme?
  - Some jobs might always be postponed.