# Greedy Algorithms and Interval Scheduling

•••

By Aviva Kosansky and Mary Boman

# What is a Greedy Algorithm?

Builds Solution in Small Steps

Does not look at big picture only at each step

At each step a decision is made to move closer to goal

Once a decision is made it is never reconsidered

All problems do not have an optimal greedy solution

Short term optimality can lead to worst-case long term solution

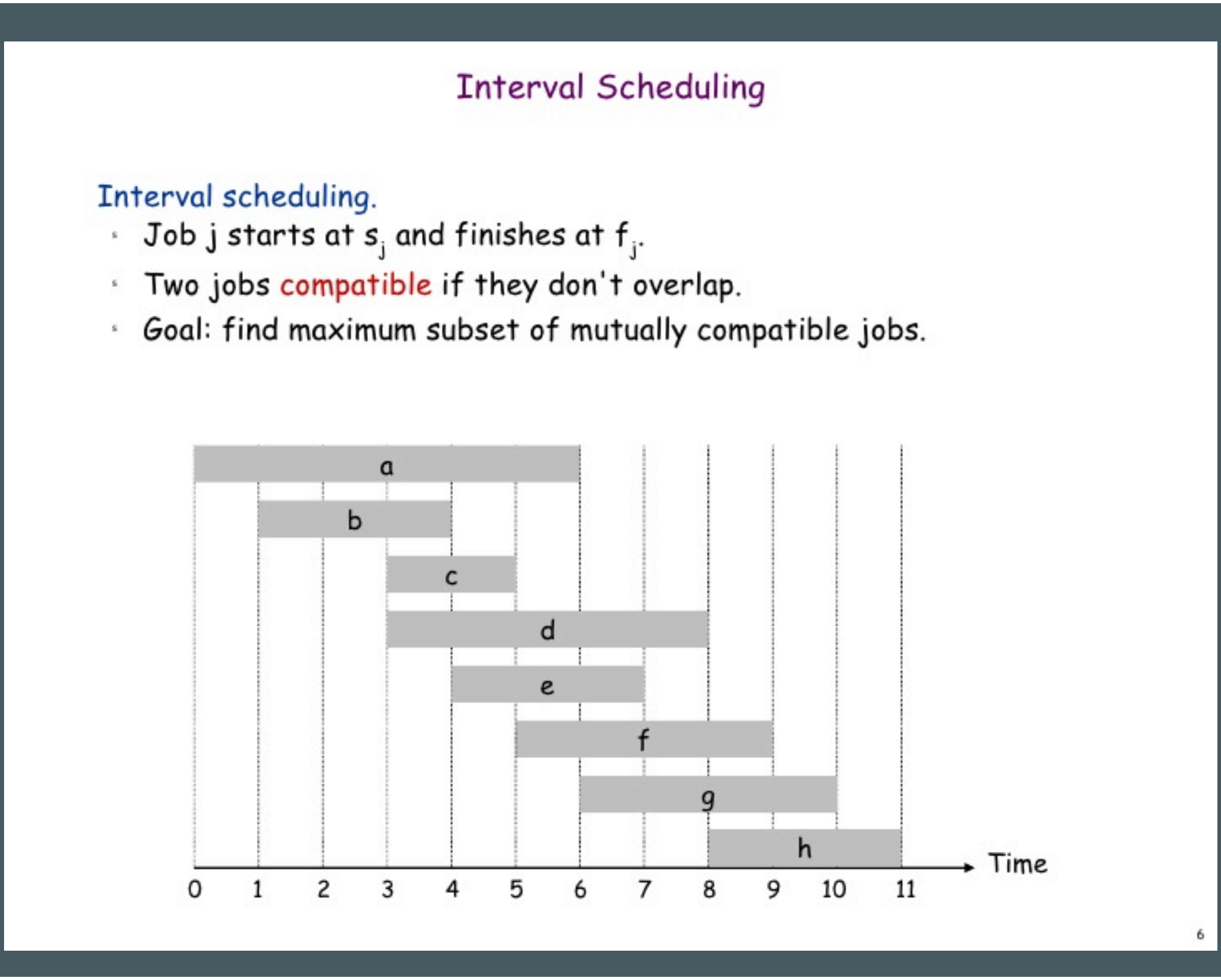
# Interval Scheduling

The Problem:

Given a set of intervals, start time and finish time, find the group of intervals where we can schedule the maximum number of jobs in non overlapping time slots.

AND

Find a simple rule (algorithm) that will allow you to do this.



# Interval Scheduling

Interval scheduling.

- Job j starts at $s_j$ and finishes at $f_j$.
- Two jobs compatible if they don't overlap.
- Goal: find maximum subset of mutually compatible jobs.

a
b
c
d
e
f
g
h
Time
0
1
2
3
4
5
6
7
8
9
10
11

# Try Solutions

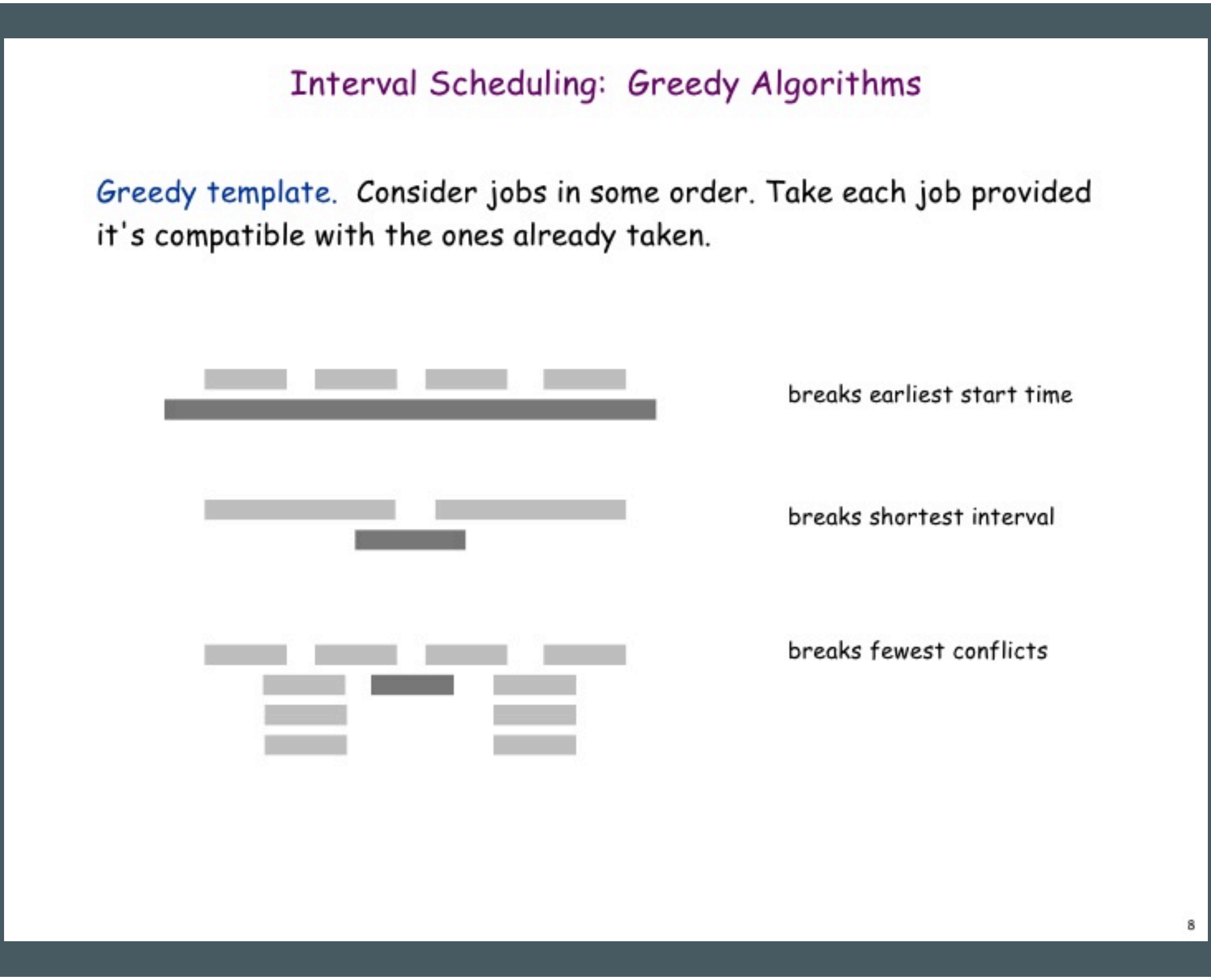
GOAL: Get Optimal Solution

Sort by Start Time

Sort by shortest interval

Sort by interval with fewest conflicts

Sort by end time



# Interval Scheduling: Greedy Algorithms

Greedy template. Consider jobs in some order. Take each job provided it's compatible with the ones already taken.

breaks earliest start time

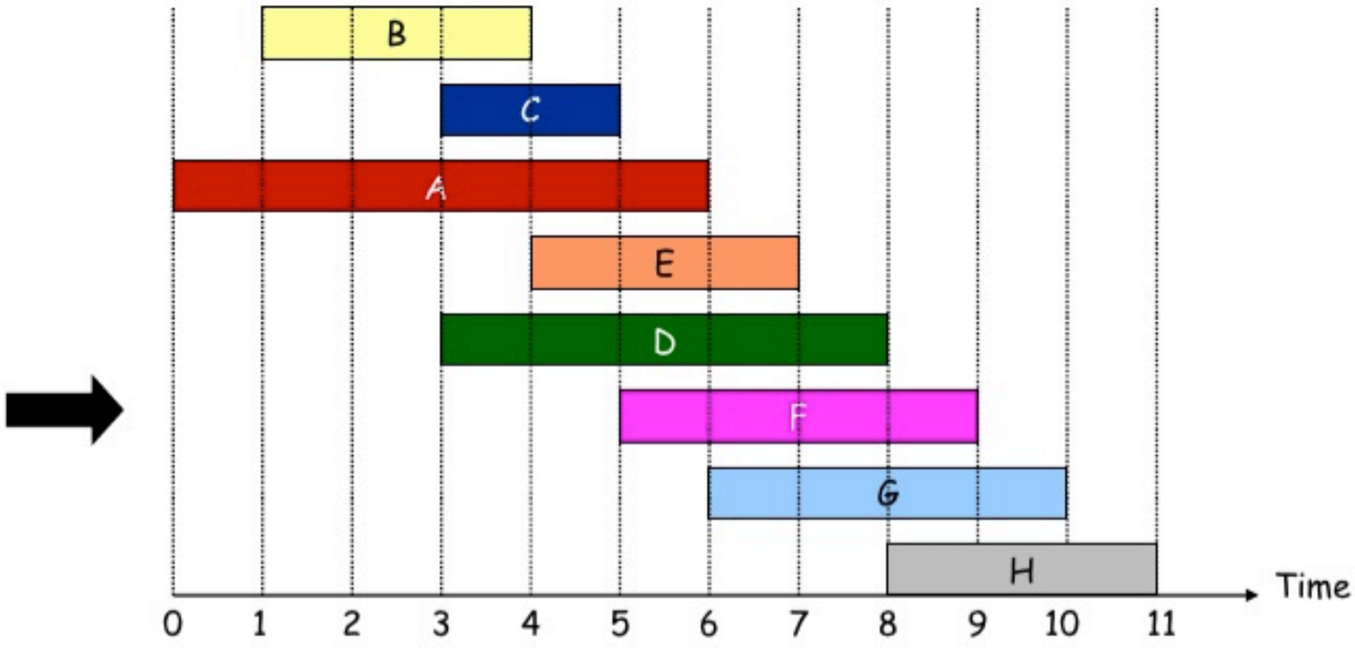
breaks shortest interval

breaks fewest conflicts

# The Algorithm

1. Select the interval, *x*, with the earliest **finishing time**.
2. Remove *x*, and all intervals intersecting *x*, from the set of candidate intervals.
3. Continue until the set of candidate intervals is empty.

# Interval Scheduling



# Analysis of Algorithm

Run time of Interval Scheduling is O(n log n) due to sorting by end time

The solution is optimal since it “stays ahead” of any other solution

This means the nth job chosen by our algorithm is the nth job chosen by the optimal solution

By construction we know that this greedy algorithm is a valid solution, all scheduled jobs are compatible