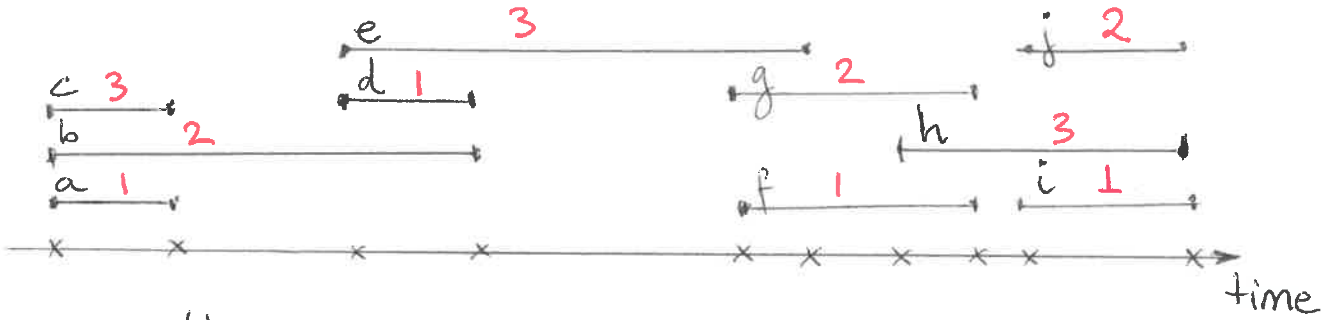
Greedy Algorithms

Scheduling all intervals

$n = 10$ intervals

depth $d = 3$



Algorithm

$O(n \lg n)$ sort the intervals by their start times, breaking ties arbitrarly

let $I_1, I_2, \ldots, I_n$ denote the intervals in this order

$O(n \lg n)$ compute depth d

$O(n^2)$

for $j = 1, 2, \ldots, n$

- $O(n)$ **for** each interval I_i that precedes I_j in sorted order and overlaps it
  - exclude the label of I_i from consideration for I_j
- $O(n)$ **if** there is any label from $\{1, 2, \ldots, d\}$ that has not been excluded then
  - assign a nonexcluded label to I_j
- **else** leave I_j unlabeled

- compute the depth d in $O(n \cdot \lg n)$
  - sort all start and finish times in monotonically increasing order
  - linear traversal to compute d
- interval $[s_i, f_i)$
- total running time: $\boxed{RT = O(n^2)}$

$\longrightarrow$ $d = 3$

labels

1	2	3
F	F	F
F	T	F
T	T	F
…		

$1 \le d \le n$

Scheduling to Minimize Lateness

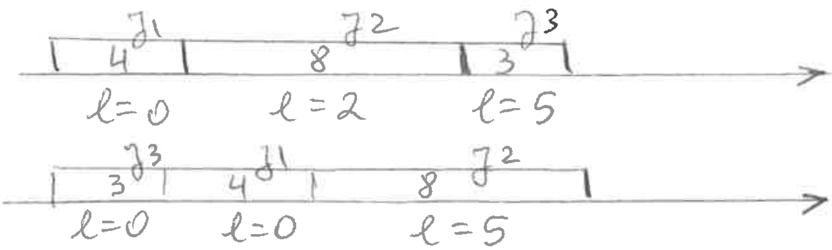
PPT presentation, slide 16

All possible orderings of the jobs with the same deadline yield the same maximum lateness.

example

$J_1(t_1=4, d_1=10)$, $J_2(t_2=8, d_2=10)$, and $J_3(t_3=3, d_3=10)$

possible schedules:



$L=5$

$L=5$

... so on

Last job sheduled always has lateness $= 15 - 10 = 5$

PPT presentation, slide 17

- examine the schedule from the beginning to find the first inversion J_i, J_j

inversion!

$J_1, J_2, J_5, J_6, J_3, \ldots$

swap

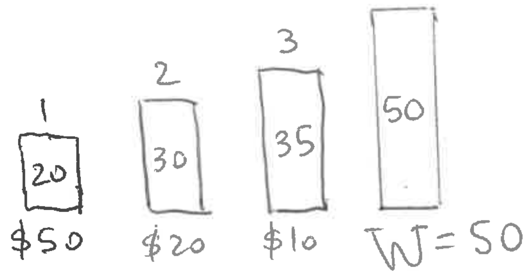
$J_1, J_2, J_5, J_3, J_6, \ldots$

- swapping does not increase the maximum lateness!

Greedy Algorithms
Fractional Knapsack Problem

* greedy choice 1: choose the object with the largest weight first
  - does not always yield an optimal solution

example

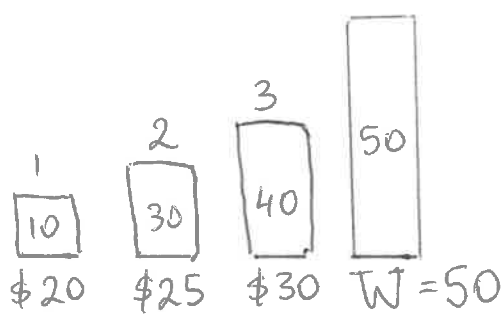


greedy choice 1: $\begin{cases} \text{item } 3 + \frac{1}{2} \text{ item } 2 \\ \text{value} = \$10 + \frac{1}{2} \cdot \$20 = \$20 \end{cases}$ → not optimal

optimal solution: $\begin{cases} \text{item } 1 + \text{item } 2 \\ \text{value} = \$50 + \$20 = \$70 \end{cases}$

* greedy choice 2: choose the object with the largest value first
  - does not always yield an optimal solution

example



greedy choice 2: item $3 + \frac{1}{3}$ item 2 → not optimal

value $= \$30 + \frac{1}{3} \cdot \$25 = \$38.33$

optimal solution: $\begin{cases} \text{item } 1 + \text{item } 2 + \frac{1}{4} \text{ item } 3 \\ \text{value} = \$20 + \$25 + \frac{1}{4} \cdot \$30 = \$52.5 \end{cases}$

* greedy choice 3 : choose the object with the largest v/w value

- always yields an optimal solution !

example

$n = 5$, $W = 100$

	2	3	1	5	4
w	10	20	30	40	50
v	20	30	66	40	60
v/w	2	1.5	2.2	1	1.2

→ order of objects in sorted order

- sort objects in decreasing order of v/w

  load = 0
  value = 0

- $i = 1$

  $x_1 = 1$
  load = 30
  value = 66

- $i = 2$

  $x_2 = 1$
  load = 30 + 10 = 40
  value = 66 + 20 = 86

- $i = 3$

  $x_3 = 1$
  load = 40 + 20 = 60
  value = 86 + 30 = 116

- $i = 4$

  $x_4 = 4/5$
  load $= 60 + 4/5 \cdot 50 = 100$
  value $= 116 + 4/5 \cdot 60 = 164$

Optimal solution : $\text{object}_1 + \text{object}_2 + \text{object}_3 + 4/5\ \text{object}_4$

total value = 164