Nov 14, 2014

Weighted Interval Scheduling

Input: n tasks (s_i, f_i, v_i) $i \in [n]$ ($v_i \geq 0$)

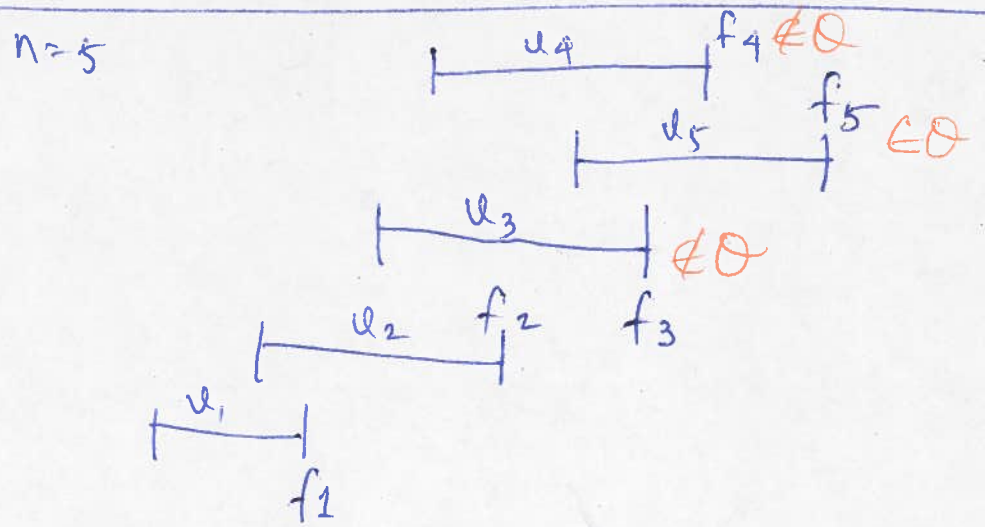
s_i: start time, f_i: finish time, v_i: value

Output: A valid schedule $S \subseteq [n]$

s.t. $v(S) = \sum_{i \in S} v_i$ is maximized.

(Interval scheduling problem $v_i = 1 \ \forall i \in [n]$)

Assume: $f_1 \leq f_2 \leq \cdots \leq f_n$ (If not, sort)



Case 1: $5 \notin O$: An optimal solution $\{1,2,3,4\}$ (~~in terms of O~~)

Claim: O is an optimal for $\{1,2,3,4\}$

Case 2: ~~$5 \in O$~~

Pf: Say not. Say O' is optimal schedule for $\{1,2,3,4\}$ BUT $v(O') > v(O)$

BUT O' is also a valid schedule for $\{1, \ldots, 5\}$ $\Rightarrow$ O is not optimal.

Claim: Any valid schedule S for $\{1, \ldots, n-1\}$ is also a valid schedule for $[n]$

Case 2: $5 \in O$: $O \setminus \{5\}$ is optimal schedule for $\{1,2\}$

$$OPT(5) = \max \{ OPT(4) , v_5 + OPT(2) \}$$

$v(O)$

value of optimal schedule $\{1,2,3,4\}$

value of optimal schedule for $\{1,2\}$

Arbitrary n tasks : $OPT(j)$ = value of optimal schedule $\{1, \ldots, j\}$

O_j = an optimal schedule for $\{1, \ldots, j\}$

Case 1: $n \notin O_n$

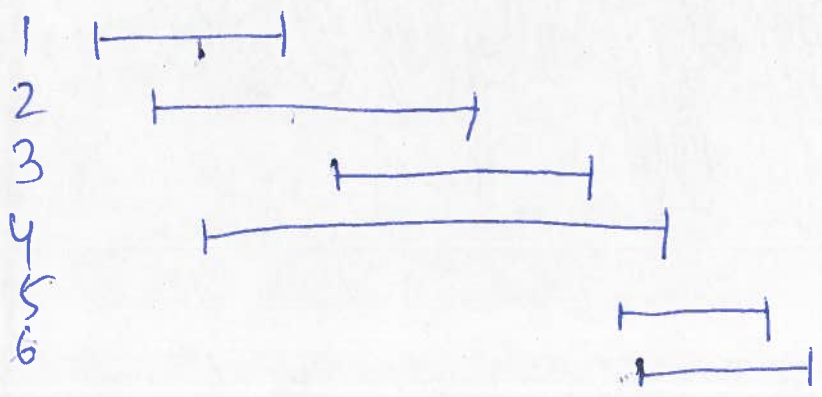
$OPT(n) = OPT(n-1)$

Case 2: $n \in O_n$ $\Rightarrow OPT(n) = \max \{ OPT(n-1), v_n + OPT(p(n)) \}$

$OPT(n) = v_n + OPT(p(n))$

Given $j \in [n]$, $p(j)$ = largest $i < j$ that doesn't conflict with j

$= 0$ otherwise



$p(1) = 0$
$p(2) = 0$
$p(3) = 1$
$p(4) = 0$
$p(5) = 3$
$p(6) = 3$