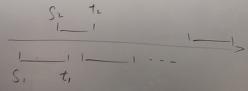


Weighted interval scheduling

加权值



foreach task i :

s_i : start time

f_i : finishing time

w_i : weight

S : set of all tasks

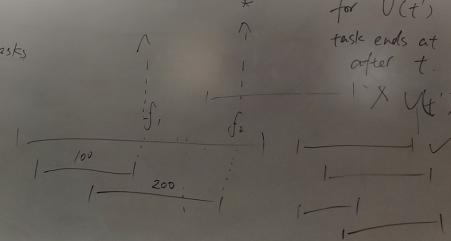
$S' \subseteq S$

$$\max \sum_{i \in S'} w_i$$

the optimal solution
 $V(t)$ stays the same
for $V(t')$ if the
task ends at t' starts
after t

$\times V(t')$

✓



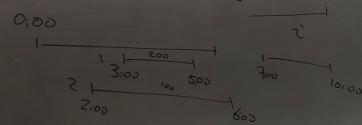
SHOT ON MI 8
AI DUAL CAMERA

" $V(f_i)$ "

|||

$V(i)$: the optimal value we can achieve at the finishing time of task i

$$V(5:00) \quad V(6:00) \quad V(5:00) \leq V(6:00)$$



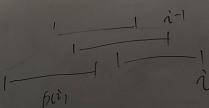
$p(i)$: task with $f(p(i))$ before S_i and is the maximum among all tasks

① : schedule task i .

$$V(i) = \max \{ V(p(i)) + w_i, V(i-1) \}$$

② : do not schedule task i .

$V(n)$



SHOT ON MI 8
AI DUAL CAMERA

WIS(n)

Compute $p(i)$ for each i .

initialize V as $1 \times n$ array

$V(i) = \underline{V(i)}$ for each i .

for $i = 2 \dots n$ do

$V(i) = \max \begin{cases} V(p(i)) + w(i) & \text{if } p(i) \text{ exists} \\ \underline{V(i)} \end{cases}$

end for.

return $(V(n))$

$O(n \log n)$

$p(i)$: the task finishes before S_i but
has the largest $f(p(i))$

$\xrightarrow{\text{from}}$ $\xrightarrow{\text{to}}$

$\xrightarrow{\text{task}}$

task 1 : (1) no $p(i)$
(2) no $i-1$

\Rightarrow take itself



SHOT ON MI 8
AI DUAL CAMERA

Initialize arrays V , T , S as $\{n\}$ arrays
 $V(i) = W(i)$, $T(i) = 0$, $S(i) = \text{True}$ for all i
for i in $2 \dots n$ do

$$V(i) = \max \begin{cases} V(p(i)) + W(i) & \text{if } p(i) \text{ exists} \\ V(i-1) \\ V(i) \end{cases}$$

if $V(i) = V(p(i)) + W(i)$: $S(i) = \text{True}$; $T(i) = p(i)$

else if $V(i) = V(i-1)$: $S(i) = \text{False}$; $T(i) = i-1$

Traceback(n, T, S)

$i = n$

while $i > 0$ do

if $S(i) = \text{True}$

output i

endif

$i = T(i)$

endwhile



SHOT ON MI 8
AI DUAL CAMERA