

COL 351 : ANALYSIS & DESIGN OF ALGORITHMS

LECTURE 15

GREEDY ALGORITHMS II : JOB SCHEDULING (CONTD.)

AUG 28, 2026

|

ROHIT VAISH

JOB SCHEDULING

JOB SCHEDULING

input : a set of n jobs with positive lengths $l_1, l_2, \dots, l_n$
and positive weights $w_1, w_2, \dots, w_n$

output: a job schedule (or sequence) that minimizes
the sum of weighted completion times $\sum_{j=1}^n w_j \cdot C_j$.

JOB SCHEDULING

input : a set of n jobs with positive lengths $l_1, l_2, \dots, l_n$
and positive weights $w_1, w_2, \dots, w_n$

output: a job schedule (or sequence) that minimizes
the sum of weighted completion times $\sum_{j=1}^n w_j \cdot C_j$.

Brute force : $O(n!)$ 😞

DEVELOPING A GREEDY ALGORITHM

DEVELOPING A GREEDY ALGORITHM

Recall : minimize $\sum_{j=1}^n w_j \cdot c_j$.

Approach : Use special cases to guide us to the algorithm

DEVELOPING A GREEDY ALGORITHM

Recall : minimize $\sum_{j=1}^n w_j \cdot C_j$.

Approach : Use special cases to guide us to the algorithm

(I) All lengths equal : schedule **heavier** or lighter job earlier ?
completion times are same for any schedule !

DEVELOPING A GREEDY ALGORITHM

Recall : minimize $\sum_{j=1}^n w_j \cdot C_j$.

Approach : Use special cases to guide us to the algorithm

- (I) All lengths equal : schedule **heavier** or lighter job earlier ?
completion times are same for any schedule !
- (II) All weights equal : schedule **shorter** or longer job earlier ?
each job negatively impacts all future jobs

RESOLVING CONFLICTING ADVICE

Q. What if heavier jobs are not shorter ($w_i > w_j$ but $l_i > l_j$)?

RESOLVING CONFLICTING ADVICE

Q. What if heavier jobs are not shorter ($w_i > w_j$ but $l_i > l_j$)?

Idea: Aggregate the weight and length parameters into a single "score"
(higher score $\Rightarrow$ scheduled earlier)

RESOLVING CONFLICTING ADVICE

Q. What if heavier jobs are not shorter ($w_i > w_j$ but $l_i > l_j$)?

Idea: Aggregate the weight and length parameters into a single "score"
(higher score $\Rightarrow$ scheduled earlier)

Consistent: Score function should be

- * increasing in weight
- * decreasing in length

RESOLVING CONFLICTING ADVICE

Q. What if heavier jobs are not shorter ($w_i > w_j$ but $l_i > l_j$)?

Idea: Aggregate the weight and length parameters into a single "score"
(higher score $\Rightarrow$ scheduled earlier)

Consistent: Score function should be

- * increasing in weight
- * decreasing in length

Proposal 1:

$$\text{score}(j) = w_j - l_j$$

Proposal 2:

$$\text{score}(j) = \frac{w_j}{l_j}$$

RESOLVING CONFLICTING ADVICE

Q. What if heavier jobs are not shorter ($w_i > w_j$ but $l_i > l_j$)?

Idea: Aggregate the weight and length parameters into a single "score"
(higher score $\Rightarrow$ scheduled earlier)

Consistent: Score function should be

- * increasing in weight
- * decreasing in length

Proposal 1:

$$\text{score}(j) = w_j - l_j$$

Proposal 2:

$$\text{score}(j) = \frac{w_j}{l_j}$$

different $\Rightarrow$ at least one
of these has to be wrong!

BREAKING A GREEDY ALGORITHM

BREAKING A GREEDY ALGORITHM

Proposal 1 : schedule in decreasing order of $w_j - l_j$

Proposal 2 : schedule in decreasing order of w_j / l_j .

BREAKING A GREEDY ALGORITHM

Proposal 1: schedule in decreasing order of $w_j - l_j$

Proposal 2: schedule in decreasing order of w_j / l_j .

E.g., $l_1 = 5$ $l_2 = 2$

$w_1 = 3$ $w_2 = 1$

larger
ratio

larger
difference

order by difference:

#1	↑ time
#2	

$$\sum_{j=1}^n w_j \cdot c_j = 1 \times 2 + 3 \times 7 = 23$$

order by ratio:

#2	↑ time
#1	

$$\sum_{j=1}^n w_j \cdot c_j = 3 \times 5 + 1 \times 7 = 22$$

Order-by-ratio does better!

CORRECTNESS OF ORDER-BY-RATIO

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : (by an exchange argument)

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : (by an exchange argument)

Two variants :

* by contradiction

* by massaging

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : (by an exchange argument)

Two variants :

* by **contradiction** : if greedy is suboptimal, then optimal can be improved

* by **massaging**

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : (by an exchange argument)

Two variants :

- * by **contradiction** : if greedy is suboptimal, then optimal can be improved
- * by **massaging** : convert any feasible solution into output of greedy algo while improving the objective

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : (by an exchange argument)

Two variants :

* **by contradiction** : if greedy is suboptimal, then optimal can be improved

* **by massaging** : convert any feasible solution into output of greedy algo while improving the objective

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Suppose, for contradiction, that order-by-ratios is not optimal on some input.

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Suppose, for contradiction, that order-by-ratios is not optimal on some input.

Let σ = greedy schedule and σ^* = optimal schedule.

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Suppose, for contradiction, that order-by-ratios is not optimal on some input.

Let σ = greedy schedule and σ^* = optimal schedule.

We will construct a schedule better than σ^* , contradicting purported optimality of σ^* .

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Two assumptions :

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Two assumptions :

① jobs are indexed in decreasing order of ratios

$$\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}.$$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Two assumptions :

① jobs are indexed in decreasing order of ratios

$$\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}.$$

② no ties between ratios

$$\frac{w_i}{l_i} \neq \frac{w_j}{l_j} \quad \text{whenever } i \neq j$$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Two assumptions :

without loss
of generality

① jobs are indexed in decreasing order of ratios

$$\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}.$$

will remove
this assumption
later

② no ties between ratios

$$\frac{w_i}{l_i} \neq \frac{w_j}{l_j} \quad \text{whenever } i \neq j$$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : By the two assumptions :

greedy schedule σ is $(1, 2, \dots, n)$ where

$$\frac{w_1}{l_1} > \frac{w_2}{l_2} > \dots > \frac{w_n}{l_n}$$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : By the two assumptions :

greedy schedule σ is $(1, 2, \dots, n)$ where

$$\frac{w_1}{l_1} > \frac{w_2}{l_2} > \dots > \frac{w_n}{l_n}$$

By contradiction assumption : $\sigma^* \neq \sigma$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

key observation

Since $\sigma^* \neq \sigma$, there must exist a consecutive inversion

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Key observation

Since $\sigma^* \neq \sigma$, there must exist a consecutive inversion

i.e., jobs i and j such that

i and j are consecutive in σ^* , and $i > j$.

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Key observation

Since $\sigma^* \neq \sigma$, there must exist a consecutive inversion

i.e., jobs i and j such that

i and j are consecutive in σ^* , and $i > j$.

[only schedule where indices always go up is $\sigma = (1, 2, \dots, n)$]

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Thought experiment

Suppose we **exchange** the order of i and j in σ^* without making any other changes.

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Thought experiment

Suppose we **exchange** the order of i and j in σ^* without making any other changes.

$i > j$

Some stuff
j
i
Other stuff

σ^*

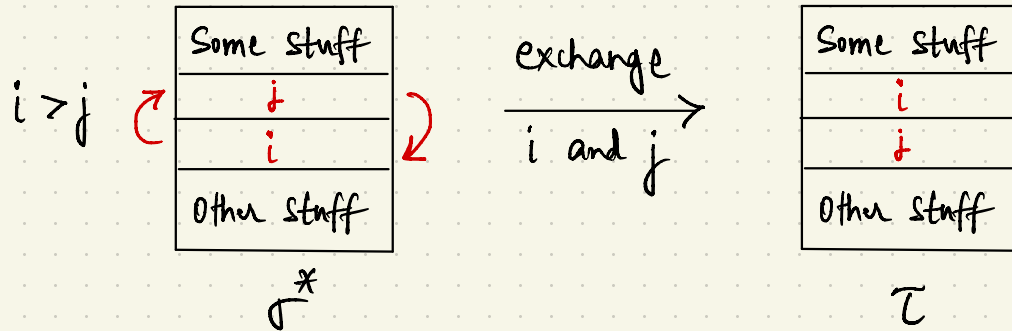
CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Thought experiment

Suppose we **exchange** the order of i and j in σ^* without making any other changes.



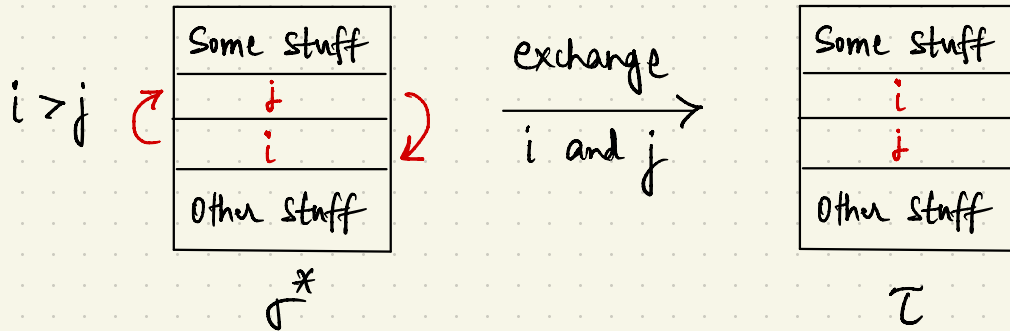
CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Thought experiment

Suppose we **exchange** the order of i and j in σ^* without making any other changes.

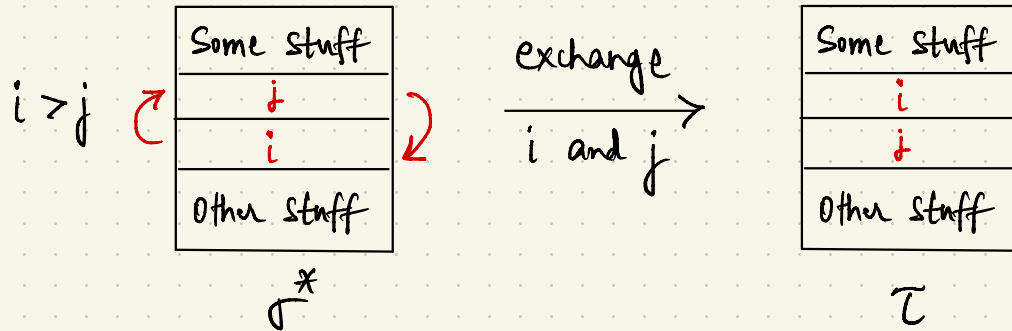


How do the completion times get affected?

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Completion time of job i ?
" " " j ?
" " " $k \neq i, j$?

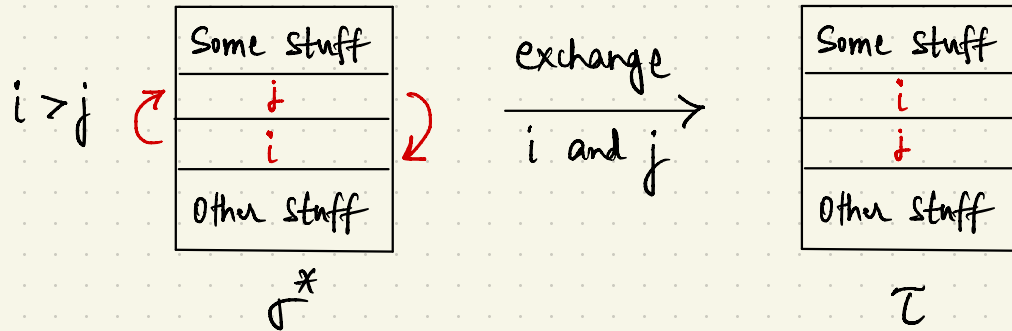


How do the completion times get affected?

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Completion time of job i increases by l_j
" " " j ?
" " " $k \neq i, j$?

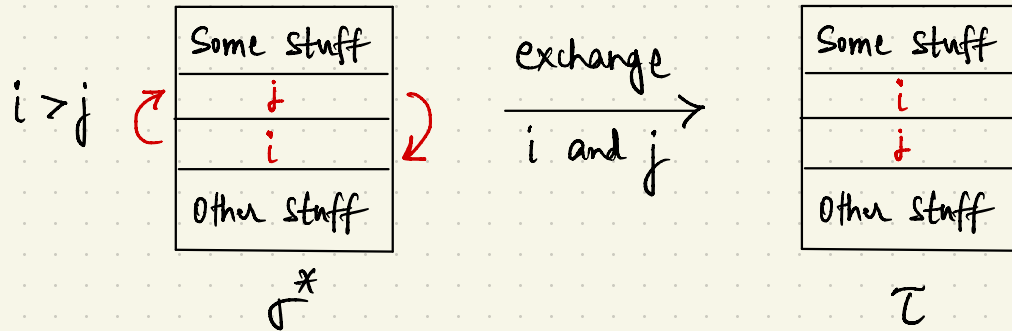


How do the completion times get affected?

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Completion time of job i increases by l_j
" " " j decreases by l_i
" " " $k \neq i, j$?

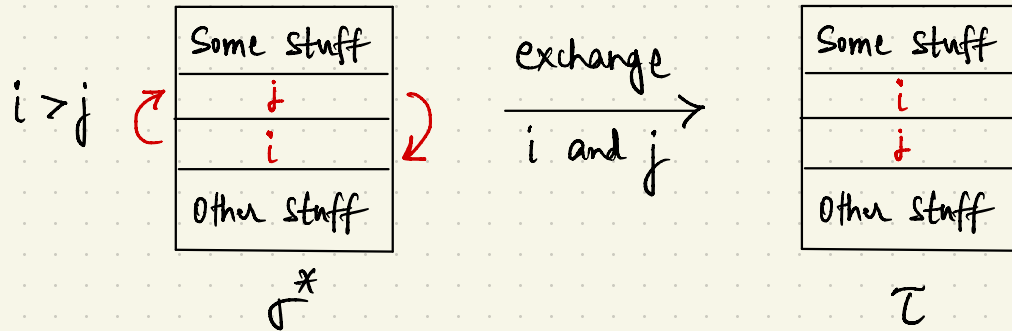


How do the completion times get affected?

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Completion time of job i increases by l_j
" " " j decreases by l_i
" " " $k \neq i, j$ stays the same



How do the completion times get affected?

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof :

Cost-benefit analysis

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

Cost-benefit analysis

cost of exchange $w_i \cdot l_j$ (C_i goes up by l_j)

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

Cost-benefit analysis

cost of exchange $w_i \cdot l_j$ (C_i goes up by l_j)

benefit of exchange $w_j \cdot l_i$ (C_j goes down by l_i)

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

Cost-benefit analysis

cost of exchange $w_i \cdot l_j$ (C_i goes up by l_j)

benefit of exchange $w_j \cdot l_i$ (C_j goes down by l_i)

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

objective for τ

objective for σ^*

effect of exchange

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

Recall : $i > j \Rightarrow \frac{w_i}{l_i} < \frac{w_j}{l_j}$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

Recall : $i > j \Rightarrow \frac{w_i}{l_i} < \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j < w_j \cdot l_i$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

Recall : $i > j \Rightarrow \frac{w_i}{l_i} < \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j < w_j \cdot l_i$
 $\Rightarrow \text{cost} < \text{benefit}$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

Recall : $i > j \Rightarrow \frac{w_i}{l_i} < \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j < w_j \cdot l_i$
 $\Rightarrow \text{cost} < \text{benefit}$
 $\Rightarrow \tau \text{ is better than } \sigma^*$

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

Recall : $i > j \Rightarrow \frac{w_i}{l_i} < \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j < w_j \cdot l_i$

$\Rightarrow \text{cost} < \text{benefit}$

$\Rightarrow \tau \text{ is better than } \sigma^*$

Contradiction!

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof :

$$\sum_{k=1}^n w_k \cdot C_k(\tau) = \sum_{k=1}^n w_k \cdot C_k(\sigma^*) + w_i l_j - w_j l_i$$

Recall : $i > j \Rightarrow \frac{w_i}{l_i} < \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j < w_j \cdot l_i$
 $\Rightarrow \text{cost} < \text{benefit}$
 $\Rightarrow \tau \text{ is better than } \sigma^*$
Contradiction!



CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Two variants :

* **by contradiction** : without ties

* **by massaging** : with ties

CORRECTNESS OF ORDER-BY-RATIO

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Two variants :

* by **contradiction** : without ties

* by **massaging** : with ties

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Recall the two assumptions :

① jobs are indexed in decreasing order of ratios

$$\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}.$$

② no ties between ratios

$$\frac{w_i}{l_i} \neq \frac{w_j}{l_j} \quad \text{whenever } i \neq j$$

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Recall the two assumptions :

① jobs are indexed in decreasing order of ratios

$$\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}.$$

~~②~~ no ties between ratios

$$\frac{w_i}{l_i} \neq \frac{w_j}{l_j} \quad \text{whenever } i \neq j$$

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Let σ = greedy schedule , τ = any other schedule
(not necessarily optimal)

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : Let σ = greedy schedule , τ = **any other** schedule
(not necessarily optimal)

Will show that σ is at least as good as τ
 $\Rightarrow$ greedy schedule is optimal.

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Greedy schedule σ is $(1, 2, \dots, n)$; thus, $\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}$.

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Greedy schedule σ is $(1, 2, \dots, n)$; thus, $\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}$.
Consider an **arbitrary** schedule τ .

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Greedy schedule σ is $(1, 2, \dots, n)$; thus, $\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}$.

Consider an **arbitrary** schedule τ .

If $\sigma = \tau$, we're done!

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Greedy schedule σ is $(1, 2, \dots, n)$; thus, $\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}$.

Consider an **arbitrary** schedule τ .

If $\sigma = \tau$, we're done!

Else τ contains a **consecutive inversion**.

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Greedy schedule σ is $(1, 2, \dots, n)$; thus, $\frac{w_1}{l_1} \geq \frac{w_2}{l_2} \geq \dots \geq \frac{w_n}{l_n}$.

Consider an **arbitrary** schedule τ .

If $\sigma = \tau$, we're done!

Else τ contains a **consecutive inversion**.

consecutive jobs i, j in τ
such that $i > j$.

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Consecutive invasion i, j
 $i > j \Rightarrow \frac{w_i}{l_i} \leq \frac{w_j}{l_j}$

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Consecutive invasion i, j
 $i > j \Rightarrow \frac{w_i}{l_i} \leq \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j \leq w_j \cdot l_i$

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Consecutive invasion i, j
 $i > j \Rightarrow \frac{w_i}{l_i} \leq \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j \leq w_j \cdot l_i$

Observe,

$$\sum_{k=1}^n w_k \cdot C_k(\tau^{\text{new}}) = \sum_{k=1}^n w_k \cdot C_k(\tau) + w_i l_j - w_j l_i$$

objective for
new schedule

objective for τ

effect of exchange

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/l_j minimizes the weighted completion time.

Proof : Consecutive invasion i, j
 $i > j \Rightarrow \frac{w_i}{l_i} \leq \frac{w_j}{l_j} \Rightarrow w_i \cdot l_j \leq w_j \cdot l_i$

Observe,

$$\sum_{k=1}^n w_k \cdot C_k(\tau^{\text{new}}) = \sum_{k=1}^n w_k \cdot C_k(\tau) + w_i l_j - w_j l_i$$

objective for new schedule objective for τ effect of exchange ≤ 0

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : τ^{new} is **no worse** than τ .

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : τ^{new} is **no worse** than τ .

After exchange of i and j , number of inversions between σ and τ **decrease** by 1.

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : τ^{new} is **no worse** than τ .

After exchange of i and j , number of inversions between σ and τ **decreases** by 1.

$\Rightarrow$ After at most nC_2 exchanges, can transform τ to σ .

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : τ^{new} is **no worse** than τ .

After exchange of i and j , number of inversions between σ and τ **decreases** by 1.

$\Rightarrow$ After at most nC_2 exchanges, can transform τ to σ .

$\Rightarrow$ σ is at least as good as **any other** schedule

HANDLING TIES

Theorem : Scheduling the jobs in decreasing order of ratios w_j/p_j minimizes the weighted completion time.

Proof : τ^{new} is **no worse** than τ .

After exchange of i and j , number of inversions between σ and τ **decreases** by 1.

$\Rightarrow$ After at most nC_2 exchanges, can transform τ to σ .

$\Rightarrow$ σ is at least as good as **any other** schedule

$\Rightarrow$ σ must be optimal