19.3
Scheduling Jobs to Minimize Average Waiting Time

The Problem

- n jobs $J_1, J_2, \ldots, J_n$.
- Each J_i has non-negative processing time p_i
- One server/machine/person available to process jobs.
- Schedule/order jobs to min. total or average waiting time
- Waiting time of J_i in schedule σ: sum of processing times of all jobs scheduled before J_i

	J_1	J_2	J_3	J_4	J_5	J_6
time	3	4	1	8	2	6

Example: schedule is $J_1, J_2, J_3, J_4, J_5, J_6$. Total waiting time is

$$0 + 3 + (3 + 4) + (3 + 4 + 1) + (3 + 4 + 1 + 8) + \ldots =$$

Optimal schedule: Shortest Job First. $J_3, J_5, J_1, J_2, J_6, J_4$.

The Problem

- n jobs $J_1, J_2, \ldots, J_n$.
- Each J_i has non-negative processing time p_i
- One server/machine/person available to process jobs.
- Schedule/order jobs to min. total or average waiting time
- Waiting time of J_i in schedule σ: sum of processing times of all jobs scheduled before J_i

	J_1	J_2	J_3	J_4	J_5	J_6
time	3	4	1	8	2	6

Example: schedule is $J_1, J_2, J_3, J_4, J_5, J_6$. Total waiting time is

$$0 + 3 + (3 + 4) + (3 + 4 + 1) + (3 + 4 + 1 + 8) + \ldots =$$

Optimal schedule: Shortest Job First. $J_3, J_5, J_1, J_2, J_6, J_4$.

The Problem

- n jobs $J_1, J_2, \ldots, J_n$.
- Each J_i has non-negative processing time p_i
- One server/machine/person available to process jobs.
- Schedule/order jobs to min. total or average waiting time
- Waiting time of J_i in schedule σ: sum of processing times of all jobs scheduled before J_i

	J_1	J_2	J_3	J_4	J_5	J_6
time	3	4	1	8	2	6

Example: schedule is $J_1, J_2, J_3, J_4, J_5, J_6$. Total waiting time is

$$0 + 3 + (3 + 4) + (3 + 4 + 1) + (3 + 4 + 1 + 8) + \ldots =$$

Optimal schedule: Shortest Job First. $J_3, J_5, J_1, J_2, J_6, J_4$.

The Problem

- n jobs $J_1, J_2, \ldots, J_n$.
- Each J_i has non-negative processing time p_i
- One server/machine/person available to process jobs.
- Schedule/order jobs to min. total or average waiting time
- Waiting time of J_i in schedule σ: sum of processing times of all jobs scheduled before J_i

	J_1	J_2	J_3	J_4	J_5	J_6
time	3	4	1	8	2	6

Example: schedule is $J_1, J_2, J_3, J_4, J_5, J_6$. Total waiting time is

$$0 + 3 + (3 + 4) + (3 + 4 + 1) + (3 + 4 + 1 + 8) + \ldots =$$

Optimal schedule: Shortest Job First. $J_3, J_5, J_1, J_2, J_6, J_4$.

Optimality of Shortest Job First (SJF)

Theorem 19.1.

Shortest Job First gives an optimum schedule for the problem of minimizing total waiting time.

Proof strategy: exchange argument

Assume without loss of generality that job sorted in increasing order of processing time and hence $p_1 \leq p_2 \leq \ldots \leq p_n$ and SJF order is $J_1, J_2, \ldots, J_n$.

Optimality of Shortest Job First (SJF)

Theorem 19.1.

Shortest Job First gives an optimum schedule for the problem of minimizing total waiting time.

Proof strategy: exchange argument

Assume without loss of generality that job sorted in increasing order of processing time and hence $p_1 \leq p_2 \leq \ldots \leq p_n$ and SJF order is $J_1, J_2, \ldots, J_n$.

Optimality of Shortest Job First (SJF)

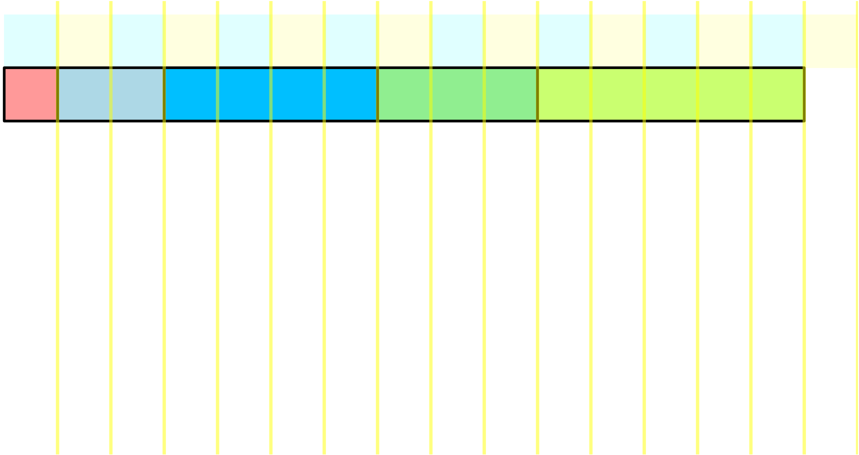
Theorem 19.1.

Shortest Job First gives an optimum schedule for the problem of minimizing total waiting time.

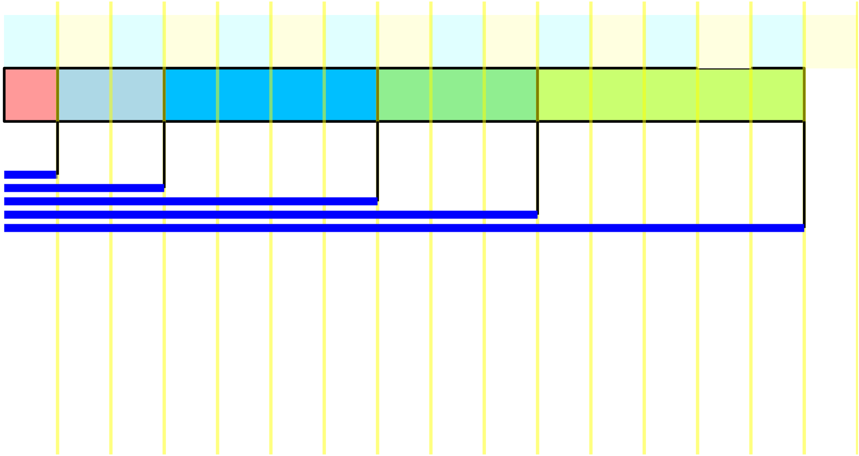
Proof strategy: exchange argument

Assume without loss of generality that job sorted in increasing order of processing time and hence $p_1 \leq p_2 \leq \ldots \leq p_n$ and SJF order is $J_1, J_2, \ldots, J_n$.

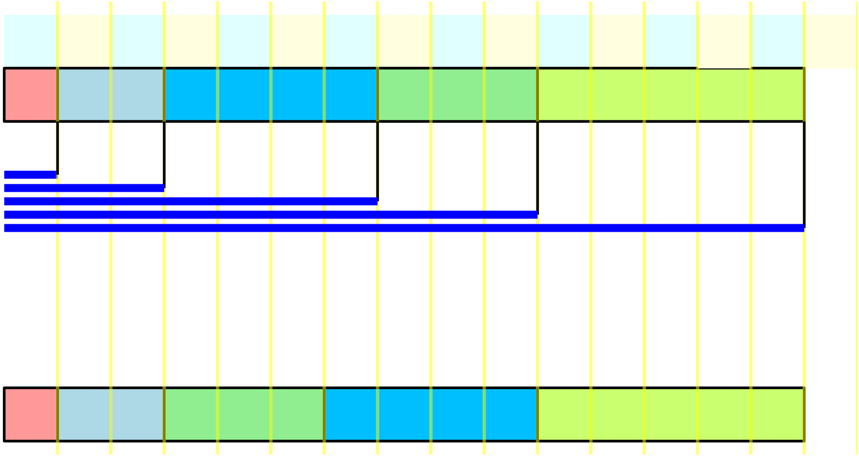
Optimality of SJF: Proof by picture



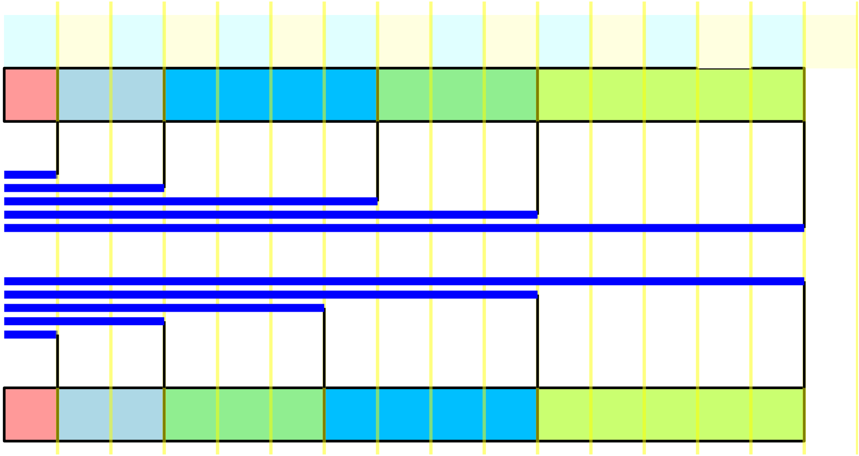
Optimality of SJF: Proof by picture



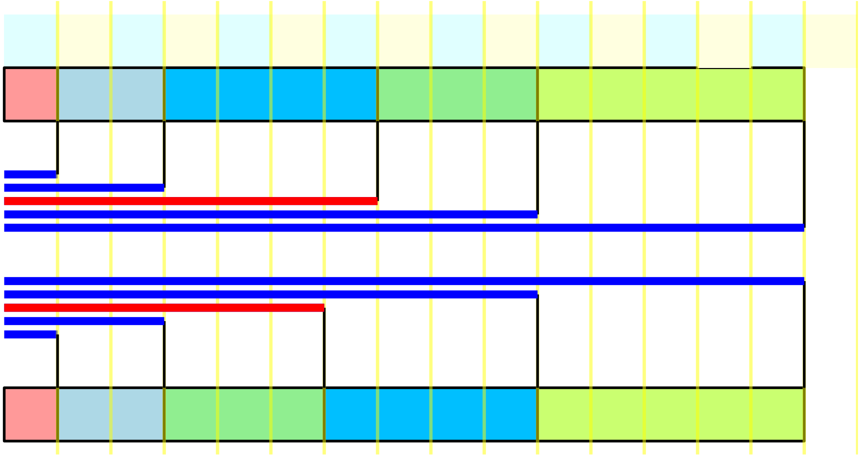
Optimality of SJF: Proof by picture



Optimality of SJF: Proof by picture



Optimality of SJF: Proof by picture



Inversions

Definition 19.2.

A schedule $J_{i_1}, J_{i_2}, \ldots, J_{i_n}$ has an inversion if there are jobs J_a and J_b such that S schedules J_a before J_b, but $p_a > p_b$.

Claim 19.3.

If a schedule has an inversion then there is an inversion between two adjacent scheduled jobs.

Proof: exercise.

Inversions

Definition 19.2.

A schedule $J_{i_1}, J_{i_2}, \ldots, J_{i_n}$ has an inversion if there are jobs J_a and J_b such that S schedules J_a before J_b, but $p_a > p_b$.

Claim 19.3.

If a schedule has an inversion then there is an inversion between two adjacent scheduled jobs.

Proof: exercise.

Proof of optimality of SJF

SJF = Shortest Job First

Recall SJF order is $J_1, J_2, \ldots, J_n$.

- Let $J_{i_1}, J_{i_2}, \ldots, J_{i_n}$ be an optimum schedule with fewest inversions.
- If schedule has no inversions then it is identical to SJF schedule and we are done.
- Otherwise there is an $1 \leq \ell < n$ such that $i_\ell > i_{\ell+1}$ since schedule has inversion among two adjacent scheduled jobs

Claim 19.4.

The schedule obtained from $J_{i_1}, J_{i_2}, \ldots, J_{i_n}$ by exchanging/swapping positions of jobs J_{i_ℓ} and $J_{i_{\ell+1}}$ is also optimal and has one fewer inversion.

Assuming claim we obtain a contradiction and hence optimum schedule with fewest inversions must be the SJF schedule.

Proof of optimality of SJF

SJF = Shortest Job First

Recall SJF order is $J_1, J_2, \ldots, J_n$.

- Let $J_{i_1}, J_{i_2}, \ldots, J_{i_n}$ be an optimum schedule with fewest inversions.
- If schedule has no inversions then it is identical to SJF schedule and we are done.
- Otherwise there is an $1 \leq \ell < n$ such that $i_\ell > i_{\ell+1}$ since schedule has inversion among two adjacent scheduled jobs

Claim 19.4.

The schedule obtained from $J_{i_1}, J_{i_2}, \ldots, J_{i_n}$ by exchanging/swapping positions of jobs J_{i_ℓ} and $J_{i_{\ell+1}}$ is also optimal and has one fewer inversion.

Assuming claim we obtain a contradiction and hence optimum schedule with fewest inversions must be the SJF schedule.

Exercise: A Weighted Version

- n jobs $J_1, J_2, \ldots, J_n$. J_i has non-negative processing time p_i and a non-negative weight w_i
- One server/machine/person available to process jobs.
- Schedule/order the jobs to minimize total or average waiting time
- Waiting time of J_i in schedule σ: sum of processing times of all jobs scheduled before J_i
- Goal: minimize total weighted waiting time.
- Formally, compute a permutation π that minimizes $\sum_{i=1}^{n}\left(\sum_{j=1}^{i-1} p_{\pi(j)}\right) w_{\pi(i)}$.

	J_1	J_2	J_3	J_4	J_5	J_6
time	3	4	1	8	2	6
weight	10	5	2	100	1	1

THE END

...

(for now)