Lateness Minimization

Bakh Khoussainov

Scheduling to minimize lateness.

We are given n requests:

$1, 2, 3, \ldots, n$.

The starting time to satisfy requests is s.

Each request i has

(1) Deadline d_i

(2) Continous time interval t_i.

Goal: Want to satisfy each request And minimize the maximum lateness.

We need to assign each request i an interval $[s(i), f(i)]$ such that

$$f(i) = s(i) + t_i$$

It is us who determine starting times $s(i)$.

Request i is late if

$$f(i) > d_i.$$

The difference

$$l_i = f(i) - d_i$$

is the lateness of i.

If $f(i) \leq d(i)$, then

set $l_i = 0$.

We want to minimize the maximum lateness

$$L = \max\{l_1, l_2, \dots, l_n\}.$$

We order the requests in order of their deadlines.

So, we assume that

$$d_1 \leq d_2 \leq \dots \leq d_n$$

Request 1 starts at time $s(1) = s$, and finishes at time $f(1) = s(1) + t_1$.

Request 2 starts at $s(2) = f(1)$, and finishes at $f(2) = s(2) + t_2$, and so on.

Here is the algorithm

Initially, set $f = s$.

For $i = 1, 2, \ldots, n$

assign job i time interval starting at $s(i) = f$, and finishing at $f(i) = s(i) + t_i$

Set $f = f(i)$

Return $[s(i), f(i)]$, $i = 1, 2, \ldots, n$

A scheduling of requests is optimal if its lateness is minimal (among all schedules).

We want to show that our algorithm produces an optimal solution.

At time t ~~for~~ a schedule is idle if at time t no job is being done.

Note that every optimal schedule has no idle time.

Property 1. The schedule A produced by our algorithm has no idle time.

Let A' be a schedule. We say that A' has inversion if some job i is scheduled before some job j yet $d_j < d_i$.

Fact 1. All schedules with no inversion and no idle time have the same maximum lateness.

Indeed, let A_1 and A_2 be two such schedules.

These schedules can only differ in the order in which jobs with identical deadlines are scheduled.

Consider a deadline d.
In A_1 and A_2, the last jobs with deadline d have the greatest AND the same lateness among these jobs. This proves the fact.

Goal: Take an optimal schedule O, and tranform it to an optimal schedule with no inversions.

This will prove the correctness of our algorithm.

Fact 2. If a schedule O has an inversion then there are two jobs i, j such that

(a) $d_j < d_i$

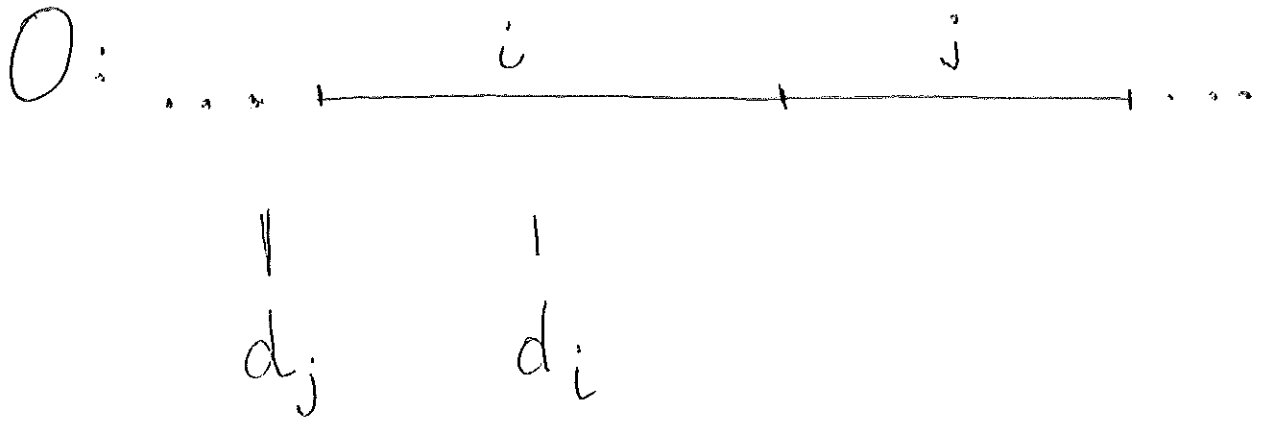
(b) j comes right after i.

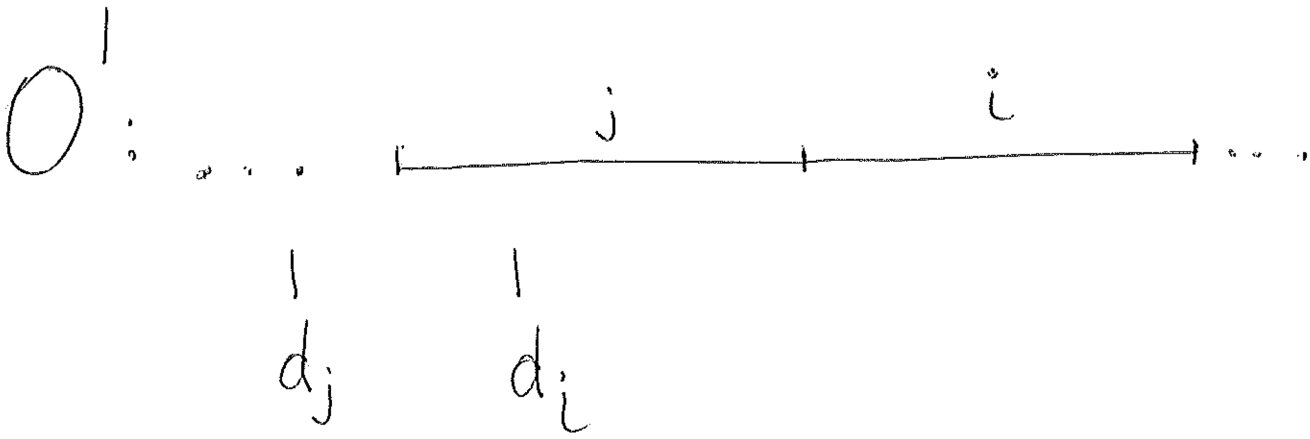
Take i, j from Fact 2. Swap them. This gives us a new schedule O'. So we have the following.

Schedule O' has one less inversion than O has.

Fact 3. O' has a maximum lateness no larger than that of O.

Picture:





$f_O(j) = f_{O'}(i)$. So, all jobs apart from i, j have the same latenesses in O and O'.

The job j finishes in O' earliear than j finishes in O.

We compare latenesses of i in O and O'.

In O' i finishes at $f_O(j)$.

So lateness of i in O' is

$f_O(j) - d_i$. But $d_j < d_i$.

So $f_O(j) - d_i < f_O(j) - d_j$.

So, lateness of i in O' is less than lateness of i in O.

Thus, given an optimal schedule O, by reducing its inversions step by step, we can transform O to O' such that

(1) O' has no idle time,

(2) O' has no inversion,

(3) O' is optimal.

All of the above show that the schedule $\underline{A}$ returned by our algorithm is optimal.