

# Schedule to Minimize Lateness

Samyak, Sangeeta, and Saurabh

University of Delhi

August 30, 2018

## 1 Problem

Given a single resource and a set of  $n$  requests, where each request  $i$  has a runtime  $t_i$  and deadline  $d_i$ , We want to schedule all requests and our objective is to minimize the maximum lateness of that schedule.

Stated more formally, we assign each request  $i$  a starting time  $s(i)$  which finishes by  $f(i) = s(i) + t_i$ . We say that a request is late if it misses its deadline  $d_i$  and its lateness is defined by  $l_i = f(i) - d_i$ . Our objective is to schedule all requests, using non-overlapping intervals so as to minimize the maximum lateness,  $L = \max_i l_i$ .

## 2 Algorithm

### 2.1 Proposed Algorithm

We propose a greedy algorithm - which we claim always gives an optimal solution - which simply schedules the request with the smallest/minimum deadline  $d_i$ , that is earliest deadline first.

We sort the requests in increasing order of their deadlines, that is

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

and we simply sort the requests in this order. This is the algorithm we get.

---

**Algorithm 1:** schedule with minimum lateness

---

**Input:** Requests in increasing order of deadline  $d_1 \leq \dots \leq d_i \leq \dots \leq d_n$

Initially,  $f = s$ ;

**begin**

    Assign request  $i$  to the time interval from  $s(i) = f$  to  $f(i) = f + t_i$   
    Let  $f = f + t_i$

**end**

Return the set of scheduled intervals  $[s(i), f(i)]$  for  $i = 1, \dots, n$

---

### 2.2 Proving Optimality

**Claim 1.** *There is an optimal solution with no idle time*

**Definition 1** (Inversion). We say that a schedule  $A'$  has an inversion if a request  $i$  with deadline  $d_i$  is scheduled before another request  $j$  with earlier deadline  $d_j < d_i$ .

A thing to be noted here is that if there are requests with identical deadlines (which will be scheduled together by our algorithm) then there can be many different schedules with no inversions since the two intervals with same deadline don't form an inversion and hence can be swapped.

**Claim 2.** *All schedules with no inversions and no idle time have the same maximum lateness*

*Proof.* If two different schedules have no inversions, then they might not produce exactly the same order of requests, but can only differ in the order of in which requests with identical deadlines are scheduled.

Consider such a deadline  $d$ . In both schedules, the requests with deadline  $d$  are scheduled consecutively. Among the requests with deadline  $d$ , the last one has the maximum lateness, and this lateness doesn't depend on the order of the requests.  $\square$

**Claim 3.** *There is an optimal solution with no inversions.*

*Proof.* Let  $O$  be any optimal solution with no idle time. If we are able to show that  $O$  has no inversion then we are done. What we will do is that we will take  $O$  and convert it into a schedule with no inversions without increasing its maximum lateness. Thus the resulting solution will be also be optimal and without any inversions.

Let us assume that  $O$  has an inversion, removing one inversion from  $O$  gives us  $O_1$ , which has one less inversion but cost remaining the same and  $O_2$  with one less inversion than  $O_1$  and so on leading to  $O^*$  with no inversion.

$$O_1 \Rightarrow O_2 \Rightarrow \dots \Rightarrow O^*$$

If there is an inversion in  $O$  then it is consecutive, i.e if  $O$  has an inversion, then there are requests  $i$  and  $j$  where  $d_i > d_j$  but  $i$  is scheduled **immediately** before  $j$ .

This is easy to show. If  $i$  and  $j$  are separated by  $n$  requests in between namely  $r_1, r_2, r_3, \dots, r_n$ , where  $r_1 = i$  and  $r_n = j$ . If  $d_{r_1} > d_{r_2}$ , then we got a pair of consecutive intervals that are in inversion, and we don't need to search for consecutive inversion any further. Else if  $d_j < d_{r_1} < d_{r_2}$  : repeat with  $(i = r_2, j)$  instead of  $(r_1, j)$ .

if  $d_{r_1} < d_{r_2} < d_{r_3} \dots < d_{r_{n-1}}$  i.e. all intervals in between are not inverted, then we will reach a point where  $d_j = d_{r_n} < d_{r_{n-1}}$ , hence we get a pair of consecutive intervals  $(r_n - 1, j)$  that are inverted.

After swapping we have  $O^*$  where  $j$  is scheduled before  $i$  and  $d_j < d_i$ , since  $d_j < d_i$  then  $T - d_j < T - d_i$  i.e. lateness, where  $T$  is the time when every request is finished. We get,  $T - d_j < T - d_i \leq \max \text{lateness of schedule}$ .  $\square$

*Optimality of our Schedule.* Since there exists an optimal solution with no inversions, and all schedules with no inversions have the same maximum lateness, hence the solution obtained by our greedy algorithm is optimal.  $\square$