CIS 575. Introduction to Algorithm Analysis
Material for April 8, 2024

Greedy Scheduling to Minimize Waiting Time

1 Minimizing Waiting Time

As a simple example of a problem that can be solved by a greedy algorithm, we shall consider the **scheduling problem** determined by the setting

- there is a single *server*, who serves one customer at a time
- there are several customers, all already present in the shop; each customer j has a job which takes $T(j)$ time units (the "duration" of the job).

Our *goal* is to **minimize the total waiting time**, that is the sum of the waiting times "suffered" by each customer until he/she is being served.

> Many variants exist on this setting, such as: more than one server, customers arriving along the way, some customers having higher priority than others, etc. Each variant results in a different problem and often a different kind of solution.

We shall propose a greedy strategy: **schedule the shortest jobs first**. That is, if $T(j) < T(i)$ then customer j will be served ahead of customer i, but if $T(j) = T(i)$ then customers i and j may be served in any order.

This strategy turns out to be optimal! To see that, assume that someone claims to have an optimal schedule S that does *not* follow our strategy. It is not hard to see that then there must be two customers i and j with $T(j) > T(i)$ such that schedule S serves customer j just before customer i. Graphically, we have

```
       jobs before         customer j          customer i    jobs after
S:   --------------------|--------------------|------------|-------------
```

But now we can construct a schedule S' by *swapping* customers i and j:

```
       jobs before        customer i       customer j         jobs after
S':  --------------------|------------|--------------------|-------------
```

What is the impact of choosing S' rather than S?

- other customers than i or j are not affected
- customer j gets delayed by $T(i)$ time units

- but customer i is served $T(j)$ time units earlier.

Since our assumption was that $T(j) > T(i)$, this shows that S' is better than S (given that the time of customer j is not considered more valuable than the time of customer i). This *contradicts* the schedule S being optimal, and shows that

> a schedule that does *not* follow our strategy can *not* be optimal.

Since an optimal schedule obviously exists, that schedule must be one that follows our strategy. (While there could be several schedules that follow our strategy, in that jobs with the same duration may be scheduled in any order, it is not hard to see that all such schedules have the same total waiting time and are thus all optimal.)