Lecture 12 – Review + Greedy Algorithms

Reading: KT Section 4.1 and 4.2

Partial content of these slides have been obtained from the official lecture slides that accompany the textbook. A complete set of slides can be found at: http://www.cs.princeton.edu/~wayne/kleinberg-tardos/

Quick review on analysis of algorithms

Let’s look at DFS again!

DFS(s):
Initialize S to be a stack with one element s
While S is not empty
    Take a node u from S
    If Explored[u] = false then
        Set Explored[u] = true
        For each edge (u, v) incident to u
            Add v to the stack S
    Endfor
Endif
Endwhile
Scheduling to minimize lateness

• Setting:
  • A single resource
  • Each job j requires \( t_j \) time units, and a deadline of \( d_j \)
  • \( j \) starts at \( s_j \), and is done at \( f_j = s_j + t_j \)
  • The lateness of a job \( j \) is \( l_j = f_j - d_j \)

• Goal:
  • Decide on the start time of each job to minimize the maximum lateness (\( L \))
  • \( L = \max_j l_j \)
Greedy algorithms to minimize lateness

- How would you order the jobs in the schedule?

Earliest deadline first

```
EARLIEST-DEADLINE-FIRST (n, l_1, l_2, ..., l_n, d_1, d_2, ..., d_n)

SORT n jobs so that d_1 ≤ d_2 ≤ ... ≤ d_n.

\( t \leftarrow 0 \)

FOR \( j = 1 \) TO \( n \)

Assign job \( j \) to interval \([t, t + t_j]\).

\( s_j \leftarrow t \); \( f_j \leftarrow t + t_j \)

\( t \leftarrow t + t_j \)

RETURN intervals \([s_1, f_1], [s_2, f_2], ..., [s_n, f_n]\).
```

max lateness = 1

```
<table>
<thead>
<tr>
<th>d_1 = 6</th>
<th>d_2 = 8</th>
<th>d_3 = 9</th>
<th>d_4 = 9</th>
<th>d_5 = 14</th>
<th>d_6 = 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>
```
Correctness – Exchange approach

What is it?
- We start with an optimal solution $O$, and turn it into the greedy solution $A$

For the earliest deadline first algorithm...
- There exists an optimal schedule with no idle time
- The earliest-deadline-first schedule has no idle time
- The earliest-deadline-first schedule has no inversions
- All schedules with no inversions and no idle time have the same maximum lateness
- If a schedule (with no idle time) has an inversion, it has one with a pair of inverted jobs scheduled consecutively
- Swapping two adjacent, inverted jobs reduces the number of inversions by one and does not increase the max lateness

There is an optimal schedule that has no inversions and no idle time.
The schedule $A$ produced by the greedy algorithm has optimal maximum lateness $L$. 