NPTEL MOOC,JAN-FEB 2015
Week 6, Module 4

DESIGN AND ANALYSIS OF ALGORITHMS

Greedy algorithms: Minimizing lateness

MADHAVAN MUKUND, CHENNAI MATHEMATICAL INSTITUTE
http://www.cmi.ac.in/~madhavan

Minimizing lateness

- A single resource, n request to use this resource
- Request i requires time t(i) to complete and has a deadline d(i)
- All requests will be scheduled
  - Request j starts at s(j) and ends at f(j) = s(j) + t(j)
  - If f(j) > d(j), request j is late by l(j) = d(j) - f(j)

Goal: Minimize maximum lateness

- Minimize the maximum value of l(j) over all j

Greedy strategies

Greedy Strategy 1

- Choose jobs in increasing order of length — t(j)

Counterexample

- Two jobs
  - t(1) = 1, d(1) = 100
  - t(2) = 10, d(2) = 10

Greedy strategies

Greedy Strategy 2

* Choose job with smaller slack times, $d(j) - t(j)$, first

Counterexample

* Two jobs
    * $t(1) = 1$, $d(1) = 2$
    * $t(2) = 10$, $d(2) = 10$

Greedy strategies

Greedy Strategy 3

- Choose job with earliest deadline d(j) first
- This strategy is correct
- How do we prove it?

Correctness

- Assume all jobs are sorted by deadline
  - Renumber so that $d(1) \leq d(2) \leq \ldots \leq d(n)$
- Schedule is simple: 1, 2, …, n
  - Job 1 starts at $s(1) = 0$ and ends at $f(1) = t(1)$
  - Job 2 starts at $s(2) = f(1)$ and ends at $f(2) = s(2)+t(2)$
  - ...

Correctness ...

- Our schedule has no gaps — idle time
  - The resource is continuously in use from s(1) to f(n)

Claim:
There is an optimum schedule with no idle time

- Shifting jobs earlier to remove idle time can only reduce lateness

Exchange argument

- Suppose O is some other optimal schedule
- Transform O step by step until it becomes identical to the schedule A found by the greedy algorithm

Inversions

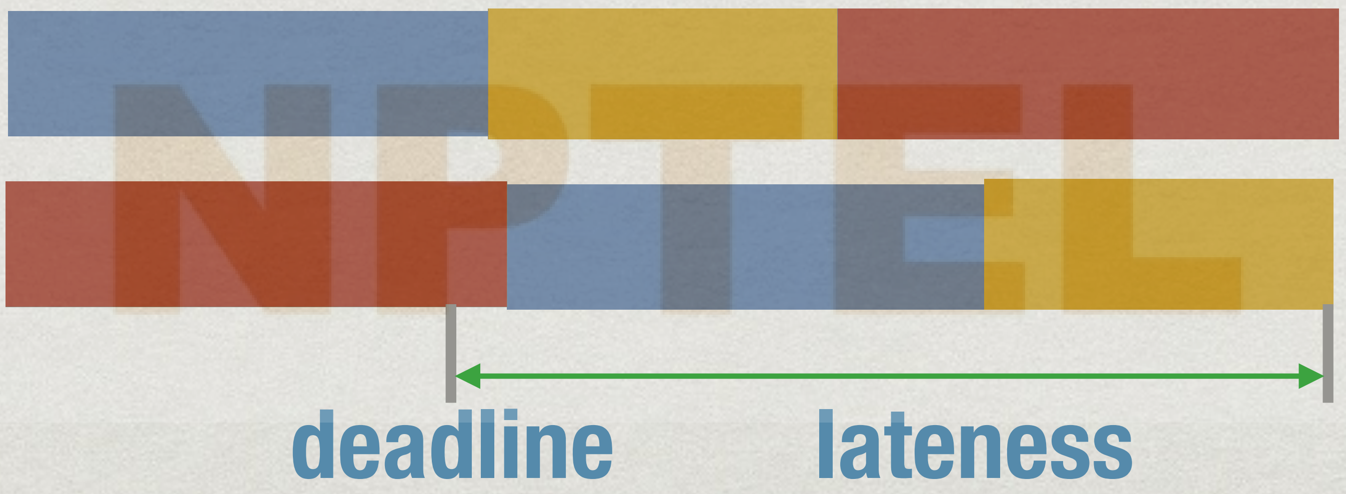
- A schedule O has an inversion if i appears before j in O but d(j) < d(i)
- By construction, the greedy solution has no inversions

Inversions ...

Claim: Any two schedules with no inversions and no idle time produce the same lateness

- No inversions, no idle time means the only difference can be in order of jobs with same deadline

- Any reordering of jobs with the same deadline produces the same lateness



Optimality …

Claim: There is an optimal schedule with no inversions and no idle time.

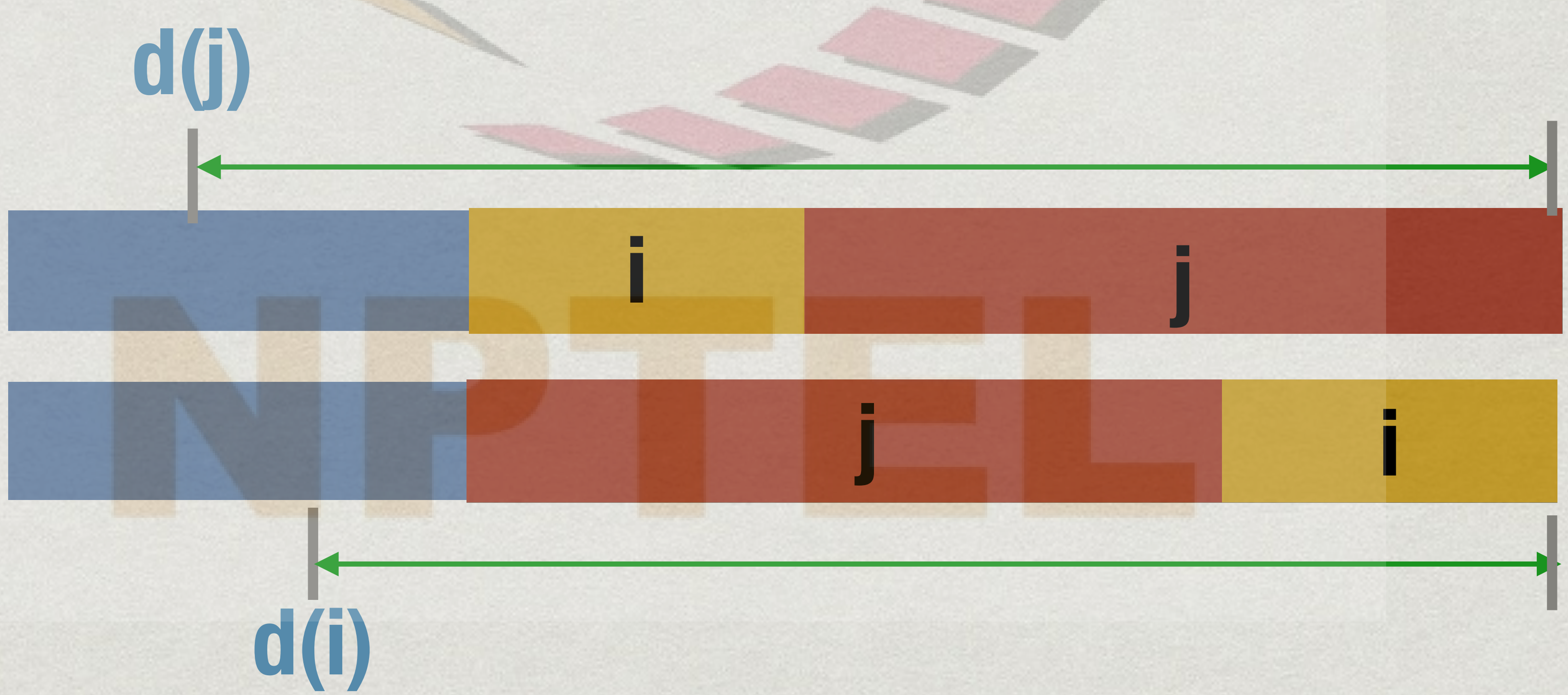
- Let O be an optimal solution with no idle time
- (A) If O has an inversion, then there is a pair of jobs i and j such that j is scheduled immediately after i and $d(j) < d(i)$
    - Find the first point where deadline decreases

Optimality …

- (B) After swapping i and j we get a solution with one less inversion
  - Obvious
- (C) After swapping i and j we get a solution whose maximum lateness is no larger than that of O
  - Not so obvious

Optimality …

- (C) After swapping i and j we get a solution whose maximum lateness is no larger than that of O
    - Recall that d(j) < d(i)
    - Lateness of i after swap cannot be more than lateness of j before swap



Optimality …

Claim: There is an optimal schedule with no inversions and no idle time.

- From (C) we can remove each adjacent inversion without increasing lateness
- At most n(n-1)/2 inversions in O to begin with
- Repeatedly remove adjacent inversions to get an optimal schedule with no inversions, no idle time

Implementation, complexity

- Sort jobs by deadline — O(n log n)
- Read off schedule in same order — O(n)
- Overall — O(n log n)