JOB SEQUENCING WITH DEADLINES

The problem is stated as below.

- There are n jobs to be processed on a machine.
- Each job i has a deadline $d_i \geq 0$ and profit $p_i \geq 0$.
- Pi is earned iff the job is completed by its deadline.
- The job is completed if it is processed on a machine for unit time.
- Only one machine is available for processing jobs.
- Only one job is processed at a time on the machine.

Divide and Conquer
Greedy Method – Job sequencing problem

Dr. K.RAGHAVA RAO, Professor in CSE, KL University,
krraocse@gmail.com, http://mcadaa.blog.com

JOB SEQUENCING WITH DEADLINES (Contd..)

- A feasible solution is a subset of jobs J such that each job is completed by its deadline.
- An optimal solution is a feasible solution with maximum profit value.

Example : Let n = 4, $(p_1,p_2,p_3,p_4) = (100,10,15,27)$, $(d_1,d_2,d_3,d_4) = (2,1,2,1)$

JOB SEQUENCING WITH DEADLINES (Contd..)

Sr.No.	**Feasible Solution**	**Processing Sequence**	**Profit value**
(i)	(1,2)	(2,1)	110
(ii)	(1,3)	(1,3) or (3,1)	115
(iii)	(1,4)	(4,1)	127 is the optimal one
(iv)	(2,3)	(2,3)	25
(v)	(3,4)	(4,3)	42
(vi)	(1)	(1)	100
(vii)	(2)	(2)	10
(viii)	(3)	(3)	15
(ix)	(4)	(4)	27

Greedy job scheduling example Number of jobs n=5. Time slots 1, 2, 3. (Slot 0 is sentinel)

Job (i)	Profit	Deadline	Profit/Time
A	100	2	100
B	19	1	19
C	27	2	27
D	25	1	25

Greedy job scheduling algorithm

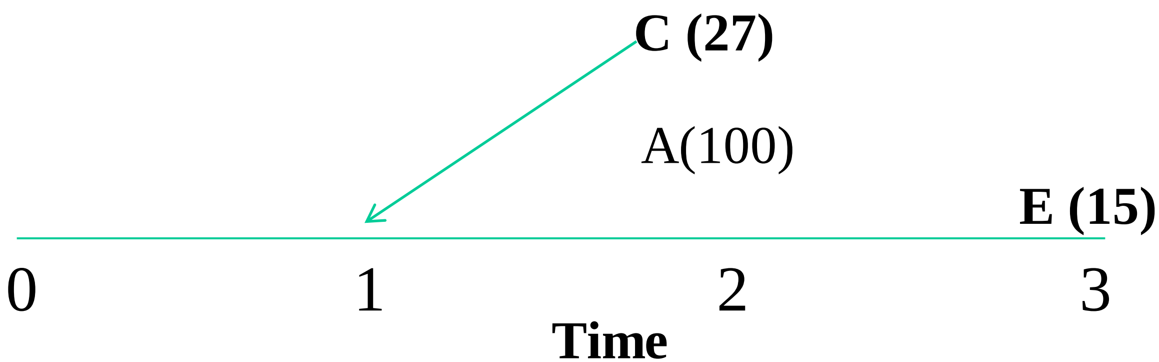
<u>Sort jobs by profit/time ratio (slope or derivative):</u>

– **A (deadline 2), C (2), D (1), B (1), E (3)**

Place each job at <u>latest time that meets its deadline</u>

Nothing is gained by scheduling it earlier, and scheduling it earlier could prevent another more profitable job from being done

Solution is {C, A, E} with profit of 142



0Time D, B infeasible

- –

Job sequencing with deadlines

- Problem: n jobs, S={1, 2, …, n}, each job i has a deadline $d_i \geq 0$ and a profit $p_i \geq 0$. We need one unit of time to process each job and we can do at most one job each time. We can earn the profit p_i if job i is completed by its deadline.

i	1	2	3	4	5
p_i	20	15	10	5	1
d_i	2	2	1	3	3

The optimal solution = {1, 2, 4}.

The total profit = 20 + 15 + 5 = 40.

Algorithm:

Step 1: Sort p_i into nonincreasing order. After sorting $p_1 \geq p_2 \geq p_3 \geq \ldots \geq p_i$.

Step 2: Add the next job i to the solution set if i can be completed by its deadline. Assign i to time slot [r-1, r], where r is the largest integer such that $1 \leq r \leq d_i$ and [r-1, r] is free.

Step 3: Stop if all jobs are examined. Otherwise, go to step 2.

Time complexity: $O(n^2)$

e.g.

i	p_i	d_i	
1	20	2	assign to [1, 2]
2	15	2	assign to [0, 1]
3	10	1	reject
4	5	3	assign to [2, 3]
5	1	3	reject

solution = {1, 2, 4}

total profit = 20 + 15 + 5 = 40

GREEDY ALGORITHM TO OBTAIN AN OPTIMAL SOLUTION

- Consider the jobs in the non increasing order of profits subject to the constraint that the resulting job sequence J is a feasible solution.
- In the example considered before, the non-increasing profit vector is

$$(100 \quad 27 \quad 15 \quad 10) \qquad (2 \quad 1 \quad 2 \quad 1)$$

$$p_1 \quad p_4 \quad p_3 \quad p_2 \qquad d_1 \quad d_4 \quad d_3 \quad d_2$$

GREEDY ALGORITHM TO OBTAIN AN OPTIMAL SOLUTION (Contd..)

J = { 1} is a feasible one

J = { 1, 4} is a feasible one with processing sequence (4,1)

J = { 1, 3, 4} is not feasible

J = { 1, 2, 4} is not feasible

J = { 1, 4} is optimal

GREEDY ALGORITHM FOR JOB SEQUENSING WITH DEADLINE

High level description of job sequencing algorithm

```
Procedure greedy job (D, J, n)
// J is the set of n jobs to be completed  by their deadlines
{
J:={1};
    for i:=2 to n do
    {
        if  (all jobs in J U{i} can be completed by their deadlines)
       then J:= ← J U {i};
    }
}
```

GREEDY ALGORITHM FOR SEQUENCING UNIT TIME JOBS

Greedy algorithm for sequencing unit time jobs with deadlines and profits

Procedure JS(d,j,n)

// d(i) ≥ 1, 1≤ i ≤ n are the deadlines, n ≥1. the jobs are ordered such that

// $p_1 \geq p_2 \geq \ldots\ldots \geq p_n$ .J[i] is the ith job in the optimal solution , i ≤ i ≤ k. Also, at termination d[J[i]]≤d [J[i+1]] ,1 ≤ i ≤ k

{

d[0]:=J[0]:=0; //initialize and J(0) is a fictious job with d(0) = 0 //

J[1]:=1; //include job 1

K:=1; // job one is inserted into J //

for i :=2 to n do // consider jobs in non increasing order of pi //

r:=k;

While ((d[J[r]]>d[i]) and (d[J[r]]#r)) do r:=r-1;

If ((d[J[r] ≤ d[i]) and d[i]>r)) then { //insert i into J[]

For q:=k to (r+1) step-1 do j[q+1]:=j[q];

J[r+1]:=i; k:=k+1;

} } return k;

}

Exercise

- Let n=5, (p1,….,p5)=(20,15,10,5,1) and (d1,……,d5) =(2,2,1,3,3). Using the above feasibility rule find optimal solution.

- Find optimal solution n=7, (p1,p2,p3..p7)=(3,5,20,18,1,6,30) and (d1,d2,d3….d7)=(1,3,4,3,2,1,2)