Scheduling

Notes 3

Moore's Algorithm

- Moore's Algorithm is used to minimize the number of tardy jobs (n_T).

- Step 1: Order the jobs according to EDD.
- Step 2: Compute the tardiness for each job in the EDD sequence.
- Set N_T=0, and let k be the first position containing a tardy job. If no
- job is tardy go to step 4.
- Step 3: Find the job with the largest processing time in positions 1
- to k. Remove job i* from the sequence, set NT=NT+1, and repeat
- Step2.
- Step 4: Place the removed N_T jobs in any order at the end of the
- sequence.

Example
Solve the problem according to Moore's Algorithm

Job	1	2	3	4	5	6
di	15	6	9	23	20	30
Pi	10	3	4	8	10	6

Order the jobs according to EDD

Job	Pi	Ci	di	Ti
2	3	3	6	0
3	4	7	9	0
1	10	17	15	2
5	10	27	20	7
4	8	35	23	12
6	6	41	30	11

Number of Tardy jobs : 4

The first tardy job is job 1. In (2,3,1) group, job 1 has the maximum processing time. We remove job 1 from the sequence and order again.

Job	Pi	Ci	di	Ti
2	3	3	6	0
3	4	7	9	0
5	10	17	20	0
4	8	25	23	2
6	6	31	30	1

The first tardy job is job 4. In (2,3,5,4) group, job 5 has the maximum processing time. We remove job 5 from the sequence and order again.

Job	Pi	Ci	di	Ti
2	3	3	6	0
3	4	7	9	0
4	8	15	23	0
6	6	21	30	0

Optimal order is 2- 3- 4- 6- 1- 5 or 2- 3- 4- 6- 5- 1

Job	Pi	Ci	di	Ti
2	3	3	6	0
3	4	7	9	0
4	8	15	23	0
6	6	21	30	0
1	10	31	15	16
5	10	41	20	21

Number of tardy jobs : 2

Lawler’s Algorithm

- Lawler’s algorithm is used in a single machine problem with precedence constraints.
- The algorithm first assigns a job to the last position, then a job to the position next to last, and so on.
- Due to precedence constraints, not all the jobs are candidates for a position.

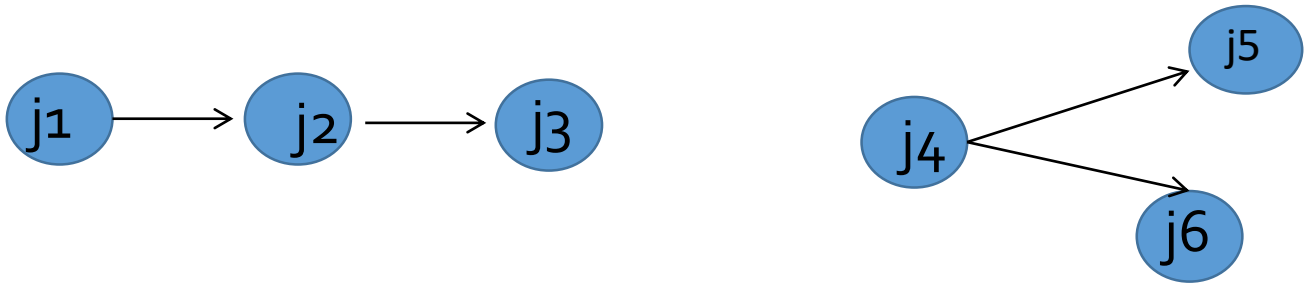
Lawler’s Algorithm

- 1|prec|fmax is the most used presentation.
- The algorithm is applicable for single machine problems with the objective of minimizing
  - Makespan
  - Maximum lateness
  - Maximum tardiness

Lawler's Algorithm

- Find the completion time of the last job.
- V is a set of the last jobs. Find the lateness of the jobs in V. Select the job that has minimum lateness.
- $V = \min \{C_{max} - d_i\}$
- Schedule this job at the end of the order.
- Remove the processing time of the scheduled job from the completion time.
- Repeat the algorithm until all jobs are scheduled.

Example



Consider 6/1/Lmax problem

Job	1	2	3	4	5	6
Pi	2	3	4	3	2	1
di	3	6	9	7	11	7

Cmax = 2+3+4+3+2+1 = 15
V = {J3,J5,J6}
V = min { (15-9), (15-11), (15-7)} = min{ 6,4,8}
Job 5 has the minimum lateness.
Job 5 is selected and scheduled in the 6. order.
_ - _ -_ -_ -_ - 5

Job 5 is removed from the list.

Cmax = 15-2 = 13

V= {J3, J6}

V = min{(13-9), 13-7)}

Job 3 is selected.

_ - _ -_ -_ -3- 5

Cmax= 13-4 =9

V= {J2, J6}

V= min{ (9-6), (9-7)}

Job 6 is selected.

_ - _ -_ -6 -3- 5

- Cmax = 9-1 = 8
- V = {J2, J4}
- V = min{ (8-6), (8-7)}
- Job 4 is selected.
- _ - _ -4-6 -3- 5

Because of the precedence constraint J1 should be processed before J2.

Optimal order is 1-2-4-6-3-5.

Job	Pi	Ci	di	Li
1	2	2	3	-1
2	3	5	6	-1
4	3	8	7	1
6	1	9	7	2
3	4	13	9	4
5	2	15	11	4

Lmax = 4

Resources

- Sıralama ve Programlama, Hüseyin Başlıgil
- Çizelgeleme Ders Notları, Prof. Dr. Hüseyin Başlıgil
- web4.uwindsor.ca/users/b/baki%20fazle/Chapter_08_Lecture_12_to_19_w08_431_scheduling.ppt – Windsor University Operations Scheduling Lecture Notes
- https://fenix.tecnico.ulisboa.pt/downloadFile/282093452004307/5.1%20-%20Scheduling.pdf
- http://business.unr.edu/faculty/ronlembke/352/ppt/16-scheduling.pdf
- www.or.uni-bonn.de/lectures/ss10/scheduling_data/sched10_2.pdf
- web-static.stern.nyu.edu/om/faculty/pinedo/scheduling/nagi/13_singlelate.ppt
- homepage.univie.ac.at/radoslava.mirkov/or2/Chapter8.ppt