Job Shop Scheduling

Contents

1. Problem Statement

2. Disjunctive Graph

3. The Shifting Bottleneck Heuristic and the Makespan

1

Literature:

1. *Scheduling, Theory, Algorithms, and Systems*, Michael Pinedo, Prentice Hall, 1995, or new: Second Addition, 2002
   Chapter 6

or

2. *Operations Scheduling with Applications in Manufacturing and Services*, Michael Pinedo and Xiuli Chao, McGraw Hill, 2000
   Chapter 5

2

Problem Statement

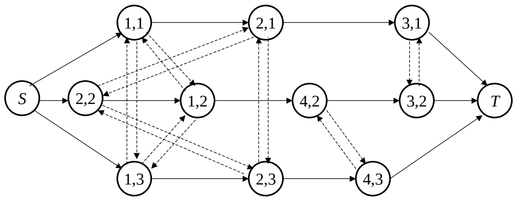
Job shop environment:

- *m* machines, *n* jobs
- objective function
- each job follows a predetermined route
- routes are not necessarily the same for each job
- machine can be visited once or more than once (recirculation)
- NP hard problems

3

Disjunctive Graph

Example of a job shop problem: 4 machines and 3 jobs



4

$Jm \mid \mid C_{max}$

(i, j)	processing of job j on machine i
p_{ij}	processing time of job j on machine i

$G = (N, A \cup B)$

$(i, j) \in N$	all the operations that must be performed on the n jobs
A	**conjunctive** (solid) arcs represent the precedence relationships between the processing operations of a single job
B	**disjunctive** (broken) arcs connect two operations which belong to two different jobs, that are to be processed on the same machine, they go in opposite directions

- Disjunctive arcs form a clique for each machine.
- Clique is a maximal subgraph in which all pairs of nodes are connected with each other.
- Operations in the same clique have to be done on the same machine.

5

How to construct a feasible schedule?

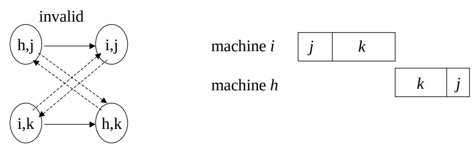
Select D - a subset of disjunctive arcs (one from each pair) such that the resulting directed graph $G(D)$ has no cycles.

Graph $G(D)$ contains conjunctive arcs + D.

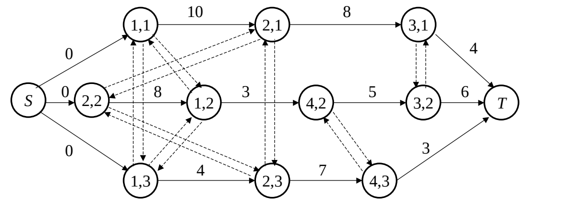
D represents a feasible schedule.

A cycle in the graph corresponds to a schedule that is infeasible.

Example



6



- The makespan of a feasible schedule is determined by the longest path in $G(D)$ from S to T.
- Minimise makespan: find a selection of disjunctive arcs that minimises the length of the longest path (the critical path).

7

The Shifting Bottleneck Heuristic and the Makespan

Idea:

- A classic idea in nonlinear programming is to hold all but one variable fixed and then optimise over that variable. Then hold all but a different one fixed, and so on.
- Furthermore, if we can do the one-variable optimisation in order of decreasing importance, there is better hope that the local optimum so found will be the global one, or close to it.
- In job shop problems fixing the value of variable means fixing the sequence in which jobs are to be processed on a given machine.

8

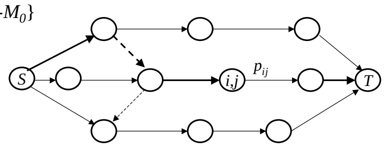
Iteration

M set of m machines

$M_0 \subset M$ machines for which sequence of jobs has already been determined in previous iterations

Analysis of machines still to be scheduled

$i \in \{M - M_0\}$



Define a single-machine problem $1 \mid r_j \mid L_{max}$ for machine i ➡

Jobs	...	j	...
p_{ij} r_{ij} d_{ij}		p_{ij} longest path from S to (i,j) $C_{max}(M_0)$ – longest path from (i,j) to $T + p_{ij}$	

9

Bottleneck selection

- A machine k with the largest maximum lateness is a *bottleneck*.

$$L_{\max}(k) = \max_{i \in \{M - M_0\}} (L_{\max}(i))$$

1. Schedule machine k according to the sequence which minimises the corresponding L_{max} (single-machine problem).
2. Insert all the corresponding disjunctive arcs in the graph.
3. Insert machine k in M_0.

$$C_{max}(M_0 \cup k) \geq C_{max}(M_0) + L_{max}(k)$$

10

Resequencing of all machines scheduled earlier

Aim: to reduce the makespan

Do for each machine $l \in \{M_0 - k\}$

delete the disjunctive arcs associated with the machine l

formulate a single machine problem for the machine l and find the sequence that minimises $L_{max}(l)$

Insert the corresponding disjunctive arcs.

11

Shifting Bottleneck Algorithm

Step 1. Set the initial conditions

$M_0 = \emptyset$ set of scheduled machines.

Graph G is the graph with all the conjunctive arcs and no disjunctive arcs.

Set $C_{max}(\emptyset)$ equal to the longest path in graph G.

Step 2. Analysis of the machines still to be scheduled

Solve the simple problem for each machine still to be scheduled: formulate a single machine problem with all operations subject to release dates and due dates.

Step 3. Bottleneck selection

The machine with the highest cost is designated the bottleneck.

Insert all the corresponding disjunctive arcs in graph G.

Insert machine which is the bottleneck in M_0 .

12

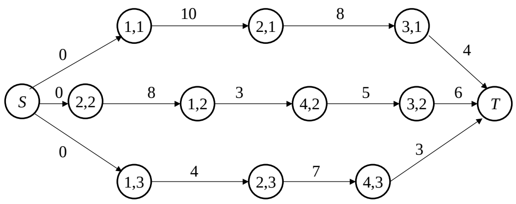
Step 4. Resequencing of all the machines scheduled earlier
Find the sequence that minimised the cost and insert the corresponding disjunctive arcs in graph *G*.

Step 5. Stopping condition
If all the machines are scheduled ($M_0 = M$) then STOP else go to Step 2.

13

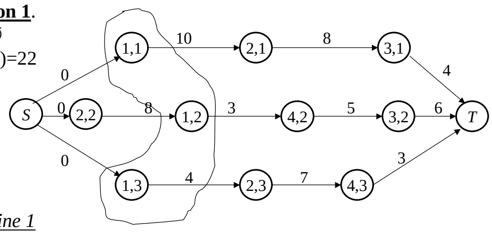
Example.

Jobs	Machine Sequence	Procesing Times
1	1, 2, 3	p_{11}=10, p_{21}=8, p_{31}=4
2	2, 1, 4, 3	p_{22}=8, p_{12}=3, p_{42}=5, p_{32}=6
3	1, 2, 4	p_{13}=4, p_{23}=7, p_{43}=3



14

Iteration 1.
$M_0 = \varnothing$
$C_{max}(\varnothing)$=22



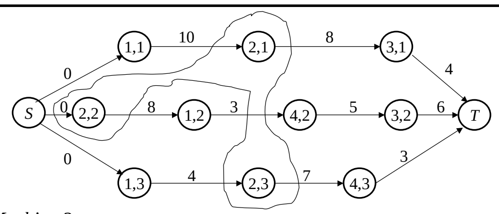
Machine 1

Jobs	1	2	3
p_{1j}	10	3	4
r_{1j}	0	8	0
d_{1j}	10	11	12

L_{max}(1, 2, 3) = max {0, 2, 5} = 5
L_{max}(1, 3, 2) = max {0, 2, 6} = 6
L_{max}(2, 1, 3) = max {0, 11,...} >11
L_{max}(2, 3, 1) = max {0, 3, 15} = 15
L_{max}(3, 1, 2) = max {-8, 4, 6} = 6
L_{max}(3, 2, 1) = max {-8, 0, 15} = 15

1,2,3 L_{max}(1)=5

15



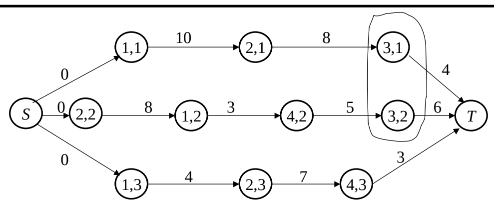
Machine 2

Jobs	1	2	3
p_{2j}	8	8	7
r_{2j}	10	0	4
d_{2j}	18	8	19

L_{max}(1, 2, 3) = max {0, 16, 14} = 16
L_{max}(1, 3, 2) = max {0, 6, 25} = 25
L_{max}(2, 1, 3) = max {0, 0, 6} = 6
L_{max}(2, 3, 1) = max {0, -4, 5} = 5
L_{max}(3, 1, 2) = max {-8, 1, 19} = 19
L_{max}(3, 2, 1) = max {-8, 7, 5} = 7

2,3,1 L_{max}(2)=5

16



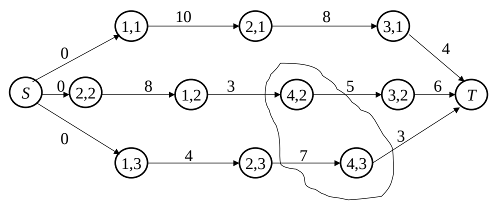
Machine 3

Jobs	1	2
p_{3j}	4	6
r_{3j}	18	16
d_{3j}	22	22

L_{max}(1, 2) = max {0, 6} = 6
L_{max}(2, 1) = max {0, 4} = <u>4</u>

L_{max}(3)=4

17



Machine 4

Jobs	2	3
p_{4j}	5	3
r_{4j}	11	11
d_{4j}	16	22

L_{max}(2, 3) = max {0, -3} = 0
L_{max}(3, 2) = max {-8, 3} = <u>3</u>

L_{max}(4)=3

18

$L_{max}(2)=5$ 2,3,1

$L_{max}(1)=5$ 1,2,3

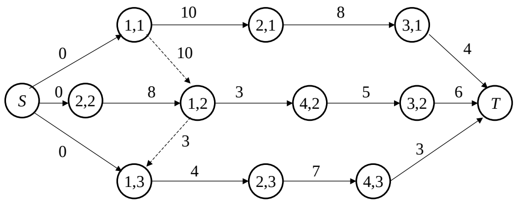
$L_{max}(3)=4$

$L_{max}(4)=3$

Machines 1 and 2 are bottlenecks.

Machine 1 is chosen as a bottleneck!

19

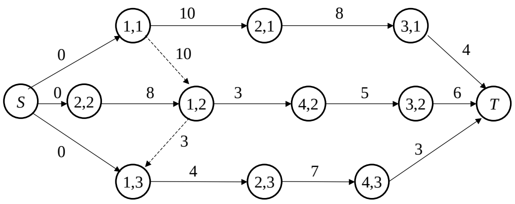


$C_{max}(\{1\}) = C_{max}(\emptyset) + L_{max}(1) = 22 + 5 = 27$

20

Iteration 2.

$M_0 = \{1\}$, $C_{max}(\{1\})=27$



Machine 2

Jobs	1	2	3
p_{2j}	8	8	7
r_{2j}	10	0	17
d_{2j}	23	10	24

2,1,3 $L_{max}(2)=1$

21

Machine 3

Jobs	1	2
p_{3j}	4	6
r_{3j}	18	18
d_{3j}	27	27

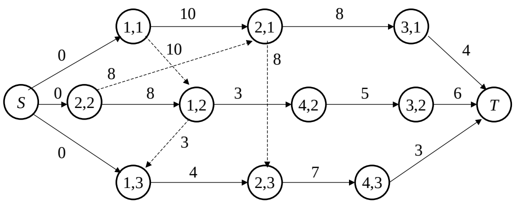
1,2 or 2,1 $L_{max}(3)=1$

Machine 4 $L_{max}(4)=0$

Machines 2 and 3 are bottlenecks.

Machine 2 is chosen as a bottleneck!

22

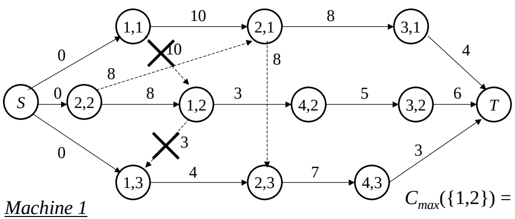


$C_{max}(\{1,2\}) = C_{max}(\{1\}) + L_{max}(2) = 27 + 1 = 28$

Can we decrease $C_{max}(\{1,2\})$?

Will resequencing machine 1 give any improvement ?

23



$C_{max}(\{1,2\}) = 28$

Machine 1

Jobs	1	2	3
p_{1j}	10	3	4
r_{1j}	0	8	0
d_{1j}	10	17	18

$L_{max}(1, 2, 3) = \max\{0, -4, -1\} = 0$
$L_{max}(1, 3, 2) = \max\{0, -4, 0\} = 0$
$L_{max}(2, 1, 3) = \max\{-6, 11, 7\} = 11$
$L_{max}(2, 3, 1) = \max\{-6, -3, 15\} = 15$
$L_{max}(3, 1, 2) = \max\{-14, 4, 0\} = 4$
$L_{max}(3, 2, 1) = \max\{-14, 2, 15\} = 15$

original sequence: 1,2,3

Resequencing machine 1 does not give any improvement.

24

Iteration 3.

$M_0 = \{1,2\}$

$C_{max}(\{1,2\})=28$

Machine 3

Jobs	1	2
p_{3j}	4	6
r_{3j}	18	18
d_{3j}	28	28

1,2 or 2,1 $L_{max}(3) = 0$

Machine 4 $L_{max}(4) = 0$

No machine is a bottleneck!

25