IO543: Scheduling

Lecture 11

Brian Denton
Associate Professor, IOE

IOE Bld 2893
btdenton@umich.edu

Announcements

- Assignment 3 graded
- Assignment 4 due Thursday

- Midterm next week:
  - No class on Tuesday
  - Midterm on ctools at 5pm Monday
  - Due no later than 10:40 on Thursday

Last Time

- Project Management:
  - CEE 536 is a full class on project management
- Flow Shops:
  - Johnson’s rule. Proof in on ctools.

Today

More about flow shops

- An IP for optimal permutation schedules for $Fm\ |prmu|\ C_{max}$
- A heuristic for $Fm\ |prmu|\ C_{max}$
- Flow shops with limited storage: $Fm\ |block|C_{max}$

$Fm \mid\mid C_{max}$

- When considering machines in parallel a schedule is defined by a **sequence at every machine**

- In a **permutation flow shop** the sequence is the same at every machine: $Fm \mid prmu \mid C_{max}$

  - There is always an optimal schedule in which the sequence between the first two and last two machines does not change

  - Therefore $F2 \mid\mid C_{max}$ and $F3 \mid\mid C_{max}$ have optimal permutation schedules

$Fm \mid prmu \mid C_{max}$

Given a **permutation schedule** J1 → J2→ ….→ Jn the completion times can be computed using **recursive equations**:

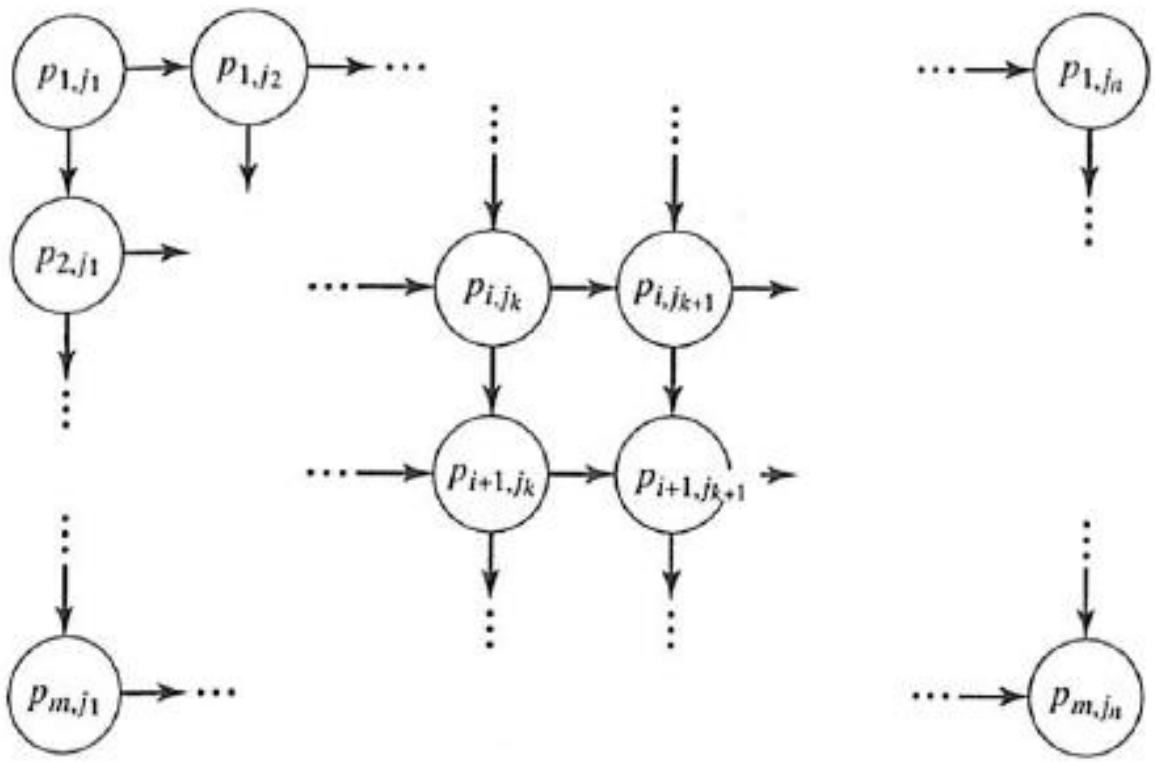
$$C_{i,J1} = \sum_{l=1}^{i} p_{l,J1}\,, \qquad i = 1, \dots, m$$

$$C_{1,Jk} = \sum_{l=1}^{k} p_{1,Jl}\,, \qquad k = 1, \dots, n$$

$$C_{i,Jk} = \max\left(C_{i-1,Jk}\,, C_{i,Jk-1}\right) + p_{i,Jk} \quad i = 2, \dots, m; k = 2, \dots, n$$

$Fm \mid prmu \mid C_{max}$

A network can be used to visualize computing the makespan for $Fm \mid prmu \mid C_{max}$ for a given sequence J1 → J2→….→ Jn



$Fm \mid prmu \mid C_{max}$ Example

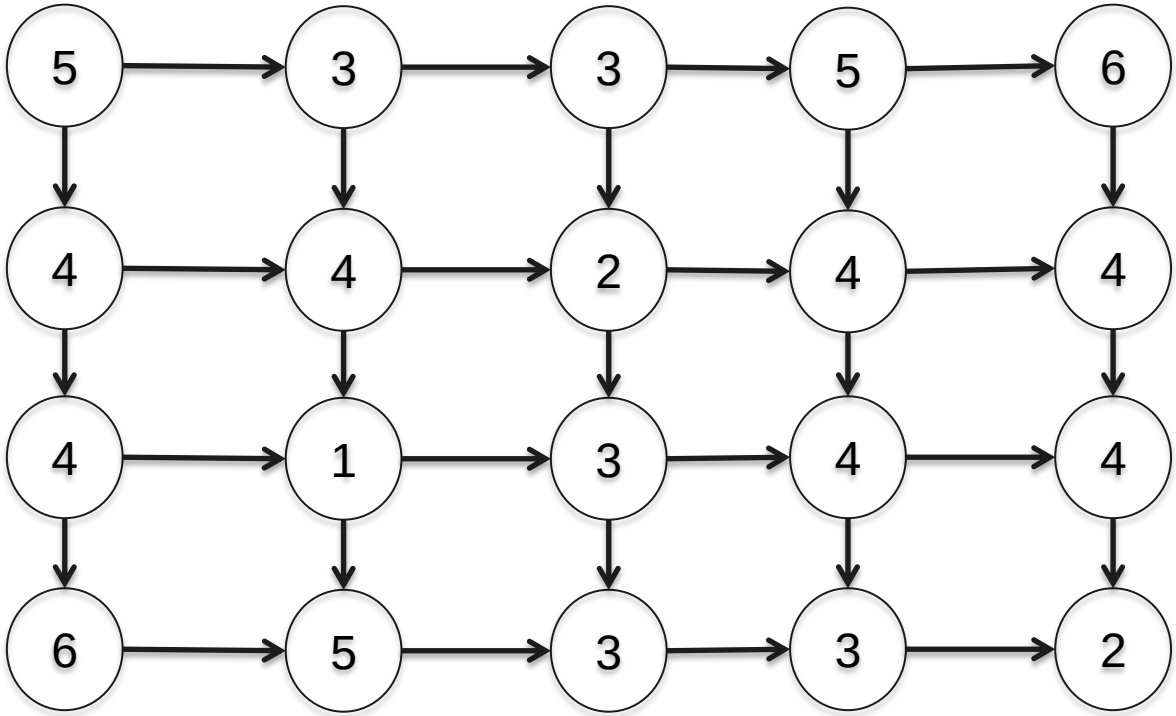
Consider the following instance of $F4 \mid prmu \mid C_{max}$:

Jobs	J1	J2	J3	J4	J5
$p_{1,Jk}$	5	5	3	6	3
$p_{2,Jk}$	4	4	2	4	4
$p_{3,Jk}$	4	4	3	4	1
$p_{4,Jk}$	3	6	3	2	5

What is the makespan for sequence J2→J5→J3→ J1→ J4?

$Fm \mid prmu \mid C_{max}$ Example

The following network expresses the previous instance of $F4 \mid prmu \mid C_{max}$:



IP Formulation of $Fm\ |prmu|\ C_{max}$

For flow shops with more than 2 machines:

- $Fm\ |prmu|\ C_{max}$ can be formulated as an **integer program**
  - Minimizing makespan is equivalent to minimizing total idle time on the last machine (why?)
  - The formulation must keep track of machine idle time between jobs

IP Formulation of $Fm\ |prmu|\ C_{max}$

Minimizing makespan is equivalent to minimizing total idle time on the last machine

$$\min \sum_{i=1}^{m-1} \sum_{j=1}^{n} x_{j1} p_{ij} + \sum_{j=1}^{n-1} I_{mj}$$

$$s.t. \quad \sum_j x_{jk} = 1, \qquad \text{for all k}$$

$$\sum_k x_{jk} = 1, \qquad \text{for all j}$$

$$\mathrm{I_{ik}} + \sum_j x_{j,k+1} p_{ij} + W_{i,k+1}$$

$$- \mathrm{W_{ik}} - \sum_j x_{jk} p_{i+1,j} - I_{i+1,k} = 0, \qquad \mathrm{k = 1..N-1; i = 1..M-1}$$

$$W_{i1} = 0, \qquad \text{for all i}$$

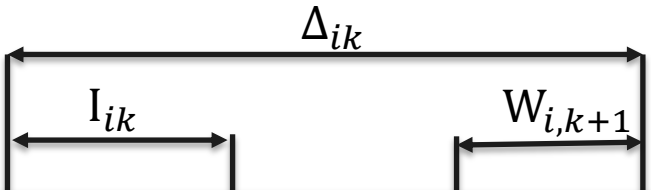
$$I_{1k} = 0, \qquad \text{for all k}$$

$$x_{jk}\ binary, for\ all\ j, k, I_{ik} W_{ik} \geq 0, for\ all\ i, k$$

IP Formulation of $Fm\ |prmu|\ C_{max}$

Balance Constraints:

$$\underbrace{\mathrm{I_{ik}} + \sum_j x_{j,k+1} p_{ij} + W_{i,k+1}}_{\Delta_{ik}} = \underbrace{\mathrm{W_{ik}} + \sum_j x_{jk} p_{i+1,j} + I_{i+1,k}}_{\Delta_{ik}}$$



IP Formulation Example

Write the MIP formulation for the following instance of $Fm\ |prmu|\ C_{max}$

Jobs	J1	J2	J3	J4	J5
$p_{1,Jk}$	5	5	3	6	3
$p_{2,Jk}$	4	4	2	4	4
$p_{3,Jk}$	4	4	3	4	1
$p_{4,Jk}$	3	6	3	2	5

Slope Heuristic for $Fm\ |prmu|\ C_{max}$

When the IP formulation becomes **difficult or impossible** to solve the Slope Heuristic can be used:

Slope Heuristic:

1) Compute the slope index for all jobs:

$$A_j = \sum_{i=1}^{m} \big(m - (2i - 1)\big) p_{ij}$$

2) Sequence jobs in increasing order of $\mathrm{A_j}$

Slope Heuristic Example

Use the Slope Heuristic to find a schedule for the following instance of $F4 \mid prmu \mid C_{max}$:

Jobs	J1	J2	J3	J4	J5
$p_{1,Jk}$	5	5	3	6	3
$p_{2,Jk}$	4	4	2	4	4
$p_{3,Jk}$	4	4	3	4	1
$p_{4,Jk}$	3	6	3	2	5

What is the makespan?

Summary

Things to remember:

- For a given schedule you can find the makespan by finding the critical path of the "**project-like"** network
- $F2\ |prmu|\ C_{max}$ can be solved to **optimality** with **Johnson's rule**
- $Fm\ |prmu|\ C_{max}$ can be solved to **optimality** using a **IP formulation** of $Fm\ |prmu|\ C_{max}$
- The **Slope Heuristic** can be used for instances that can't be solved by AMPL

$Fm\,|block|C_{max}$

- Flow shops with limited intermediate storage can result in **blocking**
  - Blocking occurs when the **buffer** between two machines becomes full
  - The job on the machine prior to the full buffer is **blocked** from completing processing and the **machine sits idle**

$Fm\,|block|C_{max}$

A flow shop with limited intermediate storage is a special case of a flow shop with **no intermediate storage**

- The storage space for each job can be treated as a machine with zero processing time
- If D_{ij} is the time job j departs machine i:

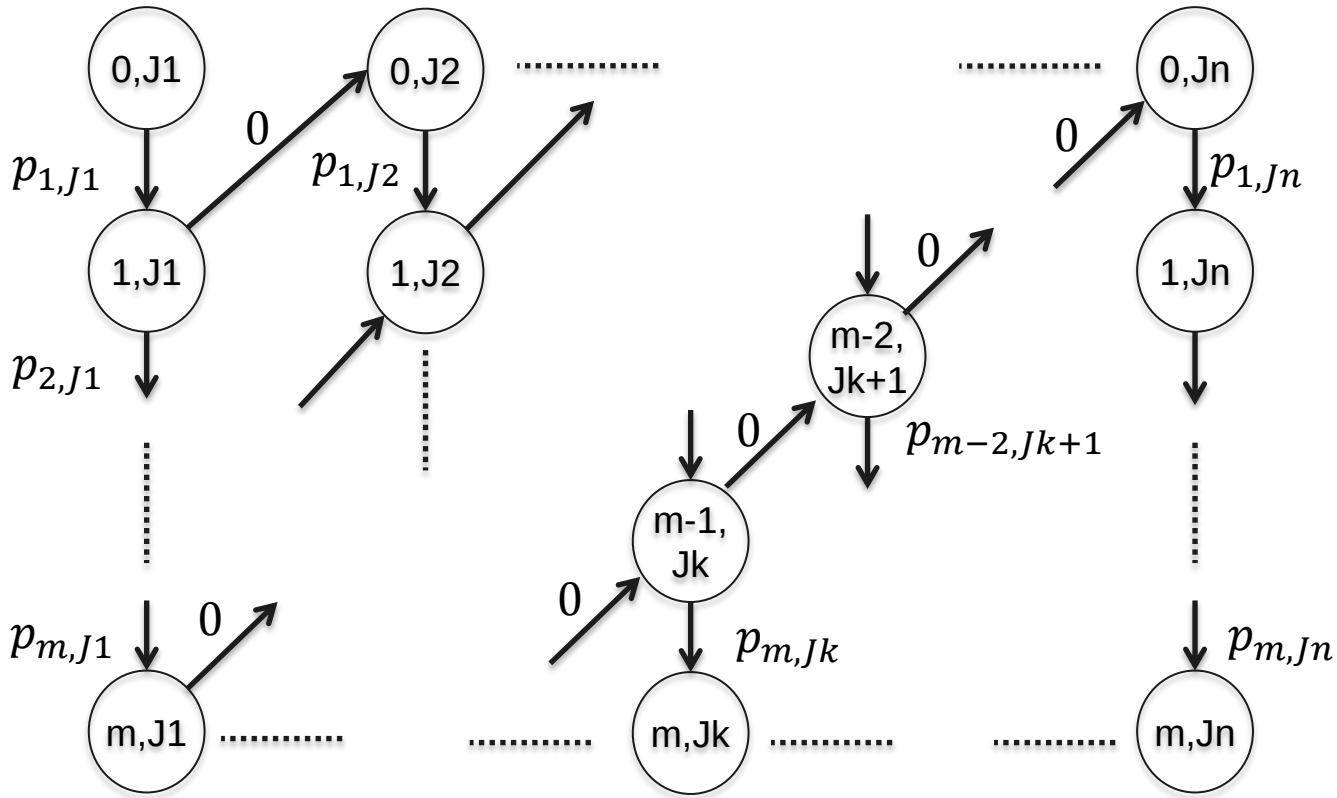
$$D_{i,J1} = \sum_{l=1}^{i} p_{l,J1}\,, \qquad i = 1, \dots, m$$

$$D_{i,Jk} = \max\left(D_{i-1,Jk} + p_{i,Jk}, D_{i+1,Jk-1}\right), \qquad i = 2, \dots, m; k = 2, \dots, n$$

$$D_{m,Jk} = D_{m-1,Jk} + p_{mJk}, \qquad k = 2, \dots, n$$

$Fm\,|block|C_{max}$ Example

A **project network** for $Fm\,|block|C_{max}$:



$Fm\,|block|C_{max}$ Example

Construct the project network and find the makespan for the following instance of $Fm\,|block|C_{max}$ for sequence J1→ J2→ J3→ J4→ J5

Jobs	J1	J2	J3	J4	J5
$p_{1,Jk}$	5	5	3	6	3
$p_{2,Jk}$	4	4	2	4	4
$p_{3,Jk}$	4	4	3	4	1
$p_{4,Jk}$	3	6	3	2	5