Aggregate Planning

Plan 1: vary inventory

plan 2: produce 10 unit/day, 20unit/day

plan 3: produce 10 unit/day + subcontract the rest

Plan 1:

Plan 1

Aggregate Planning Example

Month	Production Days (a)	Demand Forecast (b)	Production @ 14/day (c)	Inventory Change (d = c-b)	Ending Balance with 0 on hand on Jan. 1 (e)	Ending Balance with 566 on hand on Jan. 1 (f)
January	22	220	308	88	88	654
February	18	90	252	162	250	816
March	21	210	294	84	334	900
April	22	396	308	-88	246	812
May	22	616	308	-308	-62	504
June	20	700	280	-420	-482	84
July	21	378	294	-84	-566	0
August	22	220	308	88	-478	88
September	20	200	280	80	-398	168
October	23	115	322	207	-191	375
November	19	95	266	171	-20	546
December	20	260	280	20	0	566
Totals	250	3500	3500			5513

7-19

$\frac{3500}{250} = 14$

Aggregate Planning Example

- Cost calculations for Plan 1
  - Maximum inventory requiring storage = 900 units
  - Average inventory balance
    $$= \frac{654 + 816 + \ldots\ldots + 566}{12} = \frac{5513}{12}$$
    $$\approx 460 \text{ units}$$
  - **Inventory cost** = carrying cost + storage cost
    $$= 20 * 460 + 0.90 * 900$$
    $$= \$10{,}010$$

7-20

Carrying Cost = 20\$/unit

storage " = 0.90/unit

10 unit/day + subcontract

Plan 3

Aggregate Planning Example

- Cost calculations for Plan 3
  - Maximum inventory requiring storage = 240 units
  - Average inventory balance
    $$= \frac{150 + 240 + \ldots\ldots + 150}{12} = \frac{1169}{12}$$
    $$= 97.42 \text{ units}$$
  - **Inventory cost** = carrying cost + storage cost
    $$= 20 * 97.42 + 0.90 * 240 = \$2{,}164$$
  - **Subcontracting cost** = 1000 * 7 = \$7,000
  - **Total costs** = 2164 + 7000 = **\$9,164**

7-35

Aggregate Planning Example

Month	Production Days	Demand Forecast	Production @ 10/day	Inventory Change	Ending Balance with 0 on hand on Jan. 1	Ending Balance with 150 on hand on Jan. 1
January	22	220	220	0	0	150
February	18	90	180	90	90	240
March	21	210	210	0	90	240
April	22	396	220	-176	0	64
May	22	616	220	-396	0	0
June	20	700	200	-500	0	0
July	21	378	210	-168	0	0
August	22	220	220	0	0	0
September	20	200	200	0	0	0
October	23	115	230	115	115	115
November	19	95	190	95	210	210
December	20	260	200	-60	150	150
Totals	250	3500	2500			1169

7-34

Plan 2

Aggregate Planning Example

Month	Production Days	Demand Forecast	Production Rate/day	Total Production	Inventory Change	Ending Balance with 0 on hand on Jan. 1	Ending Balance with 150 on hand on Jan. 1
January	22	220	10	220	0	0	150
February	18	90	10	180	90	90	240
March	21	210	20	420	210	300	450
April	22	396	20	440	44	344	494
May	22	616	20	440	-176	168	318
June	20	700	20	400	-300	-132	18
July	21	378	20(15d)+10(6d)	360	-18	-150	0
August	22	220	10	220	0	-150	0
September	20	200	10	200	0	-150	0
October	23	115	10	230	115	-35	115
November	19	95	10	190	95	60	210
December	20	260	10	200	-60	0	150
Totals	250	3500		3500			2145

7-27

Aggregate Planning Example

- Cost calculations for Plan 2
  - Maximum inventory requiring storage = 494 units
  - Average inventory balance
    $$= \frac{150 + 240 + \ldots\ldots + 150}{12} = \frac{2145}{12}$$
    $$\approx 179 \text{ units}$$
  - **Inventory cost** = carrying cost + storage cost
    $$= 20 * 179 + 0.90 * 494 = \$4{,}025$$
  - **Shift change cost** = 3500 * 2 = \$7,000
  - **Total costs** = 4025 + 7000 = \$11,025

7-36

Using Transportation Algorithm to Solve an Aggregate Planning Problem

Example

Period	Regular time capacity	Overtime capacity	Demand
1	2	5	4
2	1	3	9
3	9	2	2

- Given
  - Initial Inventory = 4
  - Final Inventory required = 3
  - Regular time production cost = $10/unit
  - Overtime production cost = $12/unit
  - Carrying cost = $3/unit-period
  - Backordering cost = $4/unit-period

7-37

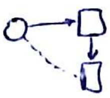
Using Transportation Algorithm to Solve an Aggregate Planning Problem

	Per. 1	Per. 2	Per. 3	Final Inv.	Slack	Capacity
Initial Inv.	0	3	6	9	0	4
Per. 1 RT	10	13	16	19	0	2
Per 1 OT	12	15	18	21	0	5
Per 2 RT	14	10	13	16	0	1
Per 2 OT	16	12	15	18	0	3
Per 3 RT	18	14	10	13	0	9
Per 3 OT	20	16	12	15	0	2
Demand	4	9	2	3	8	26 / 26

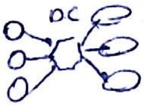
7-38

Transportaion

milk-run



cross-docking



inventory Cost:

cycle $\rightarrow Q/2$

Safety $\rightarrow F^{-1}(CSL)\sqrt{T+L}\,\sigma$

in-transit $\rightarrow \frac{T}{365} \times$ annual demand

L = lead time

T = reorder interval

$$F^{-1}(CSL)\sqrt{(T+E(L))\sigma_D^2+(E(D))^2\sigma_L^2}$$

$$\text{CK: Safety} = \frac{L+T}{2} \times \frac{\text{annual dem}}{365}$$

Pricing and Revenue Management

3 prices:

P	d	R
1800	100	180000
1600	200	–
1000	500	–

$$R = 100 \times 1800 + (200-100) \times 1600 + (500-100-100) \times 1000 = 640000$$

in general:

$$\text{demand} = A - Bp \quad (p \rightarrow \text{Price})$$

$$\text{Cost} = c^*(A-Bp)$$

$$\Rightarrow \frac{\partial \text{Profit}}{\partial p} = 0 \Rightarrow p^* = \left(\frac{A}{2B} + \frac{c}{2}\right)$$

Sourcing Decisions

manufact retailer

manu [illegible] ← 125 \$ [illegible]

80 \$ [illegible]

[illegible] ← 20 \$ discount store

manuf: fixed Pro Cost = 100,000; Cost/unit = 35

Make to Order (no contract)

retailer: If the retailer orders 12,000 units from the manufacturer, the retailer's expected profit is calculated as follows:

Demand	Probability	Revenue	Cost	Profit
8,000	0.11	$1,080,000	$960,000	$120,000
10,000	0.11	$1,290,000	$960,000	$330,000
12,000	0.28	$1,500,000	$960,000	$540,000
14,000	0.22	$1,500,000	$960,000	$540,000
16,000	0.18	$1,500,000	$960,000	$540,000
18,000	0.10	$1,500,000	$960,000	$540,000
				Expected Profit = $470,700

– Profit of the **manufacturer** = 12,000 * $80 – {$100,000 + 12,000*$35} = $440,000

10-27

Make to Order (buy-back contract)

– If the retailer orders 12,000 units from the manufacturer, the retailer's expected profit is calculated as follows:

Demand	Probability	Revenue	Cost	Profit
8,000	0.11	$1,200,000	$960,000	$240,000
10,000	0.11	$1,350,000	$960,000	$390,000
12,000	0.28	$1,500,000	$960,000	$540,000
14,000	0.22	$1,500,000	$960,000	$540,000
16,000	0.18	$1,500,000	$960,000	$540,000
18,000	0.10	$1,500,000	$960,000	$540,000
				Expected Profit = $490,500

mana. buy unsold by 50 \$/unit

10-38

Scanned by CamScanner

Make to Order (revenue-sharing contract)

– If the retailer orders 12,000 units from the manufacturer, the **retailer**'s expected profit is calculated as follows:

Demand	Probability	Revenue	Cost	Profit
8,000	0.11	$930,000	$720,000	$210,000
10,000	0.11	$1,102,500	$720,000	$382,500
12,000	0.28	$1,275,000	$720,000	$555,000
14,000	0.22	$1,275,000	$720,000	$555,000
16,000	0.18	$1,275,000	$720,000	$555,000
18,000	0.10	$1,275,000	$720,000	$555,000
				Expected Profit = $498,075

10-52

80\$ → 60\$

Revenue 10% ob

guest to manufactures

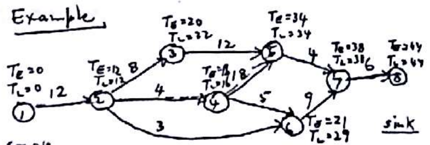
930,000 = (8000 × 125) × 0.85 + 4000 × 20

Scheduling and Sequencing

① CPM :

CPM

Example,



Find all paths from source to sink

Path	Length	
1-2-3-5-7-8	42	
1-2-4-5-7-8	44*	← Critical Path
1-2-4-6-7-8	36	
1-2-6-7-8	30	Project duration = 44

11-8

② PERT

β-distribution:

a: most optimistic duration

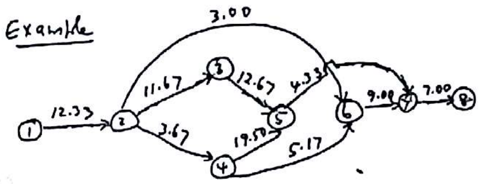
m: most likely duration

b: most pessimistic "

$$\mu = \frac{a+4m+b}{6} \qquad Var = \left(\frac{b-a}{6}\right)^2$$

PERT

Example



11-10

PERT

Activity	a	m	b	μ	σ^2	
1-2	10	12	16	12.33	1.00	*
2-3	2	8	36	11.67	32.11	*
2-4	1	4	5	3.67	0.44	
2-6	2	3	4	3.00	0.11	
3-5	8	12	20	12.67	4.00	*
4-5	15	18	30	19.50	6.25	
4-6	3	5	8	5.17	0.69	
5-7	2	4	8	4.33	1.00	*
6-7	6	9	12	9.00	1.00	
7-8	4	6	14	7.00	2.78	*

11-11

PERT

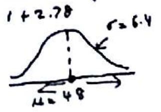
Path	Duration	
1-2-3-5-7-8	48.00 *	← critical Path
1-2-4-5-7-8	46.83	
1-2-4-6-7-8	37.17	
1-2-6-7-8	31.33	

∴ The mean project duration = 48 days

Standard deviation of the project duration

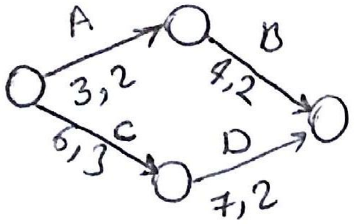
$= \sqrt{1 + 32.11 + 4 + 1 + 2.78}$

$=$ 6.4 days



11-12

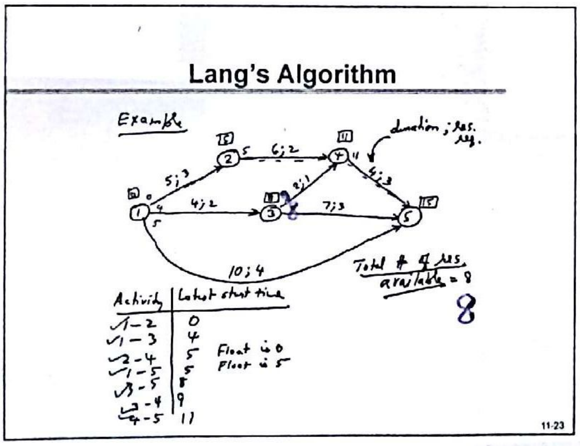
Project planning with Resource Constraints



Total # of resource availble = 4

① Lang's Algorithm:

- order activities by (Latest start times) first
- if a tie ⇒ lest float first
  - ⇒ longest duration "
  - ⇒ largest resource first
  - ⇒ Alphabetical

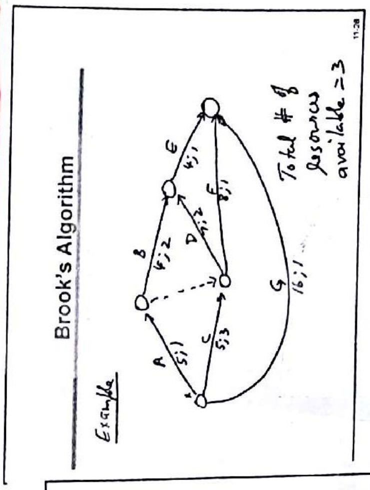


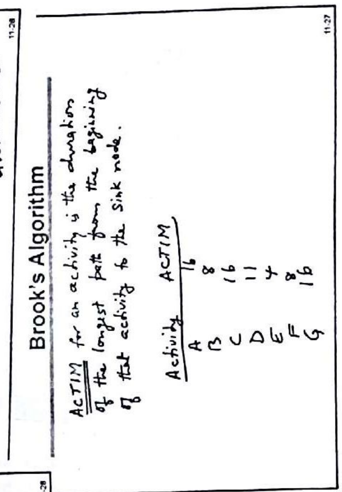
② Brook's algorithm:

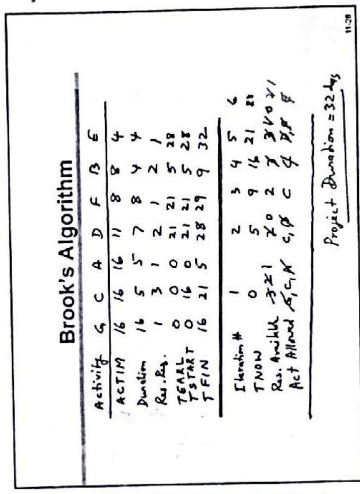
- Calculate ACTIM for each activity
- Schedual activities in decreasing order of Actim

tie? - longest duration first
- largest resource first
- Alphabeticall

Actim = duration of longest path from the beging to sink





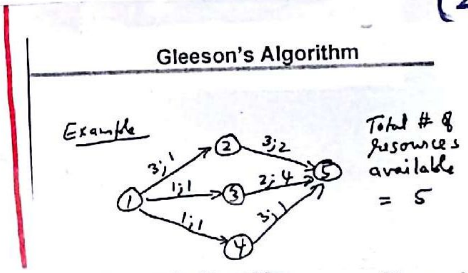


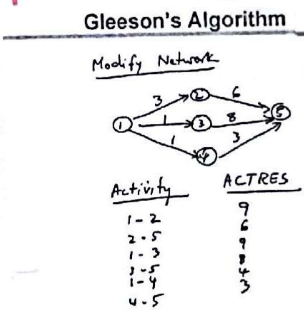
③ Gleeson Alg:

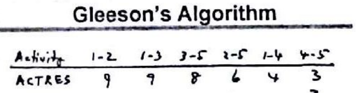
use Actress:

① 6,2 ↳ 12

② Sort base on Actim







Activity	1-2	1-3	3-5	2-5	1-4	4-5
ACTRES	9	9	8	6	4	3
Duration	3	1	2	3	1	3
Res. Req.	1	1	4	2	1	1
TEARL	0	0	1	3	0	1
TSTART	0	0	1	3	0	3
TFIN	3	1	3	6	1	6

Itn. #	1	2	3
TNOW	0	1	3
Res. Avail.	~~5~~ ~~4~~ ~~3~~ 2	~~4~~ 0	~~4~~ ~~3~~ 2
Act. Allowed	~~1-2~~, ~~1-3~~, ~~1-4~~	3-5, 4-5	2-5, 4-5

Project Duration = 6

Sequencing:

Sequencing :

Sequencing

Processing Time, t | Due Date, d

Lateness (+ve or −ve), L | Tardiness (measure of Positive Lateness), T = Max(0,L)

Flow Time, F_i = Span between, task i Available for Processing & time at which Job i is completed

Completion Time, C_i = Span between beginning of 1st Job and when Job, i is Finished

If all Jobs are available at t=0, then $C_i = F_i$

Makespan = Span of Time when we start working on the 1st Job on the 1st Machine till we finish working on the Last Job on the Last Machine

Makespan for n tasks in Sequence s, $M_s = \sum_{i=1}^{n} t_i$

Mean Flow Time is Sequence s, $\overline{F_s} = \frac{1}{n}\sum_{i=1}^{n} F_{i,s}$

$L_{i,s} = C_{i,s} - d_i$ | $T_{i,s} = Max\{0, L_{i,s}\} = Max\{0, C_{i,s} - d_i\}$

Mean Lateness in Sequence s, $\overline{L_s} = \frac{1}{n}\sum_{i=1}^{n} L_{i,s}$ | Mean Tardiness in Sequence s, $\overline{T_s} = \frac{1}{n}\sum_{i=1}^{n} T_{i,s}$

No. of Tardy Jobs, $N_t = \sum_{i=1}^{n} \delta_i$, where $\delta_i = 1$ *if* $T_i > 0, \delta_i = 0$ *otherwise*

$T_{Max} = \max\{0, L_{Max}\}$ | $L_{Max} = \max\{L_{i,s}\} \forall i$ *in n*

Sequencing *n* jobs on one machine

Example

All tasks are available at t=0

Task i	Proc. time t_i	Due date d_i	Flowtime F_i	Lateness L_i
1	5	15	5	-10
2	8	10	13	3
3	6	15	19	4
4	3	25	22	-3
5	10	20	32	12
6	14	40	46	6
7	7	45	53	8
8	3	50	56	6

$\overline{F_s} = 30.75$ $\overline{L_s} = 3.25$

Sequencing *n* jobs on one machine

Use Shortest Processing time (SPT) Rule

Task i	t_i	d_i	F_i	L_i
4	3	25	3	-22
8	3	50	6	-44
1	5	15	11	-4
3	6	15	17	2
7	7	45	24	-21
2	8	10	32	22
5	10	20	42	22
6	14	40	56	16

$\overline{F_s} = 23.875$ $\overline{L_s} = -3.625$

Sequencing *n* jobs on one machine

Minimize Maximum Lateness

Use EDD (earliest due date) Rule

Task i	t_i	d_i	F_i	L_i
2	8	10	8	-2
1	5	15	13	-2
3	6	15	19	4
5	10	20	29	9
4	3	25	32	7
6	14	40	46	6
7	7	45	53	8
8	3	50	56	6

Sequencing n Jobs in One Machine

Hodgson's Algorithm (Minimize the number of Tardy Jobs)

Example

Task	1	2	3	4	5	6	7	8
t_i	5	8	6	3	10	14	7	3
d_i	15	10	15	25	20	40	45	50

Arrange the tasks in the order of Earliest Due Date rule

Task	2	1	3	5	4	6	7	8
t_i	8	5	6	10	3	14	7	3
C_i	8	13	19	29	32	46	53	56
d_i	10	15	15	20	25	40	45	50
L_i	-2	-2	4	9	7	6	8	6

Set aside Task 2 (since it is the 1st task with positive Lateness from the Left)

Task	1	3	5	4	6	7	8
t_i	5	6	10	3	14	7	3
C_i	5	11	21	24	38	45	48
d_i	15	15	20	25	40	45	50
L_i	-10	-14	1	-1	-2	0	-2

Set aside Task 5 (since it is the 1st task with positive Lateness from the Left)

Task	1	3	4	6	7	8	2	5
t_i	5	6	3	14	7	3	8	10
C_i	5	11	14	28	35	38	46	56
d_i	15	15	25	40	45	50	10	20
L_i	-10	-4	-11	-12	-10	-12	36	36

Only 2 Tardy Jobs

Sequencing n Jobs in Two Machine

Johnson's Algorithm

- Create a list of processing times of all jobs on machine 1 (*M1*) and machine 2 (*M2*).
- Identify the shortest processing time in this list. Break ties arbitrarily.
- If the shortest processing time is on *M1*, then assign the corresponding job to the next available position starting at the beginning of the sequence. Go to step 4. If it is on *M2*, then assign the corresponding job to the next available position starting from the end of the sequence. Go to step 4. .
- Remove the assigned job from the list. Repeat steps 2 and 3 until all jobs are assigned.

Sequencing n Jobs in Three Machine

Convert this into a Two Machine Problem

$Machine\ 1' = Machine\ 1 + Machine\ 2$

$Machine\ 2' = Machine\ 2 + Machine\ 3$

Condition for Optimality

The solution to the three machine problem will be optimal using the above method if

Either $Min\ T_{i_1} \geq Max\ t_{i_2}$ or $Min\ T_{i_3} \geq Max\ t_{i_2}$ is satisfied