2

Single Machine Scheduling

CHAPTER CONTENTS

2.1 INTRODUCTION

Single machine scheduling has attained most attention in theoretical scheduling studies. The understanding of the theories paves way for analyzing and better designing multi- machine systems. The following assumptions generally apply to build single machine scheduling models.

1. Machine is continuously available during scheduling period.
2. The machine processes jobs one at a time.
3. The process time of each job on the machine is accurately known and, it does not depend upon prior jobs.
4. The process time includes both set up time and actual machining time.
5. The other job related information is known before hand. This information may include due date of the job (d_j) and release time of the job (r_j).
6. In non-preemptive scheduling, jobs finish processing without interruption. In preemptive scheduling, jobs may be removed from the machine without finishing the operation.

2.2 SCHEDULING MATHEMATICS

The scheduling parameters for a typical job, say job j, are defined as follows:

- p_j = the processing time Job j
- S_j = the start time of job j
- W_j = the waiting time of job j
- W_j = the waiting time of job j
- D_j = the due date of job j
- Ej = the earliness job j
- r_j = the release time job j
- C_j = the completion time of job j
- F_j = the flow time of job j
- L_j = the lateness of job j
- T_j = the tardiness of job j

2.2.1 Gantt Chart

Gantt chart is a popular way of graphically presenting a schedule of jobs on machines. X-axis of the chart represents time and, rectangular block on y-axis represents machine. A horizontal bar shows each job's start and finish time on a particular machine. The job number is inscribed in a rectangle. The length of the rectangle is scaled to represent job's process time. The start and finish time of a job are indicated at the starting and terminating vertical sides of the job rectangle. Bars representing machines also indicate idle intervals on the machine. The following Gantt chart presents single machine with n jobs.

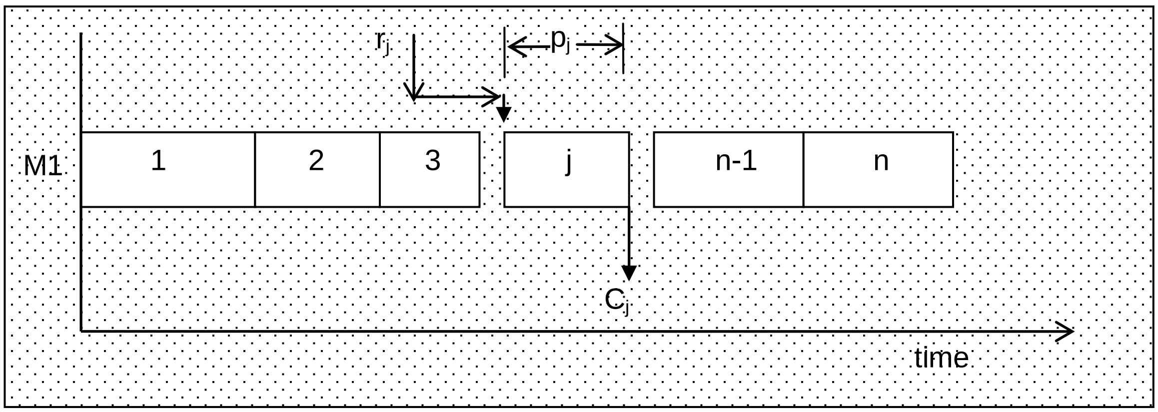


Figure 2.1 Gantt chart.

From the Gantt chart, waiting time for job j will be: $W_j = C_j - r_j$ Similarly, the lateness (L_j) of the job j will be; $L_j = C_j - d_j$. Job tardiness T_j is defined as positive lateness. In mathematical terms, tardiness (T_j) is expressed as follows:

$$\begin{aligned} T_j &= L_j, \quad \text{if } L_j > 0 \\ &= 0, \quad \text{otherwise} \\ T_j &= \max(L_j, 0) \end{aligned}$$

Similarly, job earliness is negative lateness. In mathematical terms, E_j is expressed as

$$\begin{aligned} E_j &= L_j, \quad \text{if } L_j < 0 \\ &= 0, \quad \text{otherwise} \\ E_j &= \max(-L_j, 0) \end{aligned}$$

Job Flow time (F_j) can be expressed in two ways:

$$\begin{aligned} F_j &= C_j - r_j \\ &= W_j + p_j \end{aligned}$$

From the relationship between r_j, W_j and p_j, expression for C_j can be deduced;

$$\begin{aligned} C_j &= r_j + W_j + p_j \\ &= S_j + p_j \end{aligned}$$

From the above relationship, $S_j = r_j + W_j$. Clearly, if $W_j = 0$ then $S_j = r_j$.

Start time for generating a schedule is;

$$S_j = C_{j-1} \quad \text{if} \quad C_{j-1} > r_j$$
$$= r_j \qquad \text{otherwise,}$$

Example 2.1

The following table contains data pertaining to $1||\bar{F}$ problem.

Job (j)	1	2	3	4
p_j	4	2	6	5

Note that the release time $r_j = 0$ for all jobs. Hence, it is a static shop environment. Use the following sequences and find average waiting time $\overline{W}$ and the average flow time $\overline{F}$.

i) Numerical (natural) Job order sequence (1-2-3-4)
ii) Shortest Process Time (SPT) Sequence (2-1-4-3)
iii) Random Sequence (1-3-4-2)

Solution:

i) Numerical Order Sequence (1-2-3-4)

Using the numerical or natural order sequence, the schedule is shown using Gantt chart below.

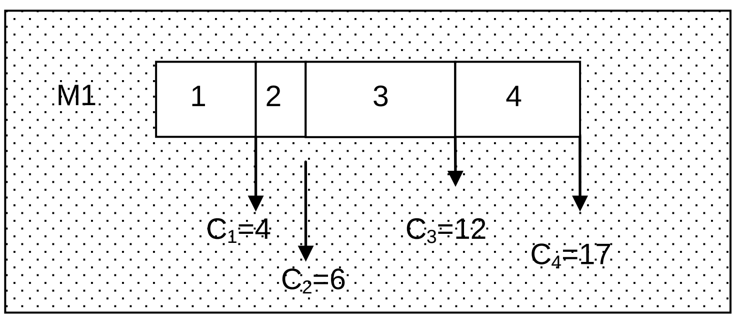


Figure 2.2 Gantt chart for numerical order sequence.

The waiting times for all the jobs is computed as shown in the following table

Job (j)	p_j	S_j	C_j	W_j	F_j
1	4	0	4	0	4
2	2	4	6	4	6
3	6	6	12	6	12
4	5	12	17	12	17

$$\overline{W} = \frac{\sum_{j=1}^{4} W_j}{4} = \frac{0+4+6+12}{4} = \frac{22}{4} = 5.5, \qquad \overline{F} = \frac{\sum F_j}{4} = \frac{39}{4} = 9.75$$

ii) SPT Sequence: (2-1-4-3)

For each job, the computations for the waiting time using the SPT sequence are shown below.

Job (j)	p_j	S_j	C_j	W_j	F_j
2	2	0	2	0	2
1	4	2	6	2	6
4	5	6	11	6	11
3	6	11	17	11	17

$$\overline{W} = \frac{\sum_{j=1}^{4} W_j}{4} = \frac{0+2+6+11}{4} = \frac{19}{4} = 4.75, \qquad \overline{F} = \frac{\sum F_j}{4} = \frac{36}{4} = 9,$$

iii) Random Sequence: (1-3-4-2)

When using the random sequence, the average waiting time calculations is given in the table below;

Job (j)	p_j	S_j	C_j	W_j	Fj
1	4	0	4	0	4
3	6	4	10	4	10
4	5	10	15	10	15
2	2	15	17	15	17

$$\overline{W} = \frac{\sum_{j=1}^{4} W_j}{4} = \frac{0+4+10+15}{4} = \frac{29}{4} = 7.25, \qquad \overline{F} = \frac{\sum F_j}{4} = \frac{46}{4} = 11.50$$

Note that the average values obtained by the SPT sequence are minimum for both $\overline{W}$ and $\overline{F}$.

Example 2.2

Consider a single machine sequencing problem with data as shown below:

Job(j)	1	2	3	4
p_j	4	2	6	5
d_j	8	12	11	10

Use the following sequences to find Makespen or maximum completion time C_{max}, Average waiting time $\overline{W}$, Average tardiness $\overline{T}$, and maximum lateness L_{max}.

i) SPT sequence, ii) EDD sequence

Solution:

i) Job sequence based on SPT is (2-1-4-3). The computations for SPT sequence are presented in the following table

Job(j)	p_j	S_j	C_j	d_j	W_j	L_j	T_j
2	2	0	2	12	0	-10	0
1	4	2	6	8	2	-2	0
4	5	6	11	10	6	1	1
3	6	11	17	11	11	6	6

$$C_{max} = \sum_{j=1}^{j=4} p_j = 17, \quad \overline{W} = \frac{\sum_{j=1}^{4} W_j}{4} = 4.75, \quad \overline{T} = \frac{\sum_{j=1}^{4} T_j}{4} = 1.75, \quad L_{max} = 6$$

ii) Job sequence based on earliest due date (EDD) is (1-4-3-2). The computations for EDD sequence are presented in table below

Job(j)	p_j	S_j	C_j	d_j	W_j	L_j	T_j
1	4	0	4	8	0	-4	0
4	5	4	9	10	4	-1	0
3	6	9	15	11	9	4	4
2	2	15	17	12	15	5	5

$$C_{max} = \sum_{j=1}^{j=4} p_j = 17, \quad \overline{W} = \frac{\sum_{j=1}^{4} W_j}{4} = 7, \quad \overline{T} = \frac{\sum_{j=1}^{4} T_j}{4} = 2.25, \quad L_{max} = 5$$

It should be noted that the value of C_{max} remains the same for both sequences. Also, the value for L_{max} for EDD sequence is smaller than the value of L_{max} for SPT sequence. In addition, the average waiting time and the average tardiness for the SPT sequence are very small when compared the values obtained by the EDD sequence. **What can you infer about these results?**

2.3 MINIMIZATION OF THE MAXIMUM LATENESS PROBLEM *($1||L_{max}$)*

For single machine problems, if due dates (d_j) are specified, then earliest due date (EDD) sequence yields an optimal solution to the maximum lateness L_{max} and the maximum tardiness T_{max}. This rule applies to the class of problems specified by $1||L_{max}$ or $1||T_{max}$ terminology. In such class of problems, it is implicitly assumed that release time of all jobs; $r_j \equiv 0$ (static shop)

Example 2.3

Find an optimal sequence for $1 || L_{max}$ problem. The data is given in the table below. Release time of all jobs is zero; i.e., $r_j = 0$ $(1 \le j \le 6)$. Compute the maximum lateness L_{max} and the average lateness $\bar{L}$.

Job (j)	1	2	3	4	5	6
p_j	10	3	4	8	10	6
d_j	15	6	9	23	20	30

Solution:

The EDD sequence of the 6-job problem from table is;

Job (j)	2	3	1	5	4	6
p_j	3	4	10	10	8	6
D_j	6	9	15	20	23	30

The calculations of Completion times (C_j) and Lateness (L_j) are shown below:

Job (j)	2	3	1	5	4	6
C_j	3	7	17	27	35	41
L_j	-3	-2	2	7	12	11

Maximum Lateness, L_{max} = max $\{ \max\{L_j, 0\};\ j = 1, \cdots 6 \}$ = 12 and the average lateness; $\bar{L} = \frac{\sum_{j=1}^{6} L_j}{6} = \frac{27}{6} = 4.5$.

2.4 MINIMIZATION OF TOTAL WEIGHTED COMPLETION TIME PROBLEM *($1||\Sigma\omega_jC_j$)*

Often the jobs in a shop have priorities attached on their tags which are specified by the term ω_j. To schedule such jobs, Weighted Shortest Processing Time (WSPT) sequence is applied. As a first step to perform WSPT sequence, calculate process time to weight ratio for each job, and, then rank jobs in increasing order of processes time to weight ratio values.

Example 2.4 (WSPT Sequence)

Assume the weight assigned ω_j for each job j (importance or priority) and then solve the following 4-jobs problem using WSPT sequence. Also, compute the total weighted completion times.

Job(j)	p_j	ω_j	p_j/ω_j	d_j
1	4	8	0.5	8
2	2	7	0.285	12
3	6	3	2	11
4	5	15	0.333	10

Solution: Jobs Sequence based on WSPT $\rightarrow$ (2-4-1-3)

Job(j)	p_j	ω_j	S_j	C_j	d_j	W_j	L_j	T_j	ω_jC_j
2	2	7	0	2	12	0	-10	0	14
4	5	15	2	7	10	2	-3	0	105
1	4	8	7	11	8	7	3	3	88
3	6	3	11	17	11	11	6	6	51

Hence, from the above table, the total weighted completion $\Sigma\omega_jC_j$ can be found which is 388.

2.5 MINIMIZATION OF TOTAL WEIGHTED COMPLETION TIME PROBLEM WITH PRECEDENCE RELATIONS *($1|\ PREC|\ \Sigma\omega_jC_j$)*

There are instances when jobs have precedence relationships. As shown in precedence network diagram below, job 3 depends on job 2 and, job 2 depends on job 1.

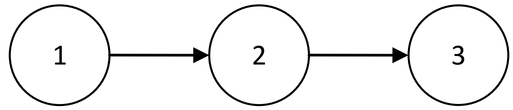


Figure 2.3 Example of precedence network.

In the following example, a single machine problem is presented where jobs have precedence relationship. The objective function is the minimization of total weighted completion time ($\sum_{j=1}^{n} \omega_j C_j$). The solution methodology is based on *Chain method* which is described below.

1. For each set of jobs in the precedence network diagram, form job sets from unscheduled jobs for each chain.
2. Find minimum value of ρ-factor for each chain, where ρ-factor = $\frac{\sum p_j}{\sum \omega_j}$
3. Select the jobs from the chain having overall minimum value of ρ -factor.
4. Include these jobs in the partial schedule, and delete them from the network diagram.
5. Repeat steps (1) to (4) until all jobs are scheduled.

Example 2.5 (1| *prec* | $\Sigma\omega_j C_j$)

Consider the following problem as an instance of the 1| *prec* | $\Sigma\omega_j C_j$. A 7-job single machine data with job precedence constraints graph is given below.

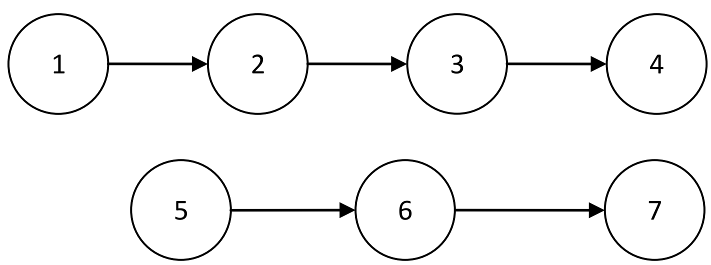


Figure 2.4 Precedence constraints graph.

The weights and process times of the jobs are given in the following table.

Jobs	1	2	3	4	5	6	7
p_j	3	6	6	5	4	8	10
ω_j	6	18	12	8	8	17	18

Solve the problem to minimize total weighted completion times using chain-method.

Solution:

Apply *Chain method* as follows:

For each job set, find ρ-factor = $\frac{\sum p_j}{\sum \omega_j}$ ratio for all job sets:

Suppose job set is: $\{1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4\}$

Then, ρ-factor = $\frac{\sum p_j}{\sum \omega_j} = \frac{\sum p_1 + p_2 + p_3 + p_4}{\sum \omega_1 + \omega_2 + \omega_3 + \omega_4} = \frac{3+6+6+5}{6+18+12+8} = \frac{20}{44}$ = 0.455

The ρ-Factor for chain 1

Job Set	1	$1 \longrightarrow 2$	$1 \longrightarrow 2 \longrightarrow 3$	$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4$
ρ-Factor	$\frac{3}{6}$	$\frac{9}{24}$	$\frac{15}{36}$	$\frac{20}{44}$
	0.5	0.375	0.416	0.455

Minimum value for ρ-Factor is 0.375 for the job set $\{1 \longrightarrow 2\}$.

The ρ-Factor for Chain 2

Job Set	5	$5 \longrightarrow 6$	$5 \longrightarrow 6 \longrightarrow 7$
ρ-Factor	$\frac{4}{8}$	$\frac{12}{25}$	$\frac{22}{43}$
	0.5	0.480	0.512

Minimum value of ρ-Factor is 0.480 for the job set $\{5 \longrightarrow 6\}$.

Comparison of the ρ-Factor of Chain-1 and Chain-2.

	Max ρ-Factor	Job Set
Chain-1	0.375	$1 \longrightarrow 2$
Chain-2	0.48	$5 \longrightarrow 6$

Overall minimum value of ρ-Factor is 0.375 for Chain-1 and job set $\{1 \longrightarrow 2\}$. Thus, the following partial sequence is obtained: {1-2-X-X-X-X-X-X-X}. Then, jobs 1 and 2 should be marked on the network diagram.

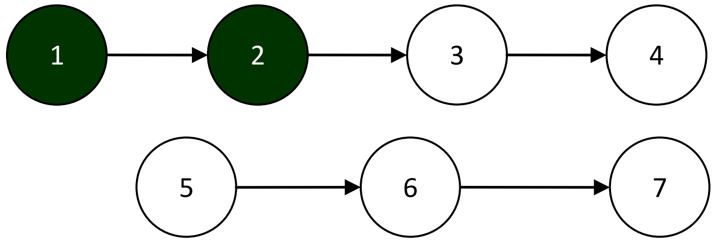


Figure 2.5 Precedence constraints graph when marking Jobs 1 and 2.

The unscheduled jobs in Chain-1 are now; 3 and 4. The corresponding two job sets are; {3} and {3 ⟶ 4}.

Then, the revised ρ-Factor values for these 2 jobs are calculated as follows:

Job Set	3	3 ⟶ 4
ρ-Factor	$\frac{6}{12}$	$\frac{11}{20}$
	0.5	0.55

Next, compare the ρ-factor for the two chains as follows:

	Max ρ-Factor	Job Set
Chain-1	0.5	3
Chain-2	0.48	5 ⟶ 6

Overall minimum value of ρ-Factor is 0.48 which belong to Chain-2 for Job Set: {5 ⟶ 6}. Thus, the Partial Sequence is {1-2-5-6-X-X-X}. Mark jobs 5 and 6 on the network diagram.

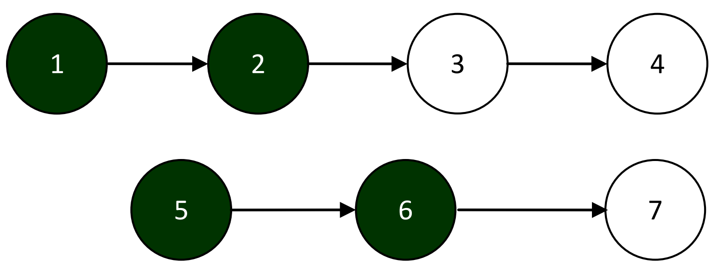


Figure 2.6 Precedence constraints graph when marking Jobs 1, 2, 5, and 6.

Chains			Max ρ_factor	Job Set
Chain-1	Job Set :{3} ρ_factor : 0.5	Job Set : {3 → 4} ρ-Factor : 0.55	0.5	{3}
Chain-2	Job Set :{7} ρ-Factor : 0.55		0.55	{7}

Minimum value of ρ-Factor is 0.5 which belong to Chain-1 for Job Set: {3}. Thus, the Update for the Partial Sequence is as follows: {1-2-5-6-3-X-X}. Next, mark job 3 on the network diagram as shown below.

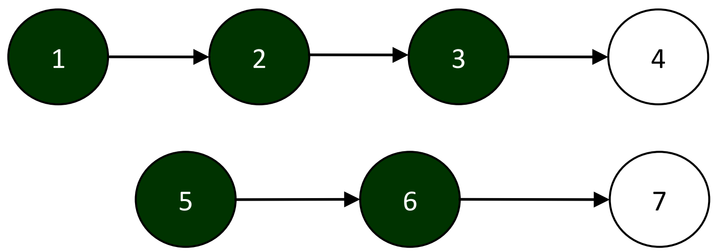


Figure 2.7 Precedence constraints graph when marking Jobs 1, 2, 5, 6, and 3.

For the remaining jobs in Chain-1 and Chain-2 are Job 4 and job 7 respectively. Thus, the Maximum value of ρ-Factor can be computed as shown below.

Chain		Max ρ_factor	Job Set
Chain-1	Job Set :{4} , ρ_factor : 0.625	0.625	{4}
Chain-2	Job Set :{7} , ρ-Factor : 0.55	0.55	{7}

The minimum value of ρ-Factor is 0.55 which belong to Chain-2 for Job Set: {7}. The partial sequence can be updated as follows: {1-2-5-6-3-7-X}. Next, mark job 7 on the network diagram as shown below.

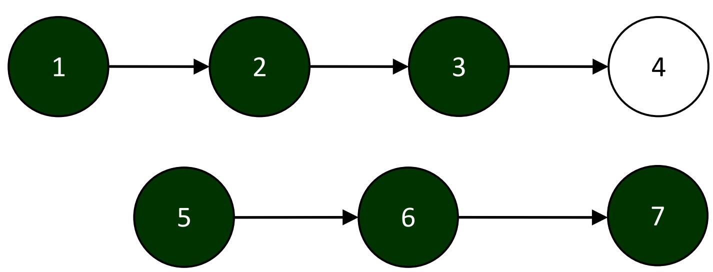


Figure 2.8 Precedence constraints graph when marking Jobs 1, 2, 5, 6, 3, and 7.

There is only one unscheduled job which is job 4. Thus, this job can be attached to the end of the partial sequence which means the final sequence is being developed and it is {1-2-5-6-3-7-4}. Using this final sequence the total weighted completion can be computed as follows:

Job(j)	p_j	ω_j	S_j	C_j	$\omega_j C_j$
1	3	6	0	3	18
2	6	18	3	9	162
5	4	8	9	13	104
6	8	17	13	21	357
3	6	12	21	27	324
7	10	18	27	37	666
4	5	8	37	42	336
				$\Sigma\omega j\ Cj$	1967

2.6 NON-PREEMPTIVE SCHEDULING

This type of scheduling is considered when release time of all jobs in the shop is same. Sequence of jobs is decided by scheduling policy. Once, the sequence is determined, the jobs are loaded on the machine accordingly. The jobs are continuously processed over the machine in the sequencing order. No change in sequencing order is made once machine starts processing the jobs. Every job is processed on the machine on its turn. No job is removed from the machine during its stay on the machine even if a high priority job arrives on that machine. Hence, no interruption is allowed during processing of the jobs and, all jobs complete their processing according to pre-defined sequence.

Example 2.6

Data pertaining to 4-job single machine problem is given in the following table.

Job(j)	p_j	r_j	d_j
1	4	0	8
2	2	3	12
3	6	3	11
4	5	5	10

Generate a non-preemptive schedule using EDD sequence, and then compute the C_{max} and the L_{max}.

Solution:

Since, jobs have distinct r_j values, the problem data presents a dynamic environment. EDD Sequence for the 4-jobs based on due dates is {1-4-3-2}. The following Gantt chart gives complete information on the EDD sequence.

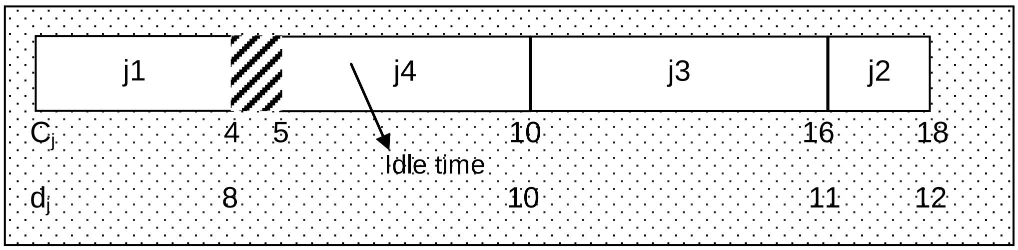


Figure 2.9 Gantt chart for the EDD sequence.

Job 4 arrives at time 5, where as the machine is free at time 4. Hence, there is an idle time from time 4 to time 5. The makespan for the sequence is 18 unit of time. The L_{max} = max {-4, 0, 5, 6} = 6. If the release time of all jobs would have been r_j = 0, then, the makespan is 17 unit of time (Σp_j = 17). Also, in this case, the maximum lateness L_{max} would have been 5.

2.7 PREEMPTIVE SCHEDULING

This type of scheduling is considered when jobs arrive in the shop at different times; i.e., the values of r_j is greater zero and it could be different for each job. In this case, jobs are scheduled according to a pre-determined sequence. However, flexibility is built in the sequencing. This means a job can be removed from the machine if a high priority job is to be processed ahead of currently scheduled job. Then, the low priority jobs or preempted jobs are processed later on.

Example 2.7

Consider example 2.6 once again. If preemption of scheduled jobs is allowed in which the priority is given for jobs with earliest due date (EDD). What will be the new schedule? Is it a better schedule (Why or why not?)

Solution:

The EDD Sequence is {1-4-3-2}

Time, t=0.

Job arrived at time zero is job 1. Job 1 is scheduled at time zero. Then, the machine will complete job1 by time 4.

Time, t=3.

Jobs 2 and 3 arrived in the shop. The machine is busy and can not process any job.

Time, t=4.

Processing of job 1 is completed and the machine becomes free and ready to process any of the waiting jobs. Referring to the EDD sequence, the next job in EDD sequence is job 4, however, it has not arrived yet because its arrival time is at time 5. Thus, the next job in the EDD sequence is picked up which is job 3. Since job3 has arrived already then it is assigned to the machine.

Time, t=5.

Job j4 has arrived in the shop. Currently the machine is processing job 3. From the EDD sequence, it is clear that job 3 has lower priority than job 4. Thus, preempt job 3 from being processed on the machine. Then, assign job 4 to the machine. Job j_4 has a process time of 5 unit of time which means it will be completed at time 10.

Time, t=10

Processing on job 4 is completed. Then, the next job in the EDD sequence is job 3. This job has already been partially processed from time t=4 to t=5. Its remaining process time is 5 unit of time. Thus, assign job 3 to the machine and will be completed at time 15.

Time, t=15

Processing on job j_3 is completed. Then, the next job in the EDD sequence is job 2. Its process time is 2 unit of time. Next, assign job 2 to the machine and be completed at time 17.

Time, t=17.

Processing on job 2 is completed. There is no other job to be processed, hence the procedure is STOP.

The following Gantt chart presents the schedule of the jobs

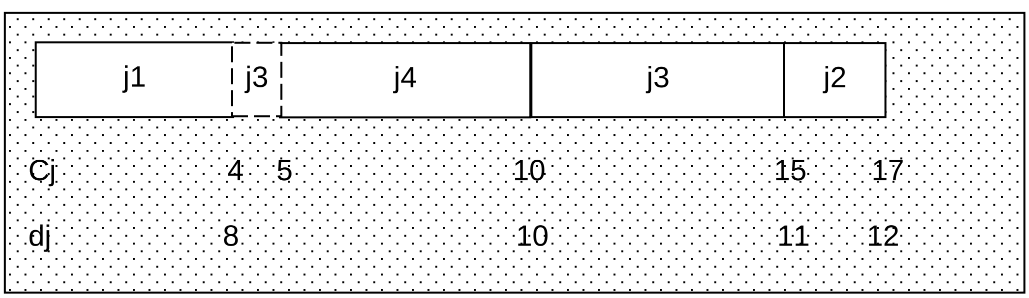


Figure 2.10 Gantt chart for the preemptive EDD schedule.

From the Gantt chart above, it should be clear that the values of makespan and maximum lateness are as follows: makespan =17 and L_{max}=5.

It should be clear that for this problem data, the preemptive scheduling provides a better solution than non-preemptive schedule.

2.8 NUMBER OF TARDY JOBS *($1 \parallel n_t$)*

A popular scheduling objective is minimization of number of tardy jobs. It is expressed as $1 \parallel n_t$ problem in scheduling terminology. Hodgson's Algorithm provides an optimal solution for this type of problem. Hodgson's Algorithm has the following steps:

Step 1: Order the jobs in EDD sequence.

Step 2: Build the Gantt chart for the EDD sequence and compute T_j values for all jobs; $(1 \leq j \leq n)$

Step 3: If T_j values for all jobs are zero (this means there is no tardy job). STOP. The EDD sequence is an optimal sequence for the scheduling problem under consideration $1 \parallel n_t$ Otherwise continue to Step 4.

Step 4: For the current sequence, find the first tardy job in the sequence, say k.

Step 5: Remove job j among the scheduled job so far $(1 \leq j \leq k)$ with longest process time from the sequence and put it in the set of tardy jobs. Then, Go To Step 2.

Example 2.8

Solve the following $1 \parallel n_t$ problem in which the following data is given:

Job (j)	1	2	3	4	5	6
p_j	10	3	4	8	10	6
d_j	15	6	9	23	20	30

Solution:

Rearrange jobs according to EDD sequence as follows;

Job (j)	2	3	1	5	4	6
p_j	3	4	10	10	8	6
d_j	6	9	15	20	23	30

The calculations of completion times (C_j) and tardiness (T_j) are shown in the table below. The number of tardy jobs; n_t is equal to 4.

Job (j)	2	3	1	5	4	6
p_j	3	4	10	10	8	6
C_j	3	7	17	27	35	41
L_j	-3	-2	2	7	12	11
T_j	0	0	2	7	12	11

From Step 4, the first tardy job in the sequence is job 1. According to Step 5, jobs 2, 3, and 1 are candidates to be removed to the set of the tardy jobs. Since job 1 has the largest processing time p_j. Then, remove job 1 from the current scheduled and put it in the set of tardy jobs which means job 1 will be attached to the end or in the last position in the sequence.

Shifting job 1 to last sequence position

Job (j)	2	3	1	5	4	6
p_j	3	4	10	10	8	6

Recalculate T_j and C_j values as shown below in Table.

Job (j)	2	3	5	4	6	1
p_j	3	4	10	8	6	10
d_j	6	9	20	23	30	15
C_j	3	7	17	25	31	41
L_j	-3	-2	-3	2	1	26
T_j	0	0	0	2	1	26

The first tardy job in new schedule is job 4 (k = 4). From the scheduled job set {2 – 3 – 5 – 4}, job 5 has largest processing time p_j. Thus, shift job 5 to last position in the sequence as shown below.

Shifting job 5 to last sequence

Job (j)	2	3	5	4	6	1
p_j	3	4	10	8	6	10

After shifting job 5 to last sequence position, recalculate Tj and Cj as shown in Table below.

Job (j)	2	3	4	6	1	5
p_j	3	4	8	6	10	10
D_j	6	9	23	30	6	20
C_j	3	7	15	21	31	41
L_j	-3	-2	-8	-9	25	21
T_j	0	0	0	0	25	21

It should be clear that there is no tardy job in the scheduled job set. Hence, there are only two tardy jobs; namely, jobs 1 and 5. Therefore, the total number of tardy jobs; n_t is equal to 2. The following are the two optimal sequences for this problem: {2 – 3 – 4 – 6 – 1 – 5} and {2 – 3 – 4 – 6 – 5 – 1}

2.9 BRANCH & BOUND METHOD

The branch and bound method uses a sequence tree. Each node in the tree contains a partial sequence of jobs. For n job problem, there are n-1 numbers of levels for a tree. At level zero, root node will be placed with all n empty sequence positions. At level 1, there will be n number of nodes. Each node will contain a partial sequence of jobs. The first position in the sequence will be occupied by a job in numerical order. Similarly, each node at $(n-1)^{th}$ level will be branched to (n-2) number of nodes. The process will continue till each node has exactly one *leaf*. The construction of a tree for generating all sequences for a 3-job problem is presented in the Figure 11.

Generation of all sequences is combinatorial in nature and, will result in enormous number of sequences even for a small number of jobs. For example, for a 10-job problem there will be 10! Sequences. To reduce the computational effort, lower bounds are calculated at every level for each node. The formula used to compute the lower bound is pertinent to objective function of the scheduling problem. Branching is carried out only from those nodes with the minimum lower bound. By doing so, only small proportion of the nodes is explored resulting in fewer amounts of computations. The branch and bound (B&B) method is applied in almost every scheduling problem. In the following paragraph, this methodology is applied to solve single machine problems where jobs have distinct release times (r_j) and objective function is to minimize L_{max}. As scheduling terminology implies, these types of problems as termed as $1 \mid r_j \mid L_{max}$ problem.

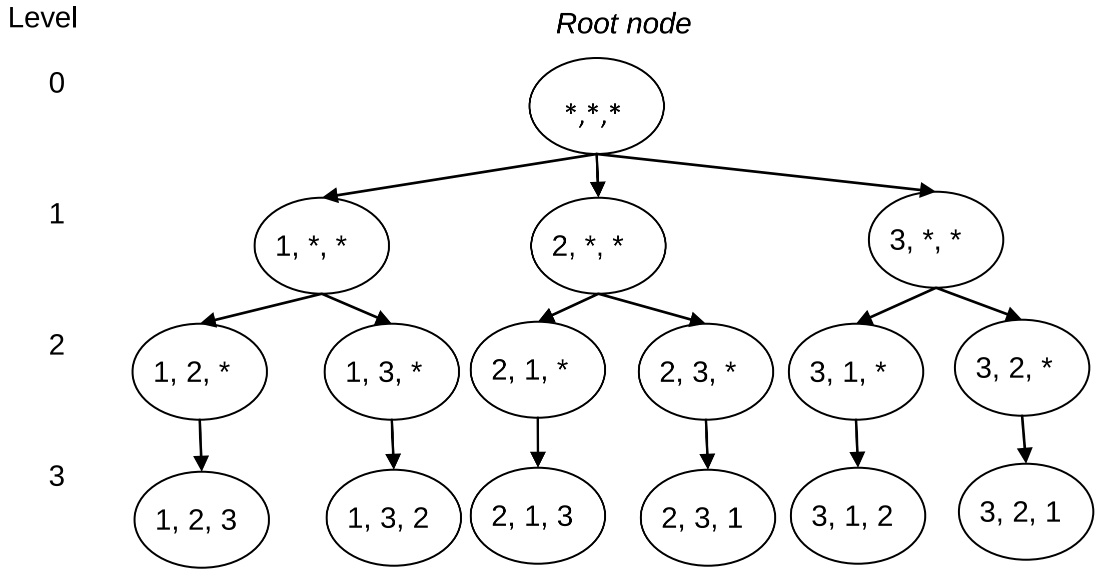


Figure 2.11 Sequence Generation tree.

2.10 MINIMIZATION OF MAXIMUM LATENESS WITH READY TIME PROBLEM *($1 \mid r_j \mid L_{max}$)*

In this section, the branch and bound (B&B) method is used to solve the following scheduling problem: $1 \mid r_j \mid L_{max}$. The following guidelines should be followed when developing the scheduling generating tree for the branch and bound for the problem under consideration:

1. At any level of the tree, generate only those child nodes for the selected parent node which satisfies the following relationship:

$$r_j < \min_{k \in J} \{\max(t, r_k) + p_k\}$$

Where:

J = set of jobs not yet scheduled

t = time at which job j is supposed to start.

This mean only those jobs which are ready to be processed will be considered.

2. Compute the lower bounds value for all nodes at any level using **preemptive EDD scheduling**. This means, at every node the L_{max} is computed using the developed preemptive EDD schedule. Then, pick the node(s) (sequence(s)) having minimum L_{max} value for further branching to lower level.

The following example will demonstrate the implementation of the branch and bound solving the following scheduling problem: $1 \mid rj \mid L_{max}$.

Example 2.9

The following Table presents an instance of 1 | rj | L_{max} problem. Find an optimal solution for this problem using the branch and bound method.

Job (j)	1	2	3	4
p_j	4	2	6	5
d_j	8	12	11	10
r_j	0	1	3	5

Solution:

At the start, there are four possible nodes at Level 1 as shown in the figure below.

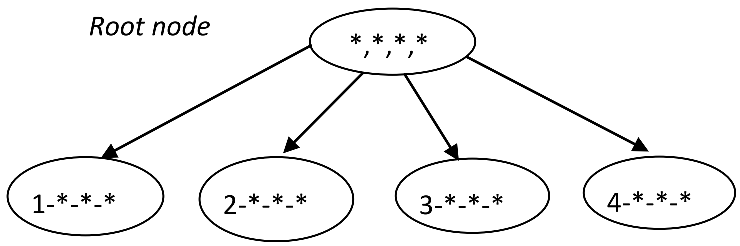


Figure 1 Level 1 Nodes for 4-job Problem.

Iteration I – Step (1)

For the four nodes at level 1, verify the condition;

$$r_j < \min_{k \in J} \{\max(t, r_k) + p_k\}$$

The following paragraphs will consider each partial sequence one by one as follows:

a) **Partial Sequence: (1-*-*-*)**

$r_1 = 0$ **t=0**

Unscheduled jobs are (2-3-4). The minimum completion time for all these jobs can be computed as follows.

K	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
2	1	1	2	3
3	3	3	6	9
4	5	5	5	10
			Min	3

Since $r_1 < 3$, then, include this node (1-*-*-*) in the tree.

b) Partial Sequence: (2-*-*-*)

r_2 = 1 t=0

Unscheduled jobs are (1-3-4). The minimum completion time for all these jobs can be computed as follows.

K	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
1	0	0	4	4
3	3	3	6	9
4	5	5	5	10
			Min	4

Since r_2 < 4, then, include this node (2-*-*-*) in the tree.

c) Partial Sequence: (3-*-*-*)

r_3 = 3 t=0

Unscheduled job are (1-2-4). The minimum completion time for all these jobs can be computed as follows:

K	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
1	0	0	4	4
2	1	1	2	3
4	5	5	5	10
			Min	3

Since r_3 is not less than 3. Then, <u>do not include</u> this node (3-*-*-*) in the tree.

d) Partial Sequence: (4-*-*-*)

r_4 = 5 t=0

Unscheduled jobs are (1-2-3). The minimum completion time for all these jobs can be computed as follows:

K	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
1	0	0	4	4
2	1	1	2	3
3	3	6	5	11
			Min	3

Since r_4 is not less than 3. Then, <u>do not include</u> this node (4-*-*-*) in the tree.

Iteration – I: Step (2)

Find L_{max} for the sequences (1-*-*-*) and (2-*-*-*) as follows:

EDD sequence based on the due dates is as follows: (1-4-3-2)

Schedule for the sequence (1-*-*-*) using preemptive scheduling is as follow:

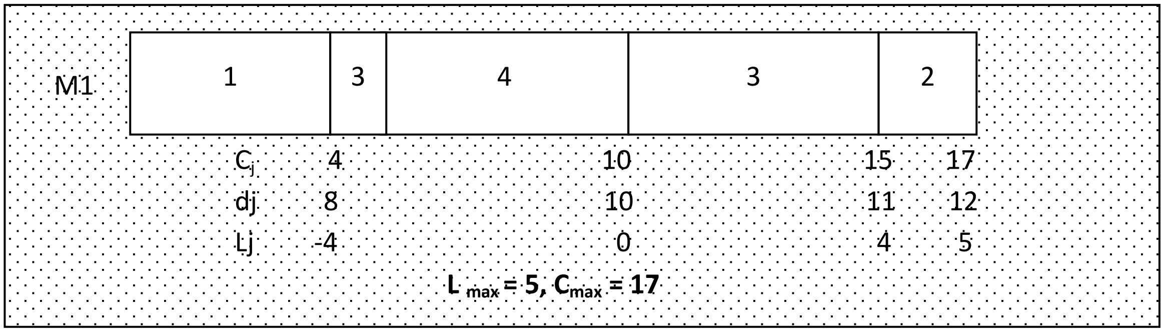


Figure 2.13 Gantt chart for the partial sequence (1-*-*-*).

Schedule for the sequence (2-*-*-*) using preemptive scheduling is as follows:

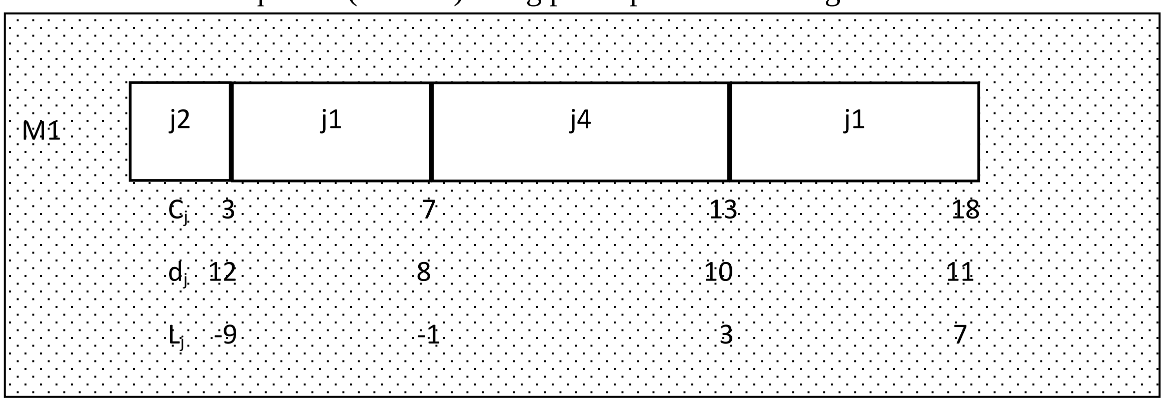


Figure 2.14 Gantt chart for the partial sequence (2-*-*-*).

From Figure 13, it should be clear that the machine has been idle during the time interval [0, 1] because the first job in sequence is job 2 and its ready time is 1. The lower bound (LB) for node (1-*-*-*) is lower than the LB for node (2-*-*-*). This means, branching should be continued from node (1-*-*-*). The following figure shows the branch from node (1-*-*-*).

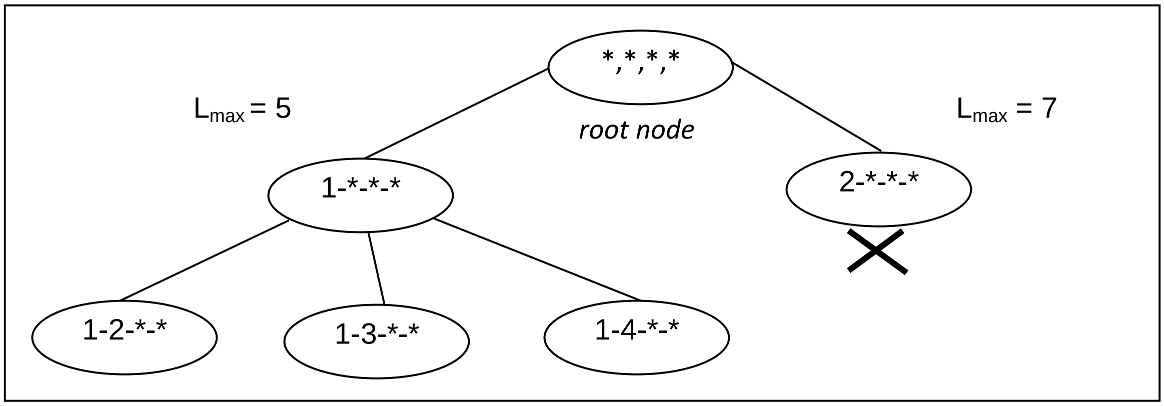


Figure 2.15 Level 2 Nodes for 4-job Problem.

It should be clear from the figure above, there are three nodes emanating from node (1-*-*-*). Next, verify the following condition in order to determine which node(s) to consider:

$$r_j < \min_{k \in J} \{\max(t, r_k) + p_k\}$$

Iteration 2 – Step (1)

a) **Partial Sequence: (1-2-*-*)**

$r_2 = 1$ t=4

Unscheduled jobs are (3-4). The minimum completion time for all these jobs can be computed as follows:

K	r_k	$S_k=\max(t,r_k)$	p_k	$S_k + p_k$
3	3	4	6	10
4	5	5	5	10
			Min	10

Since $r_2 < 10$, then, include this node (1-2-*-*) in the tree.

b) **Partial Sequence: (1-3-*-*)**

$r_3 = 3$ t=4

Unscheduled jobs are (2-4). The minimum completion time for all these jobs can be computed as follows:

K	r_k	$S_k=\max(t,r_k)$	p_k	$S_k + p_k$
2	1	4	2	6
4	5	5	5	10
			Min	6

Since $r_3 < 6$, then, include this node (1-3-*-*) in the tree.

c) Partial Sequence: (1-4-*-*)

$r_4 = 5$ t=4

Unscheduled jobs are {2-3}. The minimum completion time for all these jobs can be computed as follows:

k	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
2	1	4	2	6
3	3	4	5	9
			Min	6

Since $r_4 < 6$, then, include the node (1-4-*-*) in the tree.

Iteration 2: Step (2)

Find L_{max} for the three sequences as follows:

In sequence (1-2-*-*), the first and the second positions are already assigned to jobs 1 and 2 respectively. For the remaining two positions, based on the EDD sequence (1-4-3-2), job 4 will be assigned to position three and job 3 will be assigned to position four. The computation for the lower bound which is L_{max} is as follows:

job (j)	p_j	r_j	S_j	C_j	d_j	L_j
1	4	0	0	4	8	-4
2	2	1	4	6	12	-6
4	5	5	6	11	10	1
3	6	3	11	17	11	6

Hence, from the table above the L_{max} is equal to 6.

In sequence (1-3-*-*), the first and the second positions are already assigned to jobs 1 and 3 respectively. For the remaining two positions, based on the EDD sequence, job 4 will be assigned to position three and job 2 will be assigned to position four. The computation for the lower bound is as follows:

job (j)	p_j	r_j	S_j	C_j	d_j	L_j
1	4	0	0	4	8	-4
3	6	3	4	10	11	-1
4	5	5	10	15	10	5
2	2	1	15	17	12	5

Hence, from the table above the L_{max} is equal to 5.

In sequence (1-4-*-*), the first and the second positions are already assigned to jobs 1 and 4 respectively. For the remaining two positions, based on the EDD sequence, job

3 will be assigned to position three and job 2 will be assigned to position four. The computation for the lower bound is as follows:

job (j)	p_j	r_j	S_j	C_j	d_j	L_j
1	4	0	0	4	8	-4
4	5	5	5	10	10	0
3	6	3	10	16	11	5
2	2	1	16	18	12	6

Hence, from the table above the L_{max} is equal to 6. When the lower bound values for the three nodes are compared, it should be clear that node (1-3-*-*) has the minimum value. Thus, from this node branching should continue as shown in the Figure below.

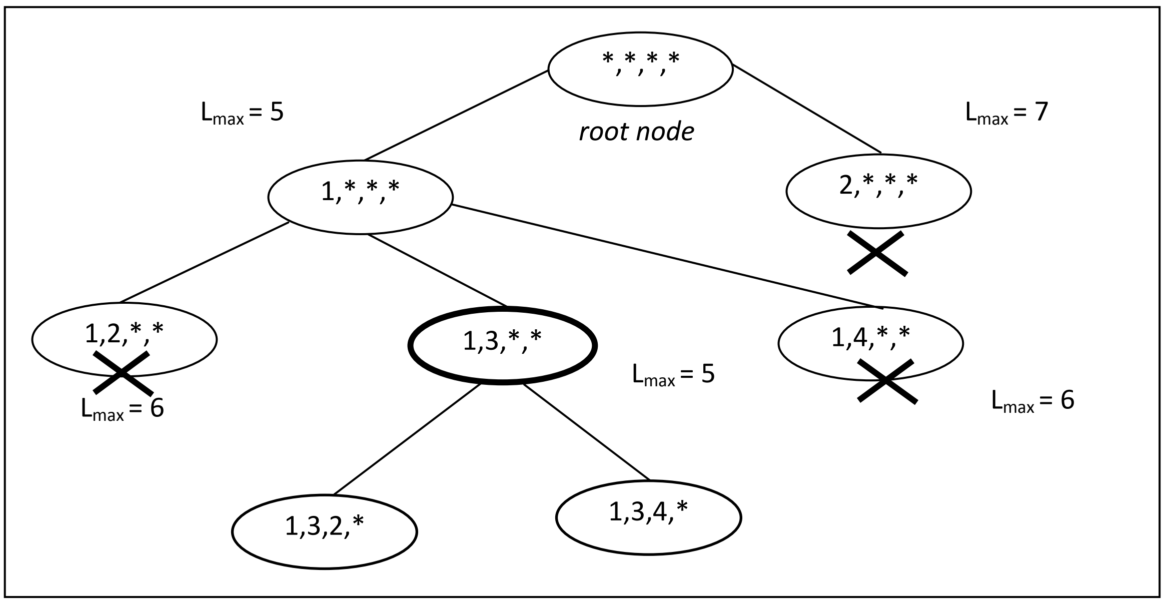


Figure 2.16 Level 3 Nodes for 4-job Problem.

Next, verify the following condition in order to determine which node(s) to consider:

$$r_j < \min_{k \in J} \{\max(t, r_k) + p_k\}$$

Iteration 3 – Step (1)

a) **Partial Sequence: (1-3-2-*)**

  $r_2 = 1$ **t=10**

  Unscheduled job is {4}. The minimum completion time for this job can be computed as follows:

K	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
4	5	10	5	15
			Min	15

Since r_2 < 15, then, include this node (1-3-2-*) in the tree.

b) Partial Sequence : (1,3,4,*)

r_2 = 5 t=10

Unscheduled job is {2}. The minimum completion time for this job can be computed as follows:

K	r_k	S_k=max(t,r_k)	p_k	$S_k + p_k$
2	1	10	2	12
			Min	12

Since r_4 < 12, then, include this node (1-3-4-*) in the tree.

Iteration 3: Step (2)

Find L_{max} for the two sequences as follows:

In sequence (1-3-2-*), the first position, the second position, the third position are already assigned to jobs 1, 3, and 2 respectively. For the remaining job which is job 4, it will be assigned to the last position. The computation for the lower bound will be as follows:

job (j)	p_j	r_j	S_j	C_j	d_j	L_j
1	4	0	0	4	8	-4
3	6	3	4	10	11	-1
2	2	1	10	12	12	0
4	5	5	12	17	10	7

Hence, from the table above the L_{max} is equal to 7.

In sequence (1-3-4-*), the first position, the second position, the third position are already assigned to jobs 1, 3, and 4 respectively. For the remaining job which is job 2, it will be assigned to the last position. The computation for the lower bound will be as follows:

job (j)	p_j	r_j	S_j	C_j	d_j	L_j
1	4	0	0	4	8	-4
3	6	3	4	10	11	-1
4	5	5	10	15	10	5
2	2	1	15	17	12	5

Hence, from the table above the L_{max} is equal to 5. When the lower bound values for the two nodes are compared, it should be clear that node (1-3-4-2) has the minimum value. This concluded the procedure of the branch and bound and also means the optimal sequence has been obtained. The optimal sequence is **{1-3-4-2}.**

2.11 MINIMIZATION OF TOTAL WEIGHTED TARDINESS PROBLEM *($1 \| \Sigma\omega_j T_j$)*

In classical scheduling theory, minimization of total weighted tardiness has been researched thoroughly. A variety of approaches have been developed. In the following pages, one of these approaches which is the branch and bound (B&B) methodology will be explored in solving this problem. In order to reduce the solution space for $\Sigma\omega_j T_j$ which means minimizing the search effort for near optimal solution in the solution space, consider the following lemma which helps in build relative relationship among jobs which produces precedence constraints among some of the jobs.

Lemma:

When minimizing $1 \| \Sigma\omega_j T_j$ problem, and for any two jobs say j and k, the following is true:

$P_j \leq P_k$

$d_j \leq d_k$ and

$W_j \geq W_k$,

Then there exists an optimal sequence that minimizes $1\| \Sigma\omega_j T_j$ problem in which job j appears before job k.

As mentioned earlier, this lemma will help in build relative relationship among jobs which produces precedence constraints among some of the jobs which consequently helps in eliminating a significant number of possible sequences which reduces the solution space and search effort.

Method:

Step 1: For the given problem data, identify job's sequence position.

Step 2: Construct a branch and bound tree with only nodes which contains possible jobs to be in the last position in the sequence. Use the Lemma mentioned above to construct those nodes at level one of the tree.

Step 3: Find the lower bound for each node developed. This means compute the $\Sigma\omega_j T_j$ as the lower bound for each node.

Step 4: Branch from the node(s) which has (have) the minimum value(s) of Lower bound.

Step 5: When branching to a lower level node, include only those jobs which satisfy the Lemma mentioned above.

Step 6: Continue branching to lowest level of the tree till all the jobs are included in the schedule.

Example 2.10

Find an optimal sequence for $1 \,||\, \Sigma\omega_j T_j$ problem given the data in the following table:

Job(j)	1	2	3	4
w_j	4	5	3	5
p_j	12	8	15	9
d_j	16	26	25	27

Solution:

The first step is to determine the set of jobs that satisfy the relationship condition in the lemma mentioned earlier in which the conditions for any jobs are as follows:

$$p_j \le p_k \quad \text{and} \quad d_j \le d_k \quad \text{and,} \quad \omega_j \ge \omega_k$$

By the applying this relationship, it can be found that in an optimal sequence job 1 will appear before job 3. Similarly, job 2 will appear before job 4. Consequently, for the B&B tree, only two nodes can be constructed which represent the partial sequences as shown in the following figure.

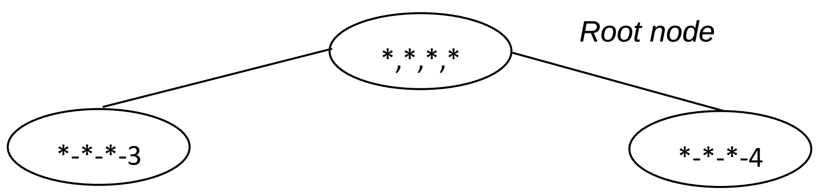
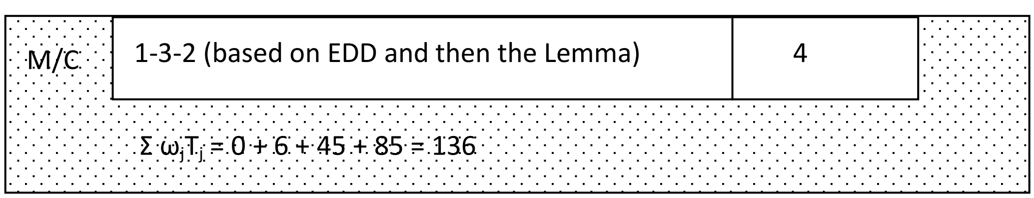


Figure 3 Level 1 Nodes for 4-job Problem.

For the node with partial sequence (*-*-*-4), the lower bound (LB) which is $\Sigma\omega_j T_j$ can be computed as follows:



Similarly, the lower bound (LB) for node with partial sequence (*-*-*-3) can be computed as follows:

M/C	1-2-4 (based on the EDD and then the Lemma)	3

$\Sigma\, \omega_j T_j = 0 + 0 + 10 + 57 = 67$

Thus, since the node with partial sequence (*-*-*-3) has a smaller lower bound value, then, branching should be continued with this node. This means, the node with partial sequence (*-*-*-4) should be fathomed. Only two nodes can be investigated with the two partial sequences (*-*-4-3) and (*-*-1-3) as shown in the figure below.

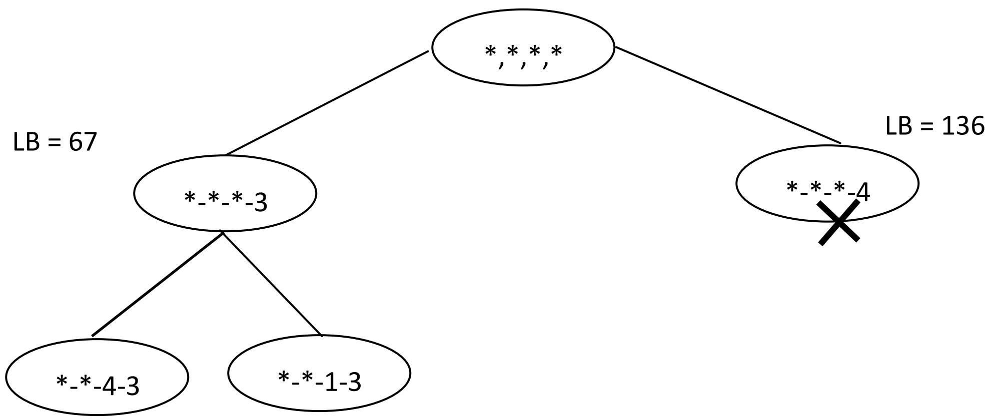


Figure 2.18 Level 2 Nodes for 4-job Problem.

The lower bound for the partial sequence (*-*-4-3) which is $\Sigma\omega_j T_j$ can be computed as follows:

M/C	1-2 (based on EDD)	4	3

$\Sigma\, \omega_j T_j = 0 + 0 + 10 + 57 = 67$

The lower bound for the partial sequence (*-*-1-3) which is $\Sigma\omega_j T_j$ can be computed as follows:

M/C	2-4(based on EDD & Lemma)	1	3

$\Sigma\, \omega_j T_j = 0 + 0 + 52 + 57 = 109$

Therefore, since the node with partial sequence (*-*-4-3) has lower bound value, then, branching should be continued from this node. This means the node with partial sequence (*-*-1-3) should be fathomed. Only two nodes can be investigated with the two partial sequences (*-1-4-3) and (*-2-1-3) as shown in the figure below.

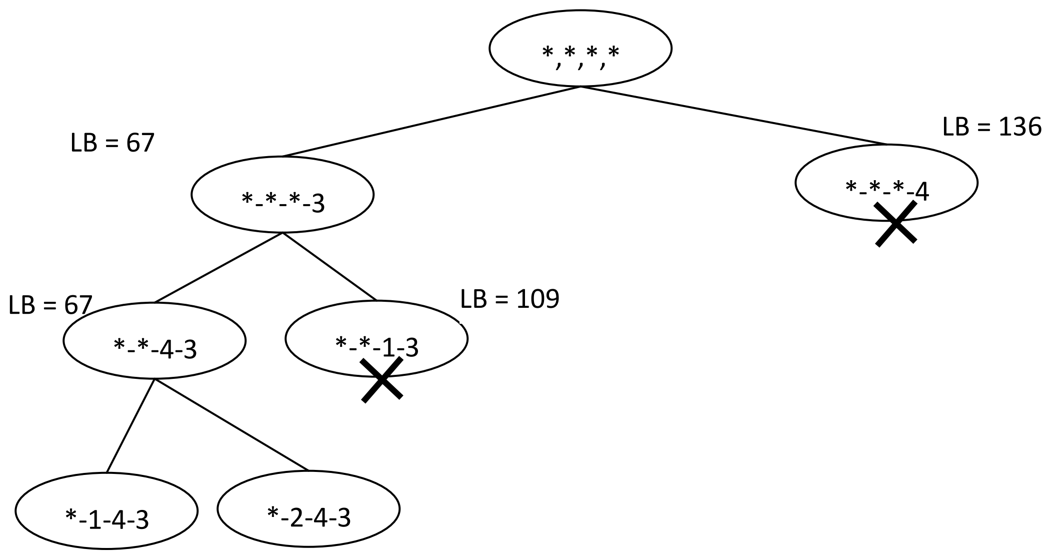


Figure 3.19 Level 3 Node for 4-job problem.

The lower bound for the partial sequence (*-1-4-3) which is $\Sigma\omega_j T_j$ can be computed as follows:

M/C	2 (only job left)	1	4	3

$\Sigma\, \omega_j T_j = 0 + 16 + 10 + 57 = 83$

The lower bound for the partial sequence (*-2-4-3)) which is $\Sigma\omega_j T_j$ can be computed as follows

M/C	1 (only job left)	2	4	3

$\Sigma\, \omega_j T_j = 0 + 0 + 10 + 57 = 67$

Thus, since the node with partial sequence (*-2-4-3) has lower bound value, then, an optimal sequence has been found which is (1-2-4-3) because no more branching can be done all jobs have been scheduled. The complete computation for the schedule is given in the following table:

Job(j)	Pj	wj	Cj	dj	Tj	ωjTj
1	12	4	12	16	0	0
2	8	5	20	26	0	0
4	9	5	29	27	2	10
3	15	3	44	25	19	57

Total weighted Tardiness, $\Sigma\omega_j T_j = 67$. The same approach can be used to solve the following scheduling problem: $1 \,||\, \Sigma\, T_j$. This means, the problem is solved with given that all jobs have equal weight (ω_j=1).

2.12 MINIMIZATION OF MAXIMUM LATENESS WITH PRECEDENCE PROBLEM *(1 | PREC | L_{max})*

In this section, the scheduling problem has jobs that have precedence relationship among them. While scheduling jobs on the single machine, the objective is to minimize the maximum lateness (L_{max}). One of the well known solution methodology to solve this problem is based on the least remaining slack (RS) rule. This rule is applied to solve 1 | prec | L_{max} problem.

The RS_j can be computed as follows:

$$RS_j = d_j - p_j - t.$$

where,

RS_j = the remaining slack of job j.

p_j = processing time of job j.

d_j = due date for job j.

t = time of the schedule,

The algorithm to implement the RS rules is as follows:

Step 1: At time zero, set t = 0.
Step 2: From the given precedence graph, form the set of schedulable jobs.
Step 3: Calculate the RS values of these jobs.
Step 4: Select a job j having minimum value of RS_j and schedule it on the machine.
Step 5: Remove the recently scheduled job from the set schedulable jobs.
Step 6: If all jobs have been scheduled, STOP. Otherwise update the set of schedulable job from precedence graph. Update value of t. Go to step 3.

Example 2.11
Consider 1| *prec* | L_{max} problem with the data given in the following table:

Job (j)	1	2	3	4	5	6	7	8
p_j	2	3	2	1	4	3	2	2
D_j	5	4	13	6	12	10	15	19

Also, the precedence network for the jobs is given in the following figure:

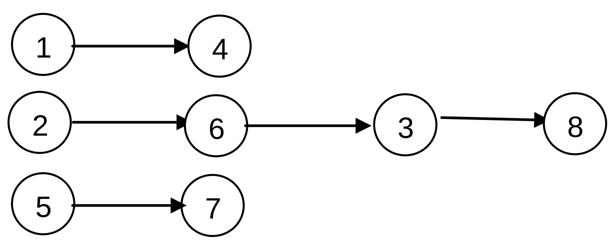


Figure 2.20 Precedence Graph of 8-Job problem

Solution:

At time zero, t = 0. Set of schedulable jobs based on precedence graph are contains the following jobs: {1-2-5}. The computation for the RS values for these jobs is given in the table:

Job(j)	d_j	P_j	t	RS_j
1	5	2	0	3
2	4	3	0	1
5	12	4	0	8
			MIN(RS_j)	1

Job 2 has the minimum RS value. Then, Job 2 is scheduled first on the machine at time 0 and it will be completed at time 3. Thus, updated the value of t to be = 3. The set S = {2}.

Next, job 6 is added to the Set of schedulable jobs. This means the set contains the following jobs: {1-5-6}. The computation for the RS values for these jobs in the set is given in the following table:

Job(j)	d_j	p_j	t	RS_j
1	5	2	3	0
5	12	4	3	5
6	10	3	3	4
			MIN(RS_j)	0

Since job 1 has the minimum RS value, then, it is scheduled second on time 3 the machine. The job will be completed ate time 5. Therefore, the updated value for t is = 5. The set S = {2-1}.

Next, job 4 is added to the schedulable jobs list. The set of schedulable jobs is {5-6-4}. Out of the three jobs in the set, job 4 has the minimum RS value as shown in the following table:

Job(j)	dj	Pj	t	RSj
4	6	1	5	0
5	12	4	5	3
6	10	3	5	2
			MIN(RSj)	0

Thus, job 4 is scheduled at time 5 and it will be completed at time 6. The set S = {2-1-4}. This will update the t value to be = 6. At this time, only two jobs are in the schedulable set. These jobs are {5-6}. Out of these two jobs, job 6 has the minimum RS value as shown in the following table:

Job(j)	d_j	p_j	t	RS_j
5	12	4	6	2
6	10	3	6	1
			MIN(RS_j)	1

Then, job 6 is scheduled at time 6 and it will be completed at time 9. The set S = {2-1-4-6}. This will updated the t value to be = 9. At this time, job 3 is added to the schedulable set. The jobs in the schedulable set are {5-3}. Out of these two jobs, job 5 has the minimum value of RS as shown in the following table:

Job(j)	d_j	p_j	t	RS_j
3	13	2	9	2
5	12	4	9	-1
			MIN(RS_j)	-1

Therefore, job 5 is scheduled at time 9 and completed at time 13. The set S = {2-1-4-6-5}. This will updated the value of t to be = 13. At this time, job 7 is added to the schedulable set of jobs. The set of schedulable jobs contains the following jobs: {3-7}. Out of these two jobs, job 3 has the minimum value of RS as shown in the following table.

Job(j)	d_j	p_j	t	RS_j
3	13	2	13	-2
7	15	2	13	0
			MIN(RS_j)	-2

Next, job 3 is scheduled at time 13 and it will be completed at time 15. The set S = {2-1-4-6-5-3}. The value of t is = 15. At this time, job 8 is added to the schedulable set. The set of schedulable jobs is {7-8}. Out of these two jobs, job 7 has the minimum value of RS as shown in the following table:

Job(j)	d_j	p_j	t	RS_j
7	15	2	15	-2
8	19	2	15	2
			MIN(RS_j)	-2

Thus, Job 7 is scheduled at time 15 and it will be completed at time 17. The set S = {2-1-4-6-5-3-7}. The value of t is = 17. The only unscheduled job is job 8 with RS value of 0 as shown in the following table:

Job(j)	d_j	P_j	t	RS_j
8	19	2	17	0
			MIN(RS_j)	0

Then, job 8 is scheduled at time 17 and it will be completed at time 19. The schedulable job set is empty. Thus, STOP. The final sequence of jobs on the machine is as follows: {2-1-4-6-5-3-7-8}. The Gantt chart for this sequence is shown below:

M/C	2	1	4	6	5	3	7	8
$C_j \rightarrow$	3	5	6	9	13	15	17	19
$d_j \rightarrow$	4	5	6	10	12	13	15	19
$L_j \rightarrow$	-1	0	0	-1	1	2	2	0

From the Gantt chart, the maximum lateness (L_{max}) is 2.

EXERCISES

2.1 A single machine facility faces problem of sequencing the production work for six customer orders described in table below.

Order	1	2	3	4	5	6
Processing time	18	26	14	8	17	22

a. What sequence will minimize the mean flow time of these orders? What is the mean flow time in this schedule?

b. Suppose that customer orders 1 and 5 are considered twice important as the rest what sequence would you propose.

2.2 Consider the following single machine scheduling problem

Jobs (j)	1	2	3	4	5
P_j	8	3	3	6	3
d_j	4	8	11	10	11

a. What sequence will minimize the average waiting time? Then compute Average completion time, Maximum flow time, and Average waiting time.

b. What sequence will minimize both maximum lateness and the maximum tardiness? Then, compute maximum lateness, maximum tardiness, maximum earliness, total tardiness, and total earliness.

2.3 Consider the following single machine scheduling problem

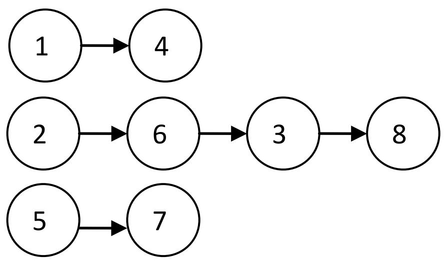
Job	1	2	3	4	5	6
P_j	8	12	7	16	9	4
ω_j	4	10	4	3	8	10

Generate a sequence to minimize weighted completion time. What is the flow time of each job in the shop?

2.4 Show that for any job i:

$\mathbf{L_i = F_i - a_i = C_i - r_i - a_i = C_i - d_i}$, and hence conclude that $\mathbf{\overline{L} = \overline{F} - \overline{a} = \overline{C} - \overline{r} - \overline{a} = \overline{C} - \overline{d}}$

2.5 Consider the following problem as an instance of the $1 | prec | \sum w_j C_j$. An 8-job single machine data with job precedence constraints graph is given below.



The weights and process times of the jobs are given in the following table.

Job	1	2	3	4	5	6	7	8
P_j	3	6	6	5	4	8	10	4
w_j	6	18	12	8	8	17	18	15

Solve the problem to minimize total weighted completion times using chain-method.

2.6 Consider the following single machine scheduling problem

Job	1	2	3	4	5	6	7
p_j	8	3	12	1	7	5	3
D_j	12	7	23	4	21	17	8

a) Use SPT sequence and, find average waiting time ($\overline{W}$).
b) Use EDD sequence and, find maximum lateness ($L_{\max}$), average Lateness ($\overline{L}$).
c) Generate a sequence to minimize number of tardy jobs (n_t).

2.7 Solve the following $1 \| n_t$ problem in which the following data is given:

Job	1	2	3	4	5	6	7
P_j	9	4	3	7	10	6	8
d_j	15	7	5	12	20	23	30

2.8 Consider 1 ‖ n_t problem with the following data:

Job	1	2	3	4	5	6	7	8	9	10
P_j	15	11	10	5	25	4	8	3	20	11
d_j	71	76	73	88	47	59	24	55	23	47

Find the sequence that minimizes the number of jobs tardy and compute n_t

2.9 Bin-laden contracting company has orders for five houses to be built. Bin-laden is well known company and has good reputation for excellence, thus, the customers will wait as long as necessary for their house to be built. The revenue in Saudi Riyals to Bin-laden for each house respectively is as follows: 145000, 290000, 910000, 1150000, and 200000. Also, the times needed in days to build each house respectively are as follows: 150, 200, 400, 450, and 1000. Assuming that Bin-laden, can only work on one house at time, what would be an appropriate measures to schedule building the houses? Using this measure, what schedule should Bin-laden?

2.10 At Toyota (Fast service) repair shop there are six cars in for repair. The car's owners will wait in the waiting and will leave when their cars are finished. At night shift, there is only one mechanic available to do the repairs whose name is Mohammed. Mohammed estimates the times needed for repair for each cars respectively as follows: 115, 145, 40, 25, 70, and 30 minutes. What schedule would you recommend for Mohammed? How would you help Mohammed to justify having another mechanic with him? (**Show all of your work and show all assumptions you make**)

2.11 Consider an instance of the $1 \mid r_j \mid L_{max}$ problem with data as follows.

Job	1	2	3	4
p_j	3	4	6	10
r_j	6	0	7	6
d_j	3	10	17	18

Find optimal sequence and show complete schedule.

2.12 Consider the following data as an instance of the $1 \parallel \Sigma w_j T_j$ problem;

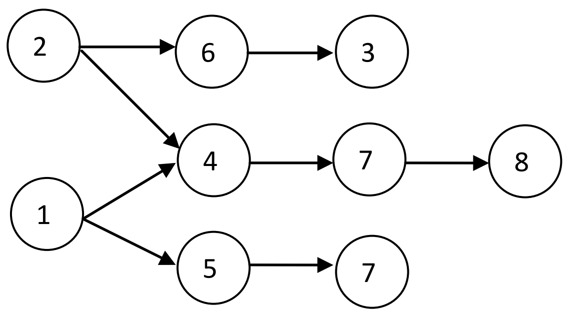
Job	1	2	3	4	5
p_j	6	3	2	4	5
w_j	3	2	1	5	3
d_j	11	4	12	7	9

Find optimal sequence and compute $\Sigma w_j T_j$.

2.13 Consider the following data as an instance of the 1| prec | L_{max} problem.

Job	1	2	3	4	5	6	7	8
p_j	2	3	2	2	4	3	2	2
d_j	5	4	13	6	12	10	15	19

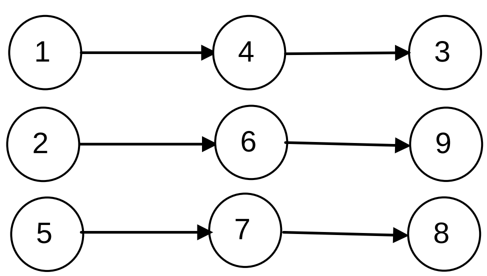
The precedence network for the problem is shown below:



Find the best efficient sequence this problem and compute L_{max} ?

2.14 Consider 1| prec | $\Sigma \omega_j C_j$ problem in which the data and precedence network are given below:

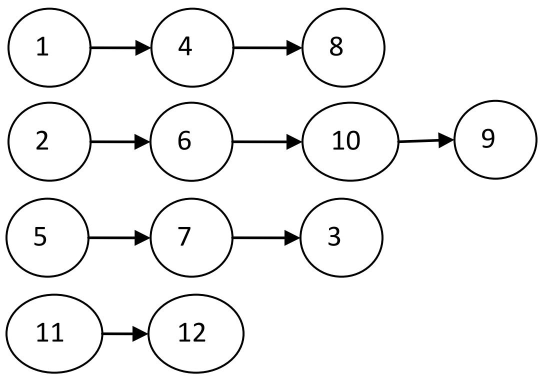
Job	1	2	3	4	5	6	7	8	9
Process time	2	3	2	1	4	3	2	2	4
Weight	5	4	6	6	12	11	12	10	13



Solve the problem to minimize total weighted completion times using chain-method. Then, constructing the Gantt chart for the solution obtained. Next, compute Average flow time, Average completion time, Maximum flow time, and Average waiting time.

2.15 Consider 1 | prec | L_{max} problem with the problem data and precedence network given below:

Job	1	2	3	4	5	6	7	8	9	10	11	12
Process time	2	3	2	1	4	3	2	2	5	4	6	4
Due dates	5	4	13	6	12	10	15	19	20	11	14	18



Solve the problem to minimize maximum lateness using remaining slack rule. Then, constructing the Gantt chart for the solution obtained. Next, compute maximum lateness, maximum tardiness, maximum earliness, total lateness, total tardiness, and total earliness.

2.16 Consider 1 | | n_t problem with the following data:

Job	1	2	3	4	5
P_i	7	8	4	6	6
d_i	9	17	18	19	21

Find optimal sequences and compute n_t, T_{max}, and $\bar{\mathbf{T}}$.

2.17 A company has a cell that can produce three parts: A, B and C. The time required to produce each part is 25, 80, and 10 minutes for each part respectively. The value to produce the parts is 5 riyals, 20 riyals, and 1 riyal respectively. How would you schedule the parts through the cell to minimize the value of work in process?

2.18 Consider the following problem: 1/ r_j / Σ F_j and find the best schedule using the appropriate dispatching rule (**<u>not optimal</u>**) which will give the minimum total flow time with release time using the following data:

Job	1	2	3	4	5	6	7	8	9	10
P_i	16	11	6	18	2	20	19	20	8	16
r_i	22	6	0	6	21	7	29	121	64	48

2.19 Use the weighted shortest processing time to find the optimal solution for the data under consideration. Then, compute total weighted completion time. Also, compute flow time for each job in the shop

Job	1	2	3	4	5	6
p_j	8	4	9	12	11	4
w_j	3	2	1	6	5	7

2.20 Consider $1 \mid \mid L_{max}$ problem with the following the processing times and due dates

Job	1	2	3	4	5	6	7
P_j	6	18	12	10	10	17	16
d_j	8	42	44	24	90	85	68

Find the optimal sequence and compute L_{max} and $\overline{\mathbf{L}}$.

2.21 Consider $1|r_j \mid L_{max}$ problem with the following data:

Job	1	2	3	4	5
P_j	6	18	12	10	10
r_j	0	18	12	8	8
d_j	8	42	44	24	90

Find the optimal sequence using the branch and bound with the preemptive due date as the lower bound and compute L_{max}

2.22 Consider an instance of the $1 \mid\mid \Sigma T_j$ problem with data as follows.

Job	1	2	3	4	5
p_j	3	9	6	5	12
d_j	13	10	7	3	5

Find optimal sequence and show complete schedule

2.23 Consider the following data as an instance of the $1 \mid\mid \Sigma w_j T_j$ problem;

Job	1	2	3	4
p_j	6	3	2	4
w_j	3	2	1	5
D_j	11	9	12	7

Check for the following relationship from the given data to make initial sequence

$$d_j \le d_k \ldots, p_j \le p_{k,\text{and},} \omega_j \ge \omega_k .$$

Use Branch and Bound method to find optimal solution.

2.24 Consider an instance of the 1 | rj | L_{max} problem with data as follows.

Job	1	2	3	4	5
p_j	3	4	6	10	2
r_j	0	1	7	6	9
d_j	3	10	17	18	15

Find optimal sequence and show complete schedule.

2.25 Consider the following data

Job	1	2	3	4	5	6	7
p_j	8	12	7	6	9	4	12
ω_j	4	1	4	3	8	1	5

Apply WSPT, and minimize $\Sigma\omega j\, Cj$. Find Flow time of each job in the shop.

2.26 Consider 1 || L_{max} problem with the following data:

Job	1	2	3	4	5	6	7	8
d_j	5	4	13	6	12	10	15	19
P_j	2	3	2	1	4	3	2	2

Given also the following precedence constraints

$2 \Rightarrow 6 \Rightarrow 3$

$1 \Rightarrow 4 \Rightarrow 7 \Rightarrow 8$

Find the optimal sequence and compute $\overline{L}, \overline{T}, \overline{F}$, T_{max}, and L_{max}

2.27 Consider 1 || n_t problem with the following data

Job	1	2	3	4	5	6	7	8	9	10
d_j	19	16	25	3	8	14	31	23	2	15
P_j	5	3	1	2	4	4	2	1	1	4

Find the optimal sequence and compute $\overline{L}, \overline{T}, \overline{F}$, T_{max}, and L_{max}

2.28 A manufacturer of charm bracelets has five jobs to schedule for a leading customer. Each job requires a stamping operation followed by a finishing operation. The finishing operation can begin immediately after its stamping is complete for any item. The table below shows operation times per item in minutes for each job. At the stamping operation, each job requires set-up before processing begins, as described in the table. Find a schedule that completes all five jobs as soon as possible.

		Operation Time per Item		
Job no.	Number in lot	Stamp	Finish	Set-up
1	20	2	8	100
2	25	2	5	250
3	100	1	2	60
4	50	4	2.5	60
5	40	3	6	80