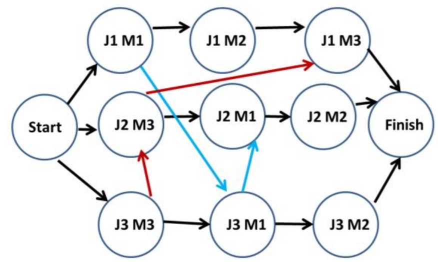
**Queuing Models**: Introduction, Kendall's notation, classification of queuing models, single server and multi-server models, Poisson arrival, exponential service, infinite population
**Sequencing Models**: Introduction, assumptions, processing n-jobs through two machines, n-jobs through three machines, n-jobs through m-machines, graphic solution for processing 2 jobs through n machines with differentorder of sequence

---

Source: http://ecoursesonline.iasri.res.in
In this unit we will look into solution of a sequencing problem.
In this lesson the solutions of following cases will be discussed:




a) n jobs and two machines A and B, all jobs processed in the order AB.
b) n jobs and three machines A, B and C all jobs processed in the order ABC
c) Problems with n jobs and m machines.

## Assumptions made to solve the sequencing problems:

a) The processing time on various machines is independent of the order in which different jobs are processed on them.
b) The time taken by different jobs in going from one machine to another is negligible.
c) A job once started on a machine would be performed to the point of completion uninterrupted.
d) A machine cannot process more than one job at a given point of time.
e) A job would start on a machine as soon as the job and the machine on which it is to be processed are both free.

## Processing of n jobs through two machines:

The simplest possible sequencing problem is that of n job two machine sequencing problem in which we want to determine the sequence in which **n**-job should be processed through two machines so as to minimize the total elapsed time T. The problem can be described as:

a) Only two machines A and B are involved.
b) Each job is processed in the order AB.
c) The exact or expected processing times $A_1, A_2, A_3, ---, A_n$; $B_1, B_2, B_3, ---, B_n$ are known and are provided in the following table

| Machine | Job(s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | -- | - | i | - - | - | n |
| A | $A_1$ | $A_2$ | $A_3$ | -- | - | $A_i$ | -- | - | $A_n$ |
| B | $B_1$ | $B_2$ | $B_3$ | -- | - | $B_i$ | -- | - | $B_n$ |

The problem is to find the sequence (or order) of jobs so as to minimize the total elapsed time T.

Johnson's procedure:

Step 1. Select the smallest processing time occurring in the list $A_1, A_2, A_3, ---, A_n$; $B_1, B_2, B_3, ---, B_n$ if there is a tie, either of the smallest processing times can be selected.

Step 2. If the least processing time is $A_r$, select the $r^{th}$ job first. If it is $B_s$, do the $s^{th}$ job last as the given order is AB

Step 3. There are now (n-1) jobs left to be ordered. Repeat steps I and II for the remaining set of processing times obtained by deleting the processing time for both the machines corresponding to the job already assigned.

Step 4. Continue in the same manner till the entire jobs have been ordered. The resulting ordering will minimize the total elapsed time T and is called the optimal sequence.

Step 5. After finding the optimal sequence as stated above find the total elapsed time and idle times on machines A and B as under:

Total elapsed time = The time between starting the first job in the optimal sequence on machine A and completing the last job in the optimal machine B.

Idle time on machine A = (Time when the last job in the optimal sequence on sequences is completed on machine B) - (Time when the last job in the optimal sequences is completed on machine A)

The Johnson's procedure can be illustrated by following examples:

Johnson's Algorithm:

Step I: Select the minimum processing time out of all Ai 's and Bi's. If it is Ar then do the rth job first. If it is Bs then do the sth job at last.

Step 2: If there is a tie in selecting the minimum of all the processing times, then such a situation is dealt with the following three ways: (i) If the minimum of all the processing times is Ar, which is also equal to Bs , that is, min(Ai ,Bi ) = Ar = Bs , then do the rth job first and sth job at last. (ii) If min (Ai, Bi) = Ar, but Ar = Ak, i.e., there is a tie for minimum among Ai 's, then select anyone. (iii) If min (Ai, Bi) = Bs, but Bs = Bt, i.e., there is a tie for minimum among Bi 's, then select anyone.

Step 3: Now, eliminate the job which has already been assigned from further consideration, and repeat steps I and 2 until an optimal sequence is found.

**Example I**

There are nine jobs, each of which must go through two machines P and Q in the order PQ, the processing times (in hours) are given below:

| Machine | Job(s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | H | I |
| P | 2 | 5 | 4 | 9 | 6 | 8 | 7 | 5 | 4 |
| Q | 6 | 8 | 7 | 4 | 3 | 9 | 3 | 8 | II |

Find the sequence that minimizes the total elapsed time T. Also calculate the total idle time for the machines in this period.

**Solution**

The minimum processing time on two machines is 2 which correspond to task A on machine P. This shows that task A will be preceding first. After assigning task A, we are left with 8 tasks on two machines

| Machine | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|
| P | 5 | 4 | 9 | 6 | 8 | 7 | 5 | 4 |
| Q | 8 | 7 | 4 | 3 | 9 | 3 | 8 | II |

Minimum processing time in this reduced problem is 3 which correspond to jobs E and G (both on machine Q). Now since the corresponding processing time of task E on machine P is less than the corresponding processing time of task G on machine Q therefore task E will be processed in the last and task G next to last. The situation will be dealt as

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | | | | | | | G | E |

The problem now reduces to following 6 tasks on two machines with processing time as follows:

| Machine | B | C | D | F | H | I |
|---|---|---|---|---|---|---|
| P | 5 | 4 | 9 | 8 | 5 | 4 |
| Q | 8 | 7 | 4 | 9 | 8 | II |

Here since the minimum processing time is 4 which occurs for tasks C and I on machine P and task D on machine Q. Therefore, the task C which has less processing time on P will be processed first and then task I and task D will be placed at the last i.e., $7^{th}$ sequence cell.

The sequence will appear as follows:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | C | I | | | | D | E | G |

The problem now reduces to the following 3 tasks on two machines

| Machine | B | F | H |
|---|---|---|---|
| P | 5 | 8 | 5 |
| Q | 8 | 9 | 8 |

In this reduced table the minimum processing time is 5 which occurs for tasks B and H both on machine P. Now since the corresponding time of tasks B and H on machine Q are same i.e. 8. Tasks B or H may be placed arbitrarily in the 4th and 5th sequence cells. The remaining task F can then be placed in the 6th sequence cell. Thus the optimal sequences are represented as

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | I | C | B | H | F | D | E | G |

or

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| A | I | C | H | B | F | D | E | G |

Further, it is also possible to calculate the minimum elapsed time corresponding to the optimal sequencing

$A \rightarrow I \rightarrow C \rightarrow B \rightarrow H \rightarrow F \rightarrow D \rightarrow E \rightarrow G$.

| Job Sequence | Machine A | | Machine B | |
|---|---|---|---|---|
| | Time In | Time Out | Time In | Time Out |
| A | 0 | 2 | 2 | 8 |
| I | 2 | 6 | 8 | 19 |
| C | 6 | 10 | 19 | 26 |
| B | 10 | 15 | 26 | 34 |
| H | 15 | 20 | 34 | 42 |
| F | 20 | 28 | 42 | 51 |
| D | 28 | 37 | 51 | 55 |
| E | 37 | 43 | 55 | 58 |
| G | 43 | 50 | 58 | 61 |

Hence the total elapsed time for this proposed sequence staring from job A to completion of job G is 61 hours .During this time machine P remains idle for 11 hours (from 50 hours to 61 hours)and the machine Q remains idle for 2 hours only (from 0 hour to 2 hour ).

## Processing of n Jobs through Three Machines:

The type of sequencing problem can be described as follows:

a) Only three machines A, B and C are involved;
b) Each job is processed in the prescribed order ABC
c) No passing of jobs is permitted i.e. the same order over each machine is maintained.
d) The exact or expected processing times $A_1, A_2, A_3, ---, A_n$; $B_1, B_2, B_3, ---, B_n$ and $C_1, C_2, C_3, ---, C_n$ are known and are denoted by the following table

| Machine | Job(s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | - | - | i | - | - | n |
| A | $A_1$ | $A_2$ | $A_3$ | - | - | $A_i$ | - | - | $A_n$ |
| B | $B_1$ | $B_2$ | $B_3$ | - | - | $B_i$ | - | - | $B_n$ |
| C | $C_1$ | $C_2$ | $C_3$ | | | $C_i$ | | | $C_n$ |

Our objective will be to find the optimal sequence of jobs which minimizes the total elapsed time. No general procedure is available so far for obtaining an optimal sequence in such case. However, the Johnson's procedure can be extended to cover the special cases where either one or both of the following conditions hold:

a) The minimum processing time on machine A $\geq$ the maximum processing time on machine B.
b) The minimum processing time on machine C $\geq$ the maximum processing time on machine B.

The method is to replace the problem by an equivalent problem involving n jobs and two machines. These two fictitious machines are denoted by G and H and the corresponding time $G_i$ and $H_i$ are defined by

$G_i = A_i + B_i$ and $H_i = B_i + C_i$

Now this problem with prescribed ordering GH is solved by the method with n jobs through two machines, the resulting sequence will also be optimal for the original problem. The above methodology is illustrated by following example:

In this section, we discuss on extension of Johnson's procedure for scheduling n jobs on three machines A, B and C in order ABC. This list of jobs with their processing times on three machines A, B and C is given below.

Processing time Job

on machine 1 2 3 ⋯ n

A $t_{11}$ $t_{12}$ $t_{13}$ ⋯ $t_{1n}$

B $t_{21}$ $t_{22}$ $t_{23}$ ⋯ $t_{2n}$

C $t_{31}$ $t_{32}$ $t_{33}$ ⋯ $t_{3n}$

An optimal solution to this problem can be obtained if either or both of the following conditions hold good:

**1.** The minimum processing time on machine A is at least as great as the maximum processing time on machine B, that is, $\min t_{1j} \geq \max t_{2j}$ , for $j = 1,2,\cdots,n$.

**2.** The minimum processing time on machine C is at least as great as the maximum processing time on machine B, that is, $\min t_{3j} \geq \max t_{2j}$ , for $j = 1,2,\cdots,n$.

**If either or both the above conditions hold good, then the algorithm can be summarized in the following steps:**

**Step 1 :** Examine the processing times of the given jobs on all three machines and if either one or both the above conditions hold, then go to Step 2; otherwise, the algorithm fails.

**Step 2 : Introduce two fictitious machines, say G and H, with corresponding processing times given by**

**(i)** $t_{Gj} = t_{1j} + t_{2j}$ , $j = 1,2,\cdots,n$, i.e., the processing time on machine G is the sum of the processing times on machines A and B.

**(ii)** $t_{Hj} = t_{2j} + t_{3j}$ , $j = 1,2,\cdots,n$, i.e., the processing time on machine H is the sum of the processing times on machines B and C

Step 3 : Determine the optimal sequence for n jobs and two machine equivalent sequencing problem with the prescribed ordering GH in the same way as discussed earlier.

**Example 2**

There are five jobs (namely 1,2,3,4 and 5), each of which must go through machines A, B and C in the order ABC. Processing Time (in hours) are given below:

| Jobs | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Machine A | 5 | 7 | 6 | 9 | 5 |
| Machine B | 2 | 1 | 4 | 5 | 3 |
| Machine C | 3 | 7 | 5 | 6 | 7 |

Find the sequence that minimum the total elapsed time required to complete the jobs.

**Solution:**

Here Min $A_i = 5$; $B_i = 5$ and $C_i = 3$ since the condition of Min. $A_i \geq$ Max. $B_i$ is satisfied the given problem can be converted into five jobs and two machines problem.

| M/C ↓ Jobs → | J1 | J2 | J3 | J4 | J5 |
|---|---|---|---|---|---|
| Machine A | 5 | 7 | 6 | 9 | 5 |
| Machine B | 2 | 1 | 4 | 5 | 3 |
| Machine C | 3 | 7 | 5 | 6 | 7 |

| M/C ↓ Jobs → | J1 | J2 | J3 | J4 | J5 |
|---|---|---|---|---|---|
| Machine G=A+B | 7 | 8 | 10 | 14 | 8 |
| Machine H=B+C | 5 | 8 | 9 | 11 | 10 |

The Optimal Sequence will be

| 2 | 5 | 4 | 3 | 1 |
|---|---|---|---|---|

Total elapsed Time will be

| Jobs | Machine A | | Machine B | | Machine C | |
|---|---|---|---|---|---|---|
| | In | Out | In | Out | In | Out |
| 2 | 0 | 7 | 7 | 8 | 8 | 15 |
| 5 | 7 | 12 | 12 | 15 | 15 | 22 |
| 4 | 12 | 21 | 21 | 26 | 26 | 32 |
| 3 | 21 | 27 | 27 | 31 | 32 | 37 |
| 1 | 27 | 32 | 32 | 34 | 37 | 40 |

Min. total elapsed time is 40 hours.
Idle time for Machine A is 8 hrs. (32-40)
Idle time for Machine B is 25 hours (0-7, 8-12, 15-21, 26-27, 31-32 and 34-40)
Idle time for Machine C is 12 hours (0-8, 22-26.)

## Problems with n Jobs and m Machines

Let there be **'n'** jobs, each of which is to be processed through m machines, say $M_1$, $M_2$, --- , $M_m$ in the order $M_1, M_2, M_3$, --- , $M_m$. Let $T_{ij}$ be the time taken by the $i^{th}$ machine to complete the $j^{th}$ job.
The iterative procedure of obtaining an optimal sequence is as follows:

**Step I:** Find (i) $\min_j(T_{1j})$  ii) $\min_j(T_{mj})$  iii) $\max_j(T_{2j}, T_{3j}, T_{4j}, --- , T_{(m-1)j})$ for j=1,2,---, n

**Step II:** Check whether

a. $\min_j(T_{1j}) \geq \max_j(T_{ij})$ for i=2,3, ----, m-1

Or

b. $\min_j(T_{mj}) \geq \max_j(T_{ij})$ for i=2,3, ---, m-1

**Step III:** If the inequalities in Step II are not satisfied, method fails, otherwise, go to next step.

**Step IV:** Convert the m machine problem into two machine problem by introducing two fictitious machines G and H, such that

$T_{Gj} = T_{1j} + T_{2j} + --- + T_{(m-1)j}$ and  $T_{Hj} = T_{2j} + T_{3j} + --- + T_{mj}$

Determine the optimal sequence of n jobs through 2 machines by using optimal sequence algorithm.

**Step V:** In addition to condition given in Step IV, if $Tij = T_{2j} + T_{3j} + --- + T_{mj} = C$ is a fixed positive constant for all i = 1, 2, 3, n then determine the optimal sequence of n jobs and two machines $M_1$ and $M_m$ in the order $M_1M_m$ by using the optimal sequence algorithm.

### Example 3

Find an optimal sequence for the following sequencing problem of four jobs and five machines when passing is not allowed, of which processing time (in hours) is given below:

| Job | Machine | | | | |
|---|---|---|---|---|---|
| | A | B | C | D | E |
| 1 | 7 | 5 | 2 | 3 | 9 |
| 2 | 6 | 6 | 4 | 5 | 10 |
| 3 | 5 | 4 | 5 | 6 | 8 |
| 4 | 8 | 3 | 3 | 2 | 6 |

Also find the total elapsed time.

**Solution**

Here Min. $A_i = 5$, Min. $E_i = 6$
Max. $(B_i, C_i, D_i) = 6, 5, 6$ respectively
Since Min. $E_i$ = Max. $(B_i, D_i)$ and Min. $A_i$ = Max. $C_i$ satisfied therefore the problem can be converted into 4 jobs and 2 fictitious machines G and H as follows:

| Job | Fictitious Machine | |
|---|---|---|
| | $G_i = A_i + B_i + C_i + D_i$ | $H_i = B_i + C_i + D_i + E_i$ |
| 1 | 17 | 19 |
| 2 | 21 | 25 |
| 3 | 20 | 23 |
| 4 | 16 | 14 |

The above sequence will be:

| | | | |
|---|---|---|---|
| 1 | 3 | 2 | 4 |

Total Elapsed Time Corresponding to Optimal Sequence can be obtained as follows:

| | Machine A | | Machine B | | Machine C | | Machine D | | Machine E | |
|---|---|---|---|---|---|---|---|---|---|---|
| Job | In | Out | In | Out | In | Out | In | Out | In | Out |
| 1 | 0 | 7 | 7 | 12 | 12 | 14 | 14 | 17 | 17 | 26 |
| 3 | 7 | 12 | 12 | 16 | 16 | 21 | 21 | 27 | 27 | 35 |
| 2 | 12 | 18 | 18 | 24 | 24 | 28 | 28 | 33 | 35 | 45 |
| 4 | 18 | 26 | 26 | 29 | 29 | 32 | 33 | 35 | 45 | 51 |

Thus, the minimum elapsed time is 51 hours.

Idle time for machine A = 25 hours (26-51)

Idle time for machine B = 33 hours (0-7,16-18,24-26,29-51)

Idle time for machine C = 37 hours (0-12,14-16,21-24,28-29,32-51)

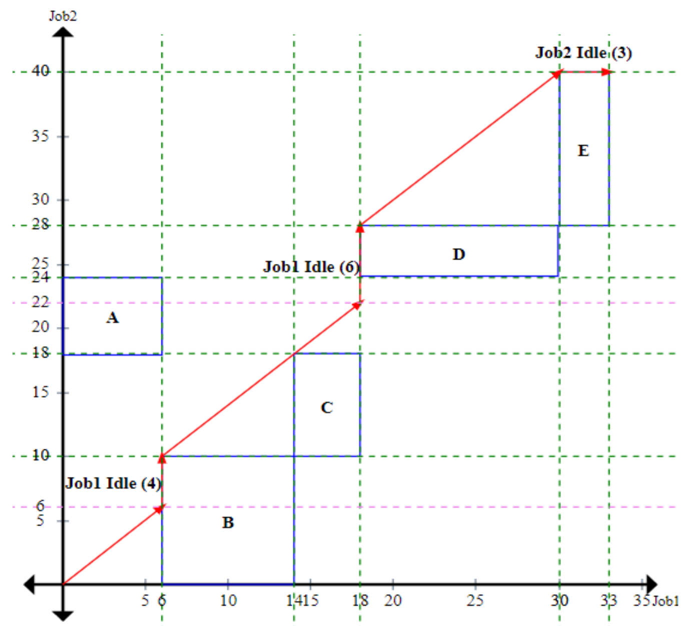
Idle time for machine D = 35 hours (0-14,17-21,27-28,35-51)

Idle time for machine E = 18 hours (0-17,26-27)

**Example 4**

**Find solution of Processing 2 Jobs Through m Machines Problem**

| | | Machine | | | | |
|---|---|---|---|---|---|---|
| Job 1 | Sequence | A | B | C | D | E |
| | Time | 6 | 8 | 4 | 12 | 3 |
| Job 2 | Sequence | B | C | A | D | E |
| | Time | 10 | 8 | 6 | 4 | 12 |

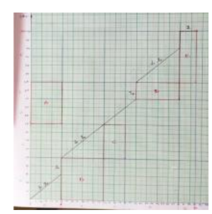



Solution:

1. We are given the job sequences and processing time of 2 jobs at 5 machines. We follow the graphical method to find the minimum total elapsed time starting the first job at the first machine to completion of the second job at the last machine.

2. We first represent the processing time of job 1 on different machines, i.e., 6,8,4,12,3 along the x-axis and the processing time of job 2, i.e., 10,8,6,4,12 along the y-axis.
We draw the first vertical line at 6 hrs, the second at 6+8=14 hrs, the third at 14+4=18 hrs, and so on.
Similarly, we draw the horizontal lines at 10 hrs, the second at 10+8=18 hrs, the third at 18+6=24 hrs, and so on.

3. We draw the rectangular blocks by pairing the same machines as shown in the Figure
(1) For machine A, 0 to 6 hours on x-axis and 18 to 24 hours on y-axis
(2) For machine B, 6 to 14 hours on x-axis and 0 to 10 hours on y-axis
(3) For machine C, 14 to 18 hours on x-axis and 10 to 18 hours on y-axis
(4) For machine D, 18 to 30 hours on x-axis and 24 to 28 hours on y-axis
(5) For machine E, 30 to 33 hours on x-axis and 28 to 40 hours on y-axis



4. Avoiding the rectangular blocks, draw the line starting from origin O to the end point, moving horizontally, vertically and diagonally along a line which makes an angle 45deg with the horizontal line.

Moving horizontally along this line indicates that the job 1 is under process while the job 2 is idle.
Similarly moving vertically along this line indicates that the job 2 is under process while the job 1 is idle.
The diagonal movement along this line shows that both jobs are under process. Since simultaneous processing of both jobs on a machine is not possible, therefore a diagonal movement through rectangle areas is not allowed.

(1) we move diagonally up to (6,6), which means both the jobs 1 and 2 are being processed simultaneously
(2) we move vertically up to (6,10), which means Job 2 is under process and Job 1 is idle for 4 hrs
(3) we move diagonally up to (18,22), which means both the jobs 1 and 2 are being processed simultaneously
(4) we move vertically up to (18,28), which means Job 2 is under process and Job 1 is idle for 6 hrs
(5) we move diagonally up to (30,40), which means both the jobs 1 and 2 are being processed simultaneously
(6) we move horizontally up to (33,40), which means Job 1 is under process and Job 2 is idle for 3 hrs

5. An optimum path minimizes the idle time for both the jobs.
Idle time of job 1 =4+6=10 hours.

Idle time of job 2 =3 hours.

6. The elapsed time is obtained by adding the idle time for either job to the processing time for that job.
Elapsed time, job 1 = processing time of job 1 + idle time of job 1 = (6+8+4+12+3) +(10) =33+10=43 hours.
Elapsed time, job 2 = processing time of job 2 + idle time of job 2 = (10+8+6+4+12) +(3) =40+3=43 hours.

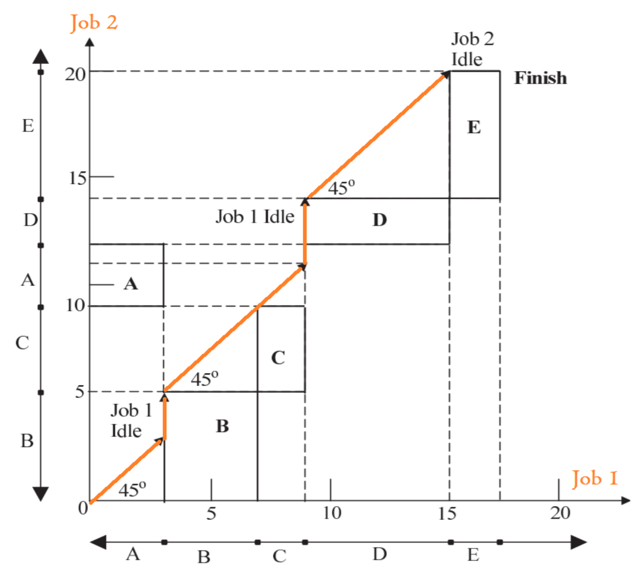
**Example 5**
Two jobs are to be performed on five machines A, B, C, D, and E. Processing times are given in the following table:

| | | Machine | | | | |
|---|---|---|---|---|---|---|
| Job 1 | Sequence | A | B | C | D | E |
| | Time | 3 | 4 | 2 | 6 | 2 |
| Job 2 | Sequence | B | C | A | D | E |
| | Time | 5 | 4 | 3 | 2 | 6 |

Find the total elapsed time.

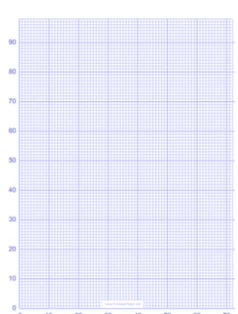
**Solution:** The following steps are to be followed to solve the problem graphically:

• Draw two perpendicular lines.
→ Horizontal line represents the processing time for job 1 while job 2 remains idle.
→ Vertical line represents processing time for job 2 while job 1 remains idle.
• Mark the processing times of job 1 and job 2 on the horizontal and vertical lines respectively according to the given order of machines.
• Draw the rectangular blocks by pairing the same machines as shown in the Figure.
• Avoiding the rectangular blocks, draw the line starting from origin 'O' to the end point, moving horizontally, vertically and diagonally along a line which makes an angle $45^0$ with the horizontal line.
• An optimum path minimizes the idle time for both the jobs. So, it is the path on which diagonal movement is maximum.




• The elapsed time is obtained by adding the idle time for either job to the processing time for that job.
For this example, idle time for job 1 is 3+2 =5 hours.
Elapsed time = processing time of job 1 + idle time of job 1 = (3+4+2+6+2) + 5 = 17 + 5 = 22 hours.

Likewise, idle time for job 2 is 2 hours.
Elapsed time = processing time of job 2 + idle time of job 2 = (5 + 4 + 3 + 2 + 6) + (2) = 20 + 2 = 22 hours.



**Game Theory:** Introduction, game with pure strategies, game with mixed strategies, dominance principle, graphical method for 2xn and mx2 games, linear programming approach for game theory.

## Introduction to Game Theory

Game theory is a type of decision theory in which one's choice of action is determined after taking into account all possible alternatives available to an opponent playing the same game, rather than just by the possibilities of several outcome results. Game theory does not insist on how a game should be played but tells the procedure and principles by which action should be selected. Thus, it is a decision theory useful in competitive situations.

Game is defined as an activity between two or more persons according to a set of rules at the end of which each person receives some benefit or suffers loss. The set of rules defines the game. Going through the set of rules once by the participants defines a play.

## Properties of a Game:

a) There are finite numbers of competitors called 'players'

b) Each player has a finite number of possible courses of action called 'strategies'

c) All the strategies and their effects are known to the players, but player does not know which strategy is to be chosen.

d) A game is played when each player chooses one of his strategies. The strategies are assumed to be made simultaneously with an outcome such that no player knows his opponent's strategy until he decides his own strategy.

e) The game is a combination of the strategies and in certain units which determines the gain or loss.

f) The figures shown as the outcomes of strategies in a matrix form are called 'pay-off matrix'.

g) The player playing the game always tries to choose the best course of action which results in optimal pay off called 'optimal strategy'.

h) The expected pay off when all the players of the game follow their optimal strategies is known as 'value of the game'. The main objective of a problem of a game is to find the value of the game.

i) The game is said to be 'fair' game if the value of the game is zero otherwise it's known as 'unfair'.

## Characteristics of Game Theory:

**a) Competitive game:** A competitive situation is called a competitive game if it has the following four properties

- → There are finite number of competitors such that $n \geq 2$. In case $n = 2$, it is called a two-person game and in case $n > 2$, it is referred as n-person game.
- → Each player has a list of finite number of possible activities.
- → A play is said to occur when each player chooses one of his activities. The choices are assumed to be made simultaneously i.e., no player knows the choice of the other until he has decided on his own.
- → Every combination of activities determines an outcome which results in a gain of payments to each player, provided each player is playing uncompromisingly to get as much as possible. Negative gain implies the loss of same amount.

**b) Strategy:** The strategy of a player is the predetermined rule by which player decides his course of action from his own list during the game.

The two types of strategy are

→ Pure strategy

→ Mixed strategy

**Pure Strategy:** If a player knows exactly what the other player is going to do, a deterministic situation is obtained, and objective function is to maximize the gain. Therefore, the pure strategy is a decision rule always to select a particular course of action.

**Mixed Strategy:** If a player is guessing as to which activity is to be selected by the other on any particular occasion, a probabilistic situation is obtained, and objective function is to maximize the expected gain. Thus, the mixed strategy is a selection among pure strategies with fixed probabilities.

**c) Number of persons:** A game is called 'n' person game if the number of persons playing is 'n'.

The person means an individual or a group aiming at a particular objective.

Two-person, zero-sum game: A game with only two players (player A and player B) is called a 'two-person, zero-sum game', if the losses of one player are equivalent to the gains of the other so that the sum of their net gains is zero.

Two-person, zero-sum games are also called rectangular games as these are usually represented by a payoff matrix in a rectangular form.

**d) Number of activities:** The activities may be finite or infinite.

**e) Payoff:** The quantitative measure of satisfaction a person gets at the end of each play is called a payoff

**f) Payoff matrix:** Suppose the player A has 'm' activities and the player B has 'n' activities.

Then a payoff matrix can be formed by adopting the following rules

- → Row designations for each matrix are the activities available to player A
- → Column designations for each matrix are the activities available to player B
- → Cell entry Vij is the payment to player A in A's payoff matrix when A chooses the activity i and B chooses the activity j.
- → With a zero-sum, two-person game, the cell entry in the player B's payoff matrix will be negative of the corresponding cell entry Vij in the player A's payoff matrix so that sum of payoff matrices for player A and player B is ultimately zero.

**g) Value of the game:** Value of the game is the maximum guaranteed game to player A (maximizing player) if both the players uses their best strategies. It is generally denoted by 'V' and it is unique.