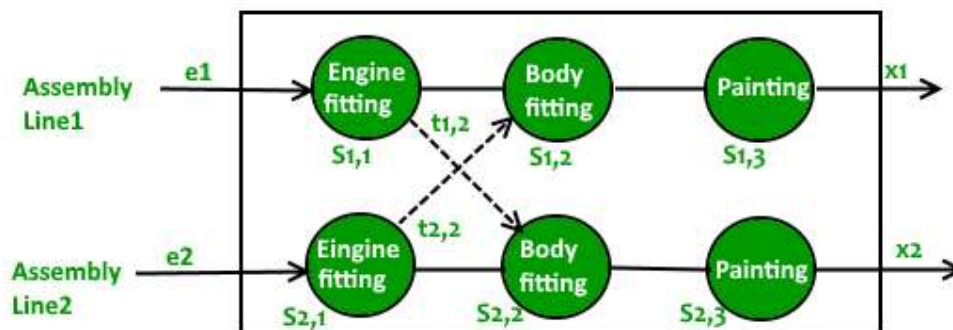


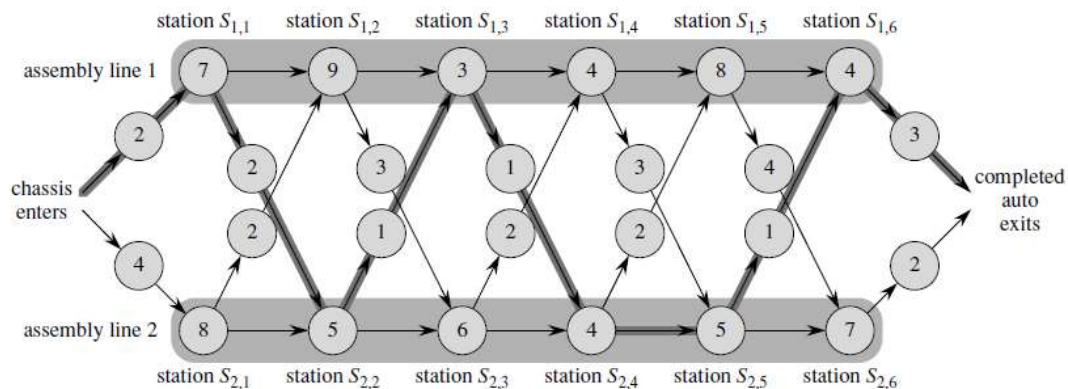
Assembly Line Scheduling

- Assembly line scheduling is a manufacturing problem.
- In automobile industries assembly lines are used to transfer parts from one station to another station.
- Manufacturing of large items like car, trucks etc. generally undergoes through multiple stations, where each station is responsible for assembling particular part only.
- Entire product be ready after it goes through predefined n stations in sequence.
- Manufacturing of car may be done through several stages like engine fitting, coloring, light fitting, fixing of controlling system, seats and many other things.
- Particular task is carried out at the station dedicated to that task only.
- Based on the requirement there may be more than one assembly line.
- In case of two assembly lines if the load at station j at assembly 1 is very high, then components are transferred to station of assembly line 2 the converse is also true. This technique helps to speed up the manufacturing process.
- The time to transfer partial product from one station to next station on the same assembly line is negligible. No cost is incurred for movement in same line.
- Transfer of partially completed product from one assembly line to another, incurs an extra cost $t_{i,j}$ to move to station j from line i.



- Terminologies are explained here :
 - S_{ij} = Station j on assembly line i .
 - a_{ij} = Time needed to assemble the partial component at station j on assembly line i .
 - t_{ij} = Time required to transfer component from one assembly to other from station j to $(j + 1)$.
 - e_i = Entry time on assembly line i .
 - x_i = Exit time from assembly line i .
 - n = number of stations on both assembly line

An instance of the assembly-line problem with costs e_i , $a_{i,j}$, $t_{i,j}$, and x_i indicated:



Assembly Line Scheduling Problem:

- Determine which station should be selected from assembly line 1 and which to choose from assembly line 2 in order to minimize the total time to build the entire product.
- Selecting stations by brute force method is quite infeasible. If there are n stations, unfortunately there are 2^n possible ways to choose stations.
- Brute force technique takes $O(2^n)$ time, it is not acceptable when n is large.

- So, we apply Dynamic Programming (DP) technique. The time complexity of the DP implementation of the assembly line scheduling is $O(n)$.

Pseudocode:

FASTEST-WAY (a, t, e, x, n)

```
1   $f_1[1] \leftarrow e_1 + a_{1,1}$ 
2   $f_2[1] \leftarrow e_2 + a_{2,1}$ 
3  for  $j \leftarrow 2$  to  $n$ 
4      do if  $f_1[j-1] + a_{1,j} \leq f_2[j-1] + t_{2,j-1} + a_{1,j}$ 
5          then  $f_1[j] \leftarrow f_1[j-1] + a_{1,j}$ 
6               $l_1[j] \leftarrow 1$ 
7          else  $f_1[j] \leftarrow f_2[j-1] + t_{2,j-1} + a_{1,j}$ 
8               $l_1[j] \leftarrow 2$ 
9      if  $f_2[j-1] + a_{2,j} \leq f_1[j-1] + t_{1,j-1} + a_{2,j}$ 
10         then  $f_2[j] \leftarrow f_2[j-1] + a_{2,j}$ 
11              $l_2[j] \leftarrow 2$ 
12         else  $f_2[j] \leftarrow f_1[j-1] + t_{1,j-1} + a_{2,j}$ 
13              $l_2[j] \leftarrow 1$ 
14 if  $f_1[n] + x_1 \leq f_2[n] + x_2$ 
15     then  $f^* = f_1[n] + x_1$ 
16          $l^* = 1$ 
17 else  $f^* = f_2[n] + x_2$ 
18      $l^* = 2$ 
```

PRINT-STATIONS (l, n)

```
1   $i \leftarrow l^*$ 
2  print "line "  $i$  ", station "  $n$ 
3  for  $j \leftarrow n$  downto 2
4      do  $i \leftarrow l_i[j]$ 
5          print "line "  $i$  ", station "  $j - 1$ 
```

Note:

- Using FASTEST-WAY algorithm, we tabulate two look-up tables ('f' table and 'l' table). As the tabulation array is used and the 'n' iteration is done to store the optimal time in the array, the time complexity of the above Dynamic Programming implementation of the assembly line scheduling is $O(n)$.

Let $f_i[j]$ denote the fastest possible time to get a chassis from the starting point through station $S_{i,j}$.

Let $li[j]$ denote the line number, 1 or 2. Here, $i = 1, 2$ and $j = 2, 3, \dots, n$.

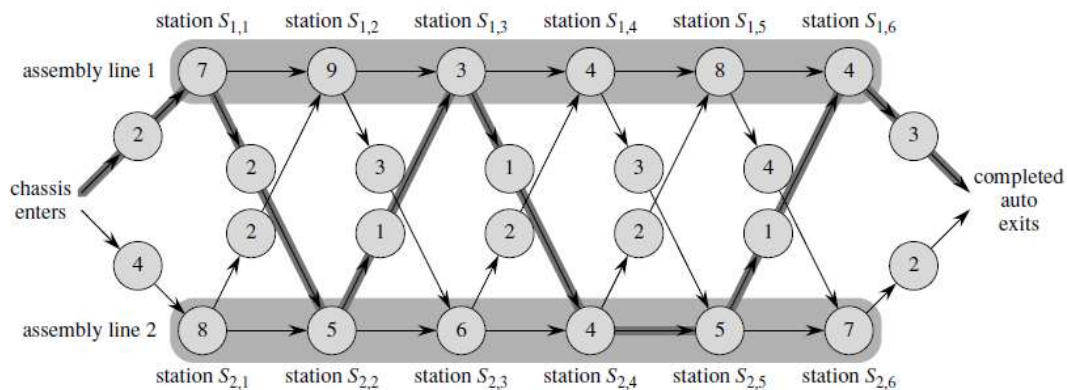
- Using PRINT_STATIONS algorithm, we print the solution.

Time Complexity: $O(n)$

Basic Formula:

$$f_1[j] = \begin{cases} e_1 + a_{1,1} & \text{if } j = 1, \\ \min(f_1[j-1] + a_{1,j}, f_2[j-1] + t_{2,j-1} + a_{1,j}) & \text{if } j \geq 2 \end{cases}$$

$$f_2[j] = \begin{cases} e_2 + a_{2,1} & \text{if } j = 1, \\ \min(f_2[j-1] + a_{2,j}, f_1[j-1] + t_{1,j-1} + a_{2,j}) & \text{if } j \geq 2 \end{cases}$$



Example: 1

Solution from text book:

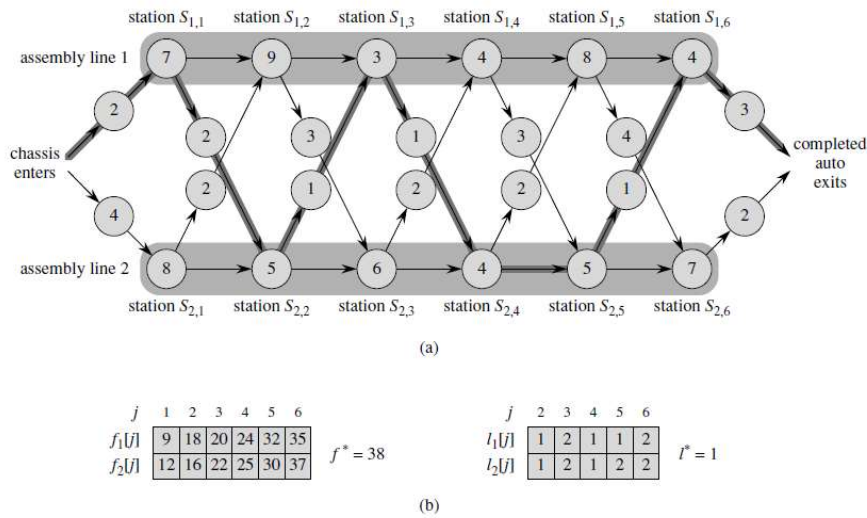
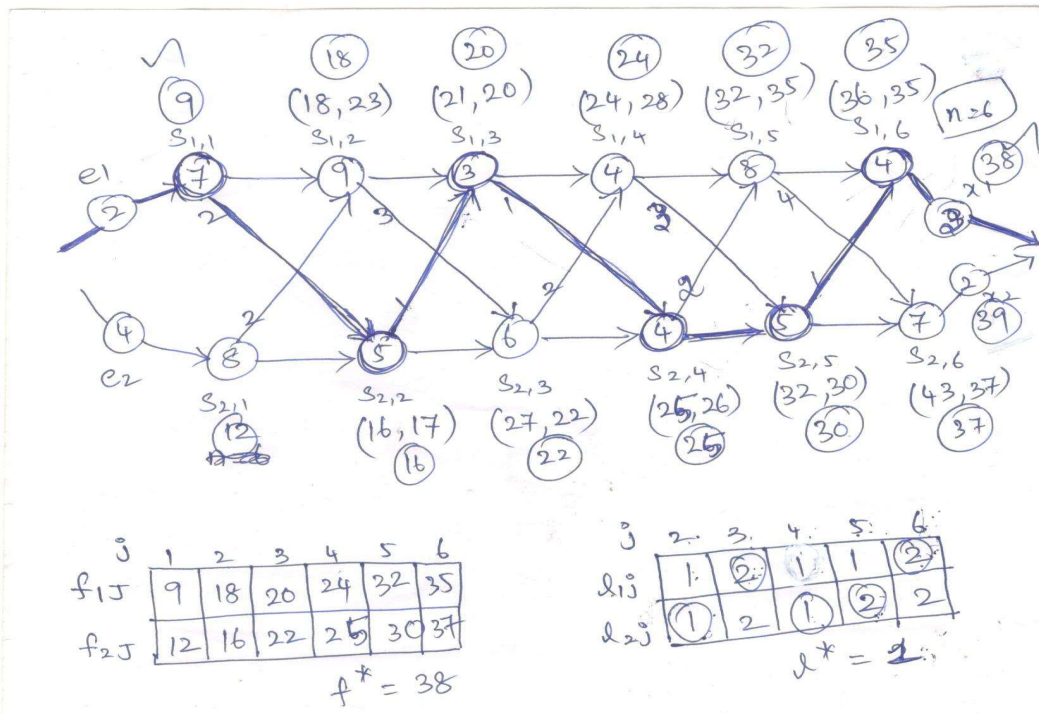


Figure 15.2 (a) An instance of the assembly-line problem with costs e_i , $a_{i,j}$, $t_{i,j}$, and x_i indicated. The heavily shaded path indicates the fastest way through the factory. (b) The values of $f_i[j]$, f^* , $l_i[j]$, and l^* for the instance in part (a).

Hand-Written solution:

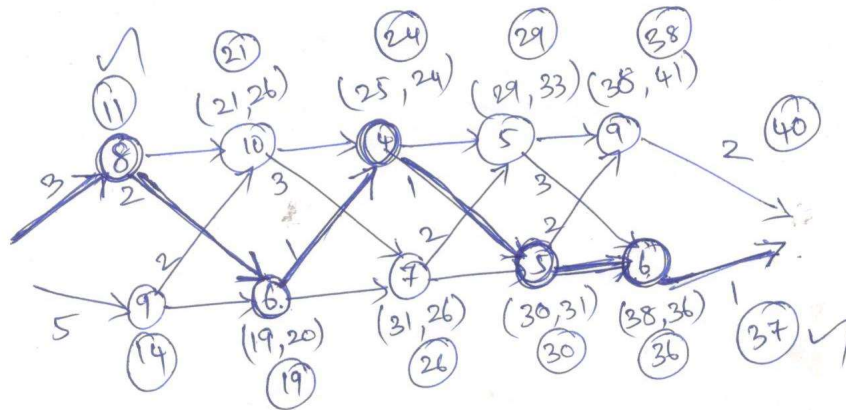


In the example of Figure 15.2, PRINT-STATIONS would produce the output

line 1, station 6
line 2, station 5
line 2, station 4
line 1, station 3
line 2, station 2
line 1, station 1

This printout is “in reverse”;
How can the print routine be
changed to get a printout from
Station 1 to Station 6 ?

Example: 2



$n = 5$

j	1	2	3	4	5
f_{1j}	11	21	24	29	38
f_{2j}	14	19	26	30	36

j	2	3	4	5
d_{1j}	1	2	1	1
d_{2j}	1	2	1	2

$f^* = 37$
 $d^* = 2$

Print stations:-

Line 2 Station 5

Line 2 Station 4

Line 1 Station 3

Line 2 Station 2

Line 1 Station 1

Practice Problem:

Example: Find optimal assembly line schedule for given data.

