طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده

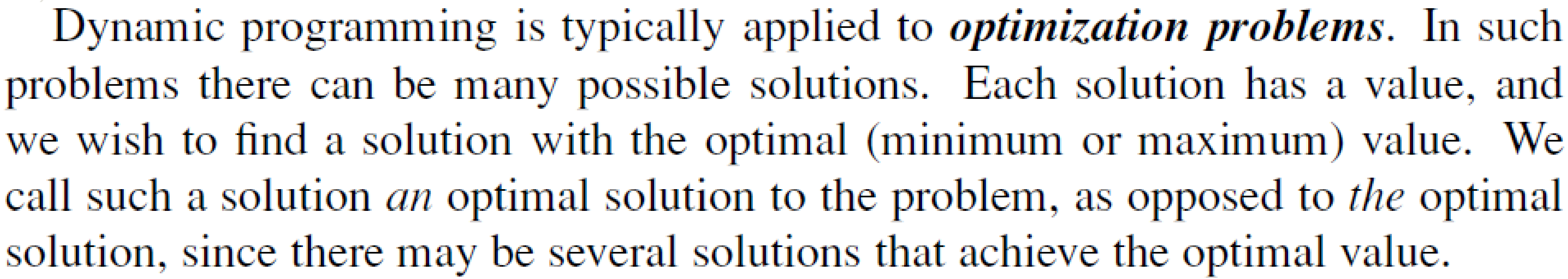
استادیار گروه مهندسی کامپیوتر دانشگاه قم

Dynamic Programming

- **ویژگی اصلی:**

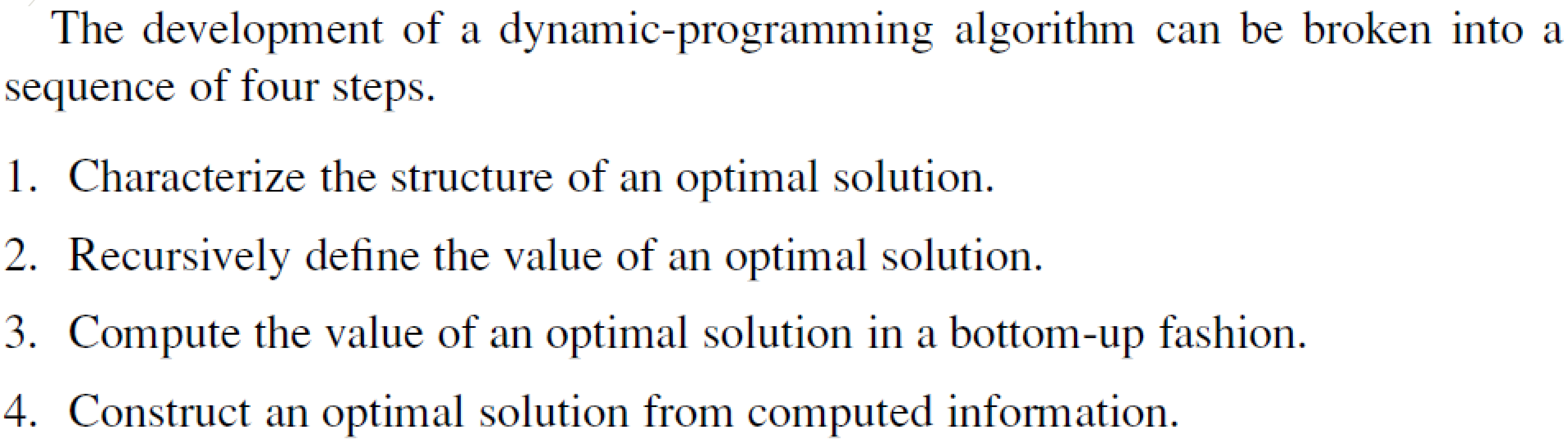
در DP، زیرمسئله ها با یکدیگر، هم پوشانی دارند ولی در D&C، زیرمسئله ها به طور مستقل حل می شوند و در نتیجه در DP، هر زیر مسئله فقط یکبار حل می شود.

Dynamic Programming



Dynamic programming is typically applied to ***optimization problems***. In such problems there can be many possible solutions. Each solution has a value, and we wish to find a solution with the optimal (minimum or maximum) value. We call such a solution *an* optimal solution to the problem, as opposed to *the* optimal solution, since there may be several solutions that achieve the optimal value.

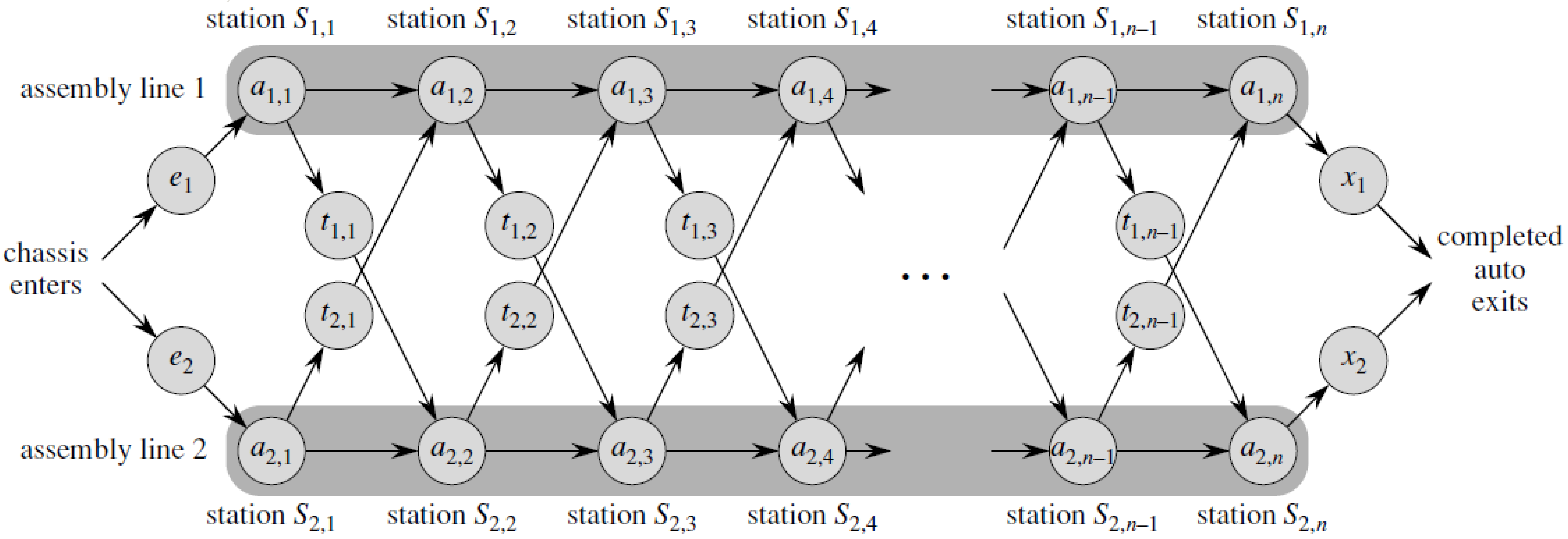
Dynamic Programming



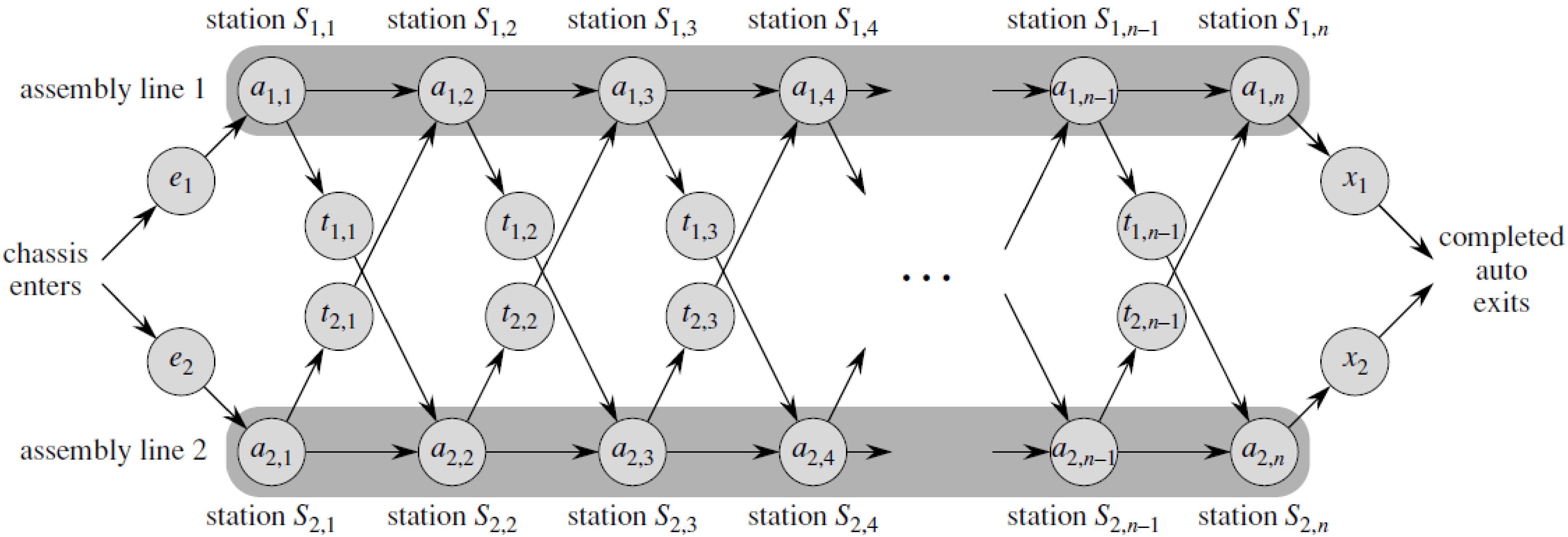
The development of a dynamic-programming algorithm can be broken into a sequence of four steps.

1. Characterize the structure of an optimal solution.
2. Recursively define the value of an optimal solution.
3. Compute the value of an optimal solution in a bottom-up fashion.
4. Construct an optimal solution from computed information.

1. Assembly-line scheduling



1. Assembly-line scheduling



- **Brute-force ?**

1. Assembly-line scheduling

$f_i[j]$

$$\begin{aligned} f_1[1] &= e_1 + a_{1,1} \;, \\ f_2[1] &= e_2 + a_{2,1} \;. \end{aligned}$$

$$f^* = \min(f_1[n] + x_1,\ f_2[n] + x_2) \;.$$

1. Assembly-line scheduling

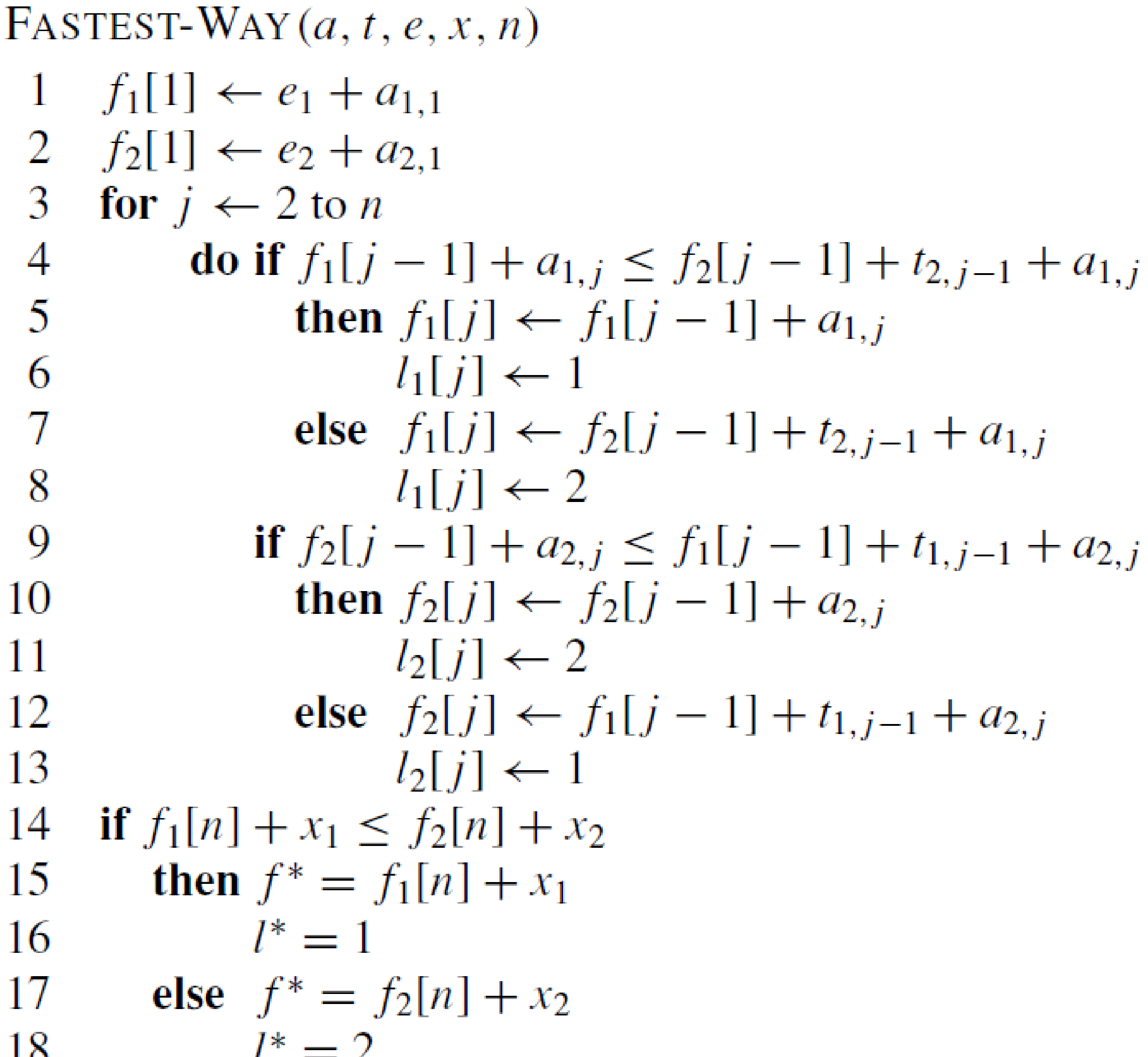
$$r_i(j)$$

$$r_1(n) = r_2(n) = 1$$

$$r_1(j) = r_2(j) = r_1(j+1) + r_2(j+1)$$

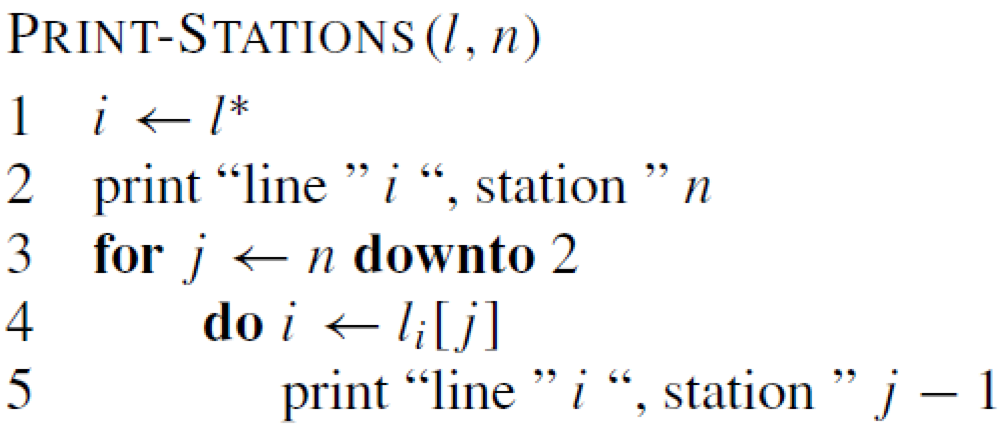
$$r_i(j) = 2^{n-j}$$

1. Assembly-line scheduling



```
FASTEST-WAY$(a, t, e, x, n)$
$f_1[1] \leftarrow e_1 + a_{1,1}$
$f_2[1] \leftarrow e_2 + a_{2,1}$
for $j \leftarrow 2$ to $n$
        do if $f_1[j-1] + a_{1,j} \le f_2[j-1] + t_{2,j-1} + a_{1,j}$
              then $f_1[j] \leftarrow f_1[j-1] + a_{1,j}$
                   $l_1[j] \leftarrow 1$
              else $f_1[j] \leftarrow f_2[j-1] + t_{2,j-1} + a_{1,j}$
                   $l_1[j] \leftarrow 2$
           if $f_2[j-1] + a_{2,j} \le f_1[j-1] + t_{1,j-1} + a_{2,j}$
              then $f_2[j] \leftarrow f_2[j-1] + a_{2,j}$
                   $l_2[j] \leftarrow 2$
              else $f_2[j] \leftarrow f_1[j-1] + t_{1,j-1} + a_{2,j}$
                   $l_2[j] \leftarrow 1$
if $f_1[n] + x_1 \le f_2[n] + x_2$
   then $f^* = f_1[n] + x_1$
        $l^* = 1$
   else $f^* = f_2[n] + x_2$
        $l^* = 2$
```

1. Assembly-line scheduling



```
PRINT-STATIONS($l$, $n$)
$i \leftarrow l^*$
print “line ” $i$ “, station ” $n$
for $j \leftarrow n$ downto 2
    do $i \leftarrow l_i[j]$
        print “line ” $i$ “, station ” $j - 1$
```