Algoritmi e Strutture Dati

Elementi di Programmazione Dinamica

Maria Rita Di Berardini, Emanuela Merelli[1]

[1]Dipartimento di Matematica e Informatica
Università di Camerino

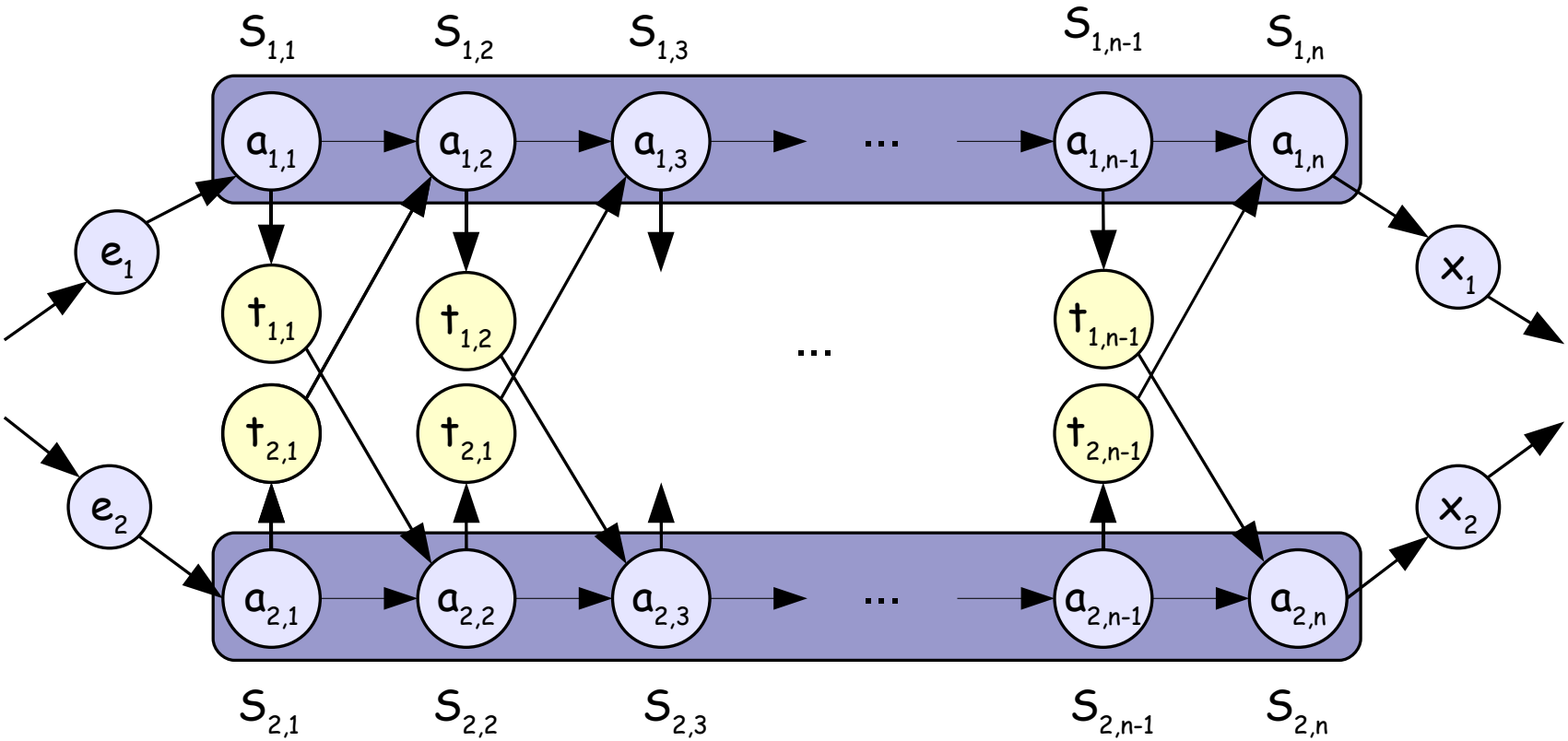
Il problema

La CMC produce automobili in uno stabilimento con due catene di montaggio

Il telaio di un'automobile entra in una catena di montaggio, riceve nuove componenti lungo n stazioni ed, infine l'auto completata esce dalla catena di montaggio

Ogni catena di montaggio ha n stazioni; $S_{i,j}$ denota la j-esima stazione della i-esima catena con $i = 1, 2$ e $j = 1, 2, \ldots, n$

$S_{1,j}$ svolge la stessa funzione di $S_{2,j}$, ma le due catene sono state costruite in periodi diversi e con tecnologie diverse e, quindi, il tempo richiesto in ciascuna stazione varia a seconda della catena; sia $a_{i,j}$ il tempo di montaggio richiesto dalla stazione $S_{i,j}$



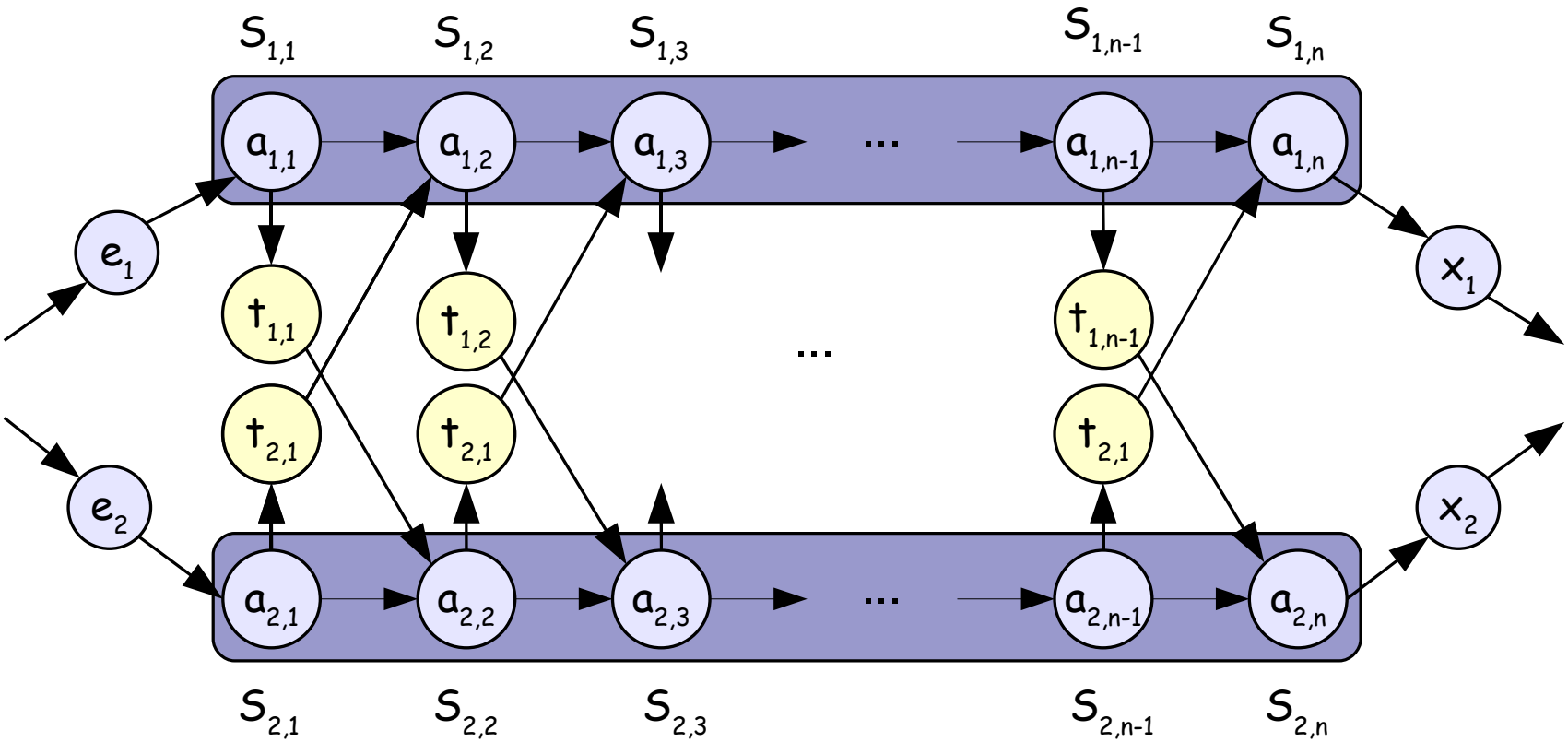
e_i rappresenta il tempo di ingresso del telaio nella catena di montaggio i, mentre x_i rappresenta il tempo di uscita della macchina completata dalla catena di montaggio i

Il problema

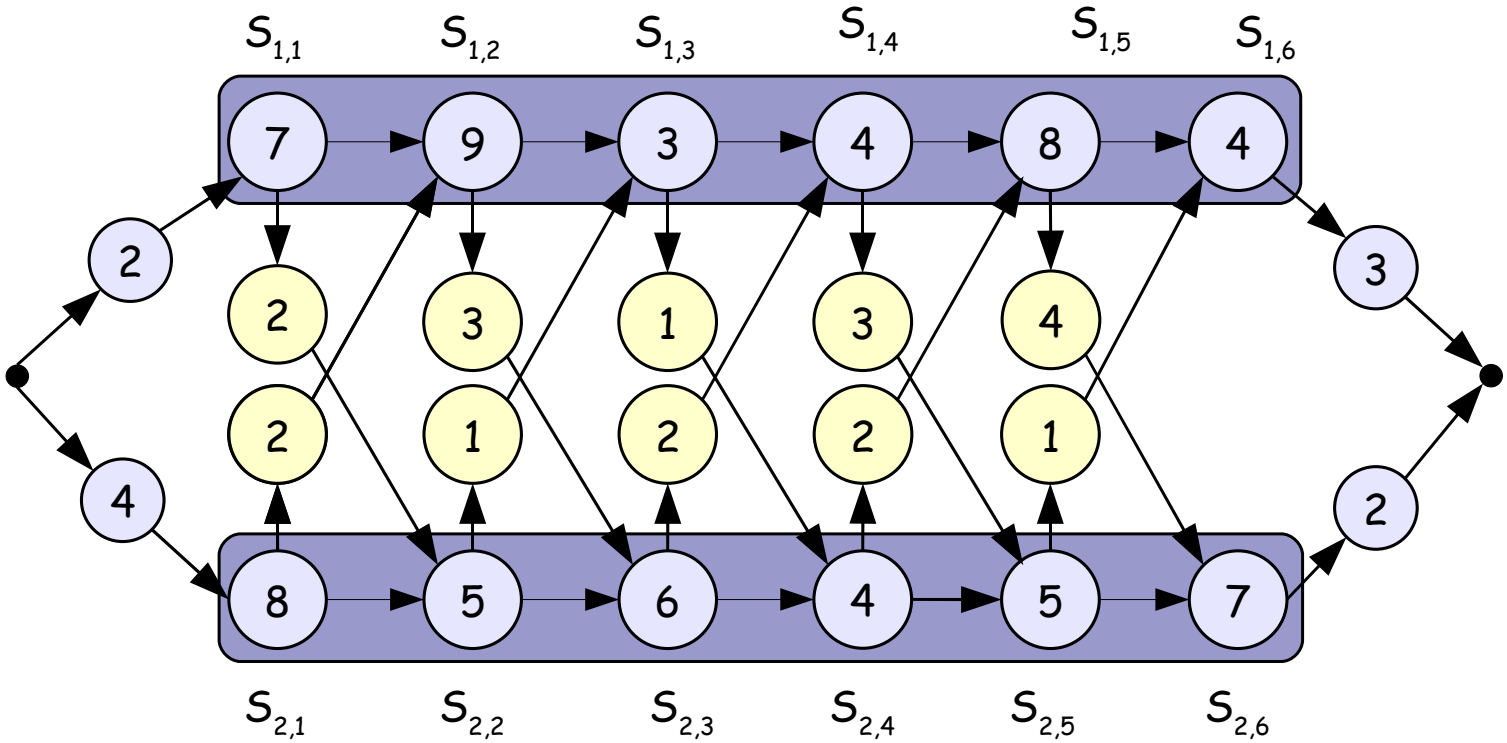
Di norma, una volta che un telaio entra in una certa catena di montaggio, prosegue lungo la stessa catena; il tempo per passare da una stazione alla successiva lungo la stessa catena è trascurabile

Tuttavia, per gestire delle situazioni particolari (come ad esempio, un ordine urgente), il responsabile della produzione può decidere di passare un'auto non ancora completata da una catena all'altra, all'uscita di una stazione qualsiasi

Il tempo per trasferire un telaio dalla catena di montaggio i dopo aver attraversato la stazione $S_{i,j}$ è $t_{i,j}$



Problema: determinare quali stazioni scegliere dalle catene 1 e 2 per **minimizzare** il tempo totale di assemblaggio di una macchina



Non possiamo usare la forza bruta, ossia calcolare il valore di **ogni** possibile soluzione e scegliere quella con valore minimo

Dobbiamo selezione n possibili stazioni ognuna delle quali può essere scelta in due modi distinti – abbiamo 2^n possibili soluzioni

Fase 1: struttura del percorso più rapido

La prima fase del paradigma di programmazione dinamica consiste nel determinare la struttura della soluzione ottima

Sia p il percorso più rapido che un telaio può seguire dal punto iniziale fino ad una data stazione, diciamo $S_{1,j}$

Come è fatto p? distinguiamo due possibili casi:

- $j = 1$: il telaio può seguire un solo percorso con valore e_1
- $j = 2, \ldots, n$: abbiamo due possibili scelte

  (1) il telaio proviene da $S_{1,j-1}$ ed va in $S_{1,j}$ impiegando un tempo trascurabile, oppure ...

Fase 1: struttura del percorso più rapido

La prima fase del paradigma di programmazione dinamica consiste nel determinare la struttura della soluzione ottima

Sia p il percorso più rapido che un telaio può seguire dal punto iniziale fino ad una data stazione, diciamo $S_{1,j}$

Come è fatto p? distinguiamo due possibili casi:

- $j = 1$: il telaio può seguire un solo percorso con valore e_1
- $j = 2, \ldots, n$: abbiamo due possibili scelte

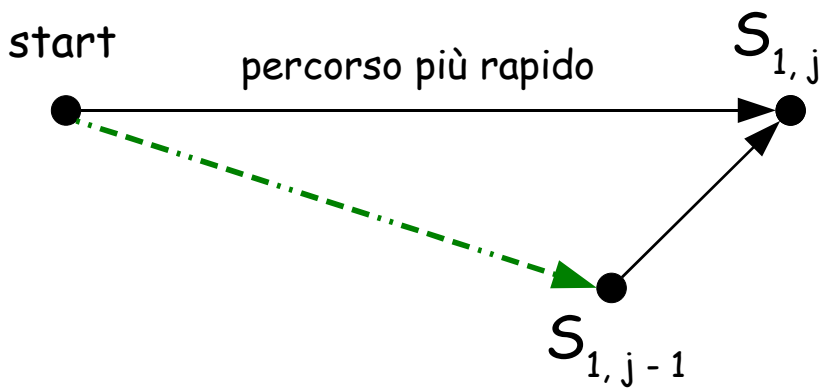
  (1) il telaio proviene da $S_{1,j-1}$ ed va in $S_{1,j}$ impiegando un tempo trascurabile, oppure ...

  (2) il telaio proviene da $S_{2,j-1}$ ed va in $S_{1,j}$ impiegando un tempo di trasferimento pari a $t_{2,j-1}$

Fase 1: struttura del percorso più rapido

Caso 1: il telaio proviene da $S_{1,j-1}$ ed va in $S_{1,j}$ impiegando un tempo trascurabile (il percorso più veloce per arrivare in $S_{1,j}$ passa per $S_{1,j-1}$)

Osservazione chiave: il telaio deve aver seguito il percorso più rapido dal punto iniziale fino a $S_{1,j-1}$

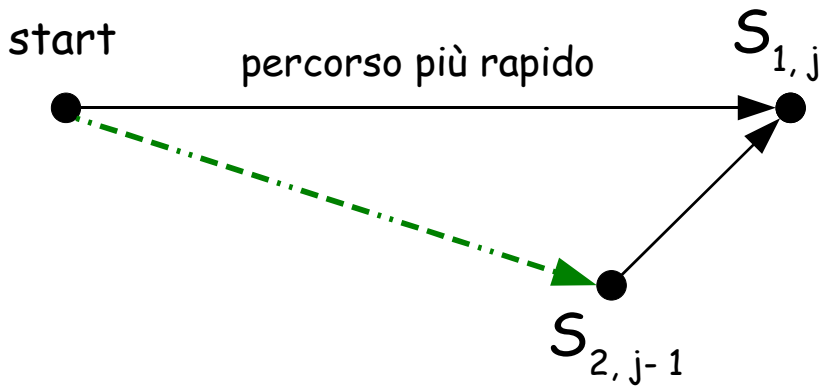


Se, per assurdo, esiste un percorso più veloce dal punto iniziale fino a $S_{1,j-1}$, allora avremmo un percorso più veloce fino a $S_{1,j}$ – impossibile

Fase 1: struttura del percorso più rapido

Caso 2: il telaio proviene da $S_{2,j-1}$ ed va in $S_{1,j}$ impiegando un tempo di trasferimento trascurabile $t_{2,j-1}$ (il percorso più veloce per arrivare in $S_{1,j}$ passa per $S_{2,j-1}$)

Osservazione chiave: il telaio deve aver seguito il percorso più rapido dal punto iniziale fino a $S_{2,j-1}$



In questo caso, l'esistenza di un percorso più veloce dal punto iniziale fino a $S_{2,j-1}$ implicherebbe l'esistenza di un percorso più veloce fino a $S_{1,j}$ – assurdo

Fase 1: struttura del percorso più rapido

Più in generale, una soluzione ottima di un problema (trovare il percorso più rapido per arrivare ad una stazione $S_{1,j}$) contiene al suo interno la soluzione ottima di sottoproblemi (trovare il percorso più rapido per raggiungere $S_{1,j-1}$ o $S_{2,j-1}$)

Faremo riferimento a questa proprietà con il termine di **sottostruttura ottima**

Possiamo usare la sottostruttura ottima per dimostrare come costruire la soluzione ottima a partire dalle soluzioni ottime dei sottoproblemi

Fase 2: una soluzione ricorsiva

Indichiamo con $f_i[j]$ il minor tempo possibile che impiega un telaio dal punto iniziale fino all'uscita della stazione $S_{i,j}$

L'obiettivo finale è quello di calcolare il tempo minimo che impiega il telaio per attraversare lo stabilimento; indichiamo con f^* questo tempo

Il telaio deve raggiungere la stazione n dalla catena 1 o 2 e poi uscire dallo stabilimento, quindi

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

Non ci resta che trovare un modo per calcolare $f_1[j]$ e $f_2[j]$ con $j = 1, 2, \ldots, n$

Fase 2: una soluzione ricorsiva

Il caso in cui $j = 1$ è abbastanza facile: infatti, $f_i[1]$ è il minor tempo possibile che impiega un telaio dal punto iniziale fino all'uscita della stazione $S_{i,1}$. Quindi:

$$f_1[1] = e_1 + a_{1,1}$$
$$f_2[1] = e_2 + a_{2,1}$$

Per il calcolo di $f_1[j]$ e $f_2[j]$ con $j = 2, 3 \ldots n$ dobbiamo ragionare per casi

$f_1[j]$ (con $j \geq 2$) è il minor tempo possibile che impiega un telaio dal punto iniziale fino all'uscita della stazione $S_{1,j}$. Abbiamo due possibili casi

- il percorso più rapido per $S_{1,j}$ passa per $S_{1,j-1}$
- il percorso più rapido per $S_{1,j}$ passa per $S_{2,j-1}$

Fase 2: una soluzione ricorsiva

$f_1[j]$ (con $j \geq 2$) è il minor tempo possibile che impiega un telaio dal punto iniziale fino all'uscita della stazione $S_{1,j}$. Abbiamo due possibili casi

- il percorso più rapido per $S_{1,j}$ passa per $S_{1,j-1}$

$$f_1[j] = \overbrace{f_1[j-1]}^{\text{tempo del percorso mimino fino a } S_{1,j-1}} + \underbrace{a_{1,j}}_{\text{tempo trascorso in } S_{1,j}}$$

- il percorso più rapido per $S_{1,j}$ passa per $S_{2,j-1}$

Fase 2: una soluzione ricorsiva

$f_1[j]$ (con $j \geq 2$) è il minor tempo possibile che impiega un telaio dal punto iniziale fino all'uscita della stazione $S_{1,j}$. Abbiamo due possibili casi

- il percorso più rapido per $S_{1,j}$ passa per $S_{1,j-1}$

$$f_1[j] = f_1[j-1] + a_{1,j}$$

- il percorso più rapido per $S_{1,j}$ passa per $S_{2,j-1}$, in maniera

$$f_1[j] = f_2[j-1] + \overbrace{t_{2,j-1}}^{\text{tempo di attraversamento}} + a_{1,j}$$

scegliamo il minimo fra i due valori

Fase 2: una soluzione ricorsiva

Quindi se $j \geq 2$

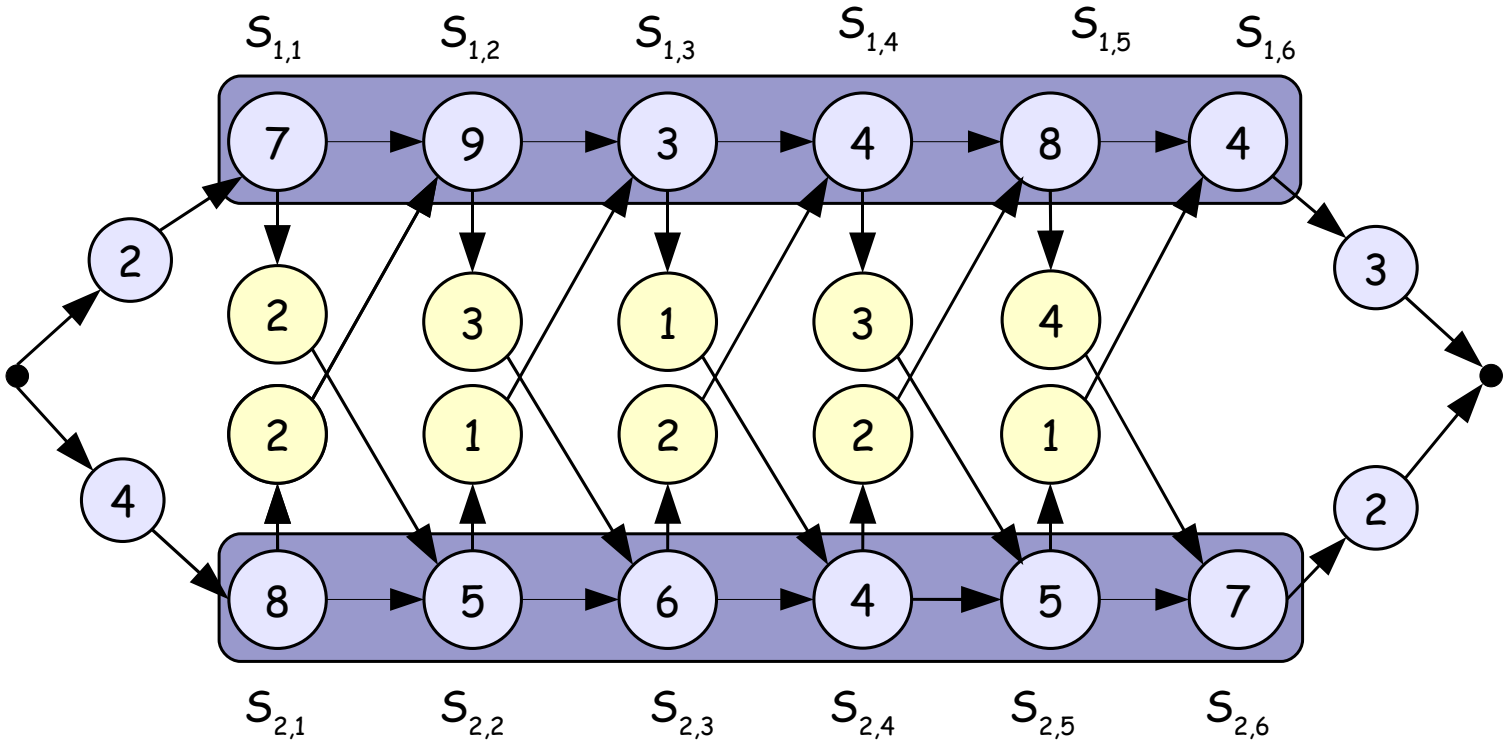
$$f_1[j] = \min\big(f_1[j-1] + a_{1,j}, f_2[j-1] + t_{2,j-1} + a_{1,j}\big)$$

Ricapitolando

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

Simmetricamente

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



j	1	2	3	4	5	6
$f_1[j]$	9	18	20	24	32	35
$f_2[j]$	12	16	22	25	30	37

$$f^* = \min(35 + 3, 37 + 2) = 38$$

Costruzione della soluzione ottima

I valori di $f_i[j]$ sono i valori delle soluzioni ottime dei sottoproblemi che ci consentono di determinare il valore della soluzione ottima

Vediamo ora come costruire la soluzione ottima; per facilitare questa costruzione definiamo:

- ℓ^* come la catena (1 o 2) la cui ultima staziona è usata nel percorso più rapido dell'intero stabilimento
- $\ell_i[j]$ come la catena (1 o 2) la cui $(j-1)$-esima stazione è usata nel percorso più rapido per arrivare ad $S_{i,j}$

  N.B: $\ell_i[1]$ non è definita perchè nessuna stazione precede la stazione 1 nelle due catene di montaggio

Costruzione della soluzione ottima

Per calcolare il valore di ℓ^* "guardiamo" come viene ottenuto il valore di f^*. Abbiamo detto che:

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

Quindi

$$\ell^* = \begin{cases} 1 & \text{se } f_1[n] + x_1 \leq f_2[n] + x_2 \\ 2 & \text{se } f_1[n] + x_1 > f_2[n] + x_2 \end{cases}$$

se $f^* = f_1[n] + x_1$ il percorso più rapido passa per $S_{1,n}$, altrimenti passa per $S_{2,n}$

Costruzione della soluzione ottima

Come calcoliamo $\ell_1[j]$ (per calcolare $\ell_2[j]$ procediamo in maniera del tutto simmetrica) per $j = 2, 3 \ldots, n$?

Sia $g_1[j]$ il minor tempo possibile per raggiungere $S_{1,j}$

$g_1[j]$ è pari al minor tempo possibile per raggiungere l'uscita della stazione $S_{1,j}$ (ossia $f_1[j]$) meno il tempo trascorso in $S_{1,j}$ (ossia $a_{1,j}$). Quindi $g_1[j] = f_1[j] - a_{1,j}$

Per $j = 2, 3 \ldots, n$

$$f_1[j] = \min\big(f_1[j-1] + a_{1,j}, f_2[j-1] + t_{2,j-1} + a_{1,j}\big)$$

Quindi

$$g_1[j] = \min\big(f_1[j-1], f_2[j-1] + t_{2,j-1}\big)$$

Costruzione della soluzione ottima

Ricapitolando il minore tempo possibile per raggiungere $S_{1,j}$ è:

$$g_1[j] = \min\big(f_1[j-1], f_2[j-1] + t_{2,j-1}\big)$$

Inoltre:

$$\ell_1[j] = \begin{cases} 1 & \text{se } g_1[j] = f_1[j-1] \\ 2 & \text{altrimenti} \end{cases}$$

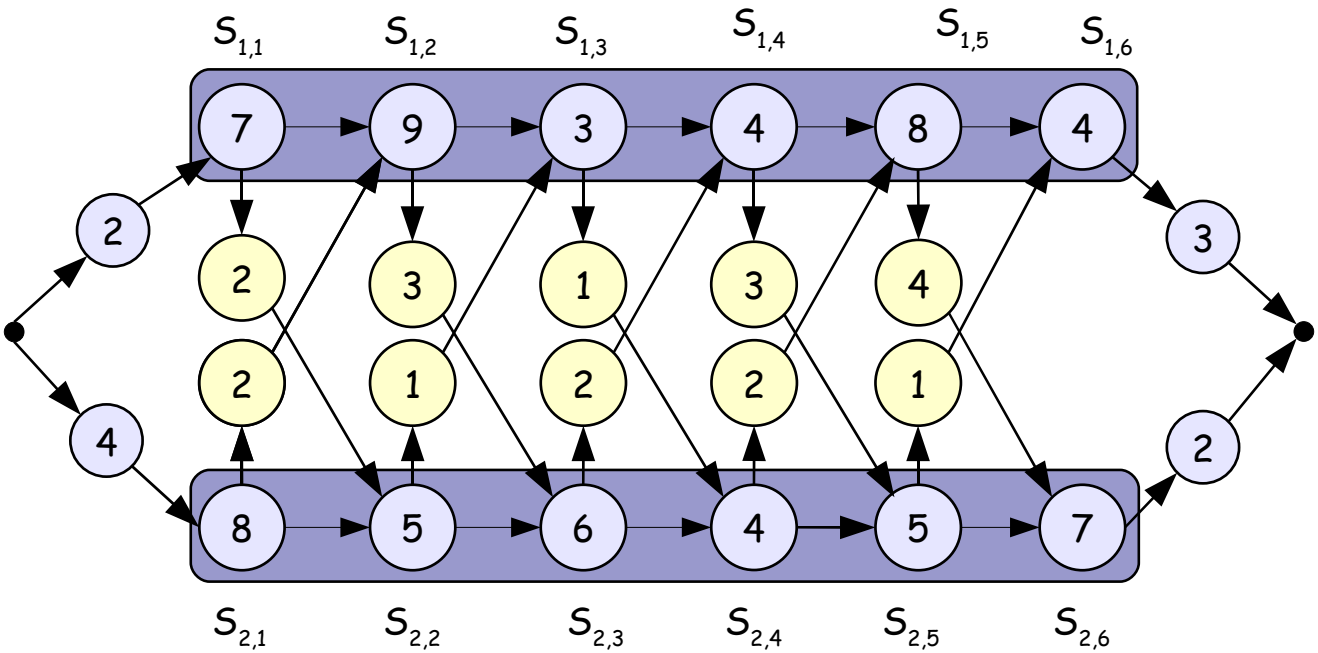
Costruzione della soluzione ottima

In maniera del tutto simmetrica, il minore tempo possibile per raggiungere $S_{2,j}$ è:

$$g_2[j] = \min\big(f_2[j-1], f_1[j-1] + t_{1,j-1}\big)$$

Inoltre:

$$\ell_2[j] = \begin{cases} 1 & \text{se } g_2[j] = f_2[j-1] \\ 2 & \text{altrimenti} \end{cases}$$



j	1	2	3	4	5	6
$f_1[j]$	9	18	20	24	32	35
$f_2[j]$	12	16	22	25	30	37
$t_{1,j}$	2	3	1	3	4	
$t_{2,j}$	2	1	2	2	1	

j	2	3	4	5	6
$\ell_1[j]$	1	2	1	1	2
$\ell_2[j]$	1	2	1	2	2

$$f^* = 38 \qquad \ell^* = 1$$

Costruzione della soluzione ottima

Identifichiamo le stazioni attraversate dal percorso minimo a partire dai valori di ℓ^*, $\ell_i[j]$ come segue:

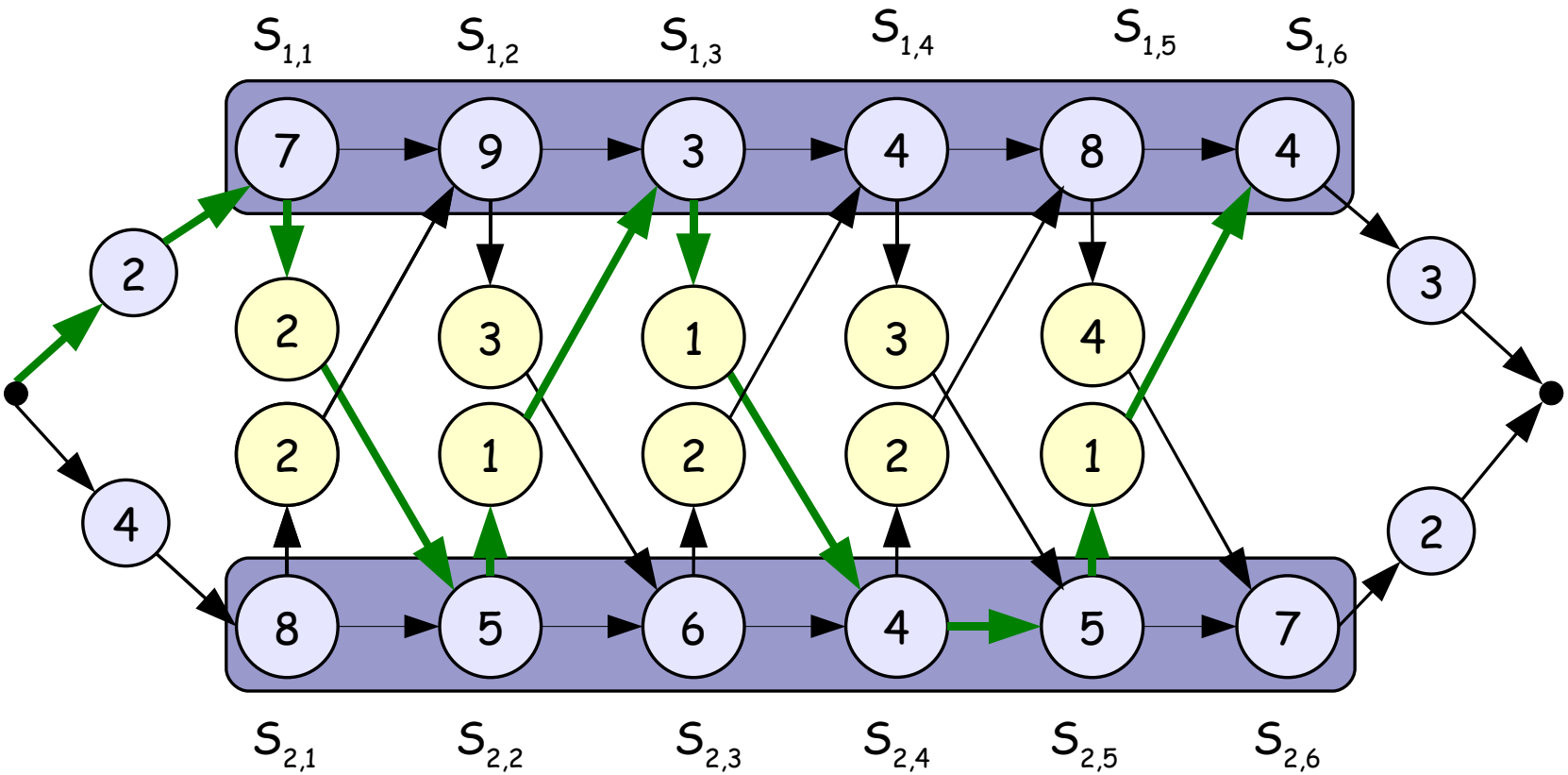
j	2	3	4	5	6
$\ell_1[j]$	1	2	1	1	2
$\ell_2[j]$	1	2	1	2	2

$\ell^* = 1$

- $\ell^* = 1 \Rightarrow$ l'ultima stazione è $S_{1,6}$
- $\ell_1[6] = 1 \Rightarrow$ passiamo sulla catena 2, stazione $S_{2,5}$
- $\ell_2[5] = 2 \Rightarrow$ rimaniamo sulla catena 2, stazione $S_{2,4}$
- $\ell_2[4] = 1 \Rightarrow$ torniamo sulla catena 1, stazione $S_{1,3}$
- $\ell_1[3] = 2 \Rightarrow$ catena 2, stazione $S_{2,2}$
- $\ell_2[2] = 1 \Rightarrow$ catena 1, stazione $S_{1,1}$

Il percorso ottimo attraversa, nell'ordine, le seguenti stazioni

$$S_{1,1}, S_{2,2}, S_{1,3}, S_{2,4}, S_{2,5}, S_{1,6}$$


$S_{1,1}$
$S_{1,2}$
$S_{1,3}$
$S_{1,4}$
$S_{1,5}$
$S_{1,6}$
7
9
3
4
8
4
2
2
3
1
3
4
3
2
2
1
2
1
4
2
8
5
6
4
5
7
$S_{2,1}$
$S_{2,2}$
$S_{2,3}$
$S_{2,4}$
$S_{2,5}$
$S_{2,6}$

L'algoritmo

FastestWay(a, t, e, x, n)

▷ a e t sono delle matrici: contengono i valori $a_{i,j}$ (con $i = 1, 2$ e $j = 1, \ldots, n$)

▷ e $t_{i,j}$ (con $i = 1, 2$ e $j = 1, \ldots, n-1$) rispettivamente, $e = \langle e_1, e_2 \rangle$, $x = \langle x_1, x_2 \rangle$

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. $\ell_1[j] \leftarrow 1$
7. **else** $f_1[j] \leftarrow f_2[j-1] + t_{2,j-1} + a_{1,j}$
8. $\ell_1[j] \leftarrow 2$
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. $\ell_2[j] \leftarrow 1$
12. **else** $f_2[j] \leftarrow f_1[j-1] + t_{1,j-1} + a_{2,j}$
13. $\ell_2[j] \leftarrow 2$

▷ questo conclude la fase di calcolo di $f_1[j]$ e $f_2[j]$

L'algoritmo

FastestWay(a, t, e, x, n)

14. **if** $f_1[n] + x_1 \leq f_2[n] + x_2$
15. **then** $f^* \leftarrow f_1[n] + x_1$
16. $\ell^* \leftarrow 1$
17. **else** $f^* \leftarrow f_2[n] + x_2$
18. $\ell^* \leftarrow 2$

PrintStation(l, n)

1. $i \leftarrow \ell^*$
2. stampa "catena" i ", stazione" n
3. **for** $j \leftarrow n$ **downto** 2
4. **do** $i \leftarrow \ell_i[j]$
5. stampa "catena" i ", stazione" $j - 1$