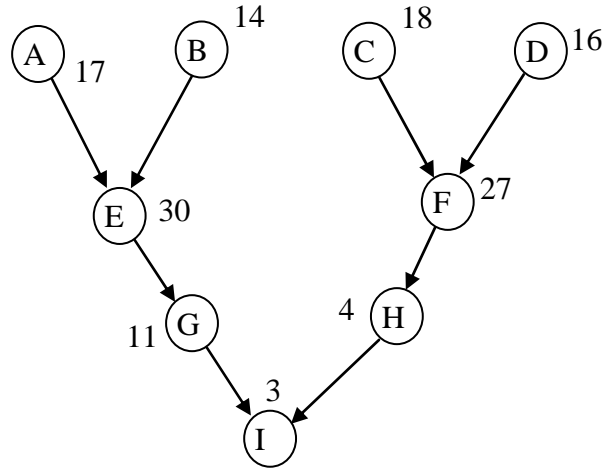


# ASSEMBLY LINE BALANCING

## Algoritmo FABLE

### DESCRIZIONE DEL PROBLEMA

E' dato il grafo di assemblaggio in figura. Si desidera progettare una linea di assemblaggio con tempo ciclo  $T_C = 40$  e minimo numero di stazioni utilizzando l'algoritmo FABLE. Costruire una soluzione iniziale con l'algoritmo RPW nella versione migliorata che assegna i task alla prima macchina fitabile. Evidenziare le soluzioni parziali costruite dall'algoritmo, la struttura dati e la sua evoluzione durante l'esecuzione.



### SOLUZIONE

1. Calcolo della soluzione iniziale con l'euristica RPW.

- a. Calcolo dei pesi posizionali. Si ricorda che  $p_i$  è la durata del task  $i$ ,  $S_i$  è l'insieme dei successori di  $i$  e che  $P_i$  è l'insieme dei predecessori di  $i$ . Utilizziamo il simbolo  $\tilde{S}_i = S_i \cup \{i\}$  per includere tra i successori anche il task  $i$ .  $PW_i = \sum_{i \in \tilde{S}_i} p_i$  è il peso

posizionale del task  $i$ .

$i$	A	B	C	D	E	F	G	H	I
$PW_i$	61	58	52	50	44	34	14	7	3

b. Ordinamento decrescente dei task: ABCDEFGHI

c. Assegnazione dei task alle stazioni:

AB CD E FG HI

d. In totale servono 5 stazioni. L'Upper Bound del problema è pertanto UB=5.

2. Calcolo del Lower Bound. FABLE prevede il calcolo di tre quantità:

a. 
$$LB_1 = \left\lceil \frac{\sum_i p_i}{T_C} \right\rceil = \left\lceil \frac{17+14+18+16+30+27+11+4+3}{40} \right\rceil = 4$$

b. 
$$LB_2 = \left\lceil \left\{ i : p_i > \frac{T_C}{2} \right\} + \left\lfloor \frac{1}{2} \left\{ i : p_i = \frac{T_C}{2} \right\} \right\rfloor \right\rceil = 2$$

c. 
$$LB_3 = \left\lceil \left\{ i : p_i > \frac{2T_C}{3} \right\} + \frac{2}{3} \left\{ i : \frac{2T_C}{3} = p_i \right\} + \frac{1}{2} \left\{ i : \frac{T_C}{3} < p_i < \frac{2T_C}{3} \right\} + \frac{1}{3} \left\{ i : \frac{T_C}{3} = p_i \right\} \right\rceil = 4$$

d. Si ottiene  $LB = \max\{LB_1, LB_2, LB_3\} = 4$ . Poiché  $UB > LB$  non possiamo dire che la soluzione trovata è ottima e procediamo con la ricerca di una soluzione migliore.

3. Calcolo dell'upper bound per ogni task, per l'applicazione della *upper bound violation*. Lo scopo è definire per ciascun task l'ultima stazione  $U_i$  compatibile con una soluzione migliore di quella disponibile. Si tratta per ciascun task  $i$  di applicare la formula  $LB$  di cui al punto precedente ai task dell'insieme  $\tilde{S}_i$ . Detto  $L_i = \max\{LB_1(\tilde{S}_i), LB_2(\tilde{S}_i), LB_3(\tilde{S}_i)\}$  tale valore, l'ultima stazione compatibile con  $i$  sarà  $U_i = UB - L_i$ . Si ottiene:

$i$	A	B	C	D	E	F	G	H	I
$L_i$	2	2	2	2	2	1	1	1	1
$U_i$	3	3	3	3	3	4	4	4	4

4. Calcolo delle coppie di task candidati all'applicazione della *task dominance*. La condizione necessaria perché  $x$  domini  $y$  ( $x \rightarrow y$ ) è che  $p_x \geq p_y$ ,  $S_x \supseteq S_y$  e che almeno una delle due relazioni sia stretta. Si ottengono le coppie

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

5. Calcolo delle etichette per l'applicazione della *labeling dominance*. L'etichetta  $l_i$  del nodo  $i$

si calcola come:  $l_i = 1 + s_i - r_i$ , dove  $s_i = \sum_{h=1}^{i-1} l_h$ ;  $r_i = \sum_{h \in P_h} l_h$ . Si ottiene:

$i$	A	B	C	D	E	F	G	H	I
$s_i$	0	1	3	7	15	28	45	75	122
$r_i$	0	0	0	0	3	12	16	29	122
$l_i$	1	2	4	8	13	17	30	47	1

### Procedura di enumerazione

1. AB | CD | E | FG –  $UBV_H$  (upper bound violation su H e I)
2. AB | CD | E | FH –  $UBV_G$
3. AB | CD | FH | E –  $UBV_G$
4. AB | D – *station dominance* (è *fitable* C, che precede D in ordine lessicografico)
5. AB | E | CD – *labeling dominance*
6. AB | E | CG –  $UBV_D$
7. AB | E | DG –  $UBV_C$
8. AB | E | G – *station dominance*
9. AC | BD – *labeling dominance*
10. AC | D – *station dominance*
11. AD – *task dominance* ( $C \rightarrow D$ , e C diventa *fitable* togliendo D)
12. BC – *task dominance* ( $A \rightarrow B$ )
13. BD – *task dominance* ( $A \rightarrow B$ )
14. CD | AB – *labeling dominance*
15. CD | B – *station dominance*
16. CD | FH | AB –  $UBV_E$
17. CD | FH | B – *station dominance*
18. D – *station dominance*

La procedura termina. La soluzione corrente di valore UB è ottima (cioè quella trovata con RPW).

Qui accanto è riportata la struttura dati al termine della procedura ma limitatamente alle prime 20 posizioni delle 123 possibili.

### Struttura dati

label	valore
1	17
2	14
3	40
4	18
5	40
6	
7	58 (al passo 9 diventa 54)
8	
9	40
10	
11	54
12	40
13	57
14	
15	80
16	80
17	
18	
19	
20	98