

# JOB SHOP SCHEDULING WITH BLOCKING

Algoritmo di tabu search

## DESCRIZIONE DEL PROBLEMA

Sono dati 4 job da eseguire su cinque macchine blocking  $M_1, M_2, M_3, M_4, M_5$ , descritti nel formato OPERAZIONE (MACCHINA, DURATA):

job 1: A ( $M_1, 10$ ) B ( $M_2, 10$ ) C ( $M_3, 6$ ) D ( $M_4, 1$ )  
job 2: E ( $M_2, 12$ ) F ( $M_5, 3$ ) G ( $M_1, 1$ ) H ( $M_3, 10$ )  
job 3: I ( $M_4, 12$ ) L ( $M_2, 15$ ) M ( $M_1, 6$ ) N ( $M_3, 10$ )  
job 4: O ( $M_4, 2$ ) P ( $M_2, 4$ ) Q ( $M_3, 18$ ) R ( $M_1, 5$ )

La soluzione iniziale  $\pi$  è data dall'ordinamento:

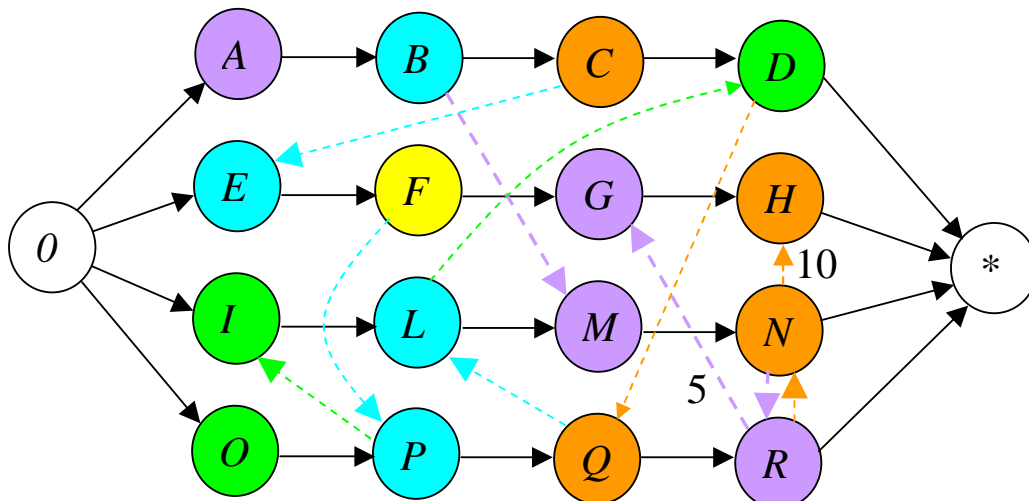
0 A B C O E F P I L D Q M R N G H \*

dove "0" e "\*" sono le operazioni fittizie *start* e *finish*.

## OBIETTIVO

1. La soluzione data è ammissibile?
2. Calcolare teste e code di ogni operazione e determinare il cammino critico.
3. determinare la mossa più vantaggiosa secondo GK09,
4. calcolare il cammino critico della nuova soluzione.

**Calcolo di teste, code e cammino critico.** *Notare che D,H,N,R non sono operazioni blocking e quindi gli archi di precedenza (N,H) e (R,G) sono pesati con le durate delle operazioni in coda (N e R), tutti gli altri archi tratteggiati hanno peso 0. Il grafo ha due cicli nulli (DQL e NR).*



	0	A	B	C	O	E	F	P	I	L	D	Q	M	R	N	G	H	*
Proc time	-	10	10	6	2	12	3	4	12	15	1	18	6	5	10	1	10	-
Teste	0	0	10	20	0	20	32	32	32	44	44	44	59	65	65	70	75	85
Code	85	75	65	59	53	53	50	49	41	26	40	23	20	15	10	10	0	0

Cammino critico 0ABCEFPILMNH\*

Blocchi: BEP; NH

Mosse ammissibili (due per blocco):  $v(B,E)$ ,  $v(E,P)$ ,  $v(N,H)$ .

NB: con la notazione  $v(x,y)$  si indica la mossa di invertire l'ordine di esecuzione di  $x$  e  $y$  sulla stessa macchina. Quindi, definendo:

$$\sigma(x) = \begin{cases} \text{l'operazione che segue } x \text{ sul suo job} & \text{se } x \text{ è blocking, es: } \sigma(B) = C \\ \text{l'operazione } x \text{ stessa} & \text{se } x \text{ non è blocking, es: } \sigma(D) = D \end{cases}$$

$v(x,y)$  richiede di:

1. rimuovere tutti gli archi disgiuntivi incidenti (entranti o uscenti in) un'operazione del job di  $x$  e lasciare invariati gli altri;
2. sostituire l'arco  $(\sigma(x),y)$  (rimosso) con  $(\sigma(y),x)$  e selezionare tutti gli archi disgiuntivi di coppie non selezionate che sono implicati da  $(\sigma(y),x)$ , cioè archi  $(\sigma(i),j)$  con  $j$  operazione appartenente allo stesso job di  $x$  e tali che  $(\sigma(j),i)$  causerebbe un ciclo positivo nel grafo.

NB: in questa fase, anche se non rappresentati sul grafo, vanno considerati come selezionati anche gli archi disgiuntivi ridondanti. Ad esempio: per la macchina  $M_2$  in figura sono rappresentati solo i tre archi  $(\sigma(B),E)$ ,  $(\sigma(E),P)$  e  $(\sigma(P),L)$  ma sono selezionati anche  $(\sigma(B),L)$ ,  $(\sigma(B),P)$  e  $(\sigma(E),L)$ . Di questi 6 archi, effettuando una mossa  $v(E,P)$  vengono rimossi  $(\sigma(B),E)$ ,  $(\sigma(E),P)$  e  $(\sigma(E),L)$  ma restano gli altri tre, compresi gli archi  $(\sigma(B),L)$  e  $(\sigma(B),P)$  non rappresentati in figura. Poi viene selezionato  $(\sigma(P),E)$  e gli archi implicati.

3. Selezionare tutti gli archi disgiuntivi non selezionati come nella soluzione di partenza.

NB: se la semplice sostituzione di  $(\sigma(x),y)$  con  $(\sigma(y),x)$  non introduce cicli positivi non è necessario eseguire i passi 2 e 3 e si possono lasciare tutti gli altri archi come nella soluzione di partenza. In altre parole il passo 2 è un passo necessario solo per evitare cicli positivi.

Per calcolare il makespan di una soluzione dopo una mossa si aggiorna l'ordine topologico dopo l'aggiunta di  $(\sigma(y),x)$  e degli archi implicati e poi si ricalcolano le teste.

**Mossa  $v(B,E)$ :** sostituendo  $(\sigma(B),E)$  con  $(\sigma(E),B)$  non viene introdotto alcun ciclo e quindi tutti gli altri archi restano selezionati come nella soluzione iniziale. Nella

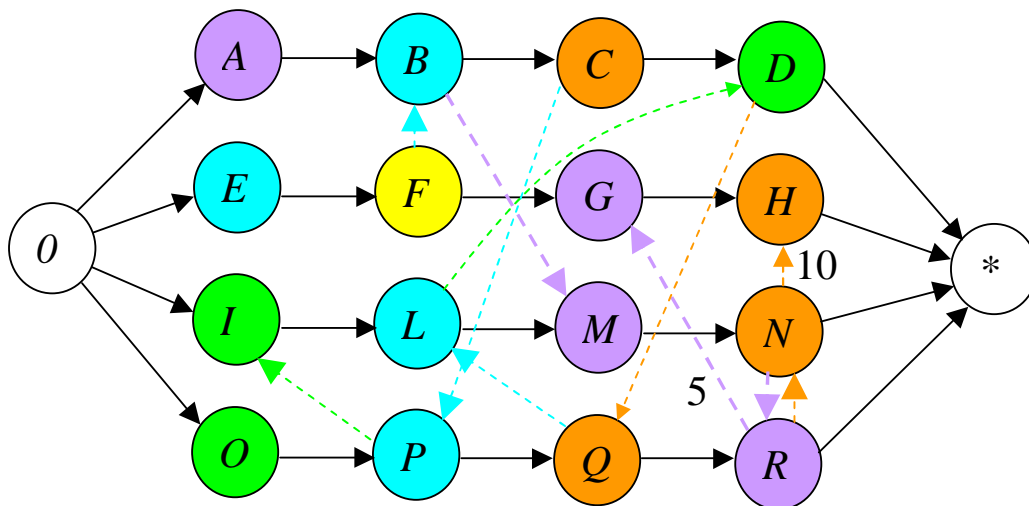
figura che segue sono rappresentati solo gli archi non ridondanti della soluzione, e solo apparentemente è stato rimosso  $(\sigma(E),P)$  e aggiunto  $(\sigma(B),P)$ .

Restano i cicli nulli LDQ e RN, che ai fini dell'ordinamento topologico saranno considerati come singoli nodi. Il nuovo ordine topologico si ottiene dal precedente:

0 A B C O E F P I L D Q M R N G H \*

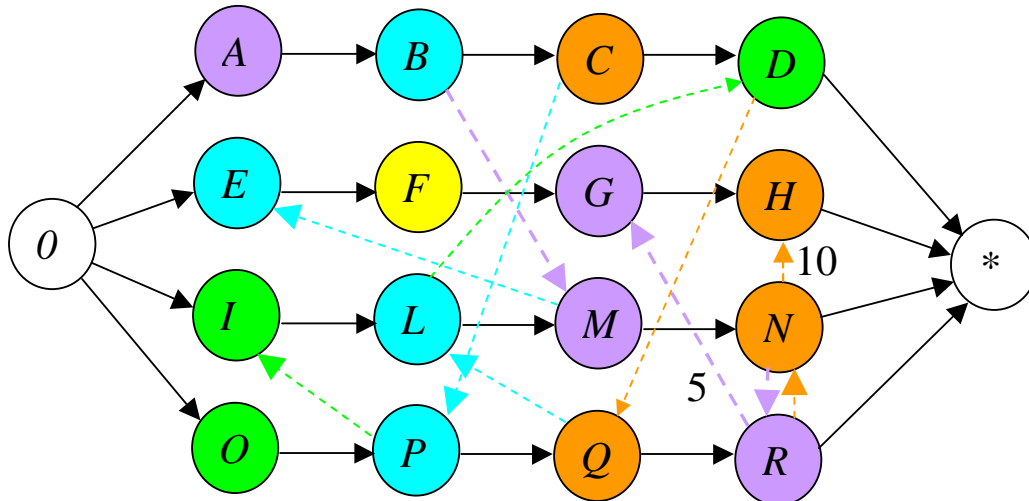
aggiungendo il nuovo arco (F,B) e aggiornando l'ordine di quelli compresi tra la posizione di F e di B (C,O,E). In particolare B e i suoi successori (solo C) si spostano dopo F. Il nuovo ordine topologico è

	0	A	O	E	F	B	C	P	I	L	D	Q	M	R	N	G	H	*
Proc time	-	10	2	12	3	10	6	4	12	15	1	18	6	5	10	1	10	-
Teste	0	0	0	0	12	12	22	22	22	34	34	34	49	55	55	55	65	75



**Mossa  $v(E,P)$ :** sostituendo  $(\sigma(E),P)$  con  $(\sigma(P),E)$  si introduce il ciclo QEFLDQ, è quindi necessario eseguire per intero l'algoritmo.

1. Si rimuovono gli archi incidenti il secondo job: oltre a (F,P) si hanno (C,E), (R,G) e (N,H) e tutti gli archi ridondanti non rappresentati in figura: (F,L), (B,G), (N,G), (D,H), (N,H).
2. Si aggiunge (Q,E) e si provano uno alla volta gli archi rimossi per vedere se introducono cicli positivi. Se si, viene selezionato il disgiuntivo, se no l'arco NON viene aggiunto in questa fase. In particolare: (C,E), (R,G) e (N,H) non introducono cicli mentre (F,L) introduce il ciclo FLDQEF, quindi viene implicato (M,E), disgiuntivo di (F,L). (B,G), (N,G), (D,H), (N,H) non introducono cicli e neanche gli archi già testati, (C,E), (R,G) e (N,H), introducono cicli dopo la selezione di (M,E).
3. Gli archi che non introducono cicli vengono selezionati come nella soluzione originale: (B,G), (N,G), (D,H), (N,H), (C,E), (R,G) e (N,H). Nella figura che segue è rappresentata la nuova soluzione, sempre omettendo gli archi ridondanti.



Il nuovo ordine topologico si ottiene dal precedente:

0 A B C O E F P I L D Q M R N G H \*

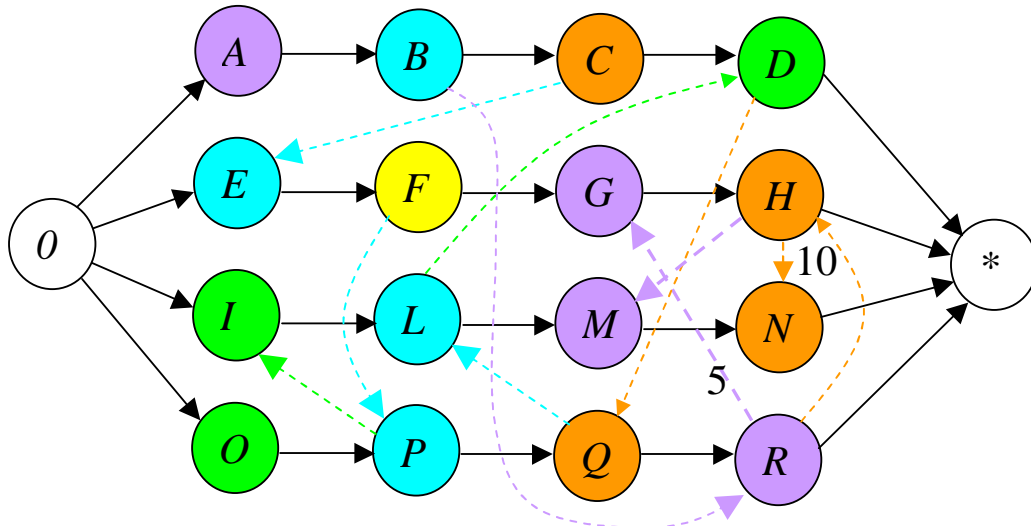
aggiungendo i nuovi archi (Q,E) e (M,E) e aggiornando l'ordine di quelli compresi tra la posizione di E e i due nuovi. In particolare E e i suoi successori (solo F) si spostano dopo M. Il nuovo ordine topologico è

	0	A	B	C	O	P	I	L	D	Q	M	E	F	R	N	G	H	*
Proc time	-	10	10	6	2	4	12	15	1	18	6	12	3	5	10	1	10	-
Teste	0	0	10	20	0	20	20	32	32	32	47	47	59	53	53	62	63	73

Si noti che nel nuovo cammino critico non troviamo l'arco aggiunto (Q,E), che è addirittura ridondante, ma un arco da questi implicato: (M,E).

**Mossa  $\nu(N,H)$ :** sostituendo  $(\sigma(N),H)$  con  $(\sigma(H),N)$  viene introdotto il ciclo HNRGH, è quindi necessario eseguire per intero l'algoritmo di aggiornamento.

1. Si rimuovono gli archi incidenti il terzo job: oltre a (N,H) si rimuovono (N,R), (R,N), (B,M), (L,D), (Q,L), (P,I) e tutti gli archi ridondanti non rappresentati in figura: (C,L), (F,L), (N,G), (D,N).
2. Si aggiunge (H,N) e si provano uno alla volta gli archi rimossi per vedere se introducono cicli positivi. Se si, viene selezionato il disgiuntivo, se no l'arco NON viene aggiunto in questa fase. In particolare: (N,R) chiude il ciclo NRGHN, per cui viene implicato (R,M). (R,N), (B,M), (L,D), (Q,L), (P,I) (C,L), (F,L) non introducono cicli mentre (N,G) chiude il ciclo NGHN, per cui viene implicato (H,M). (D,N) non introduce cicli e termina la fase di implicazione.
3. Gli archi che non introducono cicli vengono selezionati come nella soluzione originale: (R,N), (B,M), (L,D), (Q,L), (P,I) (C,L), (F,L) e (D,N). Nella figura che segue è rappresentata la nuova soluzione, sempre omettendo gli archi ridondanti.



Il nuovo ordine topologico si ottiene dal precedente:

0 A B C O E F P I L D Q R G H M N \*

aggiungendo gli archi (H,N) (R,M) (H,M) e aggiornando l'ordine. Nota che è scomparso il ciclo NR e quindi va controllato anche l'ordine dei nodi. Nel caso R deve precedere N (va bene com'è). va bene LDQ per un ciclo mentre deve essere NR per il secondo. Il nuovo ordine topologico è

	0	A	B	C	O	E	F	P	I	<u>L D Q</u>	R	G	H	M	N	*		
Proc time	-	10	10	6	2	12	3	4	12	15	1	18	5	1	10	6	10	-
Teste	0	0	10	20	0	20	32	32	32	44	44	44	62	62	62	62	68	78

La **mossa più vantaggiosa** secondo GK09 è quindi la mossa  $v(E,P)$ , che consente di ridurre il makespan a **73**. Per calcolare il cammino critico è necessario calcolare le code:

	0	A	B	C	O	P	I	<u>L D Q</u>	M	E	F	<u>R N</u>	G	H	*			
Proc time	-	10	10	6	2	4	12	15	1	18	6	12	3	5	10	1	10	-
Teste	0	0	10	20	0	20	20	32	32	32	47	47	59	50	50	62	63	73
Code	73	63	53	47	53	49	41	26	40	23	20	14	11	15	10	10	0	0

Cammino critico 0ABCPILMEFGH\* (in effetti anche D e Q sono critiche)

