ESERCIZI SULLO SCHEDULING DEL CORSO DI AUTOMAZIONE INDUSTRIALE

Spt-Edd

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alla lezione n° 10 del corso di Automazione Industriale tenute per il consorzio Nettuno)

Problema 3.1

Stavolta siete il proprietario di una macchina utensile. Vi hanno commissionato 5 lotti di pezzi, che voi (alquanto incautamente) avete promesso per certe date, prima di considerare quanto tempo ci sarebbe voluto a produrle. Per ciascun lotto, ecco il tempo p_i necessario per produrlo (in giorni) e la data d_i per la quale avete promesso la consegna.

i	1	2	3	4	5
p_i	5	4	7	6	5
d_i	4	2	3	2	4

Per ogni giorno di ritardo sulla consegna di ciascun lotto, dovete pagate una penale di 100,000 lire. Come sequenziare i lotti al fine di minimizzare i costi?

SOLUZIONE

Come si vede, il problema è quello di minimizzare la somma dei ritardi dei vari lotti. Tale problema (da non confondersi con quello in cui si vuole minimizzare il massimo ritardo) non è, in generale, affatto semplice (ne è stata dimostrata l'NP-completezza nel 1990). Tuttavia, dando uno sguardo ai dati, osserviamo che ciascun lotto ha una durata tale da farlo terminare senz'altro in ritardo, qualunque sia la sua posizione nella sequenza. In altri termini, la tardiness viene in questo caso a coincidere con la lateness. Il problema di minimizzare la somma delle lateness dei vari lotti è semplice, in quanto può essere risolto tramite applicazione della regola SPT:

2 1 5 4 3

cui corrisponde un ritardo totale di 2 + 5 + 10 + 18 + 24 = 59 giorni e dunque un costo di 5,900,000 lire.

Problema 3.2

La vostra officina deve produrre 5 lotti di pezzi, ciascuno con un suo tempo di processamento (in giorni) p_i. Per quel che riguarda le modalità di consegna, il cliente vi propone il seguente contratto: egli ha fissato, per ogni lotto, una data d_i. Se tale data è violata, la penale è di 100,000 lire per ogni giorno di ritardo. D'altro canto, il cliente è disposto a pagare un premio di 100,000 lire per ogni giorno in anticipo rispetto alla data d_i, per ogni lotto.

Vi sembra un contratto conveniente?

i	1	2	3	4	5
p_i	4	7	2	7	8
d_i	6	8	16	20	20

SOLUZIONE

Come si vede, l'obiettivo è quello di minimizzare la somma delle *lateness*, e quindi ancora una volta il problema è risolto applicando la regola SPT. Si noti che ciò non sarebbe stato possibile se, ad esempio, il premio per ogni giorno di anticipo fosse stato diverso dalla penale per ogni girono di ritardo. Un sequenziamento ottimo dei lotti è dunque:

3 1 2 4 5

cui corrisponde un costo di: -14 + 0 + 5 + 0 + 8 = –1, ossia complessivamente guadagnamo 100,000 lire, e dunque il contratto è conveniente.

Problema 3.3

Una società che produce dolciumi deve soddisfare un ordine per la produzione di cinque lotti di tavolette di cioccolata. Per quanto concerne la consegna dei pezzi, il cliente ha dettato delle date di consegna e vi pone di fronte alla scelta tra due tipi di contratto. In un primo caso, nel caso di ritardi nelle consegne, vi viene chiesto di pagare una cifra pari a un milione moltiplicato il massimo ritardo (in giorni) di un lotto (indipendentemente quindi dal numero di lotti in eventuale ritardo). Un secondo contratto invece è concepito nel seguente modo: nel caso la somma dei ritardi sia positiva, pagherete un numero di milioni pari a tale ritardo complessivo; se invece la somma dei ritardi è negativa (ossia, complessivamente vi è un anticipo rispetto alle date di consegna), un numero di milioni pari a tale anticipo complessivo verrà invece ricevuto in premio. Considerando che ciascun lotto i richiede l'uso di un centro di lavorazione per un tempo p_i (in giorni) riportato più sotto, quale contratto è più conveniente e perché (e in quale ordine vanno sequenziati i lotti)?

i	1	2	3	4	5
p_i	2	3	5	4	6
d_i	9	10	7	8	15

SOLUZIONE

Per valutare la bontà del primo contratto, dobbiamo evidentemente utilizzare la regola EDD. Sequenziando i lotti nell'ordine:

3 4 1 2 5

il ritardo massimo T_{max} si ha in corrispondenza del lotto 5, e vale 5. Dovremmo dunque pagare 5 milioni.

Per valutare il secondo contratto, dobbiamo sequenziare i lotti in modo da minimizzare la somma delle lateness, dal momento che il premio per ogni giorno di anticipo coincide con la penale per ogni giorno di ritardo. Basta dunque sequenziare i lotti in ordine SPT:

1 2 4 3 5

cui corrisponde un valore di lateness pari a $-7-5+1+7+5 = 1$, e dunque complessivamente pagheremmo 1 milione.

Il secondo contratto è dunque più conveniente.

Moore

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 10 e n° 11 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

Problema 3.4

Siete i proprietari di un centro di lavorazione per la produzione di pezzi meccanici. Uno dei vostri clienti vi ha appena ordinato un insieme di lotti, specificando, per ciascuno di essi, una data di consegna d_i (espressa in giorni a partire da oggi). In base all'entità e al tipo di lavorazioni, voi calcolate che il lotto *i*–esimo richiede p_i giorni per essere prodotto.

i	p_i	d_i
A	2	8
B	3	6
C	3	9
D	3	10
E	5	12
F	4	7

A questo punto, il cliente vi propone di scegliere tra due tipi di contratto: il primo prevede che voi paghiate una penale di 2 milioni per ogni lotto non consegnato in tempo. Il secondo, invece, non prevede penale per i lotti consegnati fino a due giorni dopo la scadenza d_i, ma successivamente la penale è di 2.5 milioni per lotto. Quale contratto è più conveniente (e, ovviamente, perché)?

SOLUZIONE

Basta evidentemente applicare due volte l'algoritmo di Moore. La prima volta con le due date indicate, la seconda con le due date aumentate di 2 unità.

Si segue dunque la regola EDD, fino ad avere il primo lotto in ritardo. I primi due lotti arrivano in tempo: B e F, mentre A giungerebbe in ritardo. Tra B, F e A il più lungo lotto è F, che viene quindi posto alla fine. La sequenza rimane:

B A

Il prossimo lotto nell'ordine EDD è C, che all'istante 8 termina in tempo. Invece D, terminerebbe in ritardo. Tra B, A, C e D il più lungo è proprio D, che viene dunque tolto e messo in fondo. La sequenza è per ora

B A C

il job E è però lungo 5, e dunque anch'esso terminerebbe in ritardo. In definitiva, si ha che solo 3 lotti giungono in tempo (B, A e C) e dunque il costo sarebbe di 6 milioni.

Con il secondo contratto, aumentiamo tutte le due date di due unità. Stavolta il primo lotto a eccedere la propria due date secondo l'ordine EDD è C. Tra B, F, A e C il più lungo è F che viene quindi eliminato. La sequenza è per ora

B A C

A questo punto, D termina in tempo mentre E eccede la propria due date. E è il più lungo e viene posto in fondo. In definitiva, i lotti B, A, C e D (in quest'ordine) giungono in tempo e dunque solo due lotti vengono terminati in ritardo. Il costo di questo contratto è dunque 5 milioni, più conveniente del precedente.

Problema 3.5

La vostra macchina utensile deve produrre 6 lotti di pezzi, ciascuno dei quali richiede la macchina per una durata p_i (espressa in giorni) e ha una data di consegna d_i:

i	1	2	3	4	5	6
p_i	4	5	6	6	8	9
d_i	31	17	19	24	15	28

La perdita di credibilità che deriverebbe dal non rispettare anche una sola data di consegna è considerata inaccettabile. Se necessario, è possibile ricorrere a subcommittenti, i quali assicurano pronta consegna, al prezzo però di 1 milione per ciascun lotto subcommissionato. Qual è il costo minimo che siamo costretti a pagare?

SOLUZIONE

L'algoritmo da applicare è quello di Moore. Infatti, l'obiettivo equivale a minimizzare il numero di lotti che arrivano in ritardo. Procedendo come nel Problema 3.7, si ottiene il sequenziamento

2 3 4 6 1

mentre il lotto 5 verrà subcommissionato.

Problema 3.6

La NASA si trova in grave ritardo nella costruzione di un nuovo propulsore, e si rivolge alla vostra azienda per la fornitura di alcuni pezzi. Vi vengono richiesti tre reattori, con scadenza al 10, 15 e 20 del mese rispettivamente; due pannelli atermici, con scadenza il 10 e il 20 del mese; e infine un alettone, per il 15 del mese. Tutte le materie prime verranno fornite all'inizio del mese. Ogni articolo che saremo in grado di consegnare entro la data richiesta verrà pagato centomila dollari. Tuttavia, la NASA non comprerà alcun articolo consegnato oltre la scadenza, in quanto ciò significherebbe un inaccettabile ritardo nei suoi piani.

L'offerta è assai allettante. Mobilitando tutte le risorse di cui disponete, giungete alla conclusione che per costruire un reattore ci vogliono 6 giorni, per un pannello 5 e per un alettone ce ne vogliono 4.

Sapreste applicare un opportuno algoritmo per massimizzare il profitto?

SOLUZIONE

Il problema consiste nel massimizzare il numero di pezzi consegnati in tempo, ovvero di minimizzare il numero di pezzi in ritardo. Ciò si risolve applicando l'algoritmo di Moore (Problema 3.7) all'istanza:

i	1	2	3	4	5	6
p_i	6	6	6	5	5	4
d_i	10	15	20	10	20	15

Una soluzione ottima è data dal produrre, nell'ordine:

4 2 6 5

mentre 1 e 3 non vengono prodotti, in quanto arriverebbero in ritardo.

Lawler

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 12 e n° 13 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

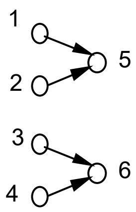
Problema 3.7

Una macchina deve eseguire 6 lavori, ciascuno dei quali è caratterizzato da una durata t_i, una data di consegna d_i e un *peso* w_i, che può pensarsi come una misura dell'importanza di ciascun lavoro. Tra i lavori esiste una relazione di precedenza, illustrata in figura.

Determinare la sequenza di lavorazione sulla macchina in modo tale da minimizzare il massimo valore che assume la quantità

$$w_i \, T_i$$

ove T_i indica la *tardiness* del lavoro *i*–esimo.



i	d_i	w_i	τ_i
1	8	2	4
2	12	1	3
3	8	2	2
4	8	1	4
5	12	4	1
6	13	3	1

SOLUZIONE

Si tratta di applicare l'algoritmo di Lawler. Allochiamo quindi i job a partire dal fondo dello schedule. La somma di tutti i tempi di processamento è 15. Inizialmente, gli unici due candidati a essere ultimi sono i job 5 e 6. Se essi terminano all'istante 15, danno come valori di funzione obiettivo 4*3=12 e 3*2=6 rispettivamente. Scegliamo quindi il job 6:

_ _ _ _ _ 6

aggiorniamo la somma dei tempi dei job non schedulati, che dà 15 – 1 = 14. Ora i candidati sono i job 3, 4 e 5. I valori di funzione obiettivo sono 2*6=12, 1*6=6 e 4*2=8 rispettivamente, e dunque scegliamo il lavoro 4:

_ _ _ _ 4 6

avendo il lavoro 4 durata pari a 4, la somma dei tempi dei job non schedulati si abbassa a 10. I candidati sono ora 3 e 5; poiché il job 5 terminerebbe entro la propria due date mentre il job 3 terminerebbe comunque in ritardo, scegliamo il job 5:

_ _ _ 5 4 6

A questo punto i candidati sono 1, 2 e 3 e la loro somma dei tempi è 9. L'unico job che termina in tempo all'istante 9 è il job 2:

_ _ 2 5 4 6

Poiché a questo punto la somma dei tempi dei job non schedulati è pari a 6, e sia 1 che 3 hanno date di consegna più alte, le due soluzioni (1 3 2 5 4 6) e (3 1 2 5 4 6) sono ugualmente ottime. Il valore ottimo della funzione obiettivo è 6 (ottenuto in corrispondenza ai job 4 e 6).

Problema 3.8

Un centro di lavorazione deve processare, in un certo mese, un insieme di lotti di pezzi. Per ogni lotto sono specificate la durata complessiva (in giorni) delle lavorazioni p_i e una data di consegna d_i. Se un lotto i viene terminato entro d_i, allora la consegna avviene a cura del cliente, altrimenti occorre accollarsi il trasporto, per il quale si rende necessario l'impiego di w_i camion (riutilizzabili poi per ogni altra consegna).

i	p_i	d_i	w_i
A	3	15	2
B	4	17	1
C	5	23	3
D	6	15	4
E	7	14	3

Il gestore dell'azienda vuole dotarsi del minimo numero di camion necessari per far fronte alle consegne ritardatarie. Qual è tale numero minimo e in che ordine vanno sequenziati i lotti?

SOLUZIONE

Il problema va risolto con l'algoritmo di Lawler. Infatti, se anche un solo lotto costringe all'uso di w_i camion, questo è ciò che dobbiamo pagare. Dunque, occorre minimizzare il massimo w_i di un lotto che termina in ritardo. Ciò equivale ad associare a ognuno dei 5 lotti una funzione $g_i(C_i)$ non decrescente del proprio tempo di completamento (e dunque regolare), che vale 0 per $C_i <= d_i$ e vale w_i per $C_i > d_i$. In questo modo, costruendo al solito la sequenza a partire dal fondo, si ha che all'istante finale T= 25 qualunque job arriva in ritardo: conviene allora sequenziare per ultimo B, che richiede un solo camion. Si aggiorna quindi T, che diviene pari a 21. Il minimo delle g_i si ha in corrispondenza di C, che all'istante 21 è ancora in tempo. Aggiornando *T*, che vale 21 – 5 = 16, si ha che tutti e tre i rimanenti lotti arrivano in ritardo. Selezioniamo allora A, che è quello avente il w_i più basso, pari a 2. A questo punto, gli altri due job possono essere sequenziati in uno qualunque dei due modi. In definitiva si ottengono le due sequenze ottime:

D E A C B oppure E D A C B.

Problema 3.9

Una società che produce ricambi per autovettura deve soddisfare un ordine per la produzione di cinque lotti di pezzi. Ciascun lotto i richiede l'uso di un centro di lavorazione per un tempo p_i (in giorni). Per quanto concerne la consegna dei pezzi, è stato siglato il seguente contratto: a ciascun lotto sono associate tre "date critiche" d_i', d_i''e d_i''', il cui significato è il seguente. Il mancato completamento *anche di un solo lotto* entro la propria data d_i' si traduce in una penale pari a x; il mancato completamento *anche di un solo lotto* entro la propria data d_i'' si traduce in una penale pari a $y > x$ e infine, il mancato completamento *anche di un solo lotto* entro la propria data d_i''' comporta un costo $z > y$ in quanto a quel punto il committente annullerebbe l'ordine nel suo complesso (restituendo anche i pezzi già consegnati). Sottolineiamo che i costi x, y e z sono indipendenti dal *numero* di lotti che superano le rispettive date d_i', d_i'' e d_i'''.

In quale ordine vanno sequenziati i lotti al fine di minimizzare i costi (e perché).

i	1	2	3	4	5
p_i	1	2	3	4	5
d_i'	2	7	8	5	6
d_i''	9	8	10	9	15
d_i'''	16	8	15	11	15

SOLUZIONE

Il problema può essere risolto in due diversi modi. Quello più elegante è forse quello consistente nell'applicare l'algoritmo di Lawler. Infatti, è possibile associare a ciascun lotto una funzione $g_i(C_i)$ che vale 0 per $C_i <= d_i'$, vale x per $d_i' < C_i <= d_i''$, vale y per $d_i'' < C_i <= d_i'''$, e vale infine z per $C_i > d_i'''$.

Partendo allora da $t = 15$, si ha che il lotto da seqeunziare per ultimo sarà il lotto 5, il quale è l'unico che ci consente di pagare una penale pari a x.

_ _ _ _ 5

Aggiornando t al valore 10, osserviamo che il lotto 3 è l'unico che consente di pagare ancora solo x, e dunque

_ _ _ 3 5

A questo punto t è pari a 7, e chiaramente, in questo caso, l'ordine in cui vengono svolti gli altri tre lotti è indifferente, in quanto tutti hanno $d_i'' > 7$. La penale da pagare è quindi pari a x.

Un metodo alternativo poteva essere quello di applicare più volte la semplice regola EDD, osservando di volta in volta se T_{max} risulta positivo o nullo. In effetti, eseguendo EDD rispetto alle date d_i' si otterrebbe $T_{max} > 0$, mentre già rispetto alle d_i'' si avrebbe $T_{max} = 0$.

Smith

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 12 e n° 13 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

Problema 3.10

Una unità operatrice deve svolgere 6 lavori, ciascuno dei quali è caratterizzato da una durata (espressa in giorni) e una due date:

i	1	2	3	4	5	6
p_i	4	2	3	4	2	5
d_i	13	13	5	14	1	18

il problema consiste nel sequenziare i 6 lavori in modo tale che la massima violazione di una due date non sia superiore a 2 giorni, e che sia minimizzato il tempo medio trascorso nel sistema da un lavoro (F). Possibilmente, indicare una sequenza ottima efficiente, ovvero tale che si possa dire con certezza che F non può migliorare senza aumentare il massimo ritardo accettabile e viceversa.

SOLUZIONE

Occorre applicare l'algoritmo di Smith-Van Wassenhove. Anzitutto, quindi, aumentiamo di 2 tutte le due date, dal momento che si richiede una sequenza (se esiste) tale che nessun job arrivi con più di 2 giorni di ritardo. La tabelmla diventa:

i	1	2	3	4	5	6
p_i	4	2	3	4	2	5
d_i	15	15	7	16	3	20

Allochiamo quindi i job a partire dal fondo dello schedule. La somma di tutti i tempi di processamento è 20. L'unico job che, secondo le nuove due date, sequenziato per ultimo, può terminare in tempo è job 6:

_ _ _ _ _ 6

A questo punto, l'istante di completamento del penultimo job è l'istante 15. Adesso sono ben tre i job candidati a essere messi al penultimo posto, in quanto hanno una due date = 15: i job 1, 2 e 4. Secondo l'algoritmo di Smith, poiché vogliamo minimizzare la somma dei tempi di completamento, la scelta va ristretta a quello o quelli più lunghi. Nel nostro caso ce ne sono due: 1 e 4. Se vogliamo poi anche una sequenza efficiente, tra questi dobbiamo scegliere quello con la due date più grande, e dunque il job 4:

_ _ _ _ 4 6

il nuovo istante di fine è ora 11. I job tra cui selezionare il predecessore del job 4 sono ancora 1 e 2. Scegliamo il job 1, in base alla regola di Smith:

_ _ _ 1 4 6

la scelta è ora ristretta ai job 2 e 3 (entrambi con due date = 7). Poiché il job 3 è più lungo, lo selezioniamo:

_ _ 3 1 4 6

infine, si ottiene:

5 2 3 1 4 6.

Problema 3.11

Questo mese la vostra officina deve produrre 6 lotti di pezzi, ciascuno con una sua data di consegna d_i e un suo tempo di processamento (in giorni) p_i. All'inizio del mese acquistate le materie prime: tra capitale immobilizzato e affitto del magazzino, voi calcolate che venite a pagare 100,000 lire al giorno per ogni lotto che non avete ancora ultimato. D'altro canto, violare le date di consegna dei job è considerato inaccettabile.

i	1	2	3	4	5	6
p_i	3	5	6	2	7	4
d_i	27	11	11	27	19	24

In quale ordine converrà sequenziare i job e perché?

SOLUZIONE

L'obiettivo del problema equivale a minimizzare la somma dei giorni di lavorazione di tutti i lotti. Questo obiettivo è perciò quello di minimizzare la somma dei tempi di flusso, ovvero, essendo tutte le release date nulle, la somma dei tempi di completamento, con il vincolo che nessun job arrivi in ritardo. Basterà allora sequenziare i lotti secondo l'algoritmo di Smith, ovvero:

2 3 5 4 6 1

cui corrisponde una spesa di 10,500,000 lire.

Problema 3.12

Un cliente vi ha commisionato 5 lotti di pezzi, ciascuno con un suo tempo di processamento (in giorni) p_i e una sua data di consegna d_i. *Per ogni giorno* in cui un lotto di pezzi non è terminato (e quindi contribuisce al work-in-process), voi pagate un costo di immagazzinamento che è uguale per tutti i lotti, e pari a £.10,000. Inoltre, avete concordato con il vostro cliente che pagherete una multa pari a £.15,000 per ogni giorno di ritardo *del lotto più in ritardo* sulla consegna.

In che ordine conviene sequenziare i lotti e perché?

i	1	2	3	4	5
p_i	2	3	3	2	4
d_i	9	6	10	10	14

SOLUZIONE

E' un problema in cui i due obiettivi, ossia il tempo totale di flusso, e il ritardo massimo, sono combinati secondo certi coefficienti di costo: precisamente, la funzione obiettivo è:

$$15{,}000\ T_{max} + 10{,}000\ \mathrm{S}_i F_i$$

vogliamo allora trovare soluzioni efficienti, partendo dal caso $T_{max} = 0$. In questo caso, applicando l'algoritmo di Smith, si ottiene la sequenza:

1 2 4 3 5

che dà un tempo totale di flusso pari a 38, e dunque un costo di £.380,000.

Aumentiamo ora tutte le due date di 1 giorno, ottenendo:

i	1	2	3	4	5
p_i	2	3	3	2	4
d_i	10	7	11	11	15

e riapplicando Smith, si ottiene:

1 4 2 3 5

che dà un tempo totale di flusso pari a 37. In effetti, il lotto 2 arriva con un ritardo pari a 1, che è proprio T_{max}. Il costo complessivo è dunque 15,000 + 370,000 = 385,000, peggiore del precedente.

A questo punto osserviamo che la sequenza ottenuta coincide con quella SPT, ovvero, in altri termini, permettere un T_{max} più elevato non porterebbe alcun miglioramento in termini di tempo totale di flusso. Dunque, la soluzione ottima è quella di costo £.380,000.

Sequenziamento di due lavori con incompatibilità

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n°15, n° 16 e n° 17 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

Problema 3.13

Un sistema di produzione è caratterizzato da una cella costituita da due centri di lavorazione, da un magazzino utensili e da due sequenze di operazioni ciascuna assegnata ad un centro. Ogni operazione di una sequenza richiede l'utilizzo di un determinato utensile. I due centri possono procedere in parallelo nell'esecuzione delle operazioni assegnate finchè non sia richiesto l'uso dello stesso utensile (operazioni incopatibili). Se questo accade si genera un conflitto ed uno dei due centri di lavorazione deve attendere che l'altro rilasci l'utensile.

Il problema consiste nel determinare, al sorgere di ogni conflitto, quale dei due centri debba attendere e quale invece debba iniziare la lavorazione.

I tempi di processamento dei due lavori sulle macchine sono:

macchina	1	2	3	4	5
A	10	3	5	3	3
B	6	5	7	5	7

Le coppie di operazioni incopatibili (quelle che richiedono, cioè, lo stesso utensile) sono:

A2,B3; A4,B4; A5,B5.

Si vuole sequenziare i due lavori su ogni macchina in modo da minimizzareil tempo di completamento totale.

SOLUZIONE

Il problema si riconduce ad un problema di cammino minimo su un grafo aciclico opportuno e può essere risolto applicando un algoritmo di programmazione dinamica.

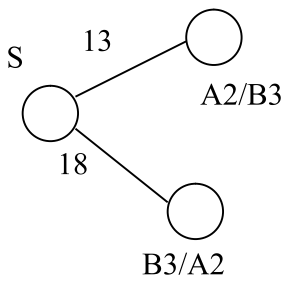
B5					
B4					
B3					
B2					
B1					
	A1	A2	A3	A4	A5

Indichiamo nel seguito con $A^{(i)}$ e con $B^{(j)}$ rispettivamente le sottosequnenze di operazioni Ai, Ai+1, … ,A5 e Bj, Bj+1, … ,B5. Siano inoltre t(Ai) e t(Bj) le durate delle corrispondenti operazioni.

All'inizio i due lavori sono eseguibili in parallelo fino ad incontare la prima coppia di operazioni incopatibili. A questo punto, come già detto, una delle due deve attendere il completamento dell'altra. Fatta una scelta, chiamiamo *milestone* l'istante di completamento della prima delle due operazioni in conflitto.

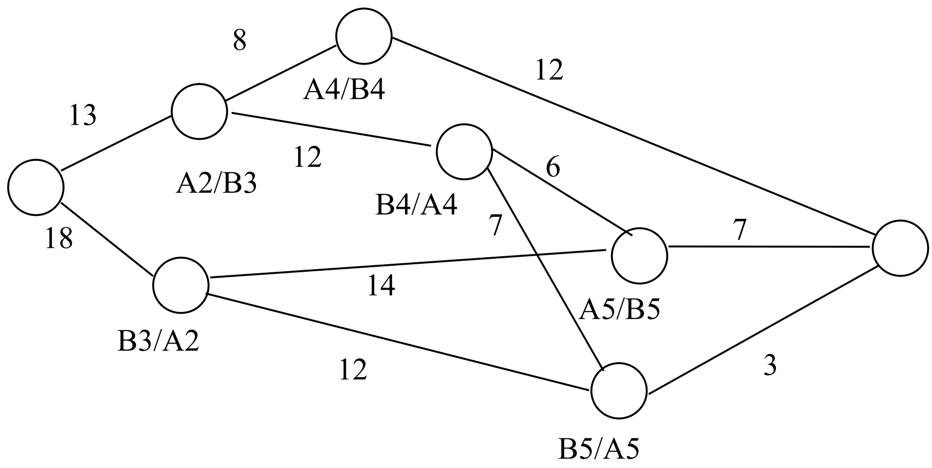
Se Ai è eseguita prima di Bj, la distanza temporale dall'inizio fino al milestone è quindi t(A1)+…+ t(Ai). Analogamente, nell'altro caso, tale distanza sarà t(B1)+…+ t(Bj).

Queste due alternative si possono rappresentare con due nodi etichettati con rispettivamente con Ai/Bj e Bj/Ai, si aggiunga, inoltre, un nodo S rappresentante il punto di inizio connesso ai due nodi precedenti. I due archi sono pesati con la corrispondente durata temporale.



Da ognuna delle due situazioni precedenti possiamo poi proseguire, considerando il nodo relativo all'alternativa scelta come il punto di inizio di un sottoproblema in cui i due lavori da eseguire, riferendosi al milestone Ai/Bj sono $A^{(i+1)}$ e $B^{(j)}$.

Reiterando il ragionamento precedente si ottiene il grafo in figura. In cui gli archi sono pesati con la distanza tra due milestone, eccetto qulli entranti in nel nodo P, che invece sono pesati con la distanza dal milestone alla fine dello schedule. Il cammino minimo su questo grafo dà lo schedule ottimo.



Gilmore-Gomory

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 13 e n° 14 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

3.14 Tempi di set-up dipendenti dalla sequenza

Si consideri una cella di produzione costituita da una macchina che deve processare un insieme di job, tutti disponibili all'istante inziale, con tempi di set-up dipendenti dalla sequenza di processamento.

In particolare ad ogni job sono assegnati i parametri a_j e b_j, che indicano lo stato di inizio e di fine della macchina. Il tempo di set-up per processare il job k dopo il job j risulta essere $|a_k - b_j|$.

I job ed i relativi parametri sono espressi nella seguente tabella:

job	0	1	2	3	4	5	6
b_j	1	15	6	24	33	29	21
a_k	3	28	46	41	14	35	18

Trovare il sequenziamento che minimizzi il tempo totale di set-up.

SOLUZIONE

Il problema può essere risolto applicando l'algolritmo di Gilmore e Gomory.

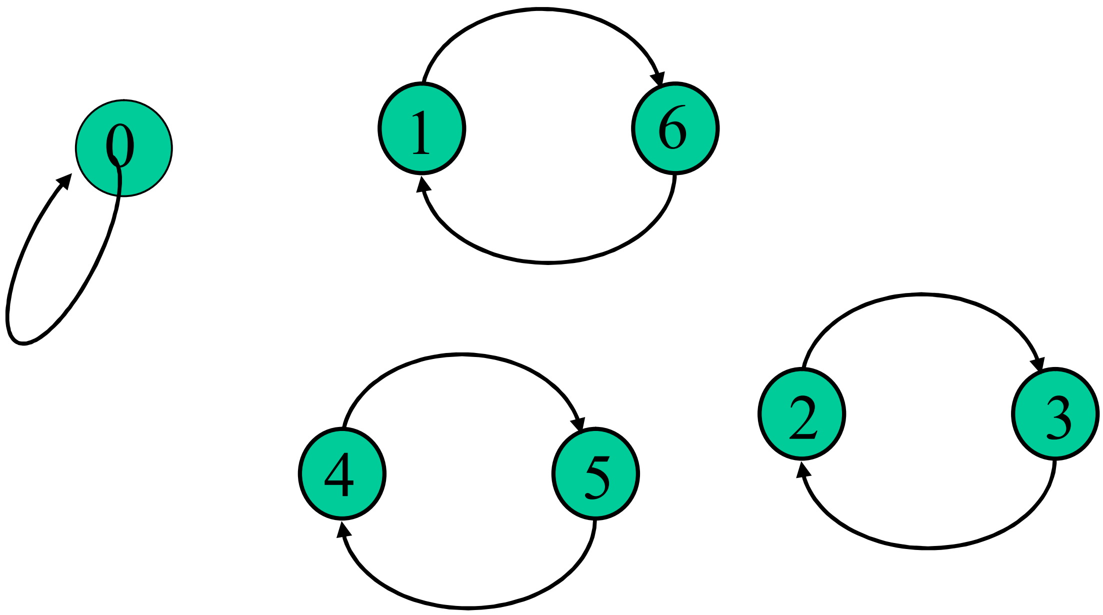
Passo 1

Riordina e rinumera i job secondo i bj crescenti. Calcola la funzione φ *.

job	0	1	2	3	4	5	6
b_j	1	6	15	21	24	29	33
a_j	3	46	28	18	41	35	14
$a_{\varphi *(j)}$	3	14	18	28	35	41	46
$\varphi *(j)$	0	6	3	2	5	4	1

Passo 2

Dai valori di φ *(j) si deduce che i nodi sono connessi l'uno all'altro nel seguente modo:



Passo 3

job	0	1	2	3	4	5	6
b_j	1	6	15	21	24	29	33
$a_{\varphi *(j)}$	3	14	18	28	35	41	46

Calcolo dei costi di interscambio $c_{\varphi *}$ I(j, j+1):

$c_{\varphi *}$ I(0, 1) = 2(6-3) = 6;

$c_{\varphi *}$ I(1, 2) = 2(15 - 14) = 2;

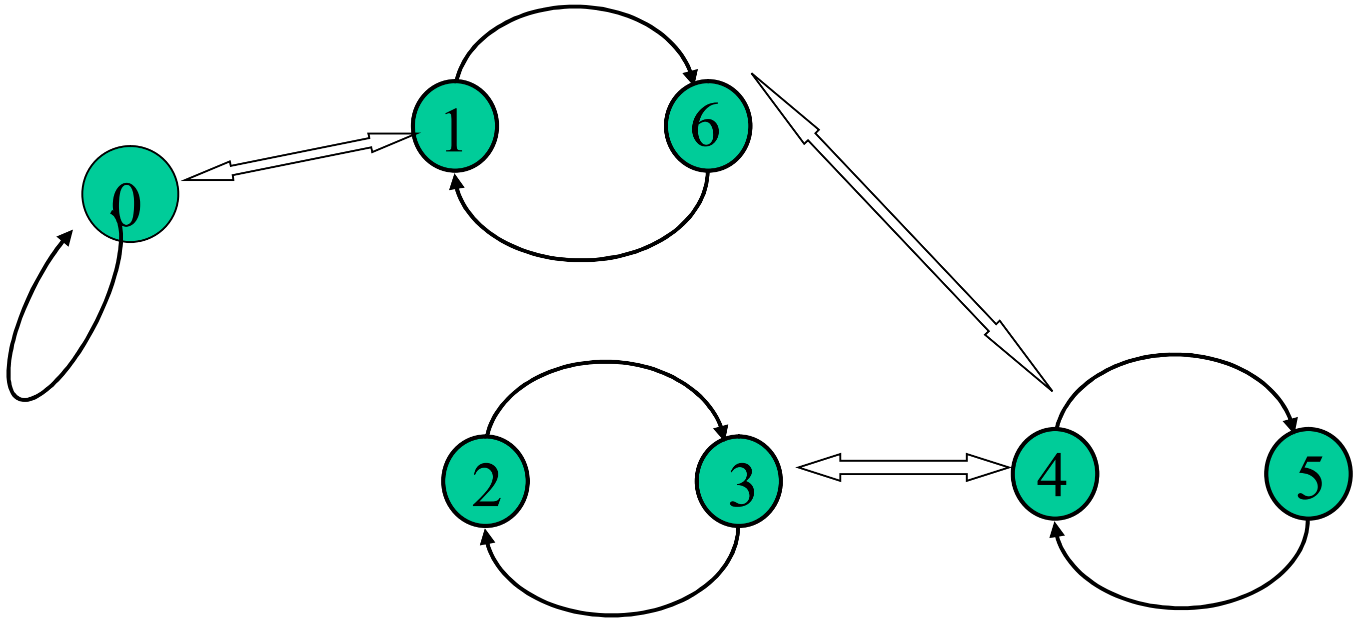
$c_{\varphi *}$ I(2, 3) = 2(21 - 18) = 6;

$c_{\varphi *}$ I(3, 4) = 0;

$c_{\varphi *}$ I(4, 5) = 0;

$c_{\varphi *}$ I(5, 6) = 0;

Passo 4



Passo 5

Vediamo a che gruppo appartengono gli scambi:

archi	bj	$a_{\varphi *(j)}$	gruppo
I(0,1)	1	3	A
I(3,4)	21	28	A
I(5,6)	29	41	A

J1=5, J2=3, J3=0.

Passo 7

Il ciclo ottimo è ottenuto dopo il seguente scambio:

$\varphi^{**}=\varphi^{*}$I(5,6) I(3,4) I(0,1) .

Si ha quindi che il ciclo ottimo e'

0 6 4 2 3 5 1 0

FABLE

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 20 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

A_ verifica che $\forall$ i $t_i \leq C$ (tempo di ciclo)

B_ calcola il LB $K_o = \lceil \sum_i t_i / C \rceil$

Regole di Taglio

TASK DOMINANCE

h domina i se:

$t_i < t_k$

$\forall j \in S(i) \Rightarrow j \in S(h)$

e h può essere scambiato con i nella sol. parziale corrente.

Se una macchina viene chiusa assegnandogli i e non h, e' possibile tagliare la sol. parziale che si stava esaminando

BOUND VIOLATION

K e' sempre il numero di macchine utilizzate nella migliore soluzione trovata

$$\mathbf{U_i = k - \lceil (t_i + \sum_{j \in S(i)} t_j)/C \rceil}$$

Se si chiude una macchina h ed esistono lavori con $UB_i = h$ non ancora assegnati, e' possibile tagliare la soluzione parziale corrente.

STATION DOMINANCE

Ogni volta che si incontra, nella costruzione di una soluzione, una sottosequenza già esaminata si può tagliare la sequenza.

EXCESSIVE IDLE TIME

$$\mathbf{T_{max} = (k-1)\, C - \sum_i t_i}$$

Se il tempo complessivo di inattività supera T_{max}, allora e' possibile tagliare la soluzione parziale corrente.

SOLUTION DOMINANCE

Se troviamo una sequenza completa che richiede K_0 macchine ci possiamo fermare:

la soluzione trovata e' quella ottima.

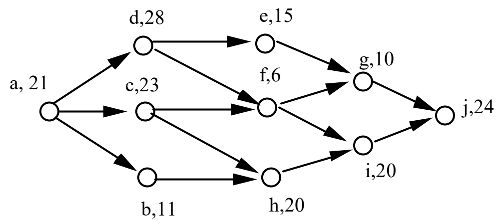
Algoritmo

0) Genera una soluzione iniziale: sia K il numero di macchine richieste dalla soluzione
1) Incrementa t_i inserendo, se possibile, i lavori non ancora considerati. Calcola gli upper bound delle stazioni per ogni lavoro e i lavori dominanti
2) Quando una stazione e' completa applica le regole di taglio e se i bounds sono violati torna indietro.
   - aggiorna gli upper bounds quando viene trovata una sol. migliore
   - fermati se il numero di macchine necessarie all'ottimo corrente sono proprio K_o

Problema 3.15

Un impianto produce ciclicamente un tipo di pezzo, caratterizzato dal grafo delle operazioni in figura. La durata delle operazioni è espressa in minuti. Il sistema è chiamato a garantire, nelle 24 ore ininterotte di lavorazione giornaliera, la produzione di 32 unità. Si tratta di assegnare le operazioni al numero minimo di stazioni di lavoro nel rispetto di questo vincolo sulla produttività.

Si applichi l'algoritmo esatto (FABLE) per il calcolo della soluzione ottima.



SOLUZIONE

Innanzitutto, occorre calcolare qual'e' la specifica sul tempo di ciclo. Dovendo produrre 32 unita' in 24 ore, evidentemente ciò vuol dire un'unita'ogni tre quarti d'ora, e quindi C = 45 minuti.

1. Determiniamo una soluzione iniziale applicando la semplice regola lessicografica.

a	b	c	d	e	f	g	h	i	j

21	11	23	28	15	6	10	20	20	24

In base al tempo di processamento e al vincolo sul tempo di ciclo delle macchine otteniamo:

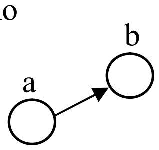
a b || c || d e || f g h || i   j

che richiede dunque 5 stazioni.

2. Procediamo, nell'enumerazione di tutte le sequenze, secondo l'algoritmo FABLE, utilizzando l'ordine lessicografico dato. In generale, per poter eliminare soluzioni parziali, occorre calcolare i valori U_i, che rappresentano l'upper bound sull'indice della stazione cui viene assegnata l'operazione i, affinché sia conveniente proseguire lungo quel ramo. Dai pesi posizionali, e considerando che conosciamo già una soluzione con 5 macchine, si ottiene, per gli U_i:

a	b	c	d	e	f	g	h	i	j
1	3	2	2	3	3	4	3	4	4

Dunque inizialmente abbiamo



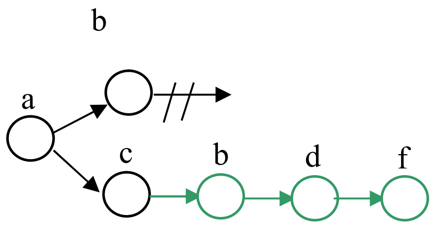
a b

a questo punto, inserire c porterebbe ad eccedere il tempo di ciclo.

Vediamo allora se esistono altre operazioni assegnabili, a parte c. L'unica assegnabile è d, che però è ancora più lunga di c.

Vediamo allora la dominanza: l'operazione c domina b (in quanto è più lunga e i successori di c includono i successori di b), e dunque possiamo sostituire b con c: abbandoniamo perciò il ramo "a b …" e proseguiamo da

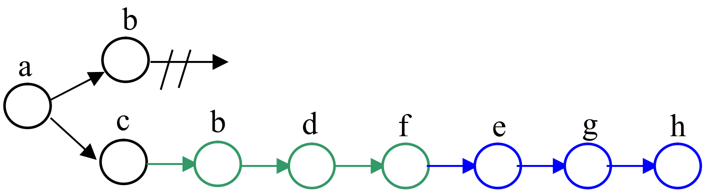
a c



osserviamo che c non è dominato da alcun nodo, e dunque chiudiamo la prima stazione. Apriamo la seconda stazione con b (seguendo l'ordine lessicografico), e quindi d. Sarebbe quindi il turno di e, che però non entra nella stazione. Osserviamo allora che c'è un'operazione assegnabile, ossia f. La situazione risulta allora, fino a questo punto:

a c || b d f

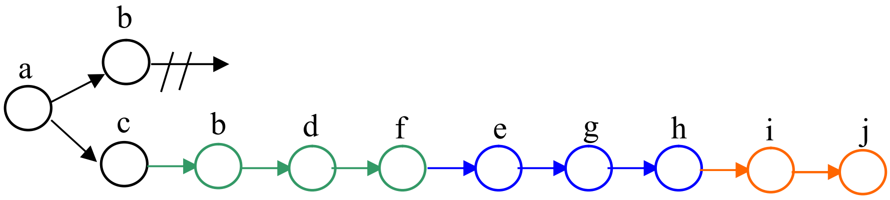
controlliamo il criterio di dominanza, e vediamo che non ci sono altre operazioni dominate. Né alcun upper bound U_i è violato, e dunque possiamo chiudere la seconda stazione. Ripartendo in ordine lessicografico, è il turno di e, seguito da g e da h:



a c || b d f || e g h

Tutti i valori U_i sono rispettati, e dunque possiamo chiudere a questo punto la terza stazione. Infine si ha:

a c || b d f || e g h || i j



poiché $K^0 = (178/45)^+ = 4$, quella trovata è anche la soluzione ottima (altrimenti, occorreva arretrare e iniziare il backtracking).

<u>**N.B.**</u> Per trovare una soluzione iniziale al problema, passo 0 dell'algoritmo, conviene applicare l'euristica RPW rispetto alla semplice regola lessicografica.

Per applicare l'euristica RPW occorre calcolare il peso posizionale di ogni operazione:

a	b	c	d	e	f	g	h	i	j

178	75	103	103	49	60	34	64	44	24

Dunque riordinando le operazioni in base al loro peso posizionale, si ottiene:

a c d b h f e i g j

Applicando l'euristica con C = 45, si ottiene la seguente soluzione:

a c || d b f || h e g || i j

che richiede 4 macchine, fornendo cosi' la soluzione ottima!

Johnson

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 21 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

Problema 3.16

Un'azienda che lavora su commessa deve produrre il prototipo di un nuovo acceleratore di particelle. Tale prodotto si compone di cinque componenti *A, B, C, D, E*, prodotti da altrettante ditte subcommittenti. Si tratta di componenti estremamente ingombranti e delicati, che richiedono, per il trasporto, particolare cura. Sicché, un TIR della nostra azienda verrà utilizzato per ritirare ciascuno dei cinque componenti dalle rispettive ditte che li hanno prodotti. A causa della delicatezza e dell'ingombro di cui sopra, viene dedicato un viaggio diverso al ritiro di ogni componente. Una volta portato ciascun componente in azienda, questo deve essere collaudato da un macchinario apposito. Quando tutti i componenti sono stati collaudati, è possibile finalmente effettuare la fase finale di assiematura. Si consideri che per andare a prendere *A, B, C, D, E* sono necessari 2, 3, 4, 5 e 7 giorni rispettivamente, che il loro collaudo richiede 6, 6, 5, 3 e 5 giorni rispettivamente, e che infine la fase finale di assiematura e la consegna richiedono, complessivamente, un altro giorno. L'azienda dispone di un solo TIR.

1) In quanti giorni l'azienda può riuscire a consegnare il prodotto finito?

2) Il fatto di avere 2 TIR anziché uno sarebbe vantaggioso?

SOLUZIONE

Si tratta di applicare l'algoritmo di Johnson, in quando il ritiro e il collaudo di ciascun componente sono assimilabili a due operazioni da effettuarsi in sequenza su ogni componente, e dunque a un flow shop costituito da due macchine. L'obiettivo è quello di minimizzare il tempo di completamento.

Com'è noto, la sequenza di Johnson si costruisce dagli estremi verso il centro. Il tempo più basso in assoluto è quello del ritiro di A (2 giorni), che dunque verrà posto all'inizio della sequenza:

A _ _ _ _

Quindi, abbiamo il ritiro di B e il collaudo di D che durano ambedue 3 giorni. Possiamo quindi porre B subito dopo A e D in fondo alla sequenza:

A B _ _ D

Il tempo più basso è ora il ritiro di C (4 giorni), che viene dunque posto in coda a B. Si ottiene perciò:

A B C E D

Il tempo di completamento, come si arguisce tracciando il diagramma di Gantt, è pari a 27.

Per rispondere al secondo punto, osserviamo che tale valore è pari alla somma dei tempi di collaudo più il più piccolo dei tempi di ritiro. La disponibilità di un secondo TIR non servirebbe evidentemente a nulla, in termini di tempo di completamento del prodotto.

Innanzitutto, occorre calcolare qual è la specifica sul tempo di ciclo. Dovendo produrre 32 unità in 24 ore, evidentemente ciò vuol dire una unità ogni tre quarti d'ora, e quindi $C = 45$ minuti.

1. Per applicare l'euristica RPW occorre calcolare il peso posizionale di ogni operazione. Ciò porta ad avere:

a	b	c	d	e	f	g	h	i	j
178	75	103	103	49	60	34	64	44	24

e dunque, riordinando le operazioni in base al loro peso posizionale, si ottiene:

a c d b h f e i g j

applicando allora l'euristica con $C = 45$, si ottiene la seguente soluzione:

a c || d b || h f e || i g || j

che richiede dunque 5 stazioni.

Problema 3.17

La vostra officina deve produrre 8 lotti di pezzi. Alcuni lotti richiedono una macchina di stampaggio (1), altri di piegatura (2), altri entrambe (prima la 1 e poi la 2). Precisamente, i lotti **A** e **B** richiedono la macchina 1 per un tempo pari a 1 e 4 giorni rispettivamente. Il lotto **C** richiede la macchina 2 per 3 giorni. Gli altri lotti richiedono di essere lavorati su entrambe le macchine, secondo i tempi riportati in tabella:

i	D	E	F	G	H
p1	2	8	5	5	7
p2	6	3	4	9	6

Qual è il tempo minimo di completamento di tutti i lotti?

SOLUZIONE

Si tratta di applicare l'algoritmo di Johnson (Problema 3.10), osservando che i primi job a essere sequenziati saranno A, B (in fondo) e C (in testa) in quanto hanno tempo di lavorazione 0 su una delle due macchine. La soluzione ottima sarà:

C D G H F E A B

cui corrisponde un tempo minimo di completamento pari a 32.

Shifting bottleneck

(La parte teorica necessaria a svolgere questi esercizi fa riferimento alle lezioni n° 24 e n° 25 del corso di Automazione Industriale tenuto per il consorzio Nettuno)

Problema 3.18

Un sistema di produzione è costituito da 4 macchine che devono eseguire 3 job.

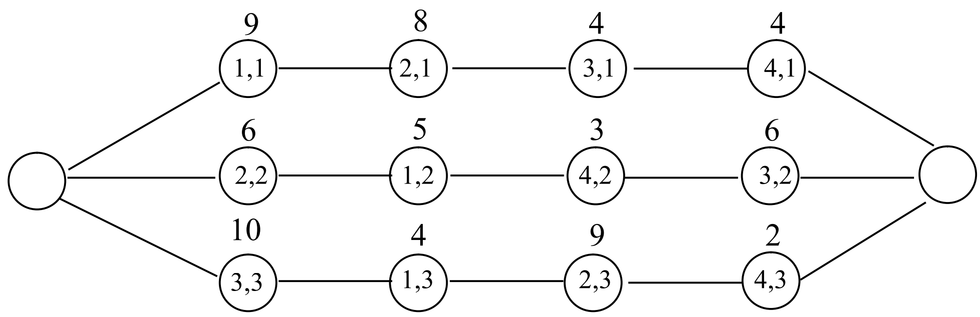
Gli instradamenti ed i tempi di processamento sono dati nella seguente tabella:

job	instradamento	tempi di proc.
1	1, 2, 3,4	p_{11}=9 p_{21}=8 p_{31}=4 p_{41}=4
2	2, 1, 4, 3	p_{22}=8 p_{12}=5 p_{42}=3 p_{32}=6
3	3, 1, 2, 4	P_{33}=10 p_{13}=4 p_{23}=9 p_{43} =2

Si vuole minimizzare C_{max}.

Soluzione

Il problema si può risolvere con l'algoritmo euristico Shifting Bottleneck.



Iterazione 1

Inizialmente, l'insieme M_0 è vuoto ed il grafo G contiene solamente archi congiuntivi. Il percorso critico ed il makespan $C_{max}(0)$ è 25 ed è raggiunto dai job 1 e 3. Per individuare quale macchina schedulare per prima, si associa ad ognuna di esse un problema $1/r_j/L_{max}$ con release date e due date determinate dal cammino più lungo in G (assumendo un makespan di 25).

Per la macchina 1 il problema $1/r_j/L_{max}$ ha i seguenti dati:

job	1	2	3
p_j	9	5	4
r_j	0	6	10
d_j	9	16	14

e la sequenza ottima è 1, 3, 2 con un $L_{max}(1)=3$.

Per la macchina 2 il problema $1/r_j/L_{max}$ ha i seguenti dati:

job	1	2	3
p_j	8	6	9
r_j	9	0	14
d_j	17	11	23

e la sequenza ottima è 2, 1, 3 con un $L_{max}(2)=3$.

Per la macchina 3 si ha:

job	1	2	3
p_j	4	6	10
r_j	17	14	0
d_j	21	25	10

e la sequenza ottima è 3,1,2 con un $L_{max}(3)=2$.

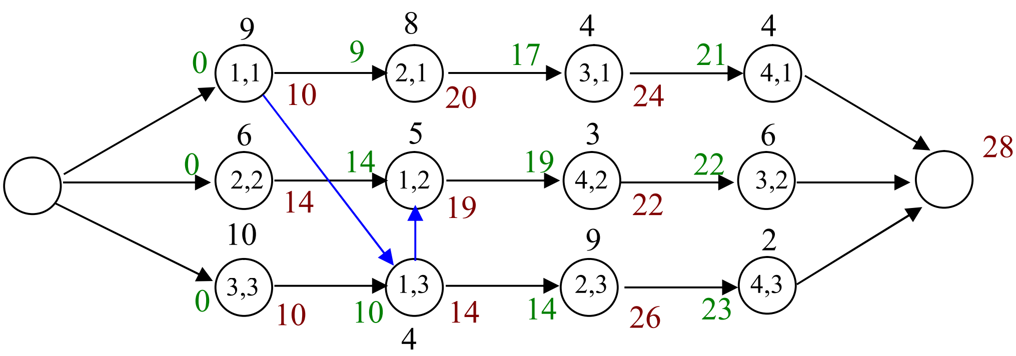
Per la macchina 4 si ha:

job	1	2	3
p_j	4	3	2

r_j	21	11	23
d_j	25	19	25

e la sequenza ottima è 2, 1, 3 con un $L_{max}(4)=2$.

Ne consegue che le macchine 1,2 possono essere considerati dei colli di bottiglia del sistema.
Scegliamo, ad esempio, la macchina 1 ed includiamola ad M_0.
Il grafo G" è ottenuto aggiungendo gli archi disgiuntivi corrispondenti alla sequenza sulla macchina 1.
Si ha che $C_{max}(\{1\})=C_{max}(0)+L_{max}(1)=25+3=28$ (vedi figura).



Iterazione 2

Poiché il makespan corrispondente a G" è 28, si può determinare il percorso critico. Le tre macchine rimanenti si possono analizzare in modo separato come problemi del tipo $1/r_j/L_{max}$ con release date e due date determinate dal cammino più lungo in G" (assumendo un makespan di 28).
Per la macchina 2 il problema $1/r_j/L_{max}$ ha i seguenti dati:

job	1	2	3
p_j	8	6	9
r_j	9	0	14
d_j	20	14	26

La sequenza ottima è 2, 1, 3 ed $L_{max}(2)=0$.

Il problema relativo alla macchina 3 ha i seguenti dati:

job	1	2	3
p_j	4	6	10
r_j	17	22	0
d_j	24	28	10

La sequenza ottima è 3, 2, 1 ed $L_{max}(3)=0$.

Il problema relativo alla macchina 4 ha i seguenti dati:

job	1	2	3
p_j	4	3	2
r_j	21	19	23
d_j	28	22	28

La sequenza ottima è 2, 1, 3 ed $L_{max}(4)=0$.

I problemi relativi alle macchine 2, 3 e 4 sono semplici poichè hanno $L_{max}= 0$.

Il sequenziamento finale risulta quindi il seguente:

macchina	Sequenza di job
1	1,3,2
2	2,1,3
3	3,2,1
4	2,1,3

Il makespan è 28.