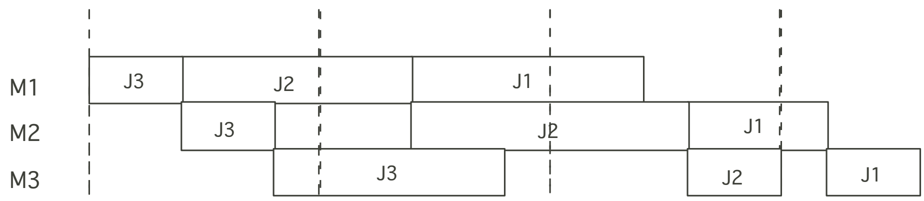
1) Si risolva la seguente istanza di Flow-Shop (J_i sono i lavori, M_i le macchine e i numeri i tempi di esecuzione). (suggerimento: per trovare l'ottimo si decrementino tutti i tempi di una quantità costante)

	M_1	M_2	M_3
J_1	5	3	2
J_2	5	6	2
J_3	2	2	5

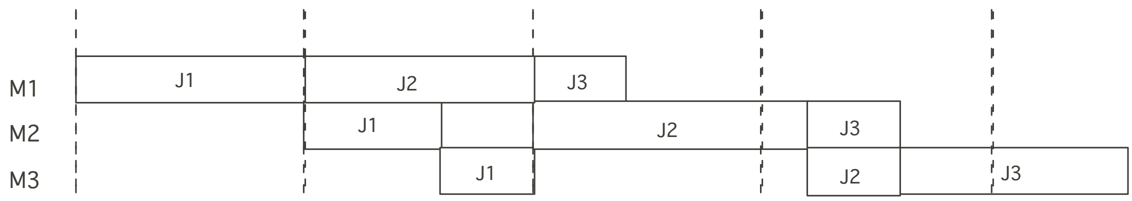
Soluzione: Dalla teoria si sa che esiste una schedulazione ottima data da una permutazione comune alle tre macchine. Quindi per ogni permutazione la lunghezza di ogni cammino contiene $(n+2)$ archi (n numero di lavori) e, decrementando di una stessa quantità ogni operazione, i cammini vengono tutti decrementati in modo uniforme, e quindi la permutazione ottima è la stessa. Se ora nell'istanza si decrementano di 2 tutte le durate abbiamo la seguente tabellina dei tempi

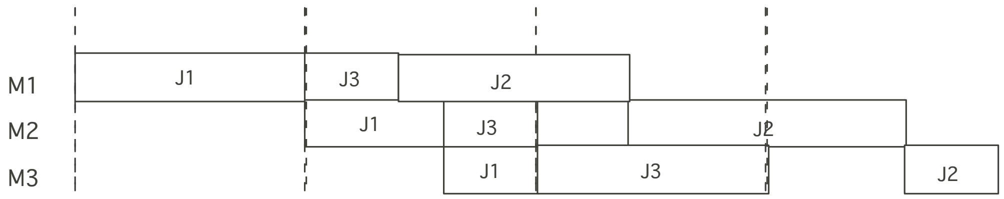
	M_1	M_2	M_3
J_1	3	2	0
J_2	3	4	0
J_3	0	0	3

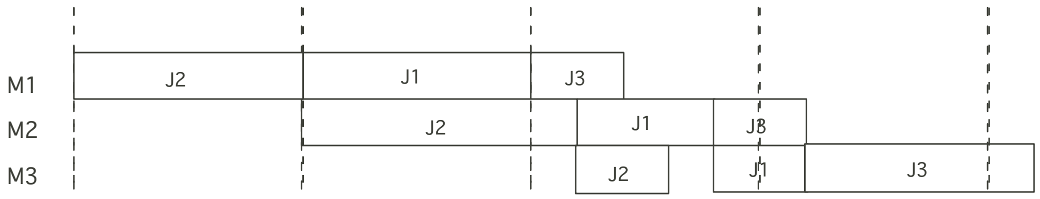
Come si vede la macchina M_3 e il lavoro J_3 sono indipendenti dalle altre macchine e dagli altri lavori e quindi J_1 e J_2 possono essere schedulati indipendentemente da J_3. Applicando la regola di Johnson a J_1 e J_2 si ha la permutazione $J_2 \to J_1$. Ovviamente J_3 deve essere schedulato prima degli altri due lavori e quindi l'ottimo è dato da $J_3 \to J_2 \to J_1$.

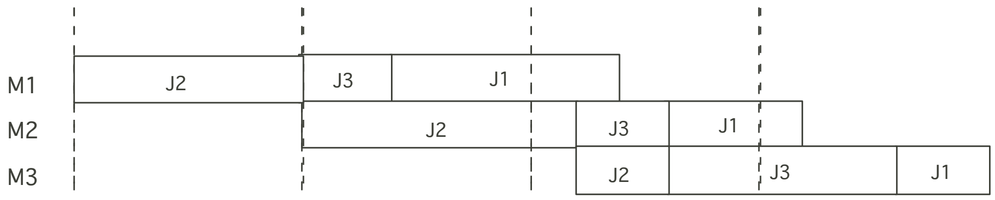


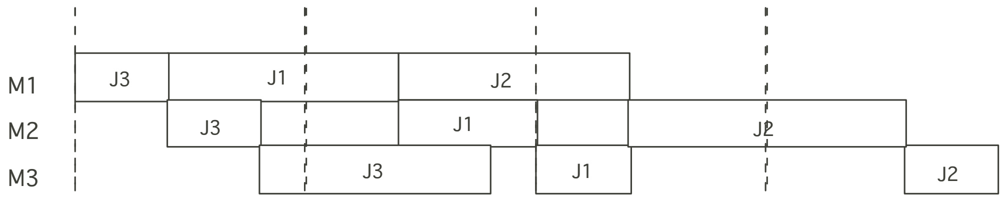
Le altre soluzioni sono:




M1
M2
M3
J1
J3
J2
J1
J3
J2
J1
J3
J2


M1
M2
M3
J2
J1
J3
J2
J1
J3
J2
J1
J3


M1
M2
M3
J2
J3
J1
J2
J3
J1
J2
J3
J1


M1
M2
M3
J3
J1
J2
J3
J1
J2
J3
J1
J2

2) Si risolva la precedente istanza di Flow-Shop con il vincolo aggiuntivo che non ci devono essere tempi morti fra le operazioni dello stesso lavoro. Si dimostri che, in generale (qualsiasi numero di macchine e di lavori) solo schedulazioni a permutazione possono essere ammissibili. C'è qualche collegamento con il problema del Commesso Viaggiatore?

Soluzione: Data una schedulazione non a permutazione esistono due macchine, siano M_1 e M_2, e due lavori, siano J_1 e J_2, per cui su M_1 viene eseguito prima il lavoro J_1 e poi J_2 e viceversa su M_2. Sia S^j_m l'inizio del lavoro J_j sulla macchina M_m e C^j_m il suo completamento. Quindi $C^1_1 \leq S^2_1$ e $C^2_2 \leq S^1_2$. Si supponga che i lavori su M_1 devono essere eseguiti prima di quelli su M_2 (si tratta di un flow-shop). Quindi $C^1_1 \leq S^1_2$ e $C^2_1 \leq S^2_2$. Inoltre $S^j_m < C^j_m$ (supponendo durate strettamente positive). Quindi $C^1_1 \leq S^2_1 < C^2_1 \leq S^2_2 < C^2_2 \leq S^1_2$. Però l'ipotesi che non ci siano tempi morti fra le operazioni di uno stesso lavoro impone $C^1_1 = S^1_2$ che è in contraddizione con le precedenti diseguaglianze.

Stabilito quindi che solo schedulazioni a permutazione sono ammissibili si considerino due lavori i e j e si supponga che i preceda immediatamente j. Per ogni macchina deve valere $C^i_m \leq S^j_m$. Per l'ipotesi fatta si ha (supponendo le macchine siano numerate secondo l'ordine naturale del flow shop)

$$C^i_m = S^i_1 + \sum_{h \leq m} p^i_h, \qquad S^j_m = S^j_1 + \sum_{h < m} p^j_h$$

e allora

$$C^i_m \leq S^j_m \Longrightarrow S^j_1 - S^i_1 \geq \sum_{h \leq m} p^i_h - \sum_{h < m} p^j_h$$

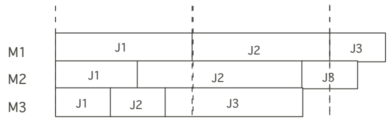
per cui, avendo l'obiettivo di rendere il più piccolo possibile $S^j_1 - S^i_1$, si ha

$$S^j_1 - S^i_1 = \max_m \sum_{h \leq m} p^i_h - \sum_{h < m} p^j_h =: c_{ij}$$

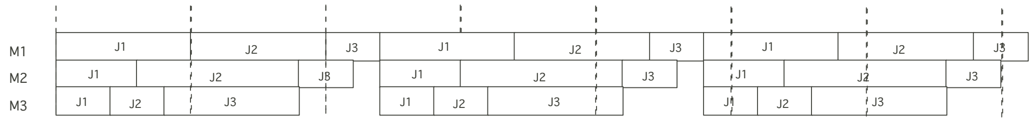
Il problema è pertanto modellabile come un TSP in cui il "costo" di schedulare j immediatamente dopo i corrisponde al ritardo c_{ij}. Data una permutazione il tempo di completamento è dato dalla somma dei termini c_{ij} più la somma delle durate su tutte le macchine dell'ultimo lavoro schedulato. Per modellare quindi il problema come un TSP asimmetrico bisogna aggiungere un nodo (indichiamolo come 0) e assegnare i costi $c_{0j} := 0$, e $c_{j0} := \sum_m p^j_m$.

3) Si supponga che nel Problema 1), invece di avere i lavori eseguiti solo una volta, ci sia un processo continuo. Più esattamente un lavoro potrebbe consistere in una lavorazione in tre fasi di un pezzo (a lavori diversi corrispondono pezzi diversi). Mentre un pezzo viene lavorato, un altro pezzo uguale al primo è in attesa che la macchina assegnata diventi disponibile, poi inizia la lavorazione e così di seguito per tutti e tre i lavori. L'interesse è nel raggiungere la più alta produttività (pezzi lavorati/unità di tempo). Si vuole anche produrre lo stesso numero di pezzi per i tre lavori. La macchina M_1 è il collo di bottiglia del sistema in quanto deve lavorare per il tempo più lungo (12 unità di tempo). Ovviamente non ci può essere nessun valore di produttività superiore a un pezzo (per ogni lavoro) ogni 12 unità di tempo. Esiste comunque una schedulazione in grado di ottenere questa produttività (asintoticamente)?

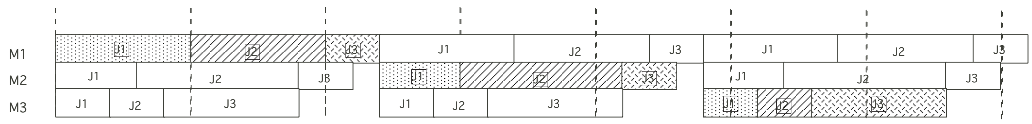
Soluzione: Si dispongano i lavori in modo arbitrario su ogni macchina, ad esempio così:



Ora si ripeta questa disposizione ogni 12 unità di tempo, nel seguente modo



Siccome uno stesso pezzo non può essere lavorato contemporaneamente su più di una macchina, le operazioni corrispondenti ad uno stesso pezzo, vengono identificate come nella seguente figura



e la schedulazione può essere ripetuta indefinitamente. Come si vede, ogni 12 unità di tempo viene prodotto un pezzo per ogni lavoro e quindi la produttività (asintotica, perché per i primi due periodi non c'è ancora produzione) è effettivamente di un pezzo ogni 12 unità di tempo.

4) Quattro studenti Anna, Bruno, Chiara e Daniele hano bisogno della strumentazione del laboratorio per portare a termine il loro homework. Anna ha bisogno dello scanner per 10 minuti, poi deve elaborare al PC l'output dello scanner per 30 minuti e infine ha bisogno di altri 10 minuti per la stampa. Bruno non ha bisogno dello scanner. Può cominciare imediatamente sul PC dove deve lavorare per 40 minuti e poi stampare i risultati per altri 15 minuti. Pure Chiara non ha bisogno dello scanner e i suoi tempi al PC e alla stampante sono di 30 e 10 minuti rispettivamente. Infine Daniele ha bisogno dello scanner per 5 minuti, poi del PC per 20 minuti e infine della stampante per 10 minuti. Il PC è un vecchio modello collegato in rete locale sia alla stampante che allo scanner. C'e' un nuovo PC più veloce, esattamante il doppio del vecchio. Tuttavia è collegato solo allo scanner e per stampare i risultati si devono copiare i files su dischetto, inserire questo nel vecchio PC e stampare i risultati da lì. Il lavoro aggiuntivo di inserire il dischetto e leggere/scrivere i files su dischetto richiede sempre 5 minuti.

Anna, Bruno, Chiara e Daniele sono tipici studenti dell'ultimo minuto e i loro compiti devono essere consegnati entro 75 minuti. Però sono anche in gamba e invece di litigare per avere la strumentazione, riescono a trovare immediatamente una schedulazione che permette loro di consegnare in tempo i lavori. Siete in grado di trovare anche voi la schedulazione (anche se non proprio immediatamente)? Possono forse avere anche del tempo extra per andarsi a prendere un caffe tutti assieme (prima della consegna)?

(nota: ci sono molte alternative, troppe per esser provate tutte; ragionando sulle proprietà del problema, il numero di alternative può esser ridotto a pochi casi che possono essere verificati con carta e penna.)

Soluzione: È conveniente far usare il PC veloce a chi ha un lavoro lungo, e quindi a Bruno. Fra Anna e Chiara si può pensare di assegnare il PC veloce ad Anna, visto che deve lavorare anche allo scanner. Quindi la tabellina dei tempi risulta essere:

	S	C_1	C_2	P
A	10	20		10
B		25		15
C			30	10
D	5		20	10

Si vede che fra Anna e Bruno è conveniente anticipare Bruno (ragionando come sulle schedulazioni a permutazione di un flow shop), mentre Chiara può precedere Daniele

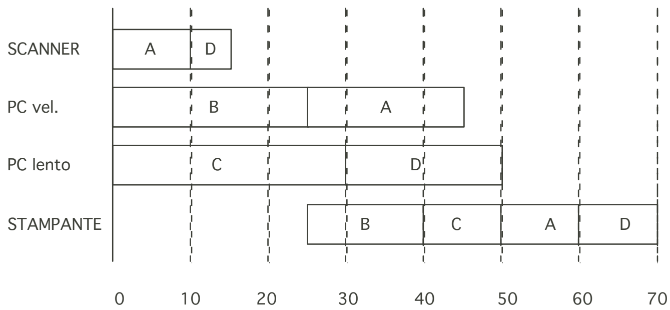
$$
\begin{array}{ccccccc}
 & S & & C_1 & C_2 & & P \\
B & & & 25 & \rightarrow & & 15 \\
 & & & \downarrow & & & \downarrow \\
A & 10 & \rightarrow & 20 & \rightarrow & & 10 \\
 & & & & & & \downarrow \\
C & \downarrow & & & 30 & \rightarrow & 10 \\
 & & & & \downarrow & & \downarrow \\
D & 5 & & \rightarrow & 20 & \rightarrow & 10
\end{array}
$$

Il cammino critico vale 75 e corrisponde alle seguenti operazioni $B-C1$, $A-C1$, $A-P$, $C-P$, $D-P$. Per fare ancora meglio si può pensare di invertire Anna e Chiara sulla stampante, ovvero

$$
\begin{array}{ccccccc}
 & S & & C_1 & C_2 & & P \\
B & & & 25 & \rightarrow & & 15 \\
 & & & & & & \downarrow \\
C & & & \downarrow & 30 & \rightarrow & 10 \\
 & & & & \downarrow & & \downarrow \\
A & 10 & \rightarrow & 20 & \rightarrow & & 10 \\
 & \downarrow & & & \downarrow & & \downarrow \\
D & 5 & & \rightarrow & 20 & \rightarrow & 10
\end{array}
$$

con un cammino critico di 70! ($B - C1$, $B - P$, $C - P$, $A - P$, $D - P$).

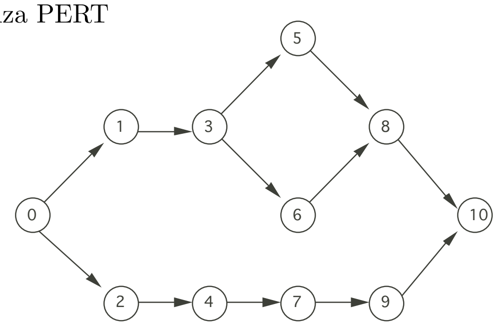
Il diagramma di Gantt illustra meglio la schedulazione. Come si vede la soluzione non è unica in quanto Anna e Daniele si possono scambiare sia allo scanner che alla stampante.



Si può fare meglio di così? Siccome la stampante deve essere usata per almeno 45 minuti, il primo che la usa dovrebbe usare il PC ed eventualmente lo scanner per meno di 25 minuti. L'unica possibilità è che o Chiara o Daniele siano i primi ad usare il PC veloce. Supponiamo sia Chiara la prima ad usare il PC veloce e successivamente la stampante. La stampante risulta disponibile al 30-mo minuto e deve essere utilizzata subito. Può utilizzarla solo chi riesca a completare il lavoro al PC lento entro 30 minuti e cioè Daniele. La stampante risulterà libera la 40-mo minuto. Bruno non è in grado di usarla perché il PC veloce è rimasto libero per 20 minuti e quello lento per soli 15 minuti, tempi insufficienti per Bruno. Può utilizzarla Anna lavorando al PC veloce subito dopo Chiara. Però a questo punto Bruno non è in grado di usare la stampante al 50-mo minuto perché nessuno dei due PC è disponibile per il tempo necessario a Bruno. Quindi far iniziare Chiara non porta ad un miglioramento.

Supponiamo inizi Daniele. Anche in questo caso la stampante è di nuovo libera al 30-mo minuto. Può usarla solo Chiara che ha precedentemente lavorato sul PC lento per 30 minuti. La stampante ritorna disponibile al 40-mo minuto. Nel frattempo il PC veloce è disponibile per 20 minuti e quello lento per 10 minuti. Solo Anna può servirsene dopo aver usato lo scanner. Tuttavia al 50-mo minuto la stampante non può iniziare a lavorare per Bruno perché questi non ha avuto tempo sufficiente su nessuno dei due PC. Quindi il risultato di 70 minuti è ottimo.

5) Si consideri la seguente istanza PERT



con durate delle operazioni $p_0 = 0$, $p_1 = 5$, $p_2 = 4$, $p_3 = 3$, $p_4 = 5$, $p_5 = 8$, $p_6 = 2$, $p_7 = 3$, $p_8 = 7$, $p_9 = 5$, $p_{10} = 5$.

— Si trovi il minimo tempo di completamento M_1 rispetto ai dati.

— Poi si prenda in considerazione il fatto che per le attività 5, 6 e 7 è richiesta la medesima risorsa e quindi non possono essere eseguite contemporaneamente. Si trovi allora il minimo tempo di completamento M_2 rispetto a questo nuovo vincolo.

— Si supponga di voler ridurre M_2 al valore M_1 e di poterlo fare in due modi alternativi: o riducendo le durate delle attività 5, 6 o 7 (altre attività non possono essere ridotte) al costo di 100 per attività per unità di tempo, o acquistando una risorsa aggiuntiva dello stesso tipo al costo di 250 in modo da poter distribuire le tre attività sulle due risorse. Quale alternativa è più conveniente?

Soluzione

Sono solo tre i cammini nel grafo ed è immediato vedere che il cammno più lungo è dato da $0 \to 1 \to 3 \to 5 \to 8 \to 10$ con lunghezza $M_1 = 28$.

Se le attività 5, 6, e 7 devono essere eseguite dalla medesima risorsa, si tratta di risolvere un problema ad una macchina con date di rilascio e code. In particolare è facile calcolare i seguenti valori $r_5 = 8$, $r_6 = 8$, $r_7 = 9$, $q_5 = 12$, $q_6 = 12$, $q_7 = 10$. Usando la regola di Jackson si schedula dapprima 5 o 6 (indifferentemente), poi l'altra delle due e infine 7. Il tempo di completamento di questa schedulazione vale $M_2 = 31$ ed è ottimo dato che

$$h(\{5,6,7\}) = \min_{j\in\{5,6,7\}} r_j + \sum_{j\in\{5,6,7\}} p_j + \min_{j\in\{5,6,7\}} q_j = 8 + 8 + 2 + 3 + 10 = 31$$

Dato che tutte e tre le attività sono sul cammino critico è necessario operare una riduzione di almeno 3 unità temporali. Ad esempio ridurre di uno ciascuna delle tre porta il tempo di completamento a $M_2 = 28$. Questo comporta un costo di 300.

Aggiungendo una risorsa, questa può essere assegnata all'attività 5. L'altra risorsa deve allora ripartirsi solo sulle attività 6 e 7 che vengono eseguite nell'ordine. È immediato vedere che il tempo di completamento è ancora 28. Questa soluzione costa 250 ed è perciò preferibile.

6) Sia dato un problema di Flowshop con due macchine con durate per le operazioni p_{1j} per l'operazione del lavoro j da eseguirsi sulla macchina 1 e p_{2j} per l'operazione del lavoro j da eseguirsi sulla macchina 2.

Si dimostri che per un problema di Flowshop con 2 macchine l'ottimo è dato dalla stessa permutazione dei lavori su entrambe le macchine (suggerimento: si usi un argomento di scambio).

Si dimostri inoltre che tale permutazione si ottiene ripartendo l'insieme J dei lavori nei due insiemi $J_1 := \{j : p_{1j} \leq p_{2j}\}$ e $J_2 := J \setminus J_1$. Dopodiché la permutazione ottima si ottiene prendendo prima i lavori in J_1 in ordine crescente di p_{1j} e poi i lavori in J_2 in ordine decrescente di p_{2j} (suggerimento: cosa succede se tutti i valori delle durate sono decrementati della stessa quantità fino a rendere nulla qualche durata?).

Si risolva la seguente istanza di un problema Flow Shop (J_i sono i lavori, M_i le macchine e i numeri le durate)

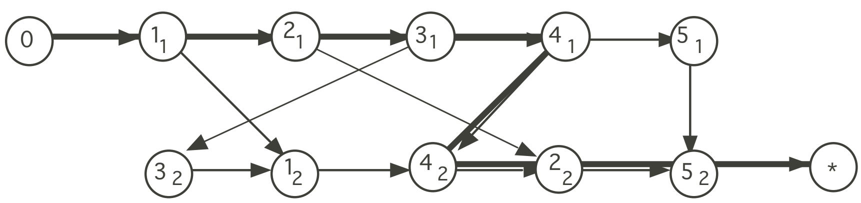
	M_1	M_2	M_3
J_1	5	4	2
J_2	3	6	2
J_3	2	1	5

tenendo presente un risultato teorico che afferma come per ogni problema Flow Shop c'è una soluzione ottima in cui la stessa permutazione dei lavori è presente sulla prima e sulla seconda macchina ed un'altra permutazione dei lavori ̀presente sull'ultima e sulla penultima macchina.

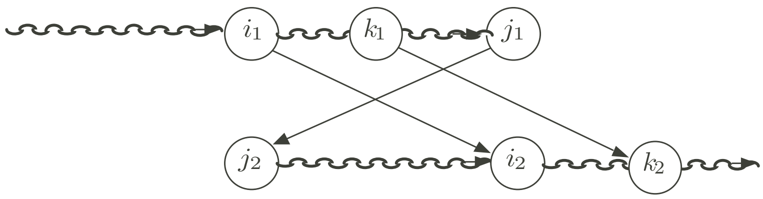
Si risolva la medesima istanza con l'ulteriore vincolo che non ci siano tempi morti fra le operazioni di un medesimo lavoro. C'è un qualche collegamento con il problema del Commesso Viaggiatore (TSP)?

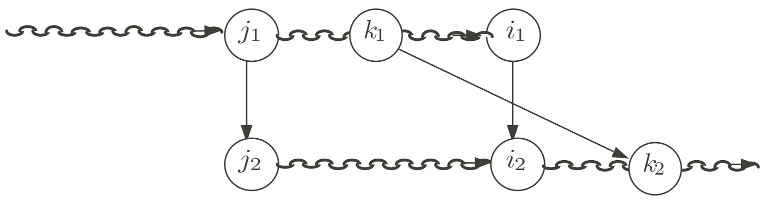
Soluzione

Si supponga che per un problema di Flow Shop con due macchine i lavori siano eseguiti con due permutazioni arbitrarie. Si noti che ci sono solo n cammini (tanti quanti i lavori) e ogni cammino è del tipo $P_i := 0 \rightsquigarrow i_1 \to i_2 \rightsquigarrow *$ (si veda la figura con evidenziato il cammino P_4).

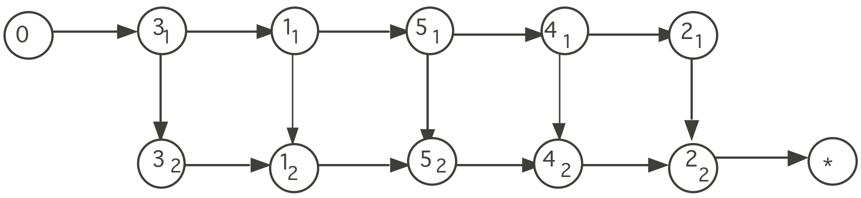


Se le permutazioni sono divere esistono due lavori i e j per cui $i \prec j$ sulla macchina 1 e $j \prec i$ sulla macchina 2. Supponiamo inoltre che per ogni $i \prec k \prec j$ sulla macchina 1 si abbia $i \prec k$ sulla macchina 2 (senza perdita di generalità, altrimenti si prenda tale k al posto di j). Quindi il cammino P_j è del tipo $0 \rightsquigarrow i_1 \rightsquigarrow j_1 \to j_2 \rightsquigarrow i_2 \rightsquigarrow *$ (si veda la figura).





Scambiando fra loro le operazioni i_1 e j_1, si alterano le lunghezze dei cammini P_k con $i_1 \prec k_1 \prec j_1$ ed è immediato vedere che le lunghezze di tali cammini P_k variano di $(p_{1j} - p_{1i})$. In ogni caso le nuove lunghezze, anche se aumentate, sono minori della lunghezza di P_j (prima dello scambio). Quindi la nuova soluzione è non peggiore della precedente e quindi esiste una soluzione ottima con la medesima permutazione su entrambe le macchine.



Essendo la medesima permutazione presente su entrambe le macchine, ogni cammino P_i ha lo stesso numero $n+1$ di archi e quindi se si decrementano le durate di una medesima quantità Δ tutti i cammini sono decrementati della stessa quantità $(n+1)\,\Delta$. Quindi la permutazione ottima è la medesima sia per i dati originali p che per quelli modificati $p - \Delta$. Si prenda allora Δ in modo che la durata più corta si annulli. Se ciò avviene per un'operazione della prima macchina la soluzione ottima può essere modificata (se tale operazione non è schedulata per prima) anticipando tale operazione in prima posizione e, a causa della sua durata nulla, senza alterare la schedulazione delle altre operazioni e senza violare il vincolo di precedenza fra la prima e la seconda macchina. Quindi la nuova schedulazione in cui tale operazione a durata nulla si trova in prima posizione è ottima anch'essa. Se invece l'operazione a durata nulla appartiene alla seconda macchina, con un simile ragionamento si trova che nella soluzione ottima verrà schedulata per ultima. Ripetendo ricorsivamente tale ragionamento si perviene all'algoritmo indicato.

In base al risultato teorico la schedulazione ottima prevede la stessa permutazione per le tre macchine. Quindi il problema può (in questo caso) essere risolto in modo esaustivo analizzando le sei permutazioni possibili (C è il tempo di completamento totale).

$$\begin{array}{cccccc} & M_1 & & M_2 & & M_3 \\ J_1 & 5 & \rightarrow & 4 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_2 & 3 & \rightarrow & 6 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_3 & 2 & \rightarrow & 1 & \rightarrow & 5 \end{array} \Longrightarrow C = 22; \quad \begin{array}{cccccc} & M_1 & & M_2 & & M_3 \\ J_2 & 3 & \rightarrow & 6 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_1 & 5 & \rightarrow & 4 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_3 & 2 & \rightarrow & 1 & \rightarrow & 5 \end{array} \Longrightarrow C = 20;$$

$$\begin{array}{cccccc} & M_1 & & M_2 & & M_3 \\ J_2 & 3 & \rightarrow & 6 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_3 & 2 & \rightarrow & 1 & \rightarrow & 5 \\ & \downarrow & & \downarrow & & \downarrow \\ J_1 & 5 & \rightarrow & 4 & \rightarrow & 2 \end{array} \Longrightarrow C = 18; \quad \begin{array}{cccccc} & M_1 & & M_2 & & M_3 \\ J_3 & 2 & \rightarrow & 1 & \rightarrow & 5 \\ & \downarrow & & \downarrow & & \downarrow \\ J_2 & 3 & \rightarrow & 6 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_1 & 5 & \rightarrow & 4 & \rightarrow & 2 \end{array} \Longrightarrow C = 17;$$

$$\begin{array}{cccccc} & M_1 & & M_2 & & M_3 \\ J_1 & 5 & \rightarrow & 4 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_3 & 2 & \rightarrow & 1 & \rightarrow & 5 \\ & \downarrow & & \downarrow & & \downarrow \\ J_2 & 3 & \rightarrow & 6 & \rightarrow & 2 \end{array} \Longrightarrow C = 18; \quad \begin{array}{cccccc} & M_1 & & M_2 & & M_3 \\ J_3 & 2 & \rightarrow & 1 & \rightarrow & 5 \\ & \downarrow & & \downarrow & & \downarrow \\ J_1 & 5 & \rightarrow & 4 & \rightarrow & 2 \\ & \downarrow & & \downarrow & & \downarrow \\ J_2 & 3 & \rightarrow & 6 & \rightarrow & 2 \end{array} \Longrightarrow C = 19;$$

Pertanto la soluzione ottima schedula i lavori secondo la sequenza $3 \rightarrow 2 \rightarrow 1$.

Nel caso di impossibilità di avere tempi morti fra le operazioni di una stesso lavoro bisogna prima dimostrare che in ogni caso (qualsiasi sia il numero di macchine e/o di lavori) solo una schedulazione con la medesima permutazione su tutte le macchine è ammissibile.

Data una schedulazione con permutazioni diverse esistono due macchine M_1 e M_2 e due lavori J_1 e J_2, per cui su M_1 viene eseguito prima il lavoro J_1 e poi J_2 e viceversa su M_2. Sia S_m^j l'inizio del lavoro J_j sulla macchina M_m e C_m^j il suo completamento. Quindi $C_1^1 \leq S_1^2$ e $C_2^2 \leq S_2^1$. Inoltre i lavori su M_1 devono essere eseguiti prima di quelli su M_2 (si tratta di un flow-shop). Quindi $C_1^1 \leq S_2^1$ e $C_1^2 \leq S_2^2$. Inoltre $S_m^j < C_m^j$ (supponendo durate strettamente positive). Quindi $C_1^1 \leq S_1^2 < C_1^2 \leq S_2^2 < C_2^2 \leq S_2^1$. Però l'ipotesi che non ci siano tempi morti fra le operazioni di uno stesso lavoro impone $C_1^1 = S_2^1$ che è in contraddizione con le precedenti diseguaglianze.

Stabilito quindi che solo schedulazioni a permutazione sono ammissibili si considerino due lavori i e j e si supponga che i preceda immediatamente j. Per ogni macchina deve valere $C_m^i \leq S_m^j$. Per l'ipotesi fatta si ha (supponendo le macchine siano numerate secondo l'ordine naturale del flow shop)

$$C_m^i = S_1^i + \sum_{h \leq m} p_h^i, \qquad S_m^j = S_1^j + \sum_{h < m} p_h^j$$

e allora

$$C_m^i \leq S_m^j \Longrightarrow S_1^j - S_1^i \geq \sum_{h \leq m} p_h^i - \sum_{h < m} p_h^j$$

per cui, avendo l'obiettivo di rendere il più piccolo possibile $S_1^j - S_1^i$, si ha

$$S_1^j - S_1^i = \max_m \sum_{h \leq m} p_h^i - \sum_{h < m} p_h^j =: c_{ij}$$

Il problema è pertanto modellabile come un TSP in cui il "costo" di schedulare j immediatamente dopo i corrisponde al ritardo c_{ij}. Data una permutazione il tempo di completamento è dato dalla somma dei termini

c_{ij} più la somma delle durate su tutte le macchine dell'ultimo lavoro schedulato. Per modellare quindi il problema come un TSP asimmetrico bisogna aggiungere un nodo (indichiamolo come 0) e assegnare i costi $c_{0j} := 0$, e $c_{j0} := \sum_m p_m^j$.

Nel caso in esame si ottiene

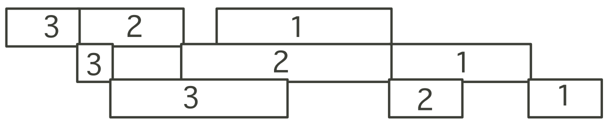
$$c = \begin{pmatrix} * & 6 & 8 \\ 4 & * & 8 \\ 2 & 1 & * \end{pmatrix}$$

e, aggiungendo il nodo 0, si ha

$$c = \begin{pmatrix} * & 0 & 0 & 0 \\ 11 & * & 6 & 8 \\ 11 & 4 & * & 8 \\ 8 & 2 & 1 & * \end{pmatrix}$$

da cui si ottengono in modo esaustivo i costi delle sei permutazioni

$$\begin{aligned} &(1 \to 2 \to 3) \Longrightarrow 22 \qquad &&(1 \to 3 \to 2) \Longrightarrow 21 \\ &(2 \to 1 \to 3) \Longrightarrow 20 \qquad &&(2 \to 3 \to 1) \Longrightarrow 20 \\ &(3 \to 1 \to 2) \Longrightarrow 18 \qquad &&(3 \to 2 \to 1) \Longrightarrow 17 \end{aligned}$$



e il costo ottimo si ha per la permutazione $3 \to 2 \to 1$, che, caso fortuito, è la medesima del flow shop normale.