Esercizio 1: Scheduling

Il problema richiede di decidere le quantità di ogni tipo di prodotto, con vincoli sugli istanti di inizio e fine delle produzioni nelle diverse fasi. A questo scopo è necessario definire variabili continue non-negative *s(p,m)* e *e(p,m)* che indicano gli istanti di inizio e fine della lavorazione di ogni prodotto *p* su ogni macchina *m*.

Su queste variabili si possono quindi imporre i vincoli di non-sovrapposizione delle lavorazioni per prodotti diversi su ogni singola macchina. Per ogni macchina *m* e per ogni prodotto *p* tranne l'ultimo si ha *e(p,m) <= s(p+1,m)*, cioè la fine di una lavorazione deve precedere l'inizio della lavorazione successiva.

Inoltre si può facilmente imporre il vincolo sulla durata complessiva della produzione: i valori di *e(p,m)* devono essere tutti non superiori a 8 ore. Per semplicità, dato l'ordinamento delle lavorazioni e l'ordinamento dei prodotti, basta imporre questo vincolo al tempo finale dell'ultima lavorazione dell'ultimo prodotto (su tutte le macchine che la possono eseguire).

Sempre tramite le stesse variabili *s* ed *e* è anche possibile imporre i vincoli sull'ordinamento delle lavorazioni per ogni prodotto. Per ogni coppia di lavorazioni consecutive, *v* e *v+1*, e per ogni coppia di macchine *m1* e *m2*, che le possono rispettivamente eseguire, si impone che per ogni prodotto *p* la fine della lavorazione di *p* su *m1* debba precedere l'inizio della lavorazione di *p* su *m2*, ossia *e(p,m1) <= s(p,m2)*.

Per imporre il vincolo solo alle macchine che possono effettivamente eseguire le lavorazioni, si può ricorrere a dati binari, che esprimono le compatibilità tra macchine e lavorazioni, come rappresentato nella Tabella 1 dei dati. Si ha perciò una matrice di coefficienti binari *w(m,v)* con tante righe quante le macchine (*m=1,...,6*) e tante colonne quante le lavorazioni (*v=1,..,3*).

Il passaggio dai tempi *s* ed *e* alle quantità prodotte si può realizzare usando i tempi di lavorazione. La durata di ogni lavorazione è la differenza tra l'istante finale e l'istante iniziale ed il rapporto tra la durata della lavorazione ed il tempo di lavorazione unitario fornisce la quantità di prodotto lavorato. Si ha quindi per ogni tipo di lavorazione *v*, per ogni macchina *m* che lo può svolgere e per ogni prototto *p*: *e(p,m)-s(p,m) = t(p,v) * y(m,p)*, dove si è indicato con *t(p,v)* il tempo unitario della lavorazione *v* del prodotto *p* e con *y(m,p)* la quantità totale di prodotto *p* lavorata sulla macchina *m*.

Occorre infine che siano coerenti le quanntità prodotte nelle diverse lavorazioni per ogni prodotto, poiché il prodotto non si crea e non si distrugge passando da una lavorazione a quella successiva. Definiamo quindi delle quantità *x(p)* che indicano il totale prodotto per ogni tipo e imponiamo che quando più macchine lavorano in parallelo lo stesso prodotto la quantità complessiva da esse lavorata sia sempre pari a *x(p).*

La funzione obiettivo si esprime facilmente come combinazione lineare delel quantità *x(p)* pesate con i profitti unitari.

Il modello risultante è di programmazione lineare. Il modello Lingo è nel file SCHEDULING.LG4 e la soluzione ottima corrispondente è nel file SCHEDULING.LGR, nel quale sono state cancellate per comodità le righe corrispondenti ai dati, mantenendo solo quelle corrispondenti alle variabili.

Le risposte alle due domande finali possono essere ricavate dall'analisi della soluzione. In particolare i tempi di fermo-macchina si ricavano dalle differenze *s(p,m)-e(p-1,m)* per ogni macchina *m*. Per evitare di doverli calcolare a mano, basta inserire opportune variabili nel modello e leggerne il valore in uscita.

La lavorazione bottleneck è quella per la quale le macchine non si fermano mai nel passaggio da un prodotto al successivo, cioè tutte le differenze *s(p,m)-e(p-1,m)* sono nulle. Si tratta della macchina 3, l'unica incaricata di eseguire la seconda lavorazione.