Assegnamento a costo minimo

Sia dato il grafo bipartito completo $G = (I, J; E)$
Problema primale:

$$\begin{aligned} \min \quad & \sum_{(ij)\in E} c_{ij}\, x_{ij} \\ & \sum_{j\in J} x_{ij} = 1 \qquad i \in I \\ & \sum_{i\in I} x_{ij} = 1 \qquad j \in J \\ & x_{ij} \geq 0 \qquad\quad (ij) \in E \end{aligned}$$

che riscriviamo come:

$$\begin{aligned} \min \quad & \sum_{(ij)\in E} c_{ij}\, x_{ij} \\ & -\sum_{j\in J} x_{ij} = -1 \qquad i \in I \\ & \sum_{i\in I} x_{ij} = \quad 1 \qquad j \in J \\ & x_{ij} \geq 0 \qquad\quad (ij) \in E \end{aligned}$$

Problema duale

$$\begin{aligned} \max \quad & -\sum_{i\in I} y_i + \sum_{j\in J} z_j \\ & -y_i + z_j \leq c_{ij} \quad (ij) \in E \end{aligned}$$

oppure

$$\begin{aligned} \max \quad & -\sum_{i\in I} y_i + \sum_{j\in J} z_j \\ & -y_i + z_j + t_{ij} = c_{ij} \quad (ij) \in E \\ & t_{ij} \geq 0 \qquad\qquad\quad (ij) \in E \end{aligned}$$

Complementarità ($\hat{x}$, $\hat{y}$, $\hat{z}$, $\hat{t}$ ottimi)

$$\hat{x}_{ij} > 0 \Longrightarrow \hat{t}_{ij} = 0; \qquad\qquad \hat{t}_{ij} > 0 \Longrightarrow \hat{x}_{ij} = 0$$

L'algoritmo esegue n iterazioni. Alla fine dell'iterazione k generica sono noti

- un accoppiamento perfetto fra i nodi $\{1, 2, \ldots, k\}$ di sinistra e i nodi di un insieme $\bar{T}$ di destra (ovviamente $|\bar{T}| = k$); sia $T := J \setminus \bar{T}$
- una soluzione primale (non ammissibile) $\bar{x}$ tale che

$$x_{ij} := \begin{cases} 1 & \text{se } i \text{ e } j \text{ sono accoppiati} \\ 0 & \text{altrimenti} \end{cases}$$

- una soluzione duale $\bar{y}$, $\bar{z}$, $\bar{t} = c + \bar{y} - \bar{z}$, tale che $\bar{t} \geq 0$ e la condizione di complementarità è soddisfatta.

Inizialmente la situazione è:

- $k = 0$ e $\bar{T} = \emptyset$
- $\bar{x} = 0$
- $\bar{y} := 0$, $\bar{z} := 0$, $\bar{t} = c$.

Consideriamo una singola iterazione. Si ripartiscano gli archi come

$$E_0 := \{(ij) \in E : \bar{x}_{ij} = 0\}, \qquad E_1 := \{(ij) \in E : \bar{x}_{ij} = 1\}$$

Si variano ora le variabili duali lasciando inalterati i valori primali. Le variazioni Δy, Δz, Δt delle variabili duali, ammissibili per mantenere la condizione di complementarità, sono (si noti che $\Delta t = \Delta y - \Delta z$):

– sugli archi in E_0 il valore di t_{ij} può essere aumentato senza limitazioni mentre può essere diminuito al massimo di $\bar{t}_{ij}$. Quindi

$$\Delta t_{ij} = \Delta y_i - \Delta z_j \geq -\bar{t}_{ij} \qquad (ij) \in E_0$$

ovvero

$$\Delta z_j - \Delta y_i \leq \bar{t}_{ij} \qquad (ij) \in E_0 \tag{1}$$

– sugli archi in E_1 il valore di t_{ij} deve essere mantenuto nullo. Quindi $\Delta t_{ij} = \Delta y_i - \Delta z_j = 0$ che riscriviamo come

$$\Delta y_i - \Delta z_j \leq 0, \qquad \Delta z_j - \Delta y_i \leq 0 \qquad (ij) \in E_1 \tag{2}$$

Le diseguaglianze (1) e (2) sono della stessa forma dei vincoli di un problema di programmazione dinamica per un problema di cammino minimo su un grafo orientato gli archi in E_0 orientati da I a J e quelli in E_1 sostituiti da una coppia di archi antiparalleli e con distanze definite da

$$d_{ij} = \begin{cases} \bar{t}_{ij} & (ij) \in E_0 \\ 0 & (ij) \in E_1 \\ 0 & (ji) \in E_1 \end{cases} \tag{3}$$

Definiamo ora come nodo sorgente s il nodo $k+1$, e calcoliamo il cammino minimo da s al più vicino nodo in T. I valori dei cammini ottimi sono gli ottimi di

$$\begin{aligned} \max \ & \sum_{t \in T} \Delta z_t \\ & \Delta z_j - \Delta y_i \leq \bar{t}_{ij} \quad (ij) \in E_0 \\ & \Delta z_j - \Delta y_i \leq 0 \quad (ij) \in E_1 \\ & \Delta y_i - \Delta z_j \leq 0 \quad (ij) \in E_1 \\ & \Delta y_s = 0 \end{aligned} \tag{4}$$

Su ogni arco del cammino minimo si ha necessariamente $-\Delta t_{ij} = \Delta z_j - \Delta y_i = d_{ij}$ per cui, una volta aggiornati i valori di $\bar{y}_i$ e di $\bar{z}_j$, si ha $\bar{t}_{ij} = 0$ su tutti gli archi del cammino minimo. In pratica basta applicare l'algoritmo di Dijkstra e interrromperlo non appena viene raggiunto un nodo w in T. Siano δ_i e δ_j le distanze minime da s ai vari nodi e sia R l'insieme dei nodi raggiunti. A quel punto l'algoritmo ha prodotto i valori di distanza minima δ_i per i nodi i raggiunti (con $\delta_w \geq \delta_i$ per ogni $i \in R$ e $\delta_w \leq \delta_i$ per ogni $i \notin R$). I valori

$$\Delta y_i = \begin{cases} \delta_i & \text{se } i \in R \cap I \\ \delta_w & \text{se } i \notin R,\ i \in I \end{cases} \qquad \Delta z_j = \begin{cases} \delta_j & \text{se } j \in R \cap J \\ \delta_w & \text{se } j \notin R,\ j \in J \end{cases}$$

sono ammissibili in (4), come si vede dalle proprietà dell'algoritmo di Dijkstra e dell'equazione di Bellman (dalla quale $\delta_j \leq \delta_i + d_{ij}$): infatti se $i, j \in R$, $\Delta y_i = \delta_i$ e $\Delta z_j = \delta_j$ e quindi si ha $\Delta z_j - \Delta y_i \leq d_{ij}$; se $i \in R$ e $j \notin R$, si ha $\delta_w \leq \delta_j \leq \delta_i + d_{ij}$ da cui $\Delta z_j - \Delta y_i \leq d_{ij}$. Se $i \notin R$ e $j \in R$ si ha $\Delta z_j - \Delta y_i = \delta_j - \delta_w \leq 0 \leq d_{ij}$. Se $i \notin R$ e $j \notin R$ si ha $\Delta z_j - \Delta y_i = \delta_w - \delta_w = 0 \leq d_{ij}$.

Siccome $\bar{t}_{ij} = 0$ sugli archi del cammino minimo, su tali archi il valore di $\bar{x}$ può essere modificato senza violare la condizione di complementarità. Inoltre il cammino minimo è necessariamente costituito da archi non accoppiati ed archi accoppiati in modo alternato (cosiddetto cammino alternante): infatti partendo da s non accoppiato il primo arco è non accoppiato. Se il successivo nodo in J non è accoppiato tale nodo è anche il nodo terminale del cammino, altrimenti il cammino ritorna ad un nodo in I. Per farlo non ci sono archi non accoppiati da percorrere all'indietro, ma solo l'arco accoppiato con il nodo, arco da percorrere verso I e di lunghezza 0. Il cammino prosegue verso J con un arco non accoppiato (l'unico arco accoppiato è appena stato usato) e continuando ricorsivamente finché non termina in un nodo in J non accoppiato.

Essendo il cammino alternante basta invertire gli accoppiamenti sul cammino ottenendo un nuovo accoppiamento con un arco accoppiato in più rispetto al caso precedente e l'iterazione è terminata.

L'algoritmo termina in condizioni di ottimalità quando tutti i nodi sono accoppiati. Siccome si eseguono n iterazioni e per ogni iterazione bisogna usare l'algoritmo di Dijkstra, la complessità è $O(n^3)$.

Esempio: sia data la seguente istanza:

$$c \quad = \quad \begin{array}{|c|c|c|c|} \hline 16 & 9 & 5 & 12 \\ \hline 15 & 13 & 3 & 17 \\ \hline 8 & 4 & 16 & 5 \\ \hline 16 & 5 & 18 & 11 \\ \hline \end{array}$$

Nella prima iterazione i valori di distanza sono esattamente uguali ai costi e il primo cammino minimo è sempre l'arco di costo minimo incidente nel nodo $1 \in I$, in questo caso l'arco $(1, 7)$ (Figura 1). Quindi le variazioni duali sono

$$\Delta y = \{0, 5, 5, 5\}, \qquad \Delta z = \{5, 5, 5, 5\}$$

e i nuovi valori duali sono:

$$\bar{y} = \{0, 5, 5, 5\}, \qquad \bar{z} = \{5, 5, 5, 5\}$$

A seguito di questo aggiornamento gli archi da considerare sono quelli nella tabella a sinistra con i seguenti valori di distanza:

$$\bar{t} \quad = \quad \begin{array}{|c|c|c|c|} \hline 11 & 4 & 0 & 7 \\ \hline 15 & 13 & 3 & 17 \\ \hline 8 & 4 & 16 & 5 \\ \hline 16 & 5 & 18 & 11 \\ \hline \end{array}$$

Con questi valori l'algoritmo di Dijkstra applicato a partire dal nodo 2 produce prima l'arco $(2, 7)$, poi l'arco accoppiato $(7, 1)$ e infine l'arco $(1, 6)$. Le variazioni duali sono quindi

$$\Delta y = \{3, 0, 7, 7\}, \qquad \Delta z = \{7, 7, 3, 7\}$$

e i nuovi valori duali sono:

$$\bar{y} = \{3, 5, 12, 12\}, \qquad \bar{z} = \{12, 12, 8, 12\}$$

A seguito di questi nuovi valori duali l'accoppiamento può essere cambiato scambiando gli accoppiamenti sul cammino minimo $2 \to 7 \to 1 \to 6$ (Figura 2). Le nuove distanze sono:

$$\bar{t} \quad = \quad \begin{array}{|c|c|c|c|} \hline 7 & 0 & 0 & 3 \\ \hline 8 & 6 & 0 & 10 \\ \hline 8 & 4 & 20 & 5 \\ \hline 16 & 5 & 22 & 11 \\ \hline \end{array}$$

Con questi valori l'algoritmo di Dijkstra applicato a partire dal nodo 3 produce in successione gli archi $(3, 6)$, $(6, 1)$, $(1, 7)$, $(7, 2)$ e $(3, 8)$. Quest'ultimo arco è il cammino minimo cercato (vedi Figura 3). Quindi

$$\Delta y = \{4, 4, 0, 5\}, \qquad \Delta z = \{5, 4, 4, 5\}$$

e

$$\bar{y} = \{7, 9, 12, 17\}, \qquad \bar{z} = \{17, 16, 12, 17\}$$

Con i valori di distanza

$$\bar{t} \quad = \quad \begin{array}{|c|c|c|c|} \hline 6 & 0 & 0 & 2 \\ \hline 7 & 6 & 0 & 9 \\ \hline 3 & 0 & 16 & 0 \\ \hline 16 & 6 & 23 & 11 \\ \hline \end{array}$$

l'algoritmo di Dijkstra applicato a partire dal nodo 4 produce gli archi $(4,6)$, $(6,1)$, $(1,7)$, $(7,2)$, $(1,8)$, $(8,3)$ e $(3,5)$, con cammino minimo $4 \to 6 \to 1 \to 8 \to 3 \to 5$ (Figura 4)

$$\Delta y = \{6,6,8,0\}\,, \qquad \Delta z = \{11,6,6,8\}$$

$$\bar{y} = \{13,15,20,17\}\,, \qquad \bar{z} = \{28,22,18,25\}$$

Questi sono i valori ottimi duali e l'accoppiamento ottimo si ottiene scambiando gli accoppiamenti sul cammino minimo (Figura 5).

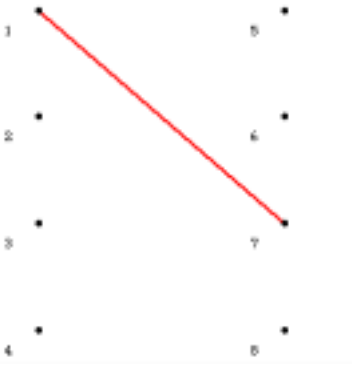

Figura 1

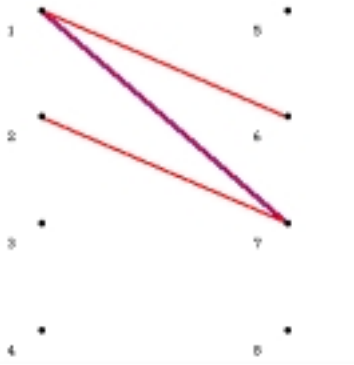

Figura 2

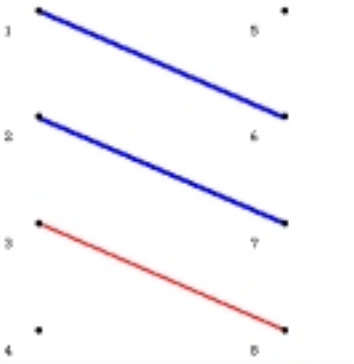

Figura 3

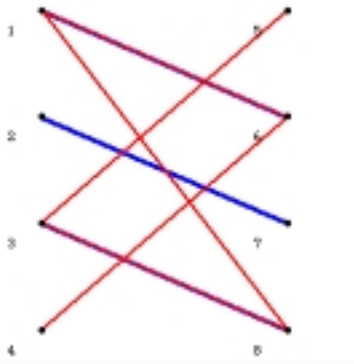

Figura 4

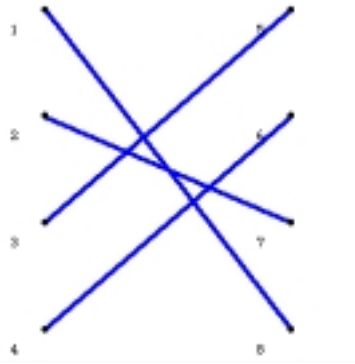

Figura 5