Assegnamento

Sia dato un grafo bipartito completo $G = (V_1 \cup V_2, E)$ con

$$V_1 = \{a_1, \ldots, a_n\}, \quad V_2 = \{b_1, \ldots, b_n\}$$

e quindi $|V_1| = |V_2| = n$. Agli archi $(i, j) \in E$ sono associati i costi $d_{ij} \geq 0$ e interi.

Il problema di assegnamento consiste nel cercare tra tutti i matching M di cardinalità n su tale grafo, quello per cui

$$\sum_{(i,j) \in M} d_{ij}$$

è minimo.

Modello per Esempio 5

Il problema nell'Esempio 5 può essere modellato come problema di assegnamento dove:

- i nodi in V_1 sono i lavori e quelli in V_2 i lavoratori;
- gli archi del grafo rappresentano il possibile assegnamento di un lavoro a un lavoratore;
- i valori d_{ij} degli archi rappresentano i costi di assegnamento del lavoro i al lavoratore j.

Risolvere il problema di individuare i lavori da assegnare ai vari lavoratori a un costo totale minimo equivale a risolvere un problema di assegnamento sul grafo appena descritto.

Nel seguito vedremo una procedura di risoluzione per questo problema, l'algoritmo ungherese.

Osservazione

Si è supposto che V_1 e V_2 abbiano la stessa cardinalità n. Vi sono però casi in cui questo non è vero ($\mid V_1 \mid > \mid V_2 \mid$ oppure $\mid V_1 \mid < \mid V_2 \mid$).

Questi casi possono **sempre** essere ricondotti al caso $\mid V_1 \mid = \mid V_2 \mid$ aggiungendo elementi fittizi nell'insieme più piccolo con costo degli accoppiamenti con elementi fittizi pari a 0 (l'accoppiamento con un elemento fittizio equivale a un non accoppiamento).

Un esempio

Tabella dei costi ($n = 4$):

	b_1	b_2	b_3	b_4
a_1	2	3	4	5
a_2	6	2	2	2
a_3	7	2	3	3
a_4	2	3	4	5

Matrice dei costi

La matrice dei costi T_0 di ordine $n \times n$ è semplicemente la matrice che ha come elemento nella posizione (i, j) il valore d_{ij}.

Riduzione della matrice - I

Il primo passo consiste nel trasformare la matrice T_0 in una nuova matrice. Si comincia a calcolare per ogni colonna j il minimo su tale colonna

$$d_j^0 = \min_i \ d_{ij};$$

tale valore verrà sottratto ad ogni elemento della colonna j e questo viene fatto per tutte le colonne. Si ottiene quindi una nuova matrice T_1 che nella posizione (i, j) ha il valore $d_{ij} - d_j^0$.

Nell'esempio

	b_1	b_2	b_3	b_4
a_1	2	3	4	5
a_2	6	2	2	2
a_3	7	2	3	3
a_4	2	3	4	5
d_j^0	2	2	2	2

Da cui T_1:

	b_1	b_2	b_3	b_4
a_1	0	1	2	3
a_2	4	0	0	0
a_3	5	0	1	1
a_4	0	1	2	3

Riduzione della matrice - II

La matrice T_1 viene ulteriormente trasformata andando a calcolare il minimo su ogni sua riga i

$$d_i^1 = \min_j \left[d_{ij} - d_j^0\right]$$

e sottraendo questo ad ogni elemento della riga i. Il risultato è una matrice T_2 che nella posizione (i, j) ha il valore

$$d_{ij}^2 = d_{ij} - d_j^0 - d_i^1 \geq 0.$$

È importante notare che, per come sono stati ottenuti, tutti gli elementi d_{ij}^2 sono non negativi.

Nell'esempio

	b_1	b_2	b_3	b_4	d_i^1
a_1	0	1	2	3	0
a_2	4	0	0	0	0
a_3	5	0	1	1	0
a_4	0	1	2	3	0

Da cui T_2:

	b_1	b_2	b_3	b_4
a_1	0	1	2	3
a_2	4	0	0	0
a_3	5	0	1	1
a_4	0	1	2	3

Rappresentazione assegnamento

Per ogni coppia $i \in V_1, j \in V_2$, si introduca la variabile

$$x_{ij} = \begin{cases} 1 & \text{se } i \text{ assegnato a } j \\ 0 & \text{altrimenti} \end{cases}$$

Un assegnamento soddisfa le condizioni:

$$\sum_{j \in V_2} x_{ij} = 1 \quad \forall\, i \in V_1$$

(a ogni $i \in V_1$ si associa uno e un solo $j \in V_2$)

$$\sum_{i \in V_1} x_{ij} = 1 \quad \forall\, j \in V_2$$

(a ogni $j \in V_2$ si associa uno e un solo $i \in V_1$).

Un lower bound per il valore ottimo

Osserviamo che

$$d_{ij} = d^2_{ij} + d^0_j + d^1_i.$$

Andando a sostituire nel valore dell'assegnamento al posto di d_{ij} si ottiene

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n}(d^2_{ij} + d^0_j + d^1_i)x_{ij} =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} d^2_{ij}x_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{n} d^0_j x_{ij} + \sum_{i=1}^{n}\sum_{j=1}^{n} d^1_i x_{ij} =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} d^2_{ij}x_{ij} + \sum_{j=1}^{n} d^0_j \sum_{i=1}^{n} x_{ij} + \sum_{i=1}^{n} d^1_i \sum_{j=1}^{n} x_{ij}.$$

Per ogni assegnamento si ha $\sum_{i=1}^{n} x_{ij} = 1$ per ogni j e $\sum_{j=1}^{n} x_{ij} = 1$ per ogni i, quindi il valore dell'assegnamento è uguale a

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^{2} x_{ij} + \sum_{j=1}^{n} d_{j}^{0} + \sum_{i=1}^{n} d_{i}^{1}.$$

Ponendo

$$D_0 = \sum_{j=1}^{n} d_{j}^{0} \quad D_1 = \sum_{i=1}^{n} d_{i}^{1},$$

si ha infine

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij}^{2} x_{ij} + D_0 + D_1.$$

Continua

Poiché, come già osservato, $d_{ij}^2 \geq 0$ per ogni i, j, e poiché si ha anche che $x_{ij} \geq 0$ per ogni i, j, si ha che

$$\sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \geq D_0 + D_1,$$

per ogni possibile assegnamento.

Nell'esempio ...

... abbiamo $D_0 = 8$ e $D_1 = 0$. Quindi, $D_0 + D_1 = 8$ fornisce un lower bound per il valore ottimo di questa istanza del problema di assegnamento.

Ma allora ...

... se trovo un assegnamento con valore pari a $D_0 + D_1$, questo è certamente anche una soluzione ottima.

Quindi la domanda che ci poniamo ora è la seguente: esiste o meno un assegnamento con valore $D_0 + D_1$?

In caso di risposta positiva, abbiamo una soluzione ottima del problema, in caso di risposta negativa ci dovremo poi porre la questione di cosa fare se non esiste.

Problema associato

Determinare un sottinsieme Δ di cardinalità massima degli 0 della matrice T_2 tale che presi due elementi qualsiasi di Δ essi sono **indipendenti***, ovvero appartengono a righe* *e* *colonne diverse.*

Se ...

... questo problema ammette una soluzione Δ con $\mid \Delta \mid = n$, consideriamo allora:

$$\overline{x}_{ij} = \begin{cases} 1 & \text{se } (i,j) \in \Delta \\ 0 & \text{altrimenti} \end{cases}$$

Per prima cosa dimostriamo che tale soluzione è un assegnamento. Supponiamo per assurdo che non lo sia. Per esempio supponiamo che per qualche j si abbia

$$\sum_{i=1}^{n} \overline{x}_{ij} \neq 1.$$

Casi possibili

- **Caso I** $\sum_{i=1}^{n} \overline{x}_{ij} = 0$: in tal caso non c'è nessun elemento di Δ nella colonna j. Quindi ve ne dovranno essere n nella restanti $n-1$ colonne. Ciò vuol dire che almeno una colonna contiene due elementi in Δ, ma questo è assurdo in quanto gli elementi di Δ devono essere tra loro indipendenti e quindi non possono appartenere ad una stessa colonna.
- **Caso II** $\sum_{i=1}^{n} \overline{x}_{ij} \geq 2$: in tal caso ci sono due elementi di Δ nella colonna j e si ha una contraddizione identica a quella vista per il Caso I.

Continua

In modo del tutto analogo si vede che le condizioni

$$\sum_{j=1}^{n} \overline{x}_{ij} = 1 \quad \forall\, i,$$

non possono essere violate. Quindi abbiamo un assegnamento.

Valore dell'assegnamento

Si nota che le $\overline{x}_{ij}$ sono uguali a 1 solo in corrispondenza di coppie (i, j) per cui si ha $d_{ij}^2 = 0$. Quindi

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\overline{x}_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2\overline{x}_{ij} + D_0 + D_1 = D_0 + D_1.$$

ovvero il valore di questo assegnamento è $D_0 + D_1$ e per quanto già osservato la soluzione è ottima.

Ma come si calcola Δ?

Si costruisca un grafo *bipartito* nel modo seguente:

* i due insiemi di vertici sono rispettivamente rappresentati dall'insieme A e dall'insieme B;
* tra il vertice a_i ed il vertice b_j si traccia un arco (non orientato) se e solo se $d_{ij}^2 = 0$.

Un insieme indipendente di 0 equivale a un matching su tale grafo bipartito. Infatti, cercare insiemi di 0 nella tabella che non siano mai sulla stessa riga e colonna equivale a cercare insiemi di archi nel grafo bipartito che non abbiano nodi in comune, ovvero equivale a cercare dei matching. Quindi, determinare il massimo insieme di 0 indipendenti equivale a risolvere un problema di matching di cardinalità massima sul grafo bipartito.

Nell’esempio

Risolvendo con l’algoritmo visto per i problemi di matching di cardinalità massima su grafi bipartiti, si ottiene la seguente soluzione:

$$\Delta = \{(a_1, b_1); (a_2, b_3); (a_3, b_2)\}$$

con $\mid \Delta \mid < n = 4$.

Che fare se $\mid \Delta \mid < n$?

L'obiettivo finale sarà quello di giungere ad un'ulteriore trasformazione della matrice T_2. Per arrivare a questa è necessario un passaggio ulteriore in cui si sfrutta l'insieme Δ trovato. Si tratterà di risolvere il seguente problema:

determinare un insieme minimo di righe e colonne tali che ricoprendole si ricoprono tutti gli 0 della matrice T_2

Nel seguito si parlerà genericamente di *linee*, dove una linea può essere indifferentemente una riga od una colonna.

Continua

Questo è strettamente legato a quello della determinazione dell'insieme Δ.

Infatti, consideriamo le etichette ottenute all'ultima iterazione dell'algoritmo per il massimo matching sul grafo bipartito costruito per determinare Δ. Si può dimostrare la seguente proprietà:

Un ricoprimento ottimo per il problema dato è formato da esattamente $\mid \Delta \mid$ *linee ed è costituito da:*

(i) Le righe a_i *corrispondenti a nodi non etichettati.*

(ii) le colonne b_i *corrispondenti a nodi etichettati.*

Nell'esempio

Le righe corrispondenti a nodi non etichettati al termine della risoluzione del problema di matching sono a_2 e a_3, mentre la sola colonna corrispondente a un nodo etichettato è la b_1.

Aggiornamento della matrice T_2

Il ricoprimento con un numero minimo di linee ottenuto con la procedura appena descritta ci serve per aggiornare la matrice T_2 e trasformarla in una nuova matrice T_3. La trasformazione avviene seguendo questi passi.

a) Determinare il valore minimo λ tra tutti gli elementi di T_2 non ricoperti da alcuna linea. Si noti che essendo gli 0 di T_2 tutti ricoperti, gli elementi non ricoperti sono tutti positivi e quindi λ stesso è positivo.

b) Gli elementi d_{ij}^3 della nuova matrice T_3 sono definiti in questo modo:

$$d_{ij}^3 = d_{ij}^2 + d_i^3 + d_j^3,$$

dove

$$d_i^3 = \begin{cases} 0 & \text{se la riga } a_i \text{ è nel ricoprimento} \\ -\lambda & \text{altrimenti} \end{cases}$$

e

$$d_j^3 = \begin{cases} \lambda & \text{se la colonna } b_j \text{ è nel ricoprimento} \\ 0 & \text{altrimenti} \end{cases}$$

Più semplicemente ...

... quanto visto equivale alla seguente regola:

- gli elementi ricoperti da due linee in T_2 devono essere incrementati di λ;
- gli elementi non ricoperti da alcuna linea vengono decrementati di λ;
- tutti gli altri (gli elementi ricoperti da una sola linea) non cambiano

Nell'esempio

Si ha $\lambda = 1$ e T_3 sarà la seguente tabella:

	b_1	b_2	b_3	b_4
a_1	0	0	1	2
a_2	5	0	0	0
a_3	6	0	1	1
a_4	0	0	1	2

Osservazione

Da questa regola si vede anche che i soli elementi a cui viene sottratto qualcosa sono quelli non ricoperti e ad essi viene sottratto il minimo di tutti gli elementi non ricoperti.

Ciò significa che tutti gli elementi d_{ij}^3 di T_3 saranno non negativi come lo erano quelli di T_2.

Un nuovo lower bound

Il valore di un assegnamento era stato riscritto in questo modo:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 x_{ij} + D_0 + D_1.$$

Osserviamo ora che $d_{ij}^2 = d_{ij}^3 - d_i^3 - d_j^3$ ed andando a sostituire otteniamo:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 x_{ij} + D_0 + D_1 = \sum_{i=1}^{n}\sum_{j=1}^{n} (d_{ij}^3 - d_i^3 - d_j^3) x_{ij} + D_0 + D_1 =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^3 x_{ij} - \sum_{j=1}^{n}\sum_{i=1}^{n} d_j^3 x_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n} d_i^3 x_{ij} + D_0 + D_1 =$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^3 x_{ij} - \sum_{j=1}^{n} d_j^3 \sum_{i=1}^{n} x_{ij} - \sum_{i=1}^{n} d_i^3 \sum_{j=1}^{n} x_{ij} + D_0 + D_1.$$

Quindi ...

... per ogni assegnamento, si avrà

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^3 x_{ij} - \sum_{j=1}^{n} d_j^3 - \sum_{i=1}^{n} d_i^3 + D_0 + D_1.$$

Continua

Vediamo ora di calcolare $\sum_{j=1}^{n} d_j^3$ e $\sum_{i=1}^{n} d_i^3$.

Indichiamo con h_1 il numero di righe nel ricoprimento e con h_2 il numero di colonne nel ricoprimento. Si noti che $h_1 + h_2 =\mid \Delta \mid$.

Si ha:

$$\sum_{i=1}^{n} d_i^3 = -\lambda \times (\text{numero righe che non sono nel ricoprimento}) = -\lambda(n - h_1),$$

e

$$\sum_{j=1}^{n} d_j^3 = \lambda \times (\text{numero colonne che sono nel ricoprimento}) = \lambda h_2.$$

Quindi...

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^{3}x_{ij} + \lambda(n - h_1) - \lambda h_2 + D_0 + D_1,$$

da cui

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^{3}x_{ij} + \lambda(n- \mid \Delta \mid) + D_0 + D_1.$$

Continua

- dal momento che $d_{ij}^3 \geq 0$ e $x_{ij} \geq 0$ si ha che un limite inferiore per il problema di assegnamento è $D_0 + D_1 + \lambda(n - \mid \Delta \mid)$;
- se riesco a trovare un assegnamento che ha come valore dell'obiettivo proprio tale limite inferiore ho determinato una soluzione ottima;
- la verifica se un tale assegnamento esista può essere fatto cercando un sottinsieme indipendente di cardinalità massima in T_3, esattamente come si era fatto in T_2. Se esso ha cardinalità n si ha un assegnamento ottimo, altrimenti sfruttando tale sottinsieme e passando attraverso la determinazione di un insieme minimo delle linee di ricoprimento di T_3 si determina una nuova matrice T_4 e si itera in questo modo la procedura fino a che si è determinato un assegnamento ottimo.

Una semplificazione

Per l'individuazione dell'insieme Δ su T_3 conviene sfruttare i calcoli già fatti su T_2.

Più precisamente, come matching iniziale sul grafo bipartito associato agli 0 di T_3 posso prendere il matching ottimo individuato sul grafo bipartito associato agli 0 di T_2.

Si può infatti dimostrare che anche dopo l'aggiornamento di T_2 in T_3 l'insieme di 0 indipendenti che era soluzione ottima del problema di matching per T_2, si ritrova ancora in T_3.

Finitezza

Ci si può chiedere se la procedura termina oppure no, cioè se si arriva infine ad una matrice T_h che contiene un sottinsieme di 0 indipendenti a due a due di cardinalità n.

Nel caso in cui tutti i d_{ij} siano interi si ha certamente terminazione finita. Lo si può vedere da come aumentano le limitazioni inferiori ad ogni iterazione.

Con T_2 avevamo una limitazione inferiore per la soluzione ottima pari a $D_0 + D_1$; con T_3 si è passati ad una limitazione inferiore pari a $D_0 + D_1 + \lambda(n- \mid \Delta \mid)$.

La limitazione inferiore cresce quindi almeno di una quantità pari a $\lambda > 0$. Nel caso in cui i d_{ij} siano interi λ deve essere anch'esso un intero e quindi è certamente maggiore o uguale a 1. Se per assurdo la procedura dovesse essere iterata infinite volte, la limitazione inferiore stessa crescerebbe all'infinito ma questo è assurdo in quanto un qualsiasi assegnamento (ad esempio l'assegnamento (a_i, b_i) per ogni $i \in \{1, \ldots, n\}$) ha un valore finito ed il minimo di tali assegnamenti non può quindi crescere all'infinito.

Complessità

Qui ci siamo limitati a dimostare che la procedura termina in un numero finito di iterazioni. In realtà si può dimostrare che essa richiede un numero $O(n^3)$ di operazioni ed è quindi una procedura di complessità polinomiale.

Nota bene

L’algpritmo ungherese può esser visto come un algoritmo costruttivo con la possibilità di rivedere decisioni passate.

Infatti, a ogni iterazione si ha un insieme Δ che definisce un assegnamento incompleto. Tale assegnamento incompleto può essere aggiornato in una data iterazione incrementandone la cardinalità, rimuovendo coppie e sostituendole con altre.