Flusso a costo minimo

Si consideri un'azienda che realizza un certo prodotto ed è formata da diversi centri che si distinguono in

- centri che fungono da magazzini, in cui il prodotto si limita a transitare;
- centri di produzione, in cui, oltre a transitare, il prodotto viene anche realizzato in quantità prefissate;
- centri di distribuzione, in cui, oltre a transitare, il prodotto viene anche rivenduto in quantità prefissate.

Valori associati ai centri

A ogni centro i si associa un valore intero b_i che sarà:

- $= 0$ per i magazzini;
- > 0 per i centri di produzione (b_i rappresenta la quantità di prodotto realizzata in quel centro);
- < 0 per i centri di distribuzione ($-b_i$ rappresenta la quantità di prodotto rivenduta in quel centro).

Si suppone che $\sum b_i = 0$, ovvero che la quantità di prodotto totale realizzata nei vari centri di produzione sia esattamente pari a quella complessivamente rivenduta in tutti i centri di distribuzione.

Collegamenti tra centri

Alcuni centri dell’azienda comunicano tra loro tramite linee di collegamento alle quali sono associati dei costi di trasporto: se abbiamo una linea di collegamento tra il centro i e il centro j, indicheremo con c_{ij} il costo di trasporto di una singola unità di prodotto lungo tale linea.

In alcuni casi alla linea è anche associato un altro valore, la sua capacità d_{ij}, che indica la massima quantità di prodotto trasportabile lungo essa.

Descrizione problema

Il problema del *flusso a costo minimo* consiste nel determinare come instradare il prodotto all'interno della rete di collegamento dell'azienda in modo che il prodotto realizzato nei centri di produzione giunga ai centri di distribuzione a un costo totale di trasporto complessivo minimo.

Osservazione

Pur avendo qui noi discusso il caso di una rete di collegamento tra centri di un'azienda, un problema di questo tipo può sorgere in altri tipi di rete, come una rete di comunicazione oppure una rete idraulica.

Rete di computer

- il "prodotto" in tal caso sono i bit di informazione;
- computer che producono informazione (con valore b_i associato positivo : rappresenta la capacità produttiva nell'unità di tempo);
- computer che devono ricevere l'informazione (con valore b_i associato negativo : $\mid b_i \mid$ rappresenta la quantità di informazione che possono ricevere nell'unità di tempo);
- computer che si limitano a smistare l'informazione (con valore b_i associato nullo) ;
- cavi di collegamento tra computer (con i c_{ij} che rappresentano i costi unitari di trasporto lungo essi).

Rete idraulica

- il "prodotto" in tal caso è l'acqua;
- centraline di immissione di acqua nella rete (con valore b_i associato positivo : rappresenta la quantità d'acqua che immettono nell'unità di tempo);
- punti di raccolta dell'acqua (con valore b_i associato negativo : $\mid b_i \mid$ rappresenta la quantità di acqua che ciascuno di essi deve ricevere nell'unità di tempo);
- punti di smistamento dell'acqua (con valore b_i associato nullo) ;
- tubi di collegamento tra punti della rete idraulica (con i c_{ij} che rappresentano i costi unitari di trasporto lungo essi).

Rappresentazione con grafi

Per slegarci dai dettagli non significativi delle applicazioni da cui sorgono questi problemi, possiamo utilizzare i grafi.

La rete (sia essa quella dell'azienda, quella di comunicazione, quella idraulica o quella proveniente da una qualche altra applicazione) viene rappresentata con un grafo orientato $G = (V, A)$ dove i nodi corrispondono ai vari centri e gli archi alle linee di collegamento tra essi.

Classificazione nodi

- **Nodi i tali che $b_i > 0$**: sono denominati nodi sorgente; in essi viene "realizzato" il "prodotto" che viaggia attraverso la rete;
- **Nodi i tali che $b_i < 0$**: sono denominati nodi destinazione; in essi viene "consumato" il "prodotto" che viaggia attraverso la rete;
- **Nodi i tali che $b_i = 0$**: sono denominati nodi transito; in essi il "prodotto" che viaggia attraverso la rete si limita a transitare.

Valori associati agli archi

Agli archi $(i, j) \in A$ sono associati i costi unitari c_{ij} ed eventualmente anche i valori di capacità d_{ij} relativi alle linee di collegamento che rappresentano.

Modello matematico-variabili

Associamo una variabile x_{ij} ad ogni arco (i,j) della rete:

$(i,j) \in A \;\; \rightarrow \;\; x_{ij} =$ quantità "prodotto" inviata lungo l'arco (i,j)

Per tali variabili sarà richiesto che:

- siano ≥ 0 (non si possono inviare quantità negative di prodotto lungo gli archi);
- siano $\leq d_{ij}$ (limite di capacità massima degli archi, se presente);
- siano intere (se il prodotto non è frazionabile).

Modello matematico - vincoli

In ogni nodo $i \in V$ si deve avere:

$$\text{(Flusso uscente da } i\text{) - (Flusso entrante in } i\text{)} = b_i$$

Se il nodo è di transito, questo vincolo dice che la quantità di prodotto uscente da i è esattamente pari a quella entrante.

Se il nodo è sorgente, questo vincolo dice che la quantità di prodotto uscente da i è pari a quella entrante in i sommata alla quantità di prodotto b_i realizzata in i.

Se il nodo è destinazione, questo vincolo dice che la quantità di prodotto entrante in i è pari a quella uscente da i sommata alla quantità di prodotto $-b_i$ che viene rivenduta in i.

Formule matematiche

Flusso uscente da i:

$$\sum_{j:(i,j)\in A} x_{ij}$$

Flusso entrante in i:

$$\sum_{j:(j,i)\in A} x_{ji}$$

Quindi, i vincoli corrisponderanno alle seguenti equazioni:

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \quad \forall\, i \in V$$

Modello matematico - obiettivo

Se il costo di trasporto di un'unità di prodotto lungo (i, j) è c_{ij}, il costo di trasporto di x_{ij} unità di prodotto è $c_{ij}x_{ij}$.

Se sommiamo su tutti gli archi otteniamo il costo totale di trasporto:

$$\sum_{(i,j)\in A} c_{ij}x_{ij}$$

che vorremo minimizzare.

Modello matematico completo

$$\min \ \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \quad \forall\ i \in V$$

$$0 \leq x_{ij} \leq d_{ij} \text{ interi} \qquad \forall\ (i,j) \in A$$

Matrice dei vincoli

Chiamiamo matrice dei vincoli di uguaglianza di tale problema la matrice avente tante righe quanti sono i vincoli (e quindi tante quante sono i nodi della rete) e tante colonne quante sono le variabili del problema (e quindi tanti quanti sono gli archi della rete).

Nella colonna relativa alla variabile x_{ij} compaiono i coefficienti di tale variabile nei vincoli.

Osservazione

La matrice dei vincoli di uguaglianza per i problemi di flusso a costo minimo coincide con la matrice di incidenza nodo-arco della rete.

Dimostrazione La colonna della matrice relativa alla variabile x_{ij} corrisponde a un arco $(i,j) \in A$. Abbiamo che x_{ij} ha coefficiente +1 nel vincolo relativo al nodo i (essendo l'arco (i,j) uscente da i, x_{ij} appare in $\sum_{j:(i,j)\in A} x_{ij}$), coefficiente -1 nell'equazione relativa al nodo j (essendo l'arco (i,j) entrante in j, x_{ij} appare in $-\sum_{i:(i,j)\in A} x_{ij}$), mentre ha coefficiente 0 in corrispondenza delle equazioni relative a tutti gli altri nodi, esattamente come la matrice di incidenza nodo-arco della rete nella colonna relativa proprio all'arco (i,j).

Esempio

Rete $G = (V, A)$ con $V = \{1, 2, 3, 4, 5\}$ e

$$A = \{(1,2); (1,3); (1,5); (2,3); (3,4); (4,2); (4,5); (5,3)\}$$

(i,j)	$(1,2)$	$(1,3)$	$(1,5)$	$(2,3)$	$(3,4)$	$(4,2)$	$(4,5)$	$(5,3)$
c_{ij}	5	-2	2	-4	0	6	3	4

i	1	2	3	4	5
b_i	2	5	1	-4	-4

Modello matematico dell'esempio

$$\min\ 5x_{12} - 4x_{23} + 6x_{42} - 2x_{13} + 0x_{34} + 2x_{15} + 4x_{53} + 3x_{45}$$

$$x_{12} + x_{13} + x_{15} = 2$$

$$x_{23} - x_{12} - x_{42} = 5$$

$$x_{34} - x_{13} - x_{23} - x_{53} = 1$$

$$x_{42} + x_{45} - x_{34} = -4$$

$$x_{53} - x_{15} - x_{45} = -4$$

$$x_{12}, x_{23}, x_{42}, x_{13}, x_{34}, x_{15}, x_{53}, x_{45} \geq 0 \text{ interi}$$

Matrice dei vincoli

$$\mathbf{A} = \begin{array}{c|cccccccc} & (1,2) & (1,3) & (1,5) & (2,3) & (4,2) & (3,4) & (5,3) & (4,5) \\ \hline \mathbf{1} & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{2} & -1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ \mathbf{3} & 0 & -1 & 0 & -1 & 0 & 1 & -1 & 0 \\ \mathbf{4} & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 \\ \mathbf{5} & 0 & 0 & -1 & 0 & 0 & 0 & 1 & -1 \end{array}$$

MIN-FLOW.MOD

INSIEMI

set $NODI$ **;**

set $ARCHI$ **within** $NODI$ **cross** $NODI$ **;**

PARAMETRI

param b{$NODI$}**;**

param c{$ARCHI$} **;**

param d{$ARCHI$} $>= 0$, **default** Infinity **;**

VARIABILI

var x{(i, j) **in** $ARCHI$} $>= 0$, $<=$ d[i,j], **integer** **;**

VINCOLI

subject to bilancio{i **in** $NODI$} **:** sum{j **in** $NODI$: (i,j) **in** $ARCHI$} x[i,j] - sum{j **in** $NODI$: (j,i) **in** $ARCHI$} x[j,i] = b[i] **;**

OBIETTIVO

minimize costo_totale **:** sum{(i,j) **in** $ARCHI$} c[i,j]*x[i,j] **;**

Parole chiave

- **within**: se dichiariamo

  set A **within** B **;**

  questo ci dice che l'insieme A è un sottinsieme di B (se al momento della definizione di A e B questa condizione non fosse soddisfatta, verrebbe segnalato un errore);
- **cross**: "A **cross** B" è il prodotto cartesiano degli insiemi A e B, ovvero l'insieme di tutte le coppie con il primo elemento in A e il secondo in B;
- **diff**: "A **diff** B" rappresenta la differenza tra l'insieme A e l'insieme B, ovvero l'insieme che contiene tutti gli elementi che sono in A ma non in B.
- **default**: utilizzata nella definizione dei parametri per indicarne i valori di default: per tutti i casi in cui non si specifica il valore di un parametro questo viene fissato al valore di default.

Nota bene - I

sum{j **in** $NODI$: (i,j) **in** $ARCHI$} x[i,j]

La somma viene eseguita su tutti gli elementi che soddisfano una determinata condizione introdotta da un **:**

È possibile che le condizioni da soddisfare siano anche più di una. In tal caso le si elenca sempre dopo i **:** separate tra loro dalla parola chiave **and**.

Nota bene - II

La frequente occorrenza di problemi di flusso nella pratica ha spinto a inserire in AMPL una sintassi speciale per tali problemi. Non ci addentreremo in questa ma si rimandano gli interessati alla lettura del manuale.

MIN-FLOW.DAT

INSIEMI

set $NODI$ **:=** n1 n2 n3 n4 n5 **;**

set $ARCHI$ **:=** (n1,n2) (n1,n3) (n1,n5) (n2,n3) (n3,n4) (n4,n2) (n4,n5) (n5,n3) **;**

PARAMETRI

param b **:=**

n1 2

n2 5

n3 1

n4 -4

n5 -4

;

```
param c :=
n1 n2 5
n1 n3 -2
n1 n5 2
n2 n3 -4
n3 n4 0
n4 n2 6
n4 n5 3
n5 n3 4 ;
```

Si noti che, non essendo specificate le capacità sugli archi, queste saranno tutte pari al valore di default ∞.

Flusso massimo

Simile al flusso a costo minimo ma con le seguenti differenze:

- l'azienda ha un solo centro di produzione e uno solo di distribuzione, tutti gli altri centri sono magazzini;
- non abbiamo costi di trasporto unitari lungo le linee di collegamento ma abbiamo sempre limiti di capacità su di esse;
- la quantità di prodotto realizzata nell'unico centro di produzione (equivalente a quella ricevuta dall'unico centro di distribuzione) non è fissata a priori ma è variabile.

Descrizione problema

Stabilire qual è la quantità massima di prodotto che può essere realizzata dal centro di produzione, instradata attraverso la rete di collegamento e fatta giungere al centro di distribuzione, tenendo conto dei limiti di capacità sulle linee di collegamento.

Naturalmente anche qui si può pensare ad altri contesti applicativi per tale problema, quali reti di computer e reti idrauliche come nel problema di flusso a costo minimo.

Problema su grafi

Anche qui possiamo svincolarci dai dettagli delle applicazioni rappresentando il problema tramite un grafo, costruito in modo identico a quello del problema di flusso a costo minimo.

In tale grafo il singolo nodo sorgente verrà convenzionalmente indicato con S e il singolo nodo destinazione con D.

Nota bene

La restrizione a un singolo nodo sorgente e un singolo nodo destinazione può non essere vera in certe applicazioni (nulla impedisce che l'azienda abbia più centri di produzione e/o più centri di distribuzione).

Con un piccolo artificio possiamo sempre ricondurci al caso di un solo nodo sorgente e un solo nodo destinazione. Se sono presenti più sorgenti e/o destinazioni, introduciamo una sorgente fittizia collegata tramite archi fittizi a capacità infinita a ciascuna sorgente reale e, analogamente, una destinazione fittizia alla quale si giunge tramite archi fittizi a capacità infinita a partire da ciascuna destinazione reale.

Le sorgenti e destinazioni reali diventano a questo punto nodi di transito e rimangono solo la singola sorgente fittizia e la singola destinazione fittizia.

Modello matematico - variabili

Associamo ad ogni arco della rete $(i, j) \in A$ una variabile:

$$x_{ij} = \text{flusso inviato lungo l'arco } (i, j)$$

Tali variabili saranno vincolate ad essere non negative (non ha senso parlare di un flusso negativo). Se indichiamo con d_{ij} la capacità dell'arco (i, j) si dovrà anche avere

$$x_{ij} \leq d_{ij} \quad \forall\, (i, j) \in A.$$

Per prodotti non frazionabili le variabili possono assumere solo valori interi.

Modello matematico - vincoli

I vincoli sono quelli di equilibrio nei nodi intermedi, che sono tutti nodi di transito:

$$\underbrace{\sum_{j:\ (k,j)\in A} x_{kj}}_{\text{flusso uscente da } k} = \underbrace{\sum_{j:\ (j,k)\in A} x_{jk}}_{\text{flusso entrante in } k} \qquad \forall\, k \in V \setminus \{S, D\}.$$

Modello matematico - obiettivo

Il nostro obiettivo è quello di massimizzare la quantità di flusso inviato dal nodo sorgente S e quindi il flusso uscente dal nodo sorgente S:

$$\sum_{j:\ (S,j)\in A} x_{Sj}$$

Equivalentemente, possiamo massimizzare il flusso entrante nel nodo destinazione D:

$$\sum_{j:\ (j,D)\in A} x_{jD}$$

(le due quantità sono uguali).

Modello matematico completo

$$
\begin{array}{lll}
\max & \sum_{j:\ (S,j)\in A} x_{Sj} & \\
& \sum_{j:\ (k,j)\in A} x_{kj} = \sum_{j:\ (j,k)\in A} x_{jk} & \forall\ k \in V \setminus \{S, D\} \\
& 0 \leq x_{ij} \leq d_{ij} & \forall\ (i,j) \in A
\end{array}
$$

Osservazione

La matrice dei vincoli di uguaglianza del problema di flusso massimo coincide con la matrice di incidenza nodo-arco della rete senza le righe relative ai nodi S e D.

Esempio

È data la rete con le seguenti capacità sugli archi:

Arco	Capacità	Arco	Capacità
$(S,1)$	3	$(2,3)$	1
$(S,2)$	2	$(2,4)$	1
$(1,3)$	1	$(3,D)$	1
$(1,4)$	4	$(4,D)$	7

Modello matematico per l'esempio

$$
\begin{array}{ll}
\max & x_{S1} + x_{S2} \\
& x_{13} + x_{14} - x_{S1} = 0 \\
& x_{23} + x_{24} - x_{S2} = 0 \\
& x_{3D} - x_{13} - x_{23} = 0 \\
& x_{4D} - x_{14} - x_{24} = 0 \\
& 0 \leq x_{S1} \leq 3 \quad 0 \leq x_{S2} \leq 2 \\
& 0 \leq x_{13} \leq 1 \quad 0 \leq x_{14} \leq 4 \\
& 0 \leq x_{23} \leq 1 \quad 0 \leq x_{24} \leq 1 \\
& 0 \leq x_{3D} \leq 1 \quad 0 \leq x_{4D} \leq 7 \\
& x_{S1}, x_{S2}, x_{13}, x_{14}, x_{23}, x_{24}, x_{3D}, x_{4D} \ \text{interi}
\end{array}
$$

Matrice dei vincoli di equilibrio

La matrice nell'esempio è la seguente:

	$(S,1)$	$(S,2)$	$(1,3)$	$(1,4)$	$(2,3)$	$(2,4)$	$(3,D)$	$(4,D)$
1	-1	0	1	1	0	0	0	0
2	0	-1	0	0	1	1	0	0
3	0	0	-1	0	-1	0	1	0
4	0	0	0	-1	0	-1	0	1

MAX-FLOW.MOD

INSIEMI

set $NODI$ **;**

set $ARCHI$ **within** $NODI$ **cross** $NODI$ **;**

PARAMETRI

param Sorgente **symbolic in** {$NODI$}**;**

param Destinazione **symbolic in** {$NODI$} , **!=** Sorgente **;**

param d{$ARCHI$} $>= 0$, **default** Infinity **;**

VARIABILI

var x{(i, j) **in** $ARCHI$} $>= 0$, $<=$ d[i,j], **integer** **;**

VINCOLI

subject to equilibrio{i **in** $NODI$ **diff** {Sorgente, Destinazione}} **:** sum{j **in** $NODI$: (i, j) **in** $ARCHI$} x[i,j] - sum{j **in** $NODI$: (j, i) **in** $ARCHI$} x[j,i] = 0 **;**

OBIETTIVO

maximize flusso_uscente **:** sum{j **in** $NODI$: (Sorgente,j) **in** $ARCHI$} x[Sorgente,j] **;**

Difficoltà

Per definire i vincoli e l'obiettivo del problema abbiamo bisogno di identificare un nodo sorgente e uno destinazione.

Ma al momento della stesura del modello non sappiamo quale nodo della rete sarà quello sorgente e quale quello destinazione (non solo, tali nodi saranno ovviamente diversi in varie istanze del problema).

Più precisamente, al momento della stesura del modello non conosciamo proprio quali siano i nodi delle rete che verranno specificati solo nel file di dati.

Per aggirare la difficoltà

param Sorgente **symbolic in** $\{NODI\}$**;**
param Destinazione **symbolic in** $\{NODI\}$, **!=** Sorgente ;

Con questa dichiarazione diciamo che *esiste* un elemento dell'insieme $NODI$ (**in** $NODI$) che identifichiamo con il simbolo (**symbolic**) Sorgente e un elemento dell'insieme $NODI$ che identifichiamo con il simbolo Destinazione diverso (**!=**) da Sorgente.

Quali saranno gli effettivi nodi sorgente e destinazione verrà specificato nel file di dati.

MAX-FLOW.DAT

```
### INSIEMI ###

set NODI := S n1 n2 n3 n4 D ;
set ARCHI := (S,n1) (S,n2) (n1,n3) (n1,n4) (n2,n3) (n2,n4) (n3,D) (n4,D) ;

### PARAMETRI ###

param Sorgente := S ;
param Destinazione := D ;
```

```
param d :=
S n1 3
S n2 3
n1 n3 1
n1 n4 4
n2 n3 1
n2 n4 1
n3 D 1
n4 D 7 ;
```

Problema del trasporto

Supponiamo di avere m *depositi* in cui è immagazzinato un prodotto e n negozi che richiedono tale prodotto.

Nel deposito i è immagazzinata la quantità a_i di prodotto. Nel negozio j si richiede la quantità b_j di prodotto.

È noto che il costo di trasporto di un'unità di prodotto dal deposito i al negozio j è pari a c_{ij}.

Il problema del trasporto consiste nel determinare quale quantità di prodotto inviare da ciascun deposito verso ciascun negozio in modo tale da minimizzare il costo complessivo di trasporto, rispettando i vincoli sulle quantità di prodotto presenti in ciascun deposito e quelli di richieste di ciascun negozio.

Ipotesi iniziale

Si suppone sempre che la quantità totale immagazzinata in tutti i depositi sia pari alla quantità totale richiesta da tutti i magazzini, ovvero

$$\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j .$$

Non si tratta comunque di un'ipotesi restrittiva dal momento che ci si può sempre ricondurre ad essa.

Infatti ...

...si supponga che

$$\sum_{i=1}^{m} a_i > \sum_{j=1}^{n} b_j.$$

cioè nei depositi vi sia più prodotto di quanto effettivamente richiesto dai negozi. Per soddisfare l'ipotesi basta aggiungere un negozio fittizio $n+1$ con

$$b_{n+1} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n} b_j,$$

e con $c_{i,n+1} = 0$ per ogni $i,\ i = 1, \ldots, m$, cioè il costo del trasporto verso il negozio fittizio è pari a 0. La quantità di prodotto che un deposito invia a un negozio fittizio resta in realtà immagazzinata nel deposito.

Analogamente ...

... si supponga che

$$\sum_{i=1}^{m} a_i < \sum_{j=1}^{n} b_j.$$

cioè nei depositi vi sia meno prodotto di quanto effettivamente richiesto dai negozi. Per soddisfare l'ipotesi basta aggiungere un deposito fittizio $m+1$ con

$$a_{m+1} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i,$$

e con $c_{m+1,j} = 0$ per ogni $j,\ j = 1, \ldots, n$, cioè il costo del trasporto dal deposito fittizio è pari a 0. La quantità di prodotto che un negozio riceve da un deposito fittizio equivale in realtà a una richiesta non soddisfatta per quel negozio.

Grafo bipartito associato

Possiamo associare al problema un grafo bipartito completo $K_{m,n}$ dove su un lato della bipartizione compaiono i nodi corrispondenti ai depositi, numerati da 1 a m, mentre sull'altro lato della bipartizione compaiono i nodi corrispondenti ai negozi, numerati da $m+1$ a $m+n$

Esempio

	Negozio 1	Negozio 2	Negozio 3	Disponibilità
Deposito 1	4	7	5	30
Deposito 2	2	4	3	20
Richiesta	15	10	25	

Modello matematico - variabili

Ad ogni coppia deposito i-negozio j associamo una variabile:

$x_{ij} =$ quantità di prodotto inviata dal deposito i verso il negozio j.

Tale quantità dovrà essere ovviamente non negativa e tipicamente intera (non frazionabilità del prodotto trasportato), ovvero:

$$x_{ij} \geq 0 \quad \text{intero} \quad i = 1, \ldots, m \;\; j = 1, \ldots, n.$$

Modello matematico - vincoli depositi

Per ogni deposito i la quantità totale di prodotto inviata da esso deve essere pari alla quantità di prodotto a_i in esso immagazzinata. La quantità totale di prodotto inviata dal deposito i è data dalla formula $\sum_{j=1}^{n} x_{ij}$ e quindi per ogni deposito i avremo il seguente vincolo:

$$\sum_{j=1}^{n} x_{ij} = a_i \quad i = 1, \ldots, m.$$

Nell'esempio

Deposito 1:

$$x_{11} + x_{12} + x_{13} = 30$$

Deposito 2:

$$x_{21} + x_{22} + x_{23} = 20$$

Modello matematico - vincoli negozi

Per ogni negozio j la quantità totale di prodotto ricevuta da esso deve essere pari alla quantità di prodotto b_j da esso richiesta. La quantità totale di prodotto ricevuta dal negozio j è data dalla formula $\sum_{i=1}^{m} x_{ij}$ e quindi per ogni negozio j avremo il seguente vincolo:

$$\sum_{i=1}^{m} x_{ij} = b_j \quad j = 1, \ldots, n.$$

Nell'esempio

Negozio 1:

$$x_{11} + x_{21} = 15$$

Negozio 2:

$$x_{12} + x_{22} = 10$$

Negozio 3:

$$x_{13} + x_{23} = 25$$

Modello matematico - obiettivo

Se inviare un'unità di prodotto dal deposito i al negozio j ha costo pari a c_{ij}, inviarne una quantità x_{ij} ha costo pari a $c_{ij}x_{ij}$. Sommando su tutte le possibili coppie deposito-negozio, abbiamo la seguente formula per l'obiettivo:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}.$$

Nell'esempio:

$$4x_{11} + 7x_{12} + 5x_{13} + 2x_{21} + 4x_{22} + 3x_{23}$$

Modello matematico finale

$$\begin{array}{lll}
\min & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & \\
& \sum_{j=1}^{n} x_{ij} = a_i & i = 1,\ldots,m \\
& \sum_{i=1}^{m} x_{ij} = b_j & j = 1,\ldots,n \\
& x_{ij} \geq 0 \ \text{interi} & i = 1,\ldots,m \ j = 1,\ldots,n
\end{array}$$

Nell'esempio

$$
\begin{array}{rr}
\min & 4x_{11} + 7x_{12} + 5x_{13} + 2x_{21} + 4x_{22} + 3x_{23} \\
 & x_{11} + x_{12} + x_{13} = 30 \\
 & x_{21} + x_{22} + x_{23} = 20 \\
 & x_{11} + x_{21} = 15 \\
 & x_{12} + x_{22} = 10 \\
 & x_{13} + x_{23} = 25 \\
 & x_{11}, x_{12}, x_{13}, x_{21}, x_{22}, x_{23} \geq 0 \ \text{ interi}
\end{array}
$$

Osservazione

La matrice dei vincoli di uguaglianza del problema del trasporto coincide con la matrice di incidenza nodo-arco del grafo bipartito completo associato al problema.

Nell'esempio

	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}
Dep. 1	1	1	1	0	0	0
Dep. 2	0	0	0	1	1	1
Neg. 1	1	0	0	1	0	0
Neg. 2	0	1	0	0	1	0
Neg. 3	0	0	1	0	0	1

TRASP.MOD

```
### INSIEMI ###
```

set $DEPOSITI$ **;**

set $NEGOZI$ **;**

```
### PARAMETRI ###
```

param $c\{DEPOSITI, NEGOZI\} >= 0$ **;**

param $a\{DEPOSITI\} >= 0$ **;**

param $b\{NEGOZI\} >= 0$ **;**

check : sum{i **in** $DEPOSITI$} a[i] = sum{j **in** $NEGOZI$} b[j] **;**

```
### VARIABILI ###
```

var $x\{DEPOSITI, NEGOZI\} >= 0$, **integer ;**

VINCOLI

subject to disp_depositi {i **in** $DEPOSITI$} **:** sum{j **in** $NEGOZI$} x[i,j] = a[i] **;**

subject to rich_negozi {j **in** $NEGOZI$} **:** sum{i **in** $DEPOSITI$} x[i,j] = b[j] **;**

OBIETTIVO

minimize costo_totale **:** sum{i **in** $DEPOSITI$, j **in** $NEGOZI$} c[i,j]*x[i,j] **;**

Nota bene

Nel modello notiamo l’uso di **check**: la condizione dopo **:** viene controllata e se non è soddisfatta viene segnalato un errore.

TRASP.DAT

INSIEMI

set $DEPOSITI$ **:=** D1 D2 **;**

set $NEGOZI$ **:=** N1 N2 N3 **;**

PARAMETRI

param a **:=**

D1 30

D2 20 **;**

param b **:=**

N1 15

N2 10

N3 25 **;**

param C **:**

	$N1$	$N2$	$N3$	**:=**
$D1$	4	7	5	
$D2$	2	4	3	**;**

Assegnamento

Siano dati due insiemi A e B entrambi di cardinalità n. Ad ogni coppia $(a_i, b_j) \in A \times B$ è associato un valore $d_{ij} \geq 0$ che misura la "incompatibilità" tra a_i e b_j, anche interpretabile come costo di assegnare a_i a b_j. Il problema di assegnamento è il seguente:

Individua n coppie di elementi appartenenti ad $A \times B$ in modo tale che ogni elemento di A e di B deve appartenere a una e una sola coppia e in modo tale da minimizzare la "incompatibilità" (o costo) totale, data dalla somma delle "incompatibilità" (o costi) di ogni singola coppia.

Possibile applicazione

Dati n lavoratori, n compiti ed un costo di assegnamento per ognuna delle n^2 coppie lavoratore-compito, individuare gli n assegnamenti di compiti ai lavoratori in modo da minimizzare i costi totali di assegnamento.

Regione ammissibile

La regione ammissibile è costituita da tutti i possibili accoppiamenti.

Per l'elemento a_1 ho n possibili scelte di accoppiamento con elementi dell'insieme B; per a_2 le scelte si riducono a $n-1$ (devo escludere l'elemento già accoppiato con a_1); per a_3 le scelte si riducono a $n-2$ (devo escludere gli elementi già accoppiati con a_1 e a_2); ...

In totale avrò quindi $n!$ diversi possibili accoppiamenti.

Osservazione

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

Questi casi possono **sempre** essere ricondotti al caso $\mid A \mid = \mid B \mid$ 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 delle incompatibilità ($n = 4$):

	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

Modello matematico - variabili

Ad ogni coppia $(a_i, b_j) \in A \times B$ si associa una variabile x_{ij} con i seguenti possibili valori:

$$x_{ij} = \begin{cases} 1 & \text{se } a_i \text{ è assegnato a } b_j \\ 0 & \text{altrimenti} \end{cases}$$

Modello matematico-vincoli

Ad ogni elemento a_i è assegnato uno ed un solo b_j e quindi avremo i seguenti n vincoli

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i \in \{1, \dots, n\}.$$

Ad ogni elemento b_j è assegnato uno ed un solo a_i e quindi avremo i seguenti n vincoli

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall\, j \in \{1, \dots, n\};$$

Nell'esempio

Vincolo per a_1:

$$x_{11} + x_{12} + x_{13} + x_{14} = 1$$

Analogo per gli altri a_i, $i = 2, 3, 4$.

Vincolo per b_1:

$$x_{11} + x_{21} + x_{31} + x_{41} = 1$$

Analogo per gli altri b_j, $j = 2, 3, 4$.

Modello matematico - obiettivo

Il contributo all'incompatibilità totale di (a_i, b_j) è d_{ij} se a_i viene assegnato a b_j, cioè se $x_{ij} = 1$, ed è 0 se a_i non viene assegnato a b_j, cioè se $x_{ij} = 0$. In entrambi i casi il contributo all'incompatibilità totale di (a_i, b_j) è $d_{ij}x_{ij}$.

Sommando su tutte le possibili coppie si ottiene l'obiettivo del problema:

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

Nell'esempio

$$2x_{11} + 3x_{12} + 4x_{13} + 5x_{14} + 6x_{21} + 2x_{22} + 2x_{23} + 2x_{24} +$$

$$7x_{31} + 2x_{32} + 3x_{33} + 3x_{34} + 2x_{41} + 3x_{42} + 4x_{43} + 5x_{44}$$

Modello matematico finale

$$
\begin{array}{lll}
\min & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} & \\
& \sum_{j=1}^{n} x_{ij} = 1 & \forall\, i \in \{1, \ldots, n\} \\
& \sum_{i=1}^{n} x_{ij} = 1 & \forall\, j \in \{1, \ldots, n\} \\
& x_{ij} \in \{0, 1\} & \forall\, i, j
\end{array}
$$

Osservazione

I vincoli $x_{ij} \in \{0,1\}$ possono essere sostituiti con i vincoli $0 \leq x_{ij} \leq 1$, x_{ij} intere, ottenendo quindi il seguente problema di PLI:

$$
\begin{array}{lll}
\min & \sum_{i=1}^n \sum_{j=1}^n d_{ij}x_{ij} & \\
& \sum_{j=1}^n x_{ij} = 1 & \forall\, i \in \{1,\ldots,n\} \\
& \sum_{i=1}^n x_{ij} = 1 & \forall\, j \in \{1,\ldots,n\} \\
& 0 \leq x_{ij} \leq 1 \ \ \text{intere} & \forall\, i,j
\end{array}
$$

Ma possiamo notare che:

$$
x_{ij} \geq 0, \quad \sum_{j=1}^n x_{ij} = 1 \ \ \forall\, i \in \{1,\ldots,n\} \ \ \Rightarrow \ \ x_{ij} \leq 1 \ \ \forall\, i,j \in \{1,\ldots,n\},
$$

il che significa che i vincoli $x_{ij} \leq 1$ sono del tutto inutili e possono quindi essere eliminati.

Quindi ...

... possiamo riscrivere il modello nella seguente forma:

$$
\begin{array}{lll}
\min & \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} & \\
& \sum_{j=1}^{n} x_{ij} = 1 & \forall\ i \in \{1, \ldots, n\} \\
& \sum_{i=1}^{n} x_{ij} = 1 & \forall\ j \in \{1, \ldots, n\} \\
& x_{ij} \geq 0 \ \ \text{intere} & \forall\ i, j
\end{array}
$$

Una volta riscritto in questo modo, notiamo che il problema di assegnamento è un caso particolare di problema del trasporto dove vi sono n depositi e n negozi, ogni deposito ha una sola unità di prodotto a disposizione e ogni negozio richiede una sola unità di prodotto.

ASSEGN.MOD

```
### INSIEMI ###

set A ;
set B ;

### PARAMETRI ###

param d{A, B} ;

check : card(A) = card(B) ;

### VARIABILI ###

var x{A, B} binary ;
```

```
### VINCOLI ###

subject to one_A {i in A} : sum{j in B} x[i,j] = 1 ;

subject to one_B {j in B} : sum{i in A} x[i,j] = 1 ;

### OBIETTIVO ###

minimize costo_totale : sum{i in A, j in B} d[i,j]*x[i,j] ;
```

Nota bene

Nel modello notiamo l'uso di **card** che, avendo come argomento un insieme, restituisce la cardinalità di tale insieme.

ASSEGN.DAT

```
### INSIEMI ###

set A := a1 a2 a3 a4 ;
set B := b1 b2 b3 b4 ;


### PARAMETRI ###

param d :
        b1  b2  b3  b4  :=
    a1  2   3   4   5
    a2  6   2   2   2
    a3  7   2   3   3
    a4  2   3   4   5   ;
```

Flusso multi-commodity

Quando abbiamo descritto il problema di flusso a costo minimo, abbiamo dato per scontato che nella rete viaggiasse un singolo tipo di prodotto.

In realtà in molte applicazioni si ha che lungo la rete, rappresentata tramite il grafo orientato $G = (V, A)$, viaggiano contemporaneamente più tipi di prodotto (multi-commodity).

Supponiamo di avere $r > 1$ di questi tipi e che per ogni tipo $k \in \{1, \ldots, r\}$ di prodotto vi sia un unico nodo sorgente, indicato con S_k, con valore associato b_k e un unico nodo destinazione D_k con valore associato $-b_k$.

Anche qui avremo un costo di trasporto unitario c_{ij} lungo ogni arco $(i, j) \in A$ e una capacità d_{ij} associata allo stesso arco.

Modello matematico - variabili

Rispetto al problema di flusso a costo minimo, ora le variabili avranno un indice in più relativo al tipo di prodotto:

$$x_{ij}^k = \text{quantità di prodotto } k \text{ inviata lungo l'arco } (i,j)$$

ovviamente sono vincolate a essere non negative e anche intere, nel caso il prodotto k non sia frazionabile.

Modello matematico - vincoli

Come vincoli dovremo includere quelli tra differenza del flusso in uscita e in entrata, distinguendo tra i flussi dei diversi tipi di prodotti:

$$\sum_{h:(i,h)\in A} x_{ih}^k - \sum_{h:(h,i)\in A} x_{hi}^k = \begin{cases} b_k & \text{se } i \equiv S_k \\ -b_k & \text{se } i \equiv D_k \\ 0 & \text{altrimenti} \end{cases}$$

per ogni $i \in V$ e per ogni $k \in \{1, \ldots, r\}$.

Avremo poi i vincoli sulla capacità degli archi:

$$\sum_{k=1}^{r} x_{ij}^k \leq d_{ij} \quad \forall\ (i,j) \in A.$$

Modello matematico - obiettivo

L'obiettivo, con le opportune modifiche rispetto al problema di flusso a costo minimo, sarà il seguente:

$$\sum_{(i,j)\in A} [c_{ij} \sum_{k=1}^{r} x_{ij}^k]$$

dove la sommatoria più interna rappresenta la quantità totale (su tutti gli r prodotti) di merce trasportata lungo l'arco (i,j). Naturalmente l'obiettivo sarà da minimizzare.

Nel caso in cui anche il costo unitario di trasporto dipenda dal tipo di prodotto k trasportato (c_{ij}^k), potremo modificare l'obiettivo in questo modo:

$$\sum_{(i,j)\in A} \sum_{k=1}^{r} c_{ij}^k x_{ij}^k.$$

Modello matematico completo

$$
\begin{array}{ll}
\min & \sum_{(i,j)\in A} [c_{ij} \sum_{k=1}^{r} x_{ij}^k] \\
 & \sum_{h:(i,h)\in A} x_{ih}^k - \sum_{h:(h,i)\in A} x_{hi}^k = \begin{cases} b_k & \text{se } i \equiv S_k \\ -b_k & \text{se } i \equiv D_k \\ 0 & \text{altrimenti} \end{cases} \qquad \forall\, i \in V,\ \forall\, k \in \{1,\ldots,r\} \\
 & \sum_{k=1}^{r} x_{ij}^k \leq d_{ij} \qquad \forall\, (i,j) \in A \\
 & x_{ij}^k \geq 0 \ \ \text{interi} \qquad \forall\, (i,j) \in A,\ \forall\, k \in \{1,\ldots
\end{array}
$$

MULTI-COMMODITY.MOD

INSIEMI

set $NODI$ **;**

set $ARCHI$ **within** $NODI$ **cross** $NODI$ **;**

set $PRODOTTI$ **;**

PARAMETRI

param Sorgente{$PRODOTTI$} **symbolic in** $NODI$ **;**

param Destinazione{k **in** $PRODOTTI$} **symbolic in** $NODI$, **!=** Sorgente[k] **;**

param b{$PRODOTTI$} $>= 0$ **;**

param c{$ARCHI$} **;**

param d{$ARCHI$} $>= 0$, **default** Infinity **;**

VARIABILI

var x{$ARCHI$, $PRODOTTI$} $>= 0$, **integer ;**

VINCOLI

subject to equilibrio{j **in** $NODI$, k **in** $PRODOTTI$ **:** j **!=** Sorgente[k] **and** j **!=** Destinazione[k] } **:** sum{h **in** $NODI$: (j, h) **in** $ARCHI$} x[j,h,k] - sum{h **in** $NODI$: (h, j) **in** $ARCHI$} x[h,j,k] = 0 ;

subject to bilancio_sorgente{k **in** $PRODOTTI$ } **:** sum{h **in** $NODI$: (Sorgente[k],h) **in** $ARCHI$} x[Sorgente[k],h,k] - sum{h **in** $NODI$: (h,Sorgente[k]) **in** $ARCHI$} x[h,Sorgente[k],k] = b[k] ;

subject to bilancio_destinazione{k **in** $PRODOTTI$ } **:** sum{h **in** $NODI$: (Destinazione[k],h) **in** $ARCHI$} x[Destinazione[k],h,k] - sum{h **in** $NODI$: (h,Destinazione[k]) **in** $ARCHI$} x[h,Destinazione[k],k] = -b[k] **;**

subject to cap_max{(i, j) **in** $ARCHI$ } **:** sum{k **in** $PRODOTTI$} x[i,j,k] $<=$ d[i,j] **;**

OBIETTIVO

minimize costo_totale **:** sum{(i, j) **in** $ARCHI$} c[i,j]*(sum{k **in** $PRODOTTI$} x[i,j,k]) **;**

Commesso viaggiatore

In una determinata zona abbiamo n centri con distanze d_{ij} tra ogni coppia di centri i e j, $i \neq j$. Un commesso viaggiatore deve partire da uno di questi centri (supporremo nel seguito sia il centro 1) e visitare gli altri $n-1$ prima di tornare a quello di partenza.

Quello che vorrebbe il commesso viaggiatore è individuare la sequenza di visita dei centri che renda minima la distanza che complessivamente percorre.

Il problema verrà indicato più avanti anche come problema TSP (Travelling Salesman Problem).

Su grafi

Costruiamo un grafo completo i cui nodi corrispondano agli n centri.

Gli archi indicano i collegamenti tra i centri e potranno essere non orientati se le distanze tra centri sono simmetriche, ovvero per ogni i, j si ha $d_{ij} = d_{ji}$, oppure orientati se non c'è simmetria nelle distanze.

Il problema sul grafo consiste nell'individuare tra tutti i circuiti hamiltoniani quello con distanza complessiva minima.

Altro ambito applicativo

Sequenziamento di una serie di n operazioni che devono essere eseguite su una certa macchina.

Il passaggio tra due operazioni eseguite una dietro l'altra non avviene in modo istantaneo ma richiede un certo tempo di setup, indicato con d_{ij}, per passare dalla configurazione della macchina per eseguire l'operazione i a quella per eseguire l'operazione j.

Ciò che si desidera è stabilire la sequenza in cui eseguire le n operazioni (ipotizzando che queste vengano svolte ciclicamente e che quindi una volta raggiunta l'ultima si debba ripartire dalla prima) in modo tale da minimizzare i tempi totali di setup.

Modello matematico - variabili

Associamo ad ogni arco $(i, j) \in A$ la variabile x_{ij} che potrà assumere i seguenti valori:

$$x_{ij} = \begin{cases} 1 & \text{se l'arco } (i, j) \text{ fa parte del circuito hamiltoniano} \\ 0 & \text{altrimenti} \end{cases}$$

Quindi $x_{ij} \in \{0, 1\}$, ovvero è binaria.

Vincoli - I

In ogni nodo c'è esattamente un arco del circuito che entra nel nodo ed esattamente uno che esce dal nodo. Avremo quindi i seguenti due insiemi di vincoli:

- Per ogni nodo $j \in V$:

  $$\sum_{i \in V,\ i \neq j} x_{ij} = 1,$$

  ovvero il circuito può contenere uno ed un solo arco entrante in j.
- Per ogni nodo $j \in V$:

  $$\sum_{i \in V,\ i \neq j} x_{ji} = 1,$$

  ovvero il circuito può contenere uno ed un solo arco uscente da j.

Esempio

In un grafo completo con 4 nodi avremo i seguenti 8 vincoli

$$x_{12} + x_{13} + x_{14} = 1 \quad x_{21} + x_{23} + x_{24} = 1$$

$$x_{31} + x_{32} + x_{34} = 1 \quad x_{41} + x_{42} + x_{43} = 1$$

$$x_{21} + x_{31} + x_{41} = 1 \quad x_{12} + x_{32} + x_{42} = 1$$

$$x_{13} + x_{23} + x_{43} = 1 \quad x_{14} + x_{24} + x_{34} = 1.$$

Nota bene

Tutti i circuiti hamiltoniani soddisfano questi vincoli.

Ma è vero anche il viceversa? Cioè è vero che tutte le soluzioni che soddisfano tali vincoli rappresentano circuiti hamiltoniani?

La risposta è no e lo si mostra attraverso il nostro esempio.

Nell'esempio

Prendiamo la soluzione

$$x_{12} = x_{21} = x_{34} = x_{43} = 1$$
$$x_{13} = x_{31} = x_{14} = x_{41} = x_{23} = x_{32} = x_{24} = x_{42} = 0.$$

Questa non rappresenta un circuito hamiltoniano ma si può verificare che soddisfa i vincoli introdotti (in ogni nodo entra ed esce un solo arco).

Ciò si verifica per la presenza di sottocircuiti (nell'esempio $1 \to 2 \to 1$ e $3 \to 4 \to 3$).

Quindi ...

... per avere un insieme di vincoli che rappresenta tutti e soli i circuiti hamiltoniani dobbiamo aggiungere ai vincoli già introdotti altri vincoli.

Vedremo due gruppi di vincoli che consentono di scartare solo i sottocircuiti, senza essere violati dai circuiti hamiltoniani.

Prima classe di vincoli

Associamo a ogni nodo i diverso dal nodo di partenza ($i \neq 1$) una variabile y_i a valori interi compresi tra 1 e $n - 1$.

Questa variabile indica la posizione del nodo i nella sequenza di visita dei nodi.

Per il nodo di partenza 1 tale posizione è fissata a 0, ovvero porremo sempre $y_1 = 0$.

Continua

Se nel circuito è presente l'arco (i, j), il nodo j segue il nodo i nel ciclo e quindi la posizione di j deve essere di almeno un'unità superiore a quella di i, ovvero avremo il seguente vincolo logico:

$$x_{ij} = 1 \quad \Rightarrow \quad y_j \geq y_i + 1 \quad \forall\, i \neq j, \quad j \neq 1.$$

A questo punto i sottocircuiti che non partono dal nodo 1 sono esclusi.

Nell'esempio

Se pensiamo alla soluzione con $x_{34} = x_{43} = 1$ e quindi con il sottocircuito (che non parte dal nodo 1)

$$3 \rightarrow 4 \rightarrow 3,$$

dai vincoli introdotti dobbiamo avere $y_4 \geq y_3 + 1$ e $y_3 \geq y_4 + 1$ che non possono essere soddisfatti contemporaneamente.

Nota bene

Notiamo che i vincoli *non* vengono imposti per $j = 1$ (altrimenti questi vincoli ci farebbero escludere non solo i sottocircuiti che non partono dal nodo 1, ma anche i circuiti hamiltoniani che partono dal nodo 1).

Vincoli logici $\rightarrow$ disequazioni

Per imporre l'implicazione, tenuto conto che abbiamo anche i vincoli

$$1 \leq y_i \leq n-1 \quad i=2,\ldots,n, \quad y_1=0$$

possiamo utilizzare vincoli di questo tipo:

$$y_j - y_i \geq x_{ij} + (1-n)(1-x_{ij}).$$

Infatti se $x_{ij}=1$ abbiamo esattamente il vincolo richiesto nell'implicazione, mentre per $x_{ij}=0$ abbiamo il vincolo

$$y_j - y_i \geq 1-n$$

ridondante alla luce dei limiti sulle y_i.

Modello matematico completo - I

$$
\begin{array}{lll}
\min & \sum_{i \neq j} d_{ij} x_{ij} & \\
& \sum_{i \in V,\ i \neq j} x_{ij} = 1 & \forall\ j \in V \\
& \sum_{i \in V,\ i \neq j} x_{ji} = 1 & \forall\ j \in V \\
& y_j - y_i \geq x_{ij} + (1-n)(1-x_{ij}) & \forall\ i \neq j,\ j \neq 1 \\
& x_{ij} \in \{0,1\} & \forall\ i,j \in \{1,\ldots,n\},\ i \neq j \\
& 1 \leq y_i \leq n-1 & i = 2,\ldots,n \\
& y_1 = 0 &
\end{array}
$$

Alternativa

I vincoli introdotti possono essere riadattati in modo tale da tener conto di altre richieste nel problema.

Per esempio, supponiamo che a un arco (i, j) sia associato, oltre alla distanza d_{ij}, anche un tempo di percorrenza T_{ij} e che esistano per ogni nodo $i \neq 1$ delle finestre temporali $[Am_i, AM_i]$ all'interno delle quali i nodi devono essere visitati.

In tal caso possiamo definire y_i non come *posizione del nodo i nella sequenza* ma come *istante in cui il nodo i viene visitato*.

Allora ...

... il vincolo logico viene modificato nel modo seguente

$$x_{ij} = 1 \quad \Rightarrow \quad y_j - y_i \geq T_{ij} \quad \forall\, i \neq j, \quad j \neq 1$$

esprimibile tramite la seguente disequazione

$$y_j - y_i \geq T_{ij} x_{ij} - (T_{\max} - T_{\min})(1 - x_{ij}),$$

dove, ad esempio, si può porre

$$T_{\min} = \min_{i \neq j} T_{ij} \quad T_{\max} = n \max_{i \neq j} T_{ij}.$$

Ora, per imporre la visita del nodo i nella finestra temporale data, è sufficiente introdurre il vincolo

$$Am_i \leq y_i \leq AM_i \quad i = 2, \ldots, n.$$

Seconda classe di vincoli

Per eliminare la formazione di sottocircuiti possiamo usare anche questi vincoli:

$$\forall\, U \subseteq V:\ 2 \leq \mid U \mid \leq \mid V \mid -2 \qquad \sum_{i \in U,\ j \in V \setminus U} x_{ij} \geq 1,$$

che richiedono che per ogni possibile partizione di V in due sottinsiemi (ciascuno di cardinalità almeno pari a 2), deve esserci almeno un arco che va da un sottinsieme all'altro.

Questo esclude tutti i possibili sottocircuiti.

Nell'esempio

$$U = \{1,2\} \quad \rightarrow \quad x_{13} + x_{14} + x_{23} + x_{24} \geq 1$$

(in particolare, questo vincolo esclude l'assegnamento di variabili precedente)

$$U = \{1,3\} \quad \rightarrow \quad x_{12} + x_{14} + x_{32} + x_{34} \geq 1$$

$$U = \{1,4\} \quad \rightarrow \quad x_{12} + x_{13} + x_{42} + x_{43} \geq 1$$

$$U = \{2,3\} \quad \rightarrow \quad x_{21} + x_{24} + x_{31} + x_{34} \geq 1$$

$$U = \{2,4\} \quad \rightarrow \quad x_{21} + x_{23} + x_{41} + x_{43} \geq 1$$

$$U = \{3,4\} \quad \rightarrow \quad x_{31} + x_{32} + x_{41} + x_{42} \geq 1$$

In realtà ...

... metà di questi vincoli è ridondante: dati due sottinsiemi tra loro complemementari, è sufficiente mettere il vincolo relativo ad uno solo di essi (ad esempio possiamo mettere uno solo dei due vincoli relativi a $U = \{1,2\}$ e $U = \{3,4\}$).

Modello matematico completo - II

$$
\begin{array}{lll}
\min & \sum_{i,j\in V,\ i\neq j} d_{ij}x_{ij} & \\
& \sum_{i\in V,\ i\neq j} x_{ij} = 1 & \forall\ j \in V \\
& \sum_{i\in V,\ i\neq j} x_{ji} = 1 & \forall\ j \in V \\
& \sum_{i\in U,\ j\in V\setminus U} x_{ij} \geq 1 & \forall\ U \subseteq V:\ 2 \leq\mid U \mid\leq\mid V \mid /2 \\
& x_{ij} \in \{0,1\} & \forall\ i,j \in \{1,\ldots,n\},\ i \neq j
\end{array}
$$

Nota bene

Da un lato quest'ultimo modello contiene un numero di vincoli estremamente elevato. Infatti il numero di vincoli è $O(2^{|V|})$ e dunque esponenziale, mentre il numero di vincoli della prima classe è $O(| V |^2)$.

D'altro lato però si può anche vedere che, in un senso che potremo chiarire solo più avanti, il modello basato sulla seconda classe di vincoli è molto più preciso di quello basato sulla prima.

TSP.MOD (prima classe vincoli)

INSIEMI

set $NODI$ **ordered ;**

set $ARCHI$ **within** $(NODI$ **cross** $NODI)$ **;**

PARAMETRI

param n **:= card**$(NODI)$ **;**

param $d\{ARCHI\}$ **default** 100000000 **;**

VARIABILI

var $x\{ARCHI\}$ **binary ;**

var $y\{NODI\} >= 0, <= n$-1, **integer ;**

VINCOLI

subject to ingresso{i **in** $NODI$} **:** sum{j **in** $NODI : j$!= i} x[j,i] = 1 **;**

subject to uscita{i **in** $NODI$} **:** sum{j **in** $NODI : j$!= i} x[i,j] = 1 **;**

subject to sequenza{(i,j) **in** $ARCHI : j$ **!= first**($NODI$)} **:** y[j]-y[i] $>=$ n*x[i,j]+1-n **;**

subject to nodo_partenza **:** y[**first**($NODI$)]=0 **;**

OBIETTIVO

minimize distanza_totale **:** sum{(i,j) **in** $ARCHI$} d[i,j]*x[i,j] **;**

Nota bene

Facciamo alcune osservazioni sul modello. L'insieme $NODI$ viene dichiarato **ordered.** L'ordine sarà definito da quello con cui vengono specificati i nodi nel file .DAT.

In questo modo nel modello possiamo fare riferimento al primo elemento dell'insieme (nel nostro caso il nodo di partenza del circuito) che è semplicemente identificato da **first**($NODI$).

Notiamo la dichiarazione dell'insieme $ARCHI$ che coincide con il prodotto cartesiano dell'insieme $NODI$ per se stesso.

Il default 100000000 per il valore d degli archi va inteso come valore abbastanza grande da escludere a priori la presenza dell'arco in una soluzione ottima del problema.

Instradamento di veicoli

Alcuni problemi sono strettamente collegati a quello del commesso viaggiatore ma richiedono alcune modifiche rispetto a esso.

Pensiamo al caso in cui si abbia a disposizione non un solo commesso viaggiatore ma fino a un numero massimo N di commessi viaggiatori (o, se pensiamo al sequenziamento di operazioni, non una singola macchina ma fino a N macchine uguali).

Supporremo che per tutti i commessi viaggiatori il nodo di partenza sia il nodo 1.

Modello

Anche qui possiamo usare le variabili binarie x_{ij} e intere y_i (con $y_1 = 0$ e le altre incluse tra 1 e $n-1$) già viste per il commesso viaggiatore.

Potremo ancora utilizzare i vincoli già visti per le variabili y_i.

Per i nodi diversi dal nodo di partenza 1 avremo sempre la richiesta di un solo arco entrante e uno solo uscente:

$$\left\{\begin{array}{l} \sum_{i \neq j} x_{ij} = 1 \\ \\ \sum_{i \neq j} x_{ji} = 1 \end{array}\right. \quad \forall\ j \neq 1$$

L'unica differenza ...

... rispetto al problema del commesso viaggiatore è che non avremo i vincoli appena visti di un solo arco entrante e uscente per quel che riguarda il nodo di partenza 1.

Per tale nodo avremo la richiesta che il numero di archi complessivamente entranti deve essere pari a quello degli uscenti:

$$\sum_{i \neq 1} x_{i1} = \sum_{i \neq 1} x_{1i}$$

(ovvero il numero di commessi viaggiatori che ritornano nel nodo 1 deve essere esattamente pari a quello dei viaggiatori che sono partiti da tale nodo).

Entrambe queste quantità dovranno essere non superiori a N:

$$\sum_{i \neq 1} x_{i1} \leq N$$

(ovvero non possono uscire ed entrare dal nodo 1 più di N commessi viaggiatori).

L'obiettivo non cambia rispetto al commesso viaggiatore tradizionale.

Variante interessante

Abbiamo dei vincoli di capacità (di merce trasportabile) per un commesso viaggiatore (si pensi al caso in cui il commesso viaggiatore utilizzi un veicolo a capacità limitata per trasportare le merci).

Supporremo che in ogni centro $i \neq 1$ il commesso viaggiatore carichi una quantità Car_i di merce.

Variabili y_i

Possiamo definire le variabili y_i in questo modo:

$y_i =$ quantità di merce presente nel veicolo al nodo i.

Su tali variabili imporremo i vincoli

$$0 \leq y_i \leq C,$$

dove C è la capacità massima di un veicolo.

Avremo quindi i vincoli logici

$$x_{ij} = 1 \quad \Rightarrow \quad y_j - y_i \geq Car_j$$

Vincoli logici $\rightarrow$ disequazioni

$$y_j - y_i \geq Car_j x_{ij} - C(1 - x_{ij})$$

Come i vincoli sulle y_i per il commesso viaggiatore originario, questi vincoli per eliminare soluzioni con sottocircuiti indesiderati (cioè sottocircuiti che non partono dal nodo di partenza 1) non sono molto efficaci dal punto di vista algoritmico.

Anche qui un’alternativa è usare un numero esponenziale di vincoli.

Vincoli alternativi

$$\sum_{i \notin S,\ j \in S} x_{ij} \geq \left\lceil \frac{\sum_{j \in S} Car_j}{C} \right\rceil \quad \forall\, S \subseteq V \setminus \{1\}.$$

Impongono che per ogni sottinsieme S di nodi non contenente il nodo di partenza 1, il numero di veicoli che dovranno entrare nei nodi in S deve essere almeno pari a:

$$\left\lceil \frac{\sum_{j \in S} Car_j}{C} \right\rceil .$$

Inoltre, tali vincoli garantiscono che i limiti di capacità dei veicoli siano rispettati (il numero di veicoli che entra in S è in grado di portare tutto il carico dei nodi in S).