“Sapienza” Università di Roma - Dipartimento di Ingegneria Informatica, Automatica e Gestionale

Problemi di Distribuzione

Renato Bruni

bruni@dis.uniroma1.it

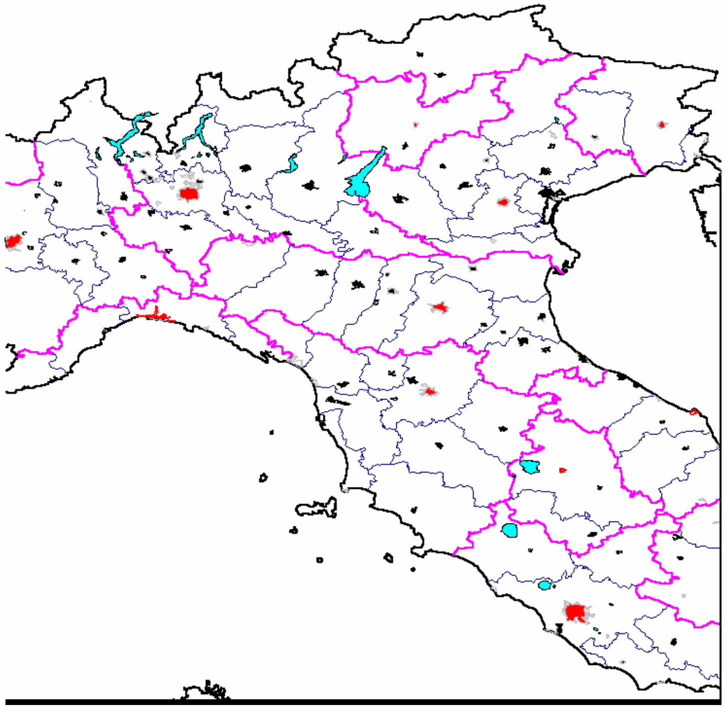
Il materiale presentato è derivato da quello dei proff. A. Sassano e C. Mannino

Problemi di Distribuzione

PROBLEMA GENERALE:

- Trasportare beni da una o più *origini* a una o più *destinazioni*
- Minimizzando dei costi di trasporto
- Rispettando una serie di vincoli:
  - vincoli di capacità dei veicoli
  - vincoli di raggiungibilità delle destinazioni
  - …

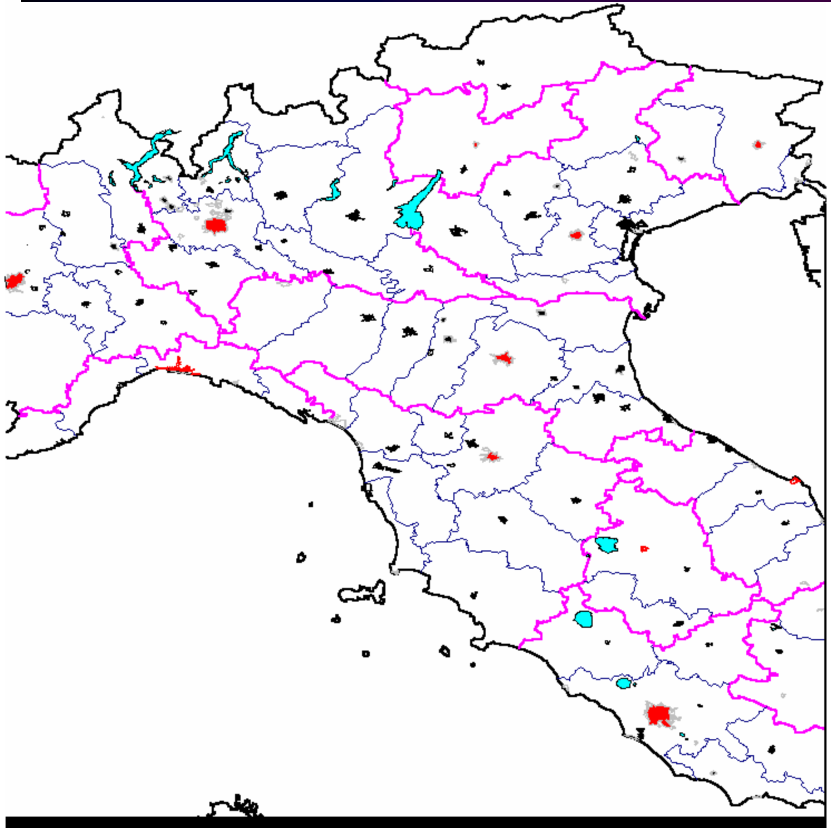
È un problema di grande rilevanza economica: la logistica costituisce in generale una spesa notevole



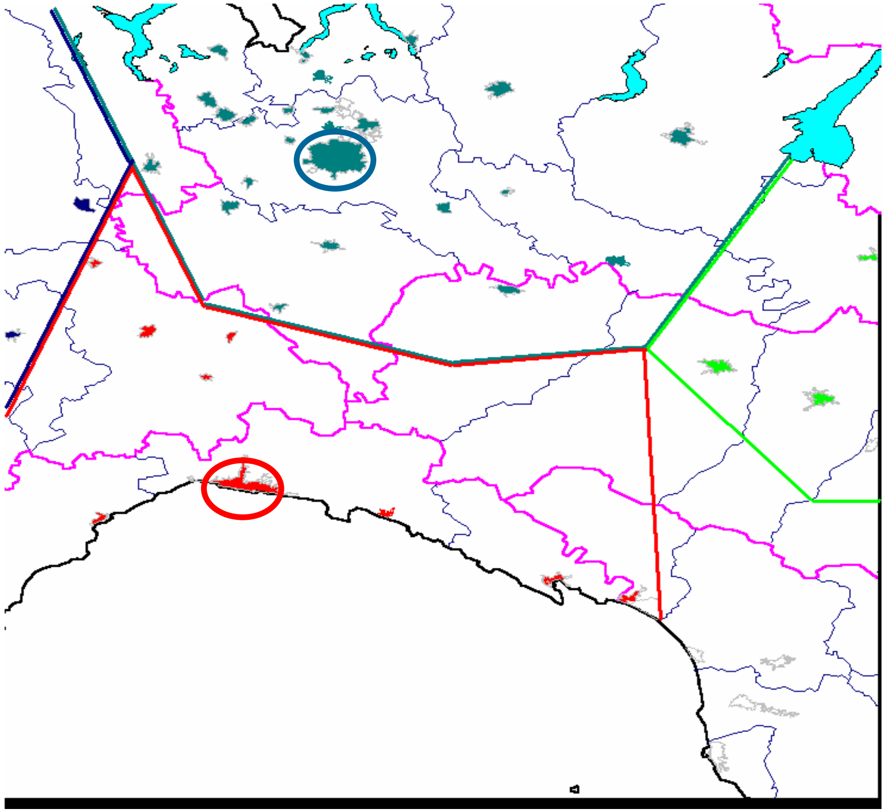
Dobbiamo conoscere aspetti relativi a:

1. *Localizzazione di origini e destinazioni (distanze, tempi, costi)*
2. *Domanda nelle destinazioni e caratteristiche dei mezzi di trasporto*

Localizzazione Origini (Centri di Distribuzione)



Localizzazione dei CD

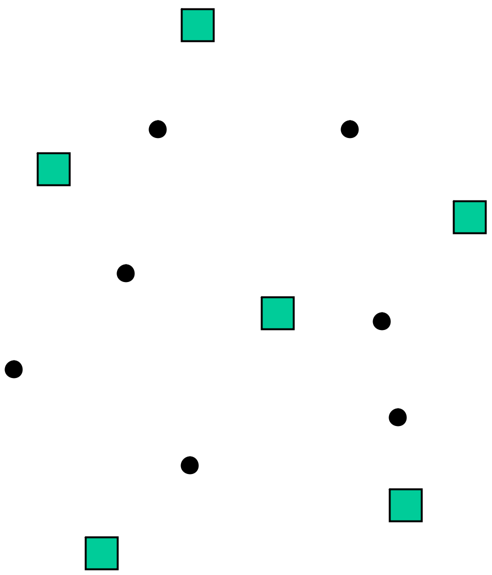


Assegnazione dei punti vendita ai CD

- Problema risolto come problema di *localizzazione di impianti*
- Decomposizione in problemi con singola origine (deposito)
- A ciascun deposito è associato un insieme di destinazioni (clienti)

Localizzazione di Impianti

- Abbiamo un insieme I di m **clienti da servire** e un insieme J di n **impianti che potrei attivare**
- L'impianto j costa f_j e
- Il collegamento ij costa c_{ij}
- Voglio minimizzare le spese



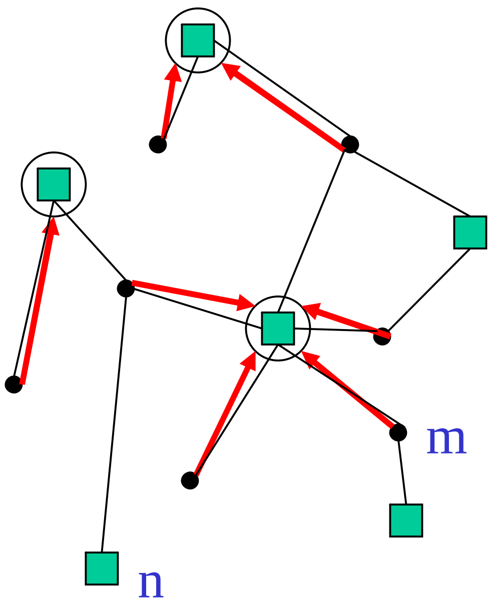
Scelte che devo fare: $\Rightarrow$ Variabili

- *Impianto j attivato o non attivato* $\Rightarrow x_j \begin{array}{l} 1 \\ 0 \end{array}$

- *Cliente i assegnato all'impianto j* $\Rightarrow y_{ij} \begin{array}{l} 1 \\ 0 \end{array}$

Soluzioni Ammissibili

- Una **soluzione ammissibile** è per esempio quella rappresentata accanto
- In altre parole, le variabili che abbiamo scelto devono rispettare dei **vincoli**:



- *Un cliente i può essere assegnato ad un impianto solo se esso è attivato* (vincoli di attivazione)

$$y_{ij} \leq x_j \qquad i \in I \qquad j \in J$$

- *Ogni cliente i deve essere assegnato ad uno e un solo impianto* (vincoli di servizio)

$$\sum_{j \in J} y_{ij} = 1 \qquad i \in I$$

- Allora posso dare una **formulazione** al problema

Modello di PLB

$$\min \sum_{j \in J} f_j x_j + \sum_{j \in J} \sum_{i \in I} c_{ij} y_{ij}$$

$$\sum_{j \in J} y_{ij} = 1 \qquad i \in I$$

$$x_j - y_{ij} \geq 0 \qquad i \in I \quad j \in J$$

$$y_{ij} \in \{0,1\} \quad x_j \in \{0,1\} \quad i \in I \quad j \in J$$

Piuttosto che risolvere direttamente questo problema binario (molto difficile) si risolve il suo rilassamento lineare (più facile). Si ottiene così la cosiddetta Formulazione Forte

Formulazione Forte

$$\min \sum_{j\in J} f_j x_j + \sum_{j\in J}\sum_{i\in I} c_{ij} y_{ij}$$

$$\sum_{j\in J} y_{ij} = 1 \qquad i\in I$$

$$\boxed{x_j - y_{ij} \geq 0 \qquad i\in I \quad j\in J}$$

$$y_{ij} \geq 0 \quad x_j \geq 0 \quad i\in I \quad j\in J$$

IN ALTERNATIVA

$$\boxed{\sum_{i\in I} y_{ij} \leq |I| x_j \quad j\in J}$$

Somma sui clienti dei vincoli di attivazione

Formulazione Debole

$$\min \sum_{j \in J} f_j x_j + \sum_{j \in J} \sum_{i \in I} c_{ij} y_{ij}$$

$$\sum_{j \in J} y_{ij} = 1 \qquad i \in I$$

$$\sum_{i \in I} y_{ij} \leq |I| x_j \quad j \in J \qquad (*)$$

$$y_{ij} \geq 0 \quad x_j \geq 0 \quad i \in I \quad j \in J$$

TEOREMA:

Se $f_j > 0$ per ogni j allora ogni soluzione ottima del problema (di PL) soddisfa all'**uguaglianza** i vincoli (*).

Sostituzione

- Allora possiamo scrivere i vincoli $(*)$ come:

$$\sum_{i \in I} y_{ij} = |I| x_j \quad j \in J$$

- e porre: $$x_j = \frac{1}{|I|} \sum_{i \in I} y_{ij} \quad j \in J$$

che possiamo **sostituire** nella formulazione debole vista eliminando così le x_j

Formulazione Debole Finale

$$\min \sum_{j\in J} (f_j / |I|) \sum_{i\in I} y_{ij} + \sum_{j\in J} \sum_{i\in I} c_{ij} y_{ij}$$

$$= \min \sum \sum (c_{ij} + f_j / |I|) y_{ij}$$

$$\sum_{j\in J} y_{ij} = 1 \qquad i \in I$$

$$y_{ij} \geq 0 \quad i \in I \quad j \in J$$

Soluzione:

Per ogni cliente $i \in I$ abbiamo:

$$y^*_{ij} = 1 \iff c_{ij} + f_j / |I| = \min_{k\in J} \{c_{ik} + f_k / |I|\}$$

Confronto Formulazioni per Localizzazione

- Formulazione **forte**: è una formulazione di migliore qualità (più stringente) quindi fornisce soluzioni migliori (più vicine a quelle del problema intero) ma ha molti vincoli di attivazione (più difficile, potrebbe essere troppo grande per risolverla)
- Formulazione **debole**: si risolve molto semplicemente ma è una formulazione di qualità peggiore e quindi la soluzione è peggiore (cioè meno vicina a quella del problema intero)
- Per risolvere la formulazione forte senza scrivere tutti i vincoli di attivazione si usa un approccio di generazione di vincoli (simplesso dinamico)

Simplesso Dinamico

Risolve un problema di Programmazione Lineare:

$$\min c^T x : \ x \in P = \{x \in \mathbf{R}^n : Ax \leq b, \ 1_n \geq x \geq 0_n\}$$

<u>Due ingredienti:</u>

$\hat{x} \in \mathbf{R}^n$

Oracolo di Separazione di P

$\hat{x} \in P$

$\hat{x} \notin P$

$a_i \hat{x} > b_i$

$\min c^T x$

$Ax \leq b,$

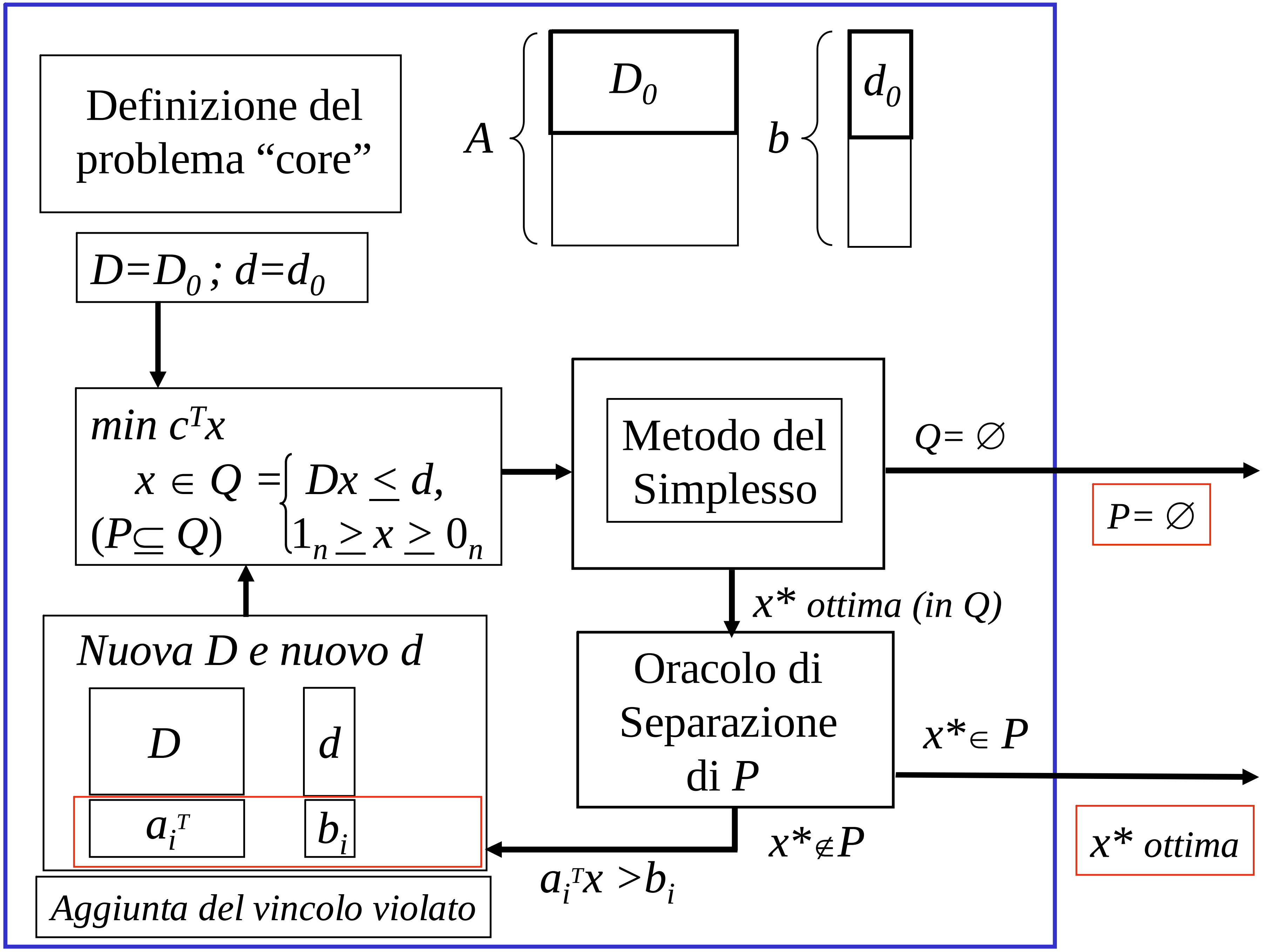
$1_n \geq x \geq 0_n$

Metodo del Simplesso

x^* *<u>soluzione ottima</u>*

~~problema illimitato~~

nessuna soluzione ($P = \emptyset$)



Definizione del problema “core”
A
D_0
b
d_0
$D=D_0\,;\; d=d_0$
$\min c^T x$
$x \in Q = \begin{cases} Dx \leq d, \\ 1_n \geq x \geq 0_n \end{cases}$
$(P \subseteq Q)$
Metodo del Simplesso
$Q = \emptyset$
$P = \emptyset$
x^* ottima (in Q)
Oracolo di Separazione di P
$x^* \in P$
x^* ottima
$x^* \notin P$
$a_i^T x > b_i$
Nuova D e nuovo d
D
d
a_i^T
b_i
Aggiunta del vincolo violato

Esempio

Consideriamo questo problema di Localizzazione Impianti (con 5 impianti e 6 clienti, completamente definito dai dati riportati sotto) e risolviamolo (col solutore Cplex) usando:

- Formulazione **Forte**
- Formulazione **Debole**
- Formulazione Forte con **Generazione Vincoli**

$$f = \begin{bmatrix} 4 & 3 & 1 & 4 & 7 \end{bmatrix}$$

$$c = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 12 & 13 & 6 & 0 & 1 \\ 8 & 4 & 9 & 1 & 2 \\ 2 & 6 & 6 & 0 & 1 \\ 3 & 5 & 2 & 10 & 8 \\ 8 & 0 & 5 & 10 & 8 \\ 2 & 0 & 3 & 4 & 1 \end{bmatrix} \end{array}$$

Formulazione Forte dell'Esempio 1/2

$$\text{Min } 12\, y(1,1) + 13\, y(1,2) + 6\, y(1,3) + y(1,5) + 8\, y(2,1) + 4\, y(2,2)$$
$$+ 9\, y(2,3) + y(2,4) + 2\, y(2,5) + 2\, y(3,1) + 6\, y(3,2) + 6\, y(3,3) + y(3,5)$$
$$+ 3\, y(4,1) + 5\, y(4,2) + 2\, y(4,3) + 10\, y(4,4) + 8\, y(4,5) + 8\, y(5,1)$$
$$+ 5\, y(5,3) + 10\, y(5,4) + 8\, y(5,5) + 2\, y(6,1) + 3\, y(6,3) + 4\, y(6,4)$$
$$+ y(6,5) + 4\, x(1) + 3\, x(2) + x(3) + 4\, x(4) + 7\, x(5)$$

Subject To

servizio(1): $y(1,1) + y(1,2) + y(1,3) + y(1,4) + y(1,5) = 1$
servizio(2): $y(2,1) + y(2,2) + y(2,3) + y(2,4) + y(2,5) = 1$
servizio(3): $y(3,1) + y(3,2) + y(3,3) + y(3,4) + y(3,5) = 1$
servizio(4): $y(4,1) + y(4,2) + y(4,3) + y(4,4) + y(4,5) = 1$
servizio(5): $y(5,1) + y(5,2) + y(5,3) + y(5,4) + y(5,5) = 1$
servizio(6): $y(6,1) + y(6,2) + y(6,3) + y(6,4) + y(6,5) = 1$

attivaz(01): $x(1) - y(1,1) >= 0$
attivaz(02): $x(1) - y(2,1) >= 0$
attivaz(03): $x(1) - y(3,1) >= 0$
attivaz(04): $x(1) - y(4,1) >= 0$
attivaz(05): $x(1) - y(5,1) >= 0$
attivaz(06): $x(1) - y(6,1) >= 0$

[continua]

Formulazione Forte dell'Esempio 2/2

[continua]

attivaz(07): x(2) – y(1,2) >= 0
attivaz(08): x(2) – y(2,2) >= 0
attivaz(09): x(2) – y(3,2) >= 0
attivaz(10): x(2) – y(4,2) >= 0
attivaz(11): x(2) – y(5,2) >= 0
attivaz(12): x(2) – y(6,2) >= 0

attivaz(13): x(3) – y(1,3) >= 0
attivaz(14): x(3) – y(2,3) >= 0
attivaz(15): x(3) – y(3,3) >= 0
attivaz(16): x(3) – y(4,3) >= 0
attivaz(17): x(3) – y(5,3) >= 0
attivaz(18): x(3) – y(6,3) >= 0

attivaz(19): x(4) – y(1,4) >= 0
attivaz(20): x(4) – y(2,4) >= 0
attivaz(21): x(4) – y(3,4) >= 0
attivaz(22): x(4) – y(4,4) >= 0
attivaz(23): x(4) – y(5,4) >= 0
attivaz(24): x(4) – y(6,4) >= 0

attivaz(25): x(5) – y(1,5) >= 0
attivaz(26): x(5) – y(2,5) >= 0
attivaz(27): x(5) – y(3,5) >= 0
attivaz(28): x(5) – y(4,5) >= 0
attivaz(29): x(5) – y(5,5) >= 0
attivaz(30): x(5) – y(6,5) >= 0

Soluzione Formulazione Forte

CPLEX 11.2.0: optimal solution; objective 11
9 dual simplex iterations (0 in phase I)

Le variabili valgono:

```
x [*] :=
1  0
2  1
3  1
4  1
5  0;

y [*,*]
:   1   2   3   4   5    :=
1   0   0   0   1   0
2   0   0   0   1   0
3   0   0   0   1   0
4   0   0   1   0   0
5   0   1   0   0   0
6   0   1   0   0   0;
```

Il costo totale e': 11 (lower bound buono: in questo caso è l'ottimo)

Risolto con Formulazione Debole

$$\min_{k\in J}\{c_{ik}+f_k/\,|I|\}$$

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ c = & \begin{bmatrix} 12 & 13 & 6 & 0 & 1 \\ 8 & 4 & 9 & 1 & 2 \\ 2 & 6 & 6 & 0 & 1 \\ 3 & 5 & 2 & 10 & 8 \\ 8 & 0 & 5 & 10 & 8 \\ 2 & 0 & 3 & 4 & 1 \end{bmatrix} \end{array}$$

$$f = \begin{bmatrix} 4 & 3 & 1 & 4 & 7 \end{bmatrix}$$

$$[\, c_{ik}+f_k/\,|I|\,]$$

$$\begin{bmatrix} 12+4/6 & 13+3/6 & 6+1/6 & \boxed{0+4/6} & 1+7/6 \\ 8+4/6 & 4+3/6 & 9+1/6 & \boxed{1+4/6} & 2+7/6 \\ 2+4/6 & 6+3/6 & 6+1/6 & \boxed{0+4/6} & 1+7/6 \\ 3+4/6 & 5+3/6 & \boxed{2+1/6} & 10+4/6 & 8+7/6 \\ 8+4/6 & \boxed{0+3/6} & 5+1/6 & 10+4/6 & 8+7/6 \\ 2+4/6 & \boxed{0+3/6} & 3+1/6 & 4+4/6 & 1+7/6 \end{bmatrix}$$

$$y^*_{14} = y^*_{24} = y^*_{34} = y^*_{43} = y^*_{52} = y^*_{62} = 1$$

$$\sum\sum (c_{ij}+f_j/\,|I|)\,y^*_{ij} = 2/3+5/3+2/3+13/6+1 = 6+1/6$$

Soluzione Formulazione Debole

Le variabili valgono:

```
x [*] :=
1  0
2  0.333333
3  0.166667
4  0.5
5  0;

y [*,*]
:   1   2   3   4   5    :=
1   0   0   0   1   0
2   0   0   0   1   0
3   0   0   0   1   0
4   0   0   1   0   0
5   0   1   0   0   0
6   0   1   0   0   0;
```

Il costo totale e': 6.166667 (lower bound di qualità peggiore)

Generazione Vincoli per l'Esempio

Si risolve inizialmente

Min 12 y(1,1) + 13 y(1,2) + 6 y(1,3) + y(1,5) + 8 y(2,1) + 4 y(2,2)
+ 9 y(2,3) + y(2,4) + 2 y(2,5) + 2 y(3,1) + 6 y(3,2) + 6 y(3,3) + y(3,5)
+ 3 y(4,1) + 5 y(4,2) + 2 y(4,3) + 10 y(4,4) + 8 y(4,5) + 8 y(5,1)
+ 5 y(5,3) + 10 y(5,4) + 8 y(5,5) + 2 y(6,1) + 3 y(6,3) + 4 y(6,4)
+ y(6,5) + 4 x(1) + 3 x(2) + x(3) + 4 x(4) + 7 x(5)
Subject To
servizio(1): y(1,1) + y(1,2) + y(1,3) + y(1,4) + y(1,5) = 1
servizio(2): y(2,1) + y(2,2) + y(2,3) + y(2,4) + y(2,5) = 1
servizio(3): y(3,1) + y(3,2) + y(3,3) + y(3,4) + y(3,5) = 1
servizio(4): y(4,1) + y(4,2) + y(4,3) + y(4,4) + y(4,5) = 1
servizio(5): y(5,1) + y(5,2) + y(5,3) + y(5,4) + y(5,5) = 1
servizio(6): y(6,1) + y(6,2) + y(6,3) + y(6,4) + y(6,5) = 1

Soluzione Generazione Vincoli

Risolvo e trovo sol:
y(1,4), y(2,4), y(3,4), y(4,3), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(5,2) + x(2) >= 0**

Risolvo e trovo sol:
x(2), y(1,4), y(2,4), y(3,4), y(4,3), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(4,3) + x(3) >= 0**

Risolvo e trovo sol:
x(2), y(1,4), y(2,4), y(3,4), y(4,1), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(4,1) + x(1) >= 0**

Risolvo e trovo sol:
x(2), x(3), y(1,4), y(2,4), y(3,4), y(4,3), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(1,4) + x(4) >= 0**

Risolvo e trovo sol:
x(2), x(3), y(1,5), y(2,4), y(3,4), y(4,1), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(2,4) + x(4) >= 0**

Soluzione Generazione Vincoli

Risolvo e trovo sol:
x(2), x(3), y(1,5), y(2,5), y(3,4), y(4,3), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(3,4) + x(4) >= 0**

Risolvo e trovo sol:
x(2), x(3), y(1,5), y(2,5), y(3,5), y(4,3), y(5,2), y(6,2) =1, tutto il resto 0

Da essa genero vincolo: **- y(1,5) + x(5) >= 0**

Risolvo e trovo sol:
x(2), x(3), x(4), y(1,4), y(2,4), y(3,4), y(4,3), y(5,2), y(6,2) =1, tutto il resto 0

A questo punto, dopo aver generato 7 vincoli di attivazione invece dei 30 visti prima, l'oracolo dice che non ci sono più vincoli di attivazione violati.

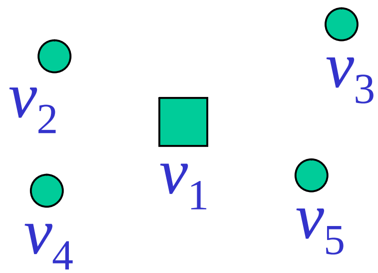
Allora ho trovato la soluzione della Formulazione Forte (in questo caso è la **Soluzione ottima**).

Per problemi più grandi fa la differenza tra risolvere o no.

Dati di Localizzazione Disponibili

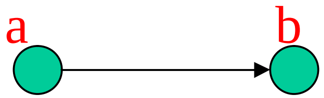
Dopo aver risolto la localizzazione, per ogni area si conosce:

1. Origine (o deposito) v_1
2. Destinazioni (o clienti) che v_1 deve servire $\{v_2, v_3, \dots, v_n\}$



3. Grafo *orientato* $G(V,E)$ deposito-clienti

   Nodi $V=\{v_1,\dots,v_n\}$

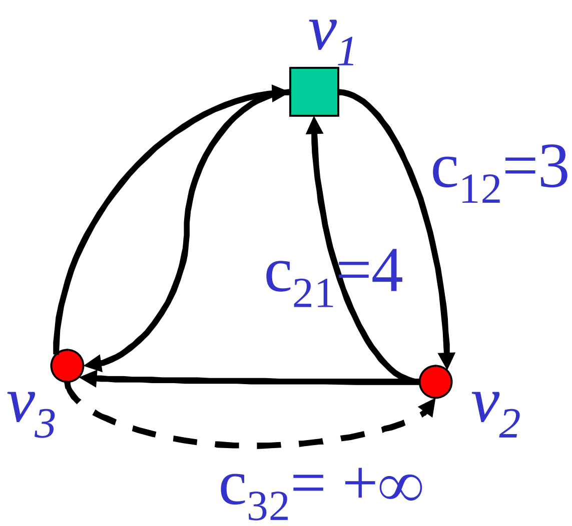
   Archi 

   se b è *accessibile* da a

4. *Costo dell'arco* da v_a a v_b: c_{ab} è il valore dell'aspetto che si vuole minimizzare, ad es. spesa, distanza, preferibilità, tempo, etc. (è importante scegliere correttamente)

Il Grafo dei Collegamenti

• Vogliamo un grafo orientato che sia completo $G(V,E)$ quindi aggiungiamo anche gli archi relativi a collegamenti che non ci sono mettendo pesi opportunamente alti

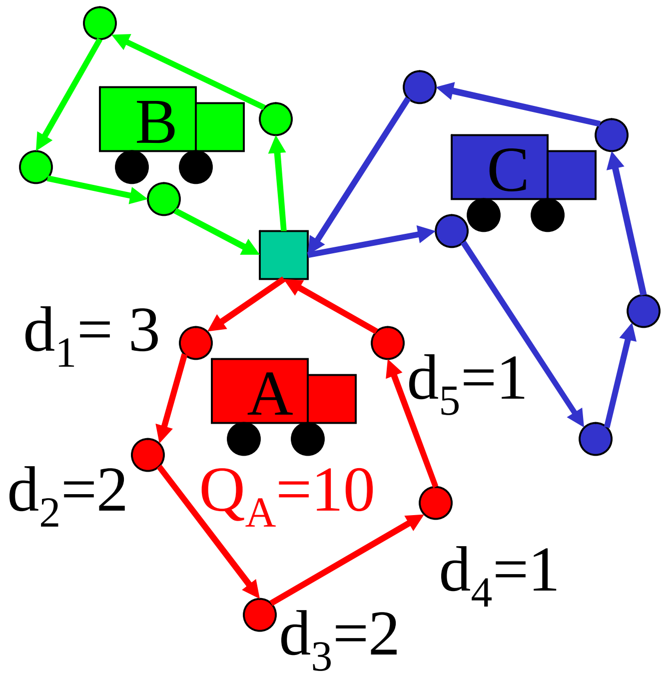


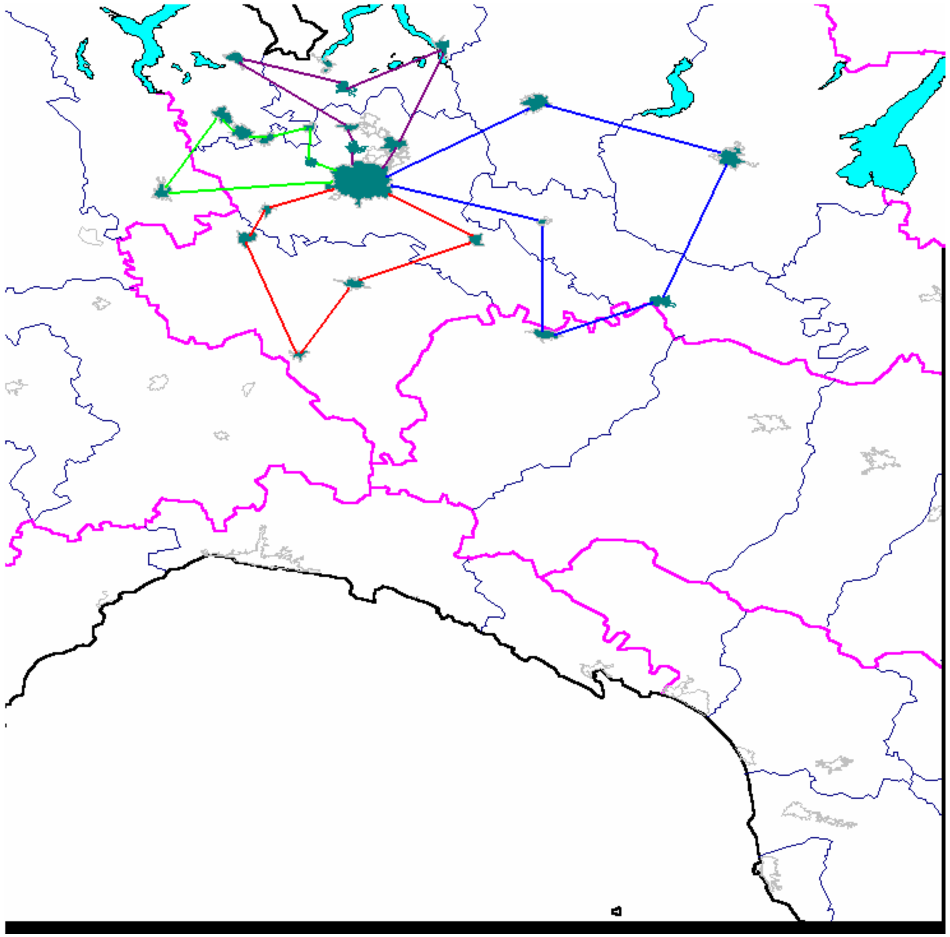
→ poniamo $c_{32} = \infty$ (o meglio un numero >> tutti gli altri costi) se 2 non è accessibile da 3

La Flotta di Veicoli

- Per servire i vari clienti abbiamo una flotta di veicoli
- Ciascun veicolo visita un insieme di clienti e <u>torna al deposito</u>

- *<u>Numero e caratteristiche dei singoli veicoli</u>* devono essere noti





Dati su Domanda e Veicoli

1. d_i: *domanda del cliente* v_i
2. m: *numero di veicoli disponibili*
3. $J=\{1, 2, \ldots, m\}$: *insieme dei veicoli*
4. Di solito Q_j: *capacità del veicolo* $j \in \{1, 2, \ldots, m\}$
5. A volte per ogni cliente bisogna distinguere se è consegna o prelievo (*Pickup or Delivery*)

Eventuali aspetti temporali: orari di apertura limitati, tempi di trasporto, di carico e scarico (tempi di transito)

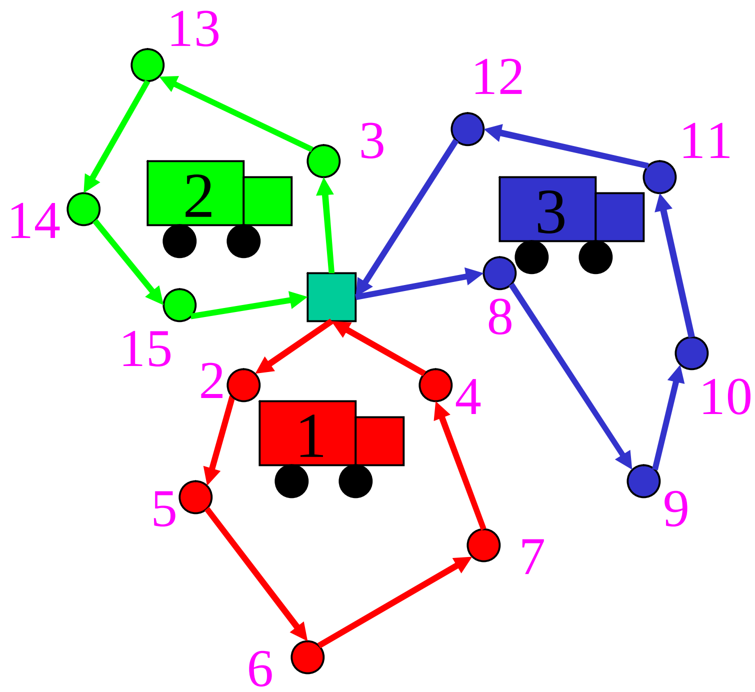
6. *Tempo di transito* da *a* a *b*: t_{ab}
7. *Tempo di servizio* di *a*: t_a
8. T_j: *tempo massimo di rientro del veicolo* $j \in \{1, 2, \ldots, m\}$
9. *Finestre temporali* associate ad ogni nodo v_i: $[inizio_i, fine_i]$

Struttura delle Soluzioni

- Famiglia dei “cluster” clienti/veicolo

$\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$

- $C_k \subseteq V \quad k \in \{1, 2, \ldots, m\}$
- $v_1 \in C_k \quad k \in \{1, 2, \ldots, m\}$
- $C_k \cap C_h = \{v_1\} \quad k,h \in \{1, 2, \ldots, m\}$
- $\cup \ C_k = V$
- $\sum_{h \in C_k} d_h \leq Q_k$

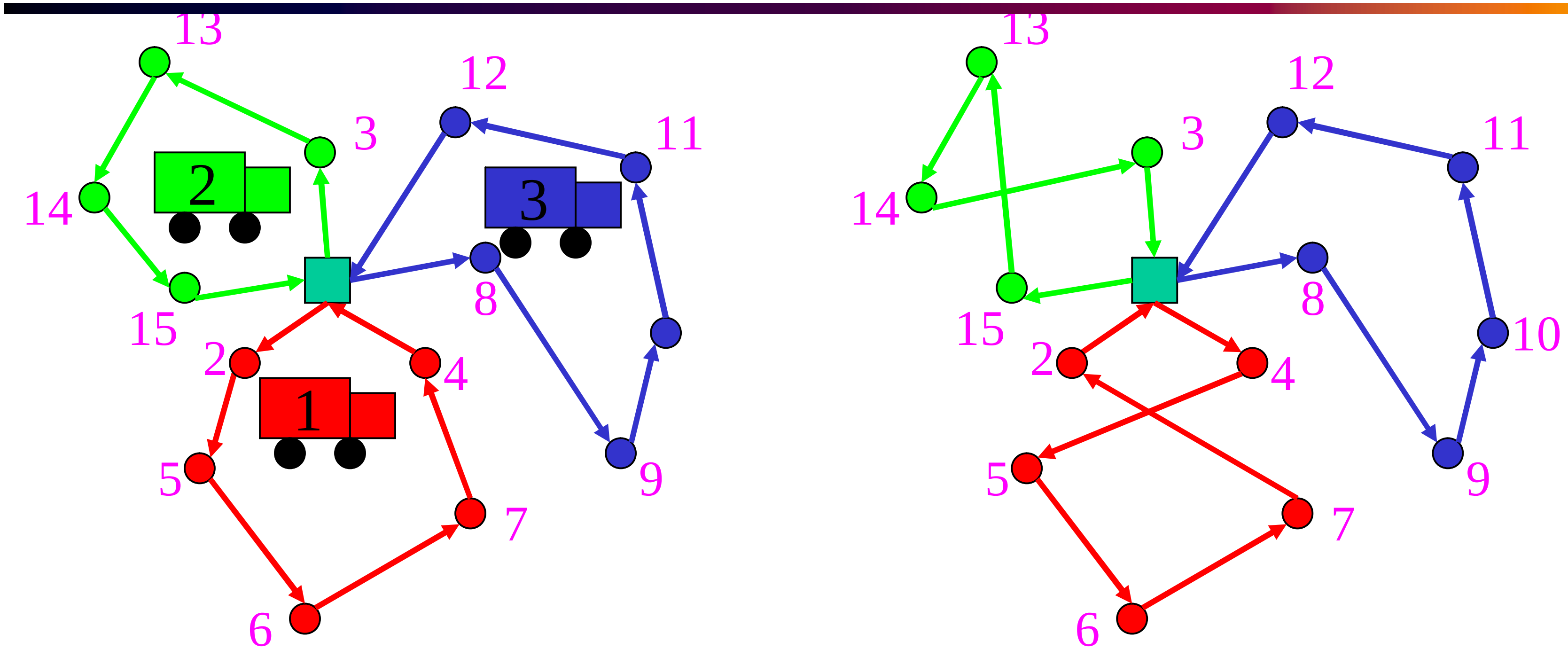


$C_1 = \{1, 2, 4, 5, 6, 7\}$
$C_2 = \{1, 8, 9, 10, 11, 12\}$
$C_3 = \{1, 3, 13, 14, 15\}$

$\mathcal{C}$ caratterizza completamente la soluzione ?

No! Ad un “cluster” posso associare diverse sequenze di visita

Clustering e Routing

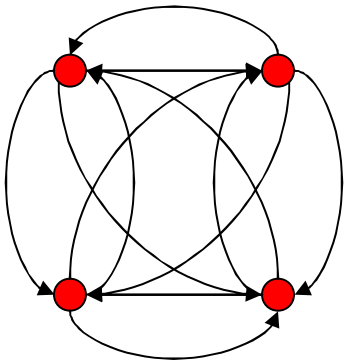


Stessi “cluster” ma sequenze (“routing”) diverse

Una sequenza è un ciclo T_k orientato che tocca tutti i nodi del “cluster” C_k (<u>*Ciclo Hamiltoniano*</u>)

<u>Conclusione</u>: *Una soluzione ammissibile è caratterizzata dai cicli hamiltoniani* $\{T_1, T_2, \ldots, T_m\}$ *definiti sui “cluster”* $\{C_1, C_2, \ldots, C_m\}$

Ciclo Hamiltoniano Minimo

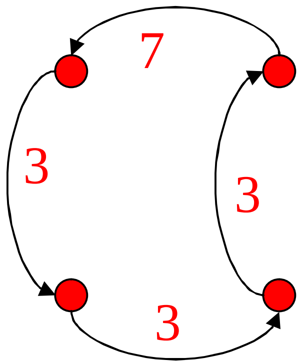


<u>DATI:</u>

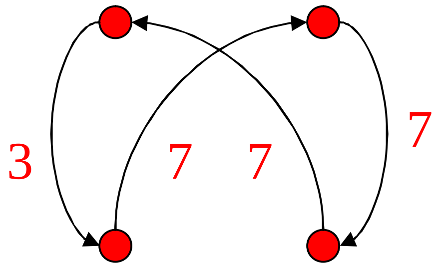
- grafo *orientato* completo $G(V,E)$
- pesi (costi) c_{hk} associati agli archi $hk \in A$

<u>TROVARE:</u>

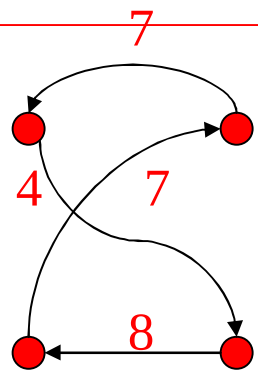
> *<u>Ciclo Hamiltoniano</u>* (che tocca tutti i nodi) *di peso minimo*



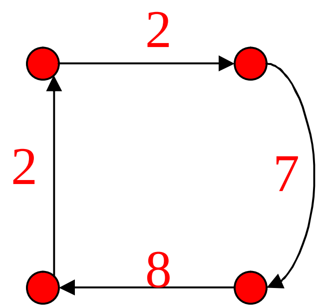
16



24



26



19

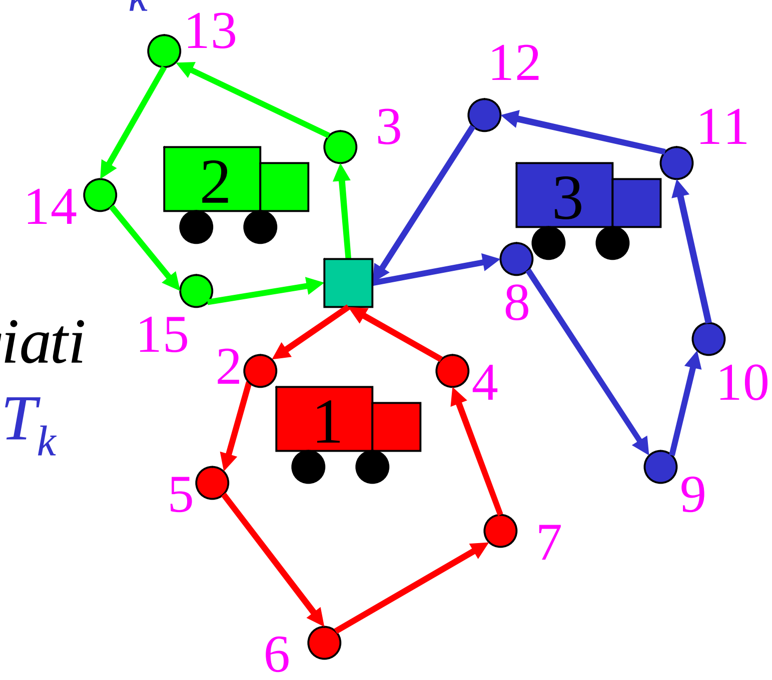
Si chiama Problema del Commesso Viaggiatore (TSP: Travelling Salesman Problem) per cui vedremo delle euristiche (**slides apposite**)

Problema Base di Distribuzione

- *Soluzione ammissibile*: $\boldsymbol{T} = \{T_1, T_2, \ldots, T_m\}$
- Un ciclo hamiltoniano T_k per ogni "cluster" C_k
- *Insieme delle soluzioni ammissibili*:

  $\mathcal{T} = \{\boldsymbol{T}_1, \boldsymbol{T}_2, \ldots, \boldsymbol{T}_q\}$

- $L(T_k)$ *somma dei costi di transito associati agli archi del ciclo hamiltoniano* T_k
- $Z(\boldsymbol{T}) = \sum_{k \in J} L(T_k)$ *costo di transito*



Problema Base di Distribuzione

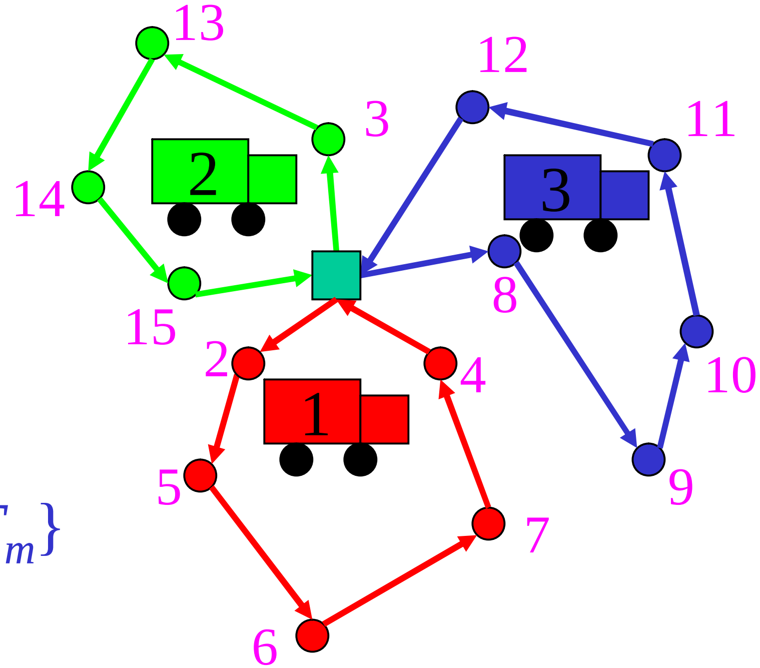
$\min \; Z(\boldsymbol{T}) : \; \boldsymbol{T} \in \mathcal{T}$

Minimizzare il costo di transito
- servendo la domanda di tutti i clienti
- rispettando la capacità dei veicoli

Formulazione del Problema Base

EVENTO ELEMENTARE:

Nodo v_h assegnato al "cluster" C_k



Variabile binaria y_{hk} *con:*

$y_{hk}=1$ *se* $v_h \in C_k$; $y_{hk}=0$ *se* $v_h \notin C_k$

- *Soluzione ammissibile:* $\mathbf{T} = \{T_1, T_2, \ldots, T_m\}$

- $C_k \subseteq V \quad k \in \{1, 2, \ldots, m\}$
- $v_1 \in C_k \quad k \in \{1, 2, \ldots, m\}$
- $C_k \cap C_h = \{v_1\} \quad k,h \in \{1, 2, \ldots, m\}$
- $\cup \; C_k = V$
- $\sum_{h \in C_k} d_h \leq Q_k$

$$\sum_{k \in J} y_{1k} = m$$

$$\sum_{k \in J} y_{hk} = 1; \quad h \in \{1, 2, \ldots, n\}$$

$$\sum_{h \in V - \{v_1\}} d_h y_{hk} \leq Q_k \quad k \in J$$

Problema di Assegnamento Generalizzato

$y_k = [y_{1k}, y_{2k}, y_{3k}, \ldots, y_{mk}]$ <u>vettore di incidenza</u> di C_k

$f(y_k)$ *costo di transito del* <u>*ciclo hamiltoniano minimo*</u> *definito sui nodi di* C_k

Problema Base di Distribuzione

$$\min \; Z(\boldsymbol{T}) : \; \boldsymbol{T} \in \mathcal{T}$$

$$\equiv \min \sum_{k \in J} L(T_k) : \; \boldsymbol{T} \in \mathcal{T}$$

$$\equiv$$

$$\min \sum_{h \in J} f(y_k)$$

$$\sum_{k \in J} y_{1k} = m$$

$$\sum_{k \in J} y_{hk} = 1$$

$$\sum_{h \in V - \{v_1\}} d_h y_{hk} \leq Q_k$$

Ma ancora non ci va bene perché l'obiettivo non è espresso linearmente

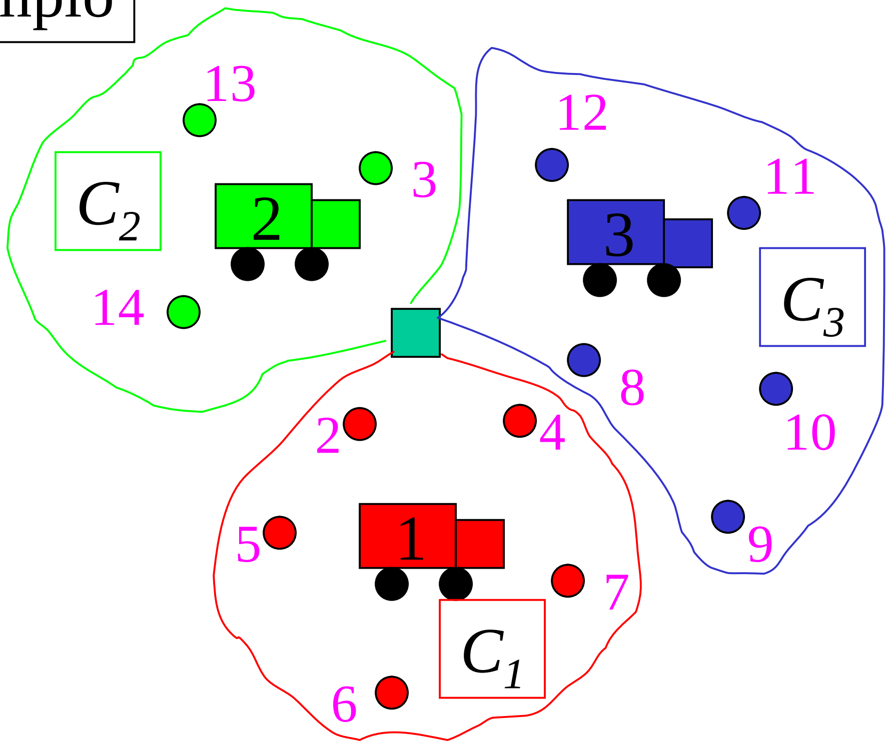
Espressione di $f(y_k)$

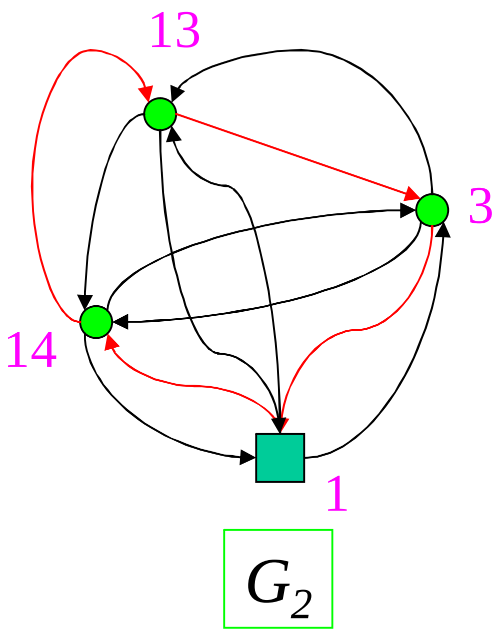
Esprimiamo $f(y_k)$ *come soluzione di un problema di PL01*

$y_k = (y_{k1}\ y_{k2}\ y_{k3}\ \ldots y_{km})^T$ <u>*vettore di incidenza*</u> *di* C_k

$f(y_k)$ *costo (di transito) del* <u>*ciclo hamiltoniano minimo*</u> *definito sul sottografo* G_k *indotto da* C_k *in* G

- Esempio



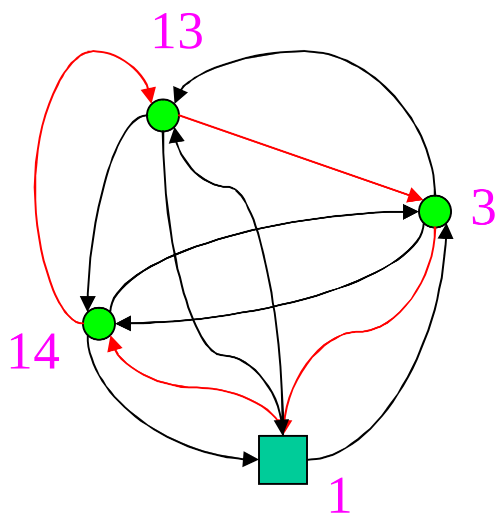


Nuove variabili binarie

$f(y_k)$ *costo (di transito) del ciclo hamiltoniano minimo definito sul sottografo* G_k *indotto da* C_k *in* G

Soluzioni ammissibili: *cicli hamiltoniani di* G_k

Evento elementare: *il veicolo* k *usa l'arco* v_iv_j *per qualche coppia* $\{v_i,v_j\} \subseteq C_k$



G_2

Variabile: x_{ij}^{k} $\begin{cases} 1 & \text{se veicolo } k \text{ usa } v_iv_j \\ 0 & \text{altrimenti} \end{cases}$

Esempio: $x_{13,3}^{2} = 1;\ x_{1,13}^{2} = 0$

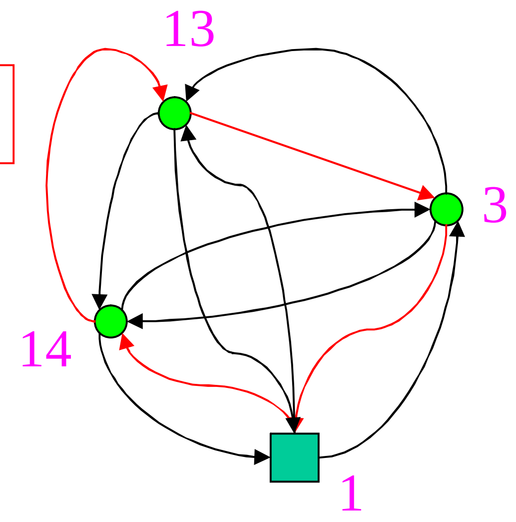
Allora potrò esprimere la funzione obiettivo come somma dei pesi degli archi usati

Condizioni di Ammissibilità 1/2

Un sottoinsieme di archi T di G_k

è l'insieme di archi di un ciclo hamiltoniano di G_k

quando *(ma basta richiedere ciò?)*



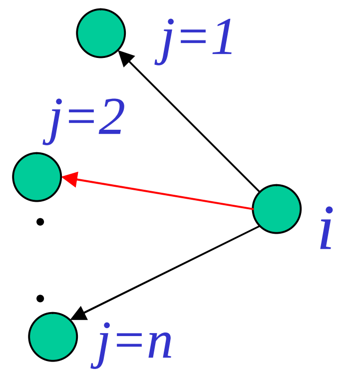
G_2

1. *Esiste un arco uscente da ogni nodo di* C_k
2. *Esiste un arco entrante in ogni nodo di* C_k

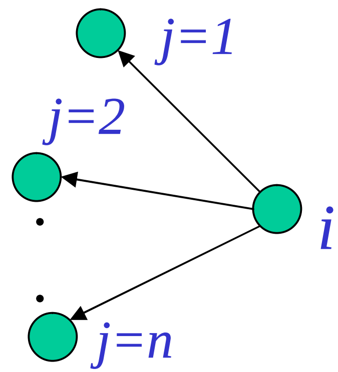
Il vettore [x_{ij}^k: $i=1,\dots,n;\ \ j=1,\dots,n$]

è il vettore di incidenza di un ciclo hamiltoniano T *di* G_k

$$\begin{cases} \boxed{\displaystyle\sum_{j=1}^{n} x_{ij}^{k} = y_{ik}} \\ \displaystyle\sum_{j=1}^{n} x_{ji}^{k} = y_{ik} \end{cases}$$



$y_{ik}=1$ se $i \in C_k$



$y_{ik}=0$ se $i \notin C_k$

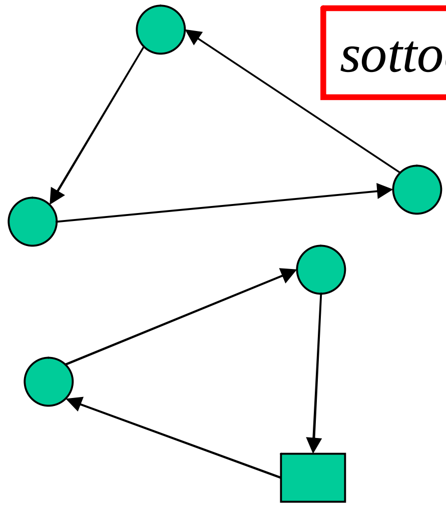
Condizioni di Ammissibilità 2/2

Il vettore [x_{ij}^{k}: $i=1,\ldots,n;\quad j=1,\ldots,n$]

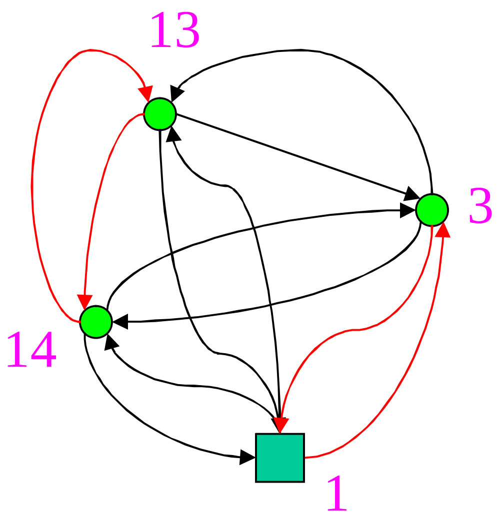
è il vettore di incidenza di un ciclo hamiltoniano T di G_k

solo se

$$\begin{cases} \sum_{j=1}^{n} x_{ij}^{k} = y_{ik} \\ \sum_{j=1}^{n} x_{ji}^{k} = y_{ik} \end{cases}$$



sottociclo

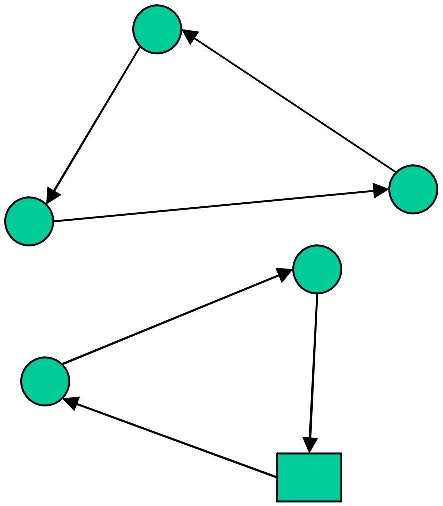


G_2

Ma questo non basta ancora: con questi vincoli i sottocicli risultano ancora ammissibili !

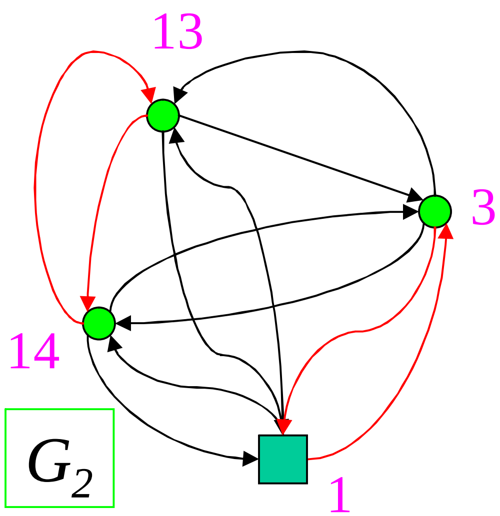
Come si eliminano ?

Eliminazione dei Sottocicli



- *Un sottociclo è un ciclo orientato su un sottoinsieme dei nodi di* C_k
- *Un ciclo orientato su un sottoinsieme* $S \subset C_k$ *contiene |S| archi*

ELIMINAZIONE DEI SOTTOCICLI:
Nessun sottoinsieme proprio S *di* C_k *può contenere |S| archi*



$$\sum_{(i,j)\in S} x_{ij}^k \leq |S|-1 \qquad \text{per ogni } S \subset C_k$$

$$2 \leq |S| \leq |C_k|-2$$

per ogni $k \in J$

o equivalentemente

$$\sum_{i\in S\, j\notin S} x_{ij}^k \geq 1$$

Ammissibilità Cicli

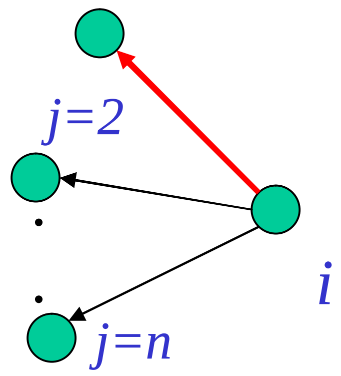
Un sottoinsieme di archi T di G_k

è l'insieme di archi di un ciclo hamiltoniano di G_k

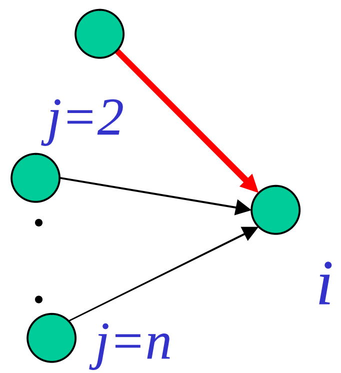
se e solo se

$$\begin{cases} \displaystyle\sum_{j=1}^{n} x_{ij}^{k} = y_{ik} \\ \displaystyle\sum_{j=1}^{n} x_{ji}^{k} = y_{ik} \\ \displaystyle\sum_{ij \in E(S)} x_{ij}^{k} \leq |S| - 1 \ \text{ per ogni } S \subset C_k \quad 2 \leq |S| \leq |C_k|\text{-}2 \end{cases}$$

1. *arco uscente da ogni nodo di* C_k



2. *arco entrante in ogni nodo di* C_k



3. *non contiene sottocicli*

Mettendo tutto assieme possiamo eliminare le y e ottenere una formulazione di PL01

Formulazione VRP come PL01

$$\min \sum_{k\in J} \sum_{i\in I} \sum_{\substack{j\in I\\ j\neq i}} c_{ij}\, x_{ij}^k$$ minimizzo somma costi

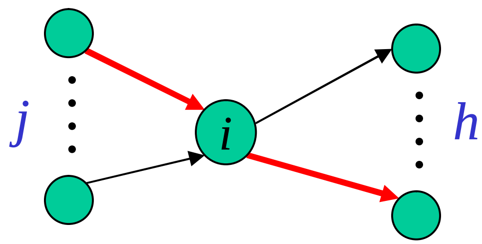
$I=\{1, 2, \ldots, n\}$: insieme dei nodi (tra cui 1 è il deposito)

$J=\{1, 2, \ldots, m\}$: insieme dei veicoli

$$\sum_{k\in J} \sum_{j\in I} x_{1j}^k = m$$ dal deposito escono m veicoli

$$\sum_{k\in J} \sum_{\substack{j\in I\\ j\neq i}} x_{ij}^k = 1 \qquad i\in\{2,\ldots,n\}$$ ogni cliente è servito

$$\sum_{\substack{j\in I\\ j\neq i}} x_{ji}^k = \sum_{\substack{h\in I\\ h\neq i}} x_{ih}^k \qquad \begin{matrix} i\in I\\ k\in J\end{matrix}$$ se veicolo k entra nel nodo i deve uscirne



$$\sum_{i\in I} \sum_{\substack{j\in I\\ j\neq i\\ j\neq 1}} d_j\, x_{ij}^k \le Q_k \qquad k\in J$$ la capacità è rispettata

$$\sum_{(i,j)\in S} x_{ij}^k \le |S|-1 \qquad S\subseteq\{2,\ldots,n\},\ |S|\ge 2 \quad k\in J$$ proibisco ogni possibile sottociclo S

$$x_{ij}^k \in \{0,1\} \qquad \begin{matrix} i,j\in I\\ k\in J\end{matrix}$$ variabili x_{ij}^k : percorro arco (i,j) col veicolo k

Numero di Veicoli variabile

- Se i veicoli disponibili sono m ma **potrei anche utilizzarne di meno** (spendendo meno), considero per ognuno un costo di attivazione f_k
- Se invece spendo sempre lo stesso anche se lascio dei veicoli fermi, non serve

$$\min \sum_{k \in J} \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} c_{ij}\, x_{ij}{}^{k} + \sum_{k \in J} f_k \sum_{\substack{j \in I \\ j \neq 1}} x_{1j}{}^{k}$$

minimizzo somma costi percorrenza + costi attivazione

$$\sum_{k \in J} \sum_{j \in I} x_{1j}{}^{k} \leq m$$

dal deposito escono fino a m veicoli

… e tutto il resto come prima

- Con meno veicoli occorre fare dei percorsi più lunghi
- Cerco il miglior compromesso tra risparmio per i veicoli ed economicità dei percorsi

Aspetti Temporali

- Se per il generico veicolo k ho un **tempo massimo di utilizzo** T_k (dovuto per es. alla disponibilità del conducente, a vincoli sull'orario di lavoro o alla chiusura del deposito)
- T_k deve essere espresso in una unità di misura facilmente sommabile e a partire dall'inizio delle operazioni (ad esempio in minuti dall'uscita dal deposito)

- Devo **considerare**: (ovviamente nella stessa unità di misura)

$$\begin{cases} \textit{Tempo di transito} \text{ da } i \text{ a } j\text{: } t_{ij} \\ \textit{Tempo di servizio} \text{ di } i\text{: } t_i \end{cases}$$

- Devo **aggiungere vincoli**:

$$\sum_{i \in I} t_i \sum_{\substack{j \in I \\ j \neq i}} x_{ij}^k + \sum_{i \in I} \sum_{\substack{j \in I \\ j \neq i}} t_{ij} \, x_{ij}^k \leq T_k \qquad k \in J$$

il tempo complessivo di utilizzo del veicolo k non supera il limite

- Tempi di transito e/o di servizio potrebbero variare da veicolo a veicolo
- In tal caso ho t_{ij}^k e t_i^k

Finestre Temporali 1/2

- Se invece per il generico cliente j ho una **finestra temporale** [$iniz_j$ $fine_j$] (posso visitarlo solo dall'istante $iniz_j$ all'istante $fine_j$, sempre espresso in una opportuna unità di misura a partire dall'inizio delle operazioni

- Devo **considerare**: un nuovo insieme di variabili $a_j \in \mathrm{R}$ con $j \in I$ che rappresentano l'istante di arrivo dal cliente j (dette variabili temporali, ovviamente nella stessa unità di misura)

- Devo **aggiungere i vincoli**:

$$iniz_j \leq a_j \leq fine_j \qquad j \in I$$

giungo dal cliente j rispettando la sua finestra temporale

$$a_j \in \mathrm{R} \qquad j \in I$$

- Però questo non basta (!): devo anche **aggiungere vincoli** tra le a e le x, altrimenti troverò dei valori per le a che soddisfano tutte le finestre, ma che non è magari possibile ottenere con i percorsi fatti secondo le x

Finestre Temporali 2/2

- Vincoli da aggiungere che legano le a e le x:

se visito il cliente i all'istante a_i e subito dopo vado dal cliente j, deve avere:

$$a_j = a_i + t_i^k + t_{ij}^k$$

raggiungo il cliente j un istante che è dato da quando arrivo in i, più il tempo di servire i e transitare da i a j

e ciò deve valere sse $x_{ij}^k = 1$

allora separo = in ≤ e ≥ e aggiungo qualcosa che impone il vincolo sse $x_{ij}^k = 1$:

$$\left.\begin{array}{l} a_j \geq a_i + t_i^k + t_{ij}^k - (1 - x_{ij}^k)\, T \\ a_j \leq a_i + t_i^k + t_{ij}^k + (1 - x_{ij}^k)\, T \end{array}\right\} \quad i,j \in I \quad k \in J$$

T è un valore maggiore di tutti i possibili valori di a, e quindi se $x_{ij}^k = 0$ i vincoli sono banalmente soddisfatti, se invece $x_{ij}^k = 1$ i vincoli devono valere

Cenni a Ulteriori Varianti

- Se in alcuni nodi devo prelevare e in altri consegnare ciò che ho prelevato, si chiama **pickup and delivery** problem

devo imporre vincoli di precedenza tra l'instante in cui visito in cliente i in cui prelievo e l'istante in cui visito il cliente j a cui consegno

$$a_i \leq a_j \quad \text{per ogni coppia } i, j \text{ di pickup e delivery}$$

Oppure, se non ho le variabili temporali a_i, uso variabili intere u_i^k che rappresentano l'ordine di visita del nodo i col veicolo k, vincoli opportuni tra le u e vincoli che colleghino le u alle x (simili a quelli già visti per le a)

Inoltre servono variabili q_i^k che sono il carico totale del veicolo k alla partenza dal cliente i, in modo da imporre che il carico totale non superi mai la capacità

- Esiste anche il **Dial a Ride** Problem: bisogna soddisfare varie richieste di trasporto da un punto ad un altro (eventualmente giunte anche durante l'esecuzione): è il problema dei taxi

Come gestire i Vincoli di Sottocicli?

- I vincoli di "eliminazione" di sottocicli sono in numero esponeziale $\sum_{ij \in E(S)} x_{ij}^k \leq |S| - 1 \text{ per ogni } S \subset C_k$

- Semplifichiamo: consideriamo il caso singolo cluster di $G(V,E)$

$$(1) \quad \sum_{j=1}^{n} x_{ij} = 1 \text{ per ogni } i \in V$$

$$(2) \quad \sum_{j=1}^{n} x_{ji} = 1 \text{ per ogni } i \in V$$

$$(3) \quad \sum_{ij \in E(S)} x_{ij} \leq |S| - 1 \text{ per ogni } S \subset V$$

- Proprietà: x vettore d'incidenza di un ciclo hamiltoniano $\rightarrow$

$$\sum_{ij \in E} x_{ij} = |V|$$

- Ottenuto sommando le $|V|$ equazioni (1)
- Ogni ciclo hamiltoniano di G *contiene esattamente* $|V|$ *archi*.

Separazione Vincoli Sottocicli

- Oracolo di separazione: dato un punto x^* che soddisfa (1) decide *se esiste un vincolo* (3) *violato da* x^* *, i.e. trova* $S \subset V$, *tale che*

$$\sum_{ij \in E(S)} x_{ij}^* > |S| - 1 \qquad \text{(A)}$$

oppure conclude che nessun vincolo (3) è violato da x^*

Oss: sommando i vincoli (1) per $i \in S$ si ottiene

$$\sum_{ij \in E(S)} x_{ij}^* + \sum_{ij \in \delta(S)} x_{ij}^* = |S|$$

La condizione di violazione (A) è quindi equivalente a

$$\sum_{ij \in \delta(S)} x_{ij}^* < 1$$

Oracolo Vincoli Sottocicli

Oracolo di Separazione

Associa a ogni arco (i,j) capacità $c_{ij} = x_{ij}^*$

Per ogni coppia di nodi u,v

Calcola l'u-v taglio a capacità minima ($u \in W$, $v \notin W$)

Se c(W) < 1 → vincolo violato e STOP

Fine ciclo

(Per ogni arco (u,v) la condizione non è verificata)

→ *Non esiste un vincolo violato*

l'*u-v taglio* a capacità minima si trova risolvendo il problema di massimo flusso *u-v*.

Quindi risolvo un problema di massimo flusso per ogni scelta *u*,*v*. Se fisso *u*, ne devo risolvere uno per ogni scelta di *v*, cioè $|C_k|$-1

Algoritmi di Soluzione

- Metodi Esatti:

Branch-and-cut: con generazione di vincoli di sottociclo

raggiunge l'ottimo per problemi di dimensione contenuta, non riesce a risolvere problemi grandi

- Metodi Approssimati (Euristiche):

Euristiche costruttive: Greedy, Clarke & Wright

Costruttive in due fasi: Cluster First Route Second (CFRS)
o Route First Cluster Second (RFCS)

Euristiche migliorative: Ricerca Locale, Taboo Search

più usate per risolvere problemi pratici (spesso grandi dimensioni)

Euristiche Costruttive

- Clarke & Wright:

Passo 0: forma soluzione iniziale $\{T_1^0, \ldots, T_{n-1}^0\}$ con $T_i^0 = \{v_1, v_{i+1}\}$ (una rotta per cliente)

Passo *i*: per ogni coppia $\{T_h^i, T_k^i\}$ tale che la somma delle sue domande è $\leq$ Q (cioè rotte unibili senza violare capacità) calcola

$c_{last(h),1} + c_{1,first(k)} - c_{last(h),first(k)}$

(cioè il risparmio ottenuto fondendo la coppia: elimino l'utlimo arco della prima e il primo della seconda e aggiungo l'arco che collega le due rotte, può essere negativo) e scegli il massimo

Se il massimo risparmio è ≤ 0 STOP
altrimenti fondi le rotte corrispondenti al massimo, $i=i+1$ e vai al Passo *i*

Se il numero di veicoli è *m*, per avere una sol. ammissibile occorre fondere rotte anche se ho massimo negativo fino ad averne non più di *m*

Metodi Costruttivi Due Fasi

- Sweep (CFRS):

Si rappresentano tutti i clienti in un piano dove l'origine è il deposito

Si immagina un raggio uscente dal deposito che ruoti in modo da spazzare tutto il piano

I clienti progressivamente incontrati dal raggio vengono messi in un cluster finchè la somma delle loro domande non supera la capacità, oltre viene creato un nuovo cluster

Per ogni cluster si trova poi il routing con una delle euristiche per il TSP

- Clarke & Wright modificato:

Il risparmio viene calcolato considerando anche le permutazioni delle due rotte che potrei fondere, e periodicamente riottimizzo le rotte di ogni cluster con una euristica per il TSP

- RFCS:

Si risolve prima il TSP per tutti i clienti per mezzo di una euristica

Poi si spezza il ciclo Hamiltoniano trovato in modo da rispettare le capacità dei veicoli

Euristiche Migliorative

- Ricerca Locale:

Calcola un intorno di una soluzione ammissibile (vari possibili criteri: spostamento di cliente da un cluster a un altro, modifica routing)

Trova la miglior soluzione dell'intorno e passa ad essa se conveniente (come visto, ci si arresta in un minimo locale)

- Tabu Search (per non arrestarsi in minimi locali)

Inizializzazione: Definisci una soluzione ammissibile iniziale attraverso un metodo costruttivo

Iterazione: genera intorni eseguendo delle "mosse" (ad esempio spostamento di cliente da un cluster a un altro). Effettuare la "mossa" inversa viene vietato (è Tabu) per un certo numero di passi

Ci si sposta in una soluzione dell'intorno accettando anche, sotto opportune condizioni, dei peggioramenti, in modo da non fermarsi su minimi locali

Il Tabu serve ad evitare di tornare subito sulla soluzione precedente quando ho accettato un peggioramento

Postottimizzazione: riottimizza il routing di ogni cluster con euristica per TSP

Ovviamente, tante possibili varianti e miglioramenti dello schema descritto