

---

**La decomposizione di Dantzig Wolfe  
applicata al problema SSTDMA con vincolo  
di cardinalità**

---

# Il problema

---

Data una matrice  $D$  in input si ricerca una decomposizione di questa in  $q$  sottomatrici

$$n = 3, l = 2 \quad D = \begin{array}{|c|c|c|} \hline 10 & 25 & 12 \\ \hline 15 & 30 & 7 \\ \hline 20 & 15 & 35 \\ \hline \end{array} \Rightarrow \{D_1, D_2, \dots, D_q\}$$

Le sottomatrici individuate devono rispettare i seguenti vincoli:

- Ogni sottomatrice deve avere al più un elemento positivo in ogni riga
- Ogni sottomatrice deve avere al più un elemento positivo in ogni colonna
- Ogni sottomatrice deve avere al più  $1 \leq l \leq n$  elementi diversi da 0
- Ogni elemento della matrice  $D$  deve essere inserito in una e una sola sottomatrice

Ad ogni sottomatrice viene associato un costo uguale al valore massimo di un suo elemento. Fra tutte le decomposizioni si cerca quella per cui la somma dei costi associati alle sottomatrici è minimo.

# Il problema

---

$$n = 3, l = 2$$

$$D = \begin{array}{|c|c|c|} \hline 10 & 25 & 12 \\ \hline 15 & 30 & 7 \\ \hline 20 & 15 & 35 \\ \hline \end{array} \rightarrow \{D_1, D_2, \dots, D_q\}$$

10		
	<b>30</b>	

	25	
		<b>35</b>

15		
	<b>15</b>	

		12
<b>20</b>		

		<b>7</b>

# Il problema

$$n = 3, l = 2$$

$$D = \begin{array}{|c|c|c|} \hline 10 & 25 & 12 \\ \hline 15 & 30 & 7 \\ \hline 20 & 15 & 35 \\ \hline \end{array} \Rightarrow \{D_1, D_2, \dots, D_q\}$$

10		
	30	

	25	
		35

15		
	15	

		12
20		

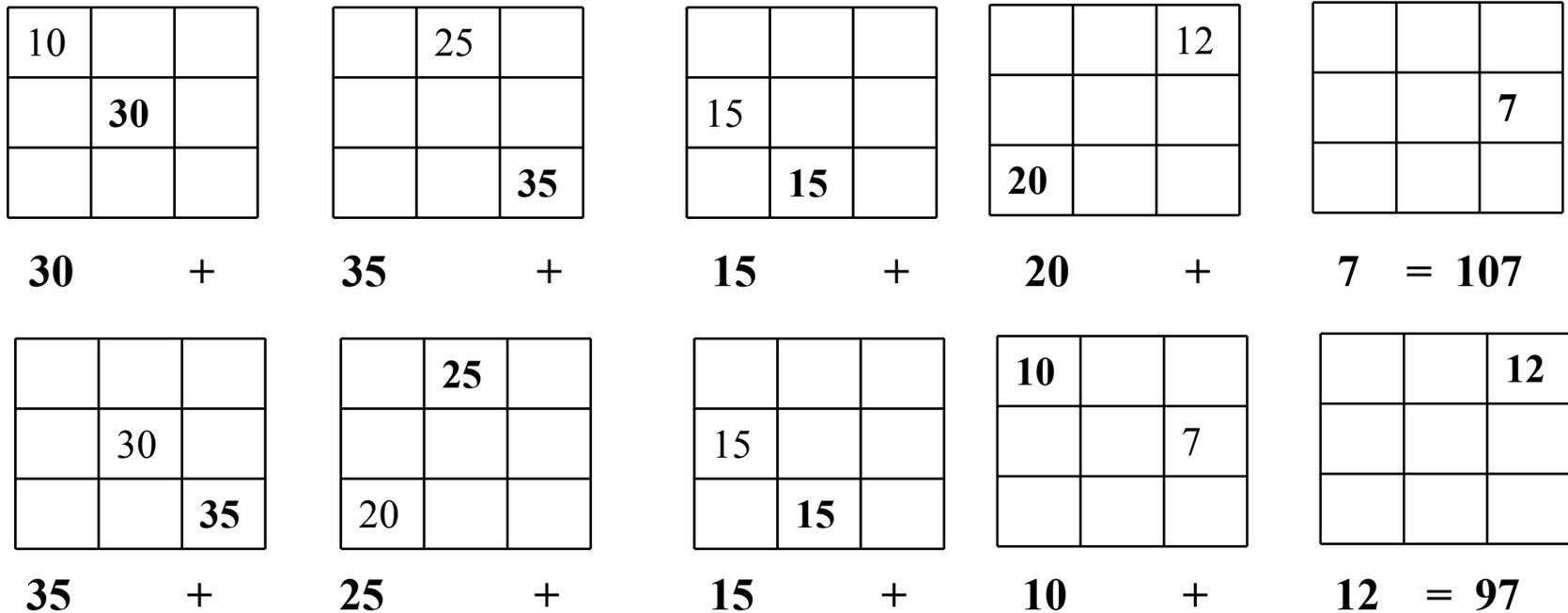
		7

$$30 + 35 + 15 + 20 + 7 = 107$$

# Il problema

$$n = 3, l = 2$$

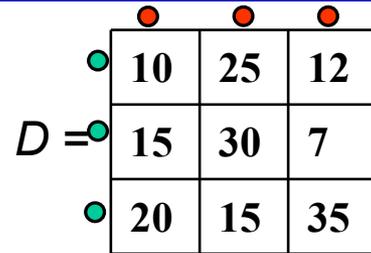
$$D = \begin{array}{|c|c|c|} \hline 10 & 25 & 12 \\ \hline 15 & 30 & 7 \\ \hline 20 & 15 & 35 \\ \hline \end{array} \rightarrow \{D_1, D_2, \dots, D_q\}$$



# Applicazione pratica

---

$$n = 3, l = 2$$



	●	●	●
●	10	25	12
●	15	30	7
●	20	15	35

## Satellite Switched Time Division Multiple Access

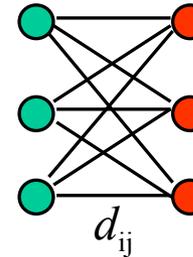
Usando lo schema SSTDMA un satellite divide le trasmissioni da effettuare in gruppi compatibili e soddisfa le richieste di un gruppo dopo l'altro. Prima di passare al gruppo successivo, aspetta che le trasmissioni del gruppo precedente siano terminate.

Se si considerano le righe della matrice  $D$  come stazioni di trasmissione, le colonne come stazioni di ricezione e il valore di un elemento come il tempo di trasmissione richiesto, il problema SSTDMA cerca la divisione in gruppi di costo minimo.

# Applicazione pratica

$$n = 3, l = 2$$

			
	10	25	12
	15	30	7
	20	15	35



Satellite  
Switched  
Time  
Division  
Multiple  
Access

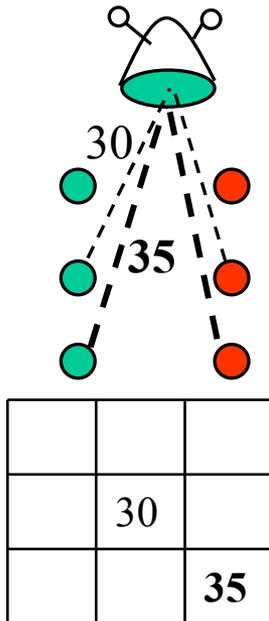
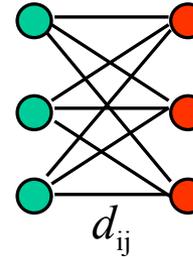
Usando lo schema SSTDMA un satellite divide le trasmissioni da effettuare in gruppi compatibili e soddisfa le richieste di un gruppo dopo l'altro. Prima di passare al gruppo successivo, aspetta che le trasmissioni del gruppo precedente siano terminate.

Se si considerano le righe della matrice  $D$  come stazioni di trasmissione, le colonne come stazioni di ricezione e il valore di un elemento come il tempo di trasmissione richiesto, il problema SSTDMA cerca la divisione in gruppi di costo minimo.

# Applicazione pratica

$n = 3, l = 2$

			
	10	25	12
	15	30	7
	20	15	35

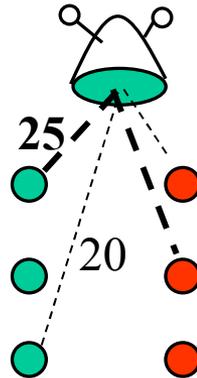
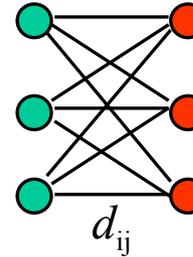


# Applicazione pratica

$n = 3, l = 2$

$D =$

	●	●	●
●	10	25	12
●	15	30	7
●	20	15	35



	30	
		35

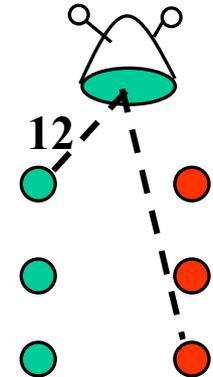
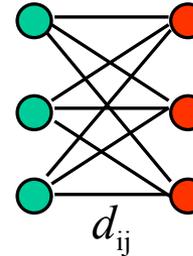
	25	
20		

# Applicazione pratica

$n = 3, l = 2$

$D =$

	●	●	●
●	10	25	12
●	15	30	7
●	20	15	35



	30	
		35

	25	
20		

15		
	15	

10		
		7

		12

35 + 25 + 15 + 10 + 12 = 97

# Lavori precedenti sul problema studiato

---

C.C. Ribeiro, M. Minoux and M.C. Penna, *An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment*, E.J.O.R., Vol. 41, 232-239 (1989).

M.Dell'Amico, F.Maffioli and M.T., *New Bounds for Optimum Traffic Assignment In Satellite Communication*, Computers Ops. Res., Vol. 25, 9, 729-743 (1998)

L.Taehan. and P.Sungsoo, *An integer programming approach to the time slot assignment problem in SS/TDMA systems with intersatellite links*, E.J.O.R., Vol. 135, 57- 66 (2001)

# Il modello: individuazione del massimo

$$\begin{aligned} P : z &= \min \sum_{k=1}^q z^k \\ d_{ij} x_{ij}^k &\leq z^k \quad \forall i, j, k \\ \sum_{i=1}^n x_{ij}^k &\leq 1 \quad \forall j, k \\ \sum_{j=1}^n x_{ij}^k &\leq 1 \quad \forall i, k \\ \sum_{i=1}^n \sum_{j=1}^n x_{ij}^k &\leq l \quad \forall k \\ \sum_{k=1}^q x_{ij}^k &= 1 \quad \forall i, j \\ x_{ij}^k &\in \{0, 1\} \quad \forall i, j, k \\ z^k &\geq 0 \quad \forall k \end{aligned}$$

$k$ -esimo frame

	25	
		35

$$z_k = 35$$

# Il modello: vincoli righe

$$P : z = \min \sum_{k=1}^q z^k$$

$$d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k$$

$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k$$

$$\sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k$$

$$\sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$$

$$z^k \geq 0 \quad \forall k$$

*k*-esimo frame

	25	
		35

Il numero  $q$  di frame non viene fornito dal problema: per semplicità può essere fissato ad  $n^2$ . In questo caso, nella soluzione ottima ci saranno molti frame completamente vuoti

# Il modello: vincoli colonne

$$P : z = \min \sum_{k=1}^q z^k$$

$$d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k$$

$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k$$

$$\sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k$$

$$\sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$$

$$z^k \geq 0 \quad \forall k$$

*k*-esimo frame

	25	
		35

# Il modello: vincoli cardinalità

$$P : z = \min \sum_{k=1}^q z^k$$

$$d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k$$

$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k$$

$$\sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k$$

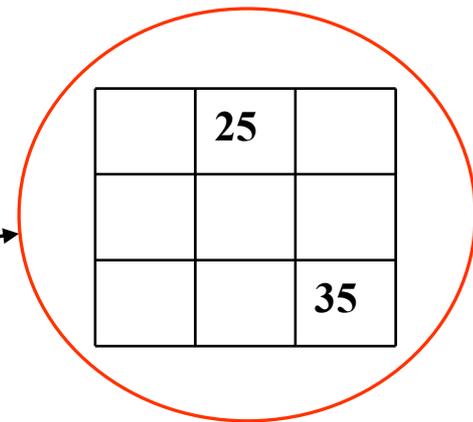
$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k$$

$$\sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$$

$$z^k \geq 0 \quad \forall k$$

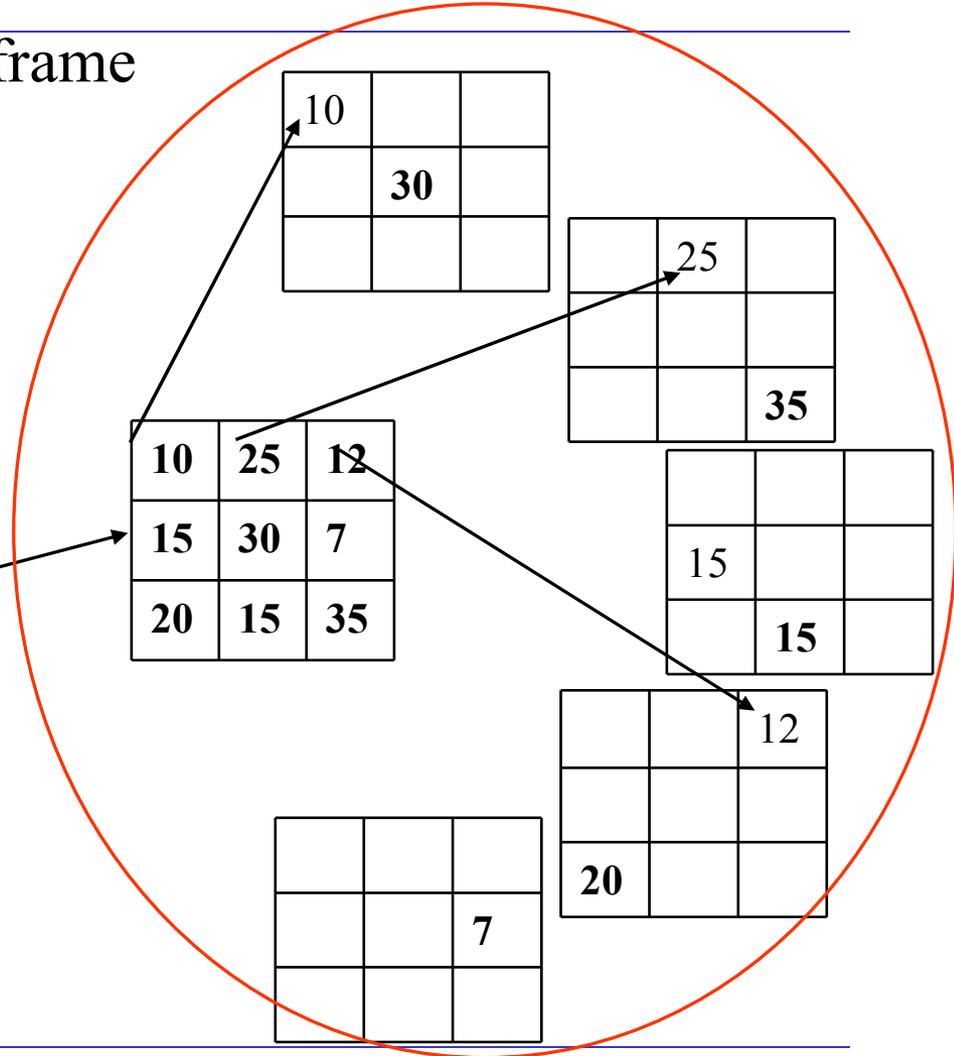
*k*-esimo frame



# Il modello: vincoli copertura

$k$ -esimo frame

$$\begin{aligned}
 P : z &= \min \sum_{k=1}^q z^k \\
 d_{ij} x_{ij}^k &\leq z^k \quad \forall i, j, k \\
 \sum_{i=1}^n x_{ij}^k &\leq 1 \quad \forall j, k \\
 \sum_{j=1}^n x_{ij}^k &\leq 1 \quad \forall i, k \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij}^k &\leq l \quad \forall k \\
 \sum_{k=1}^q x_{ij}^k &= 1 \quad \forall i, j \\
 x_{ij}^k &\in \{0, 1\} \quad \forall i, j, k \\
 z^k &\geq 0 \quad \forall k
 \end{aligned}$$



# Lower Bounds: immediati

$$\begin{aligned}
 P : z &= \min \sum_{k=1}^q z^k & L01 &= \frac{\sum_{i=1}^n \sum_{j=1}^n d_{ij}}{l} = 85 \\
 d_{ij} x_{ij}^k &\leq z^k \quad \forall i, j, k \\
 \sum_{i=1}^n x_{ij}^k &\leq 1 \quad \forall j, k & L02 &= \max \left( \max_{i=1..n} \sum_{j=1}^n d_{ij}, \max_{j=1..n} \sum_{i=1}^n d_{ij} \right) = 70 \\
 \sum_{j=1}^n x_{ij}^k &\leq 1 \quad \forall i, k \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij}^k &\leq l \quad \forall k \\
 \sum_{k=1}^q x_{ij}^k &= 1 \quad \forall i, j \\
 x_{ij}^k &\in \{0, 1\} \quad \forall i, j, k \\
 z^k &\geq 0 \quad \forall k
 \end{aligned}$$

10	25	12
15	30	7
20	15	35

# Lower Bounds: eliminazione di vincoli

$$P : z = \min \sum_{k=1}^q z^k$$

$$d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k$$

$$L01 = \frac{\sum_{i=1}^n \sum_{j=1}^n d_{ij}}{l} = 85$$

10	25	12
15	30	7
20	15	35

~~$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k$$~~

$$L02 = \max \left( \max_{i=1..n} \sum_{j=1}^n d_{ij}, \max_{j=1..n} \sum_{i=1}^n d_{ij} \right) = 70$$

~~$$\sum_{j=1}^n x_{ij}^k < 1 \quad \forall i, k$$~~

$$L1 = 94$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k$$

$$\sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$$

$$z^k \geq 0 \quad \forall k$$

Dell'Amico et al. '98

# Lower Bounds: eliminazione di vincoli

$$\begin{aligned}
 P : z &= \min \sum_{k=1}^q z^k \\
 d_{ij} x_{ij}^k &\leq z^k \quad \forall i, j, k \\
 \sum_{i=1}^n x_{ij}^k &\leq 1 \quad \forall j, k
 \end{aligned}$$

$$L01 = \frac{\sum_{i=1}^n \sum_{j=1}^n d_{ij}}{l} = 85$$

10	25	12
15	30	7
20	15	35

$$L02 = \max \left( \max_{i=1..n} \sum_{j=1}^n d_{ij}, \max_{j=1..n} \sum_{i=1}^n d_{ij} \right) = 70$$

~~$$\sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k$$~~

$$L1 = 94$$

$$L2^c = 75$$

~~$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k$$~~

$$\begin{aligned}
 \sum_{k=1}^q x_{ij}^k &= 1 \quad \forall i, j \\
 x_{ij}^k &\in \{0, 1\} \quad \forall i, j, k \\
 z^k &\geq 0 \quad \forall k
 \end{aligned}$$

Dell'Amico et al. '98

# Lower Bounds: eliminazione di vincoli

$$P : z = \min \sum_{k=1}^q z^k$$

$$d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k$$

~~$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k$$~~

$$\sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k$$

~~$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k$$~~

$$\sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$$

$$z^k \geq 0 \quad \forall k$$

$$L01 = \frac{\sum_{i=1}^n \sum_{j=1}^n d_{ij}}{l} = 85$$

10	25	12
15	30	7
20	15	35

$$L02 = \max \left( \max_{i=1..n} \sum_{j=1}^n d_{ij}, \max_{j=1..n} \sum_{i=1}^n d_{ij} \right) = 70$$

$$L1 = 94$$

$$L2^c = 75$$

$$L2^R = 70$$

Dell'Amico et al. '98

# Lower Bounds: combinazione

10	25	12
15	30	7
20	15	35

$$P : z = \min \sum_{k=1}^q z^k \quad L01 = \frac{\sum_{i=1}^n \sum_{j=1}^n d_{ij}}{l} = 85$$

$$d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k$$

$$\sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k \quad L02 = \max \left( \max_{i=1..n} \sum_{j=1}^n d_{ij}, \max_{j=1..n} \sum_{i=1}^n d_{ij} \right) = 70$$

$$\sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k \quad L1 = 94$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k \quad L2^C = 75$$

$$\sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j \quad L2^R = 70$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j, k$$

$$z^k \geq 0 \quad \forall k$$

$$\widehat{LB} = \sum_{k=1}^q \max (L1_k, L2_k^C, L2_k^R) = 94$$

Dell'Amico et al. '98

# Lower Bounds: lagrangiano

$$\begin{array}{l}
 P : z = \min \sum_{k=1}^q z^k \\
 d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k \\
 \sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k \\
 \sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k \\
 \sum_{k=1}^q x_{ij}^k = 1 \quad \forall i, j \\
 x_{ij}^k \in \{0, 1\} \quad \forall i, j, k \\
 z^k \geq 0 \quad \forall k
 \end{array}$$

$$\begin{array}{l}
 L : z = \min \sum_{k=1}^q z^k - \sum_{i=1}^n \sum_{j=1}^n \lambda_{ij} \left( \sum_{k=1}^q x_{ij}^k - 1 \right) \\
 d_{ij} x_{ij}^k \leq z^k \quad \forall i, j, k \\
 \sum_{i=1}^n x_{ij}^k \leq 1 \quad \forall j, k \\
 \sum_{j=1}^n x_{ij}^k \leq 1 \quad \forall i, k \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij}^k \leq l \quad \forall k \\
 x_{ij}^k \in \{0, 1\} \quad \forall i, j, k \\
 z^k \geq 0 \quad \forall k
 \end{array}$$

# Lower Bounds: lagrangiano

---

Se si riscrive la f.o. si ottiene:

$$\sum_{k=1}^q \left( z^k - \sum_{i=1}^n \sum_{j=1}^n \lambda_{ij} x_{ij}^k \right) + \sum_{i=1}^n \sum_{j=1}^n \lambda_{ij}$$

Da cui risulta evidente che il lower bound può essere ottenuto risolvendo il problema a fianco e moltiplicando il risultato per q

Questo problema riguarda la ricerca di un frame ammissibile che minimizza una certa f.o. funzione dei moltiplicatori lagrangiani.

$$LSP : w = \min z - \sum_{i=1}^n \sum_{j=1}^n \lambda_{ij} x_{ij}$$

$$d_{ij} x_{ij} \leq z \quad i, j = 1..n$$

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

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

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} \leq l$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1..n$$

$$z \geq 0$$

# Decomposizione di Dantzig-Wolfe per PLI: una sola famiglia di vincoli

---

$$P : z = \min c^T \underline{x}$$

$$A\underline{x} \geq \underline{b}$$

$$D\underline{x} \geq \underline{d}$$

$$\underline{x} \in \mathbb{N}^n$$

Un problema di PLI complesso può essere risolto decomponendo il problema in sottoproblemi più semplici.

Ad esempio è possibile considerare i vincoli  $D\underline{x} \geq \underline{d}$  separatamente

Se l'insieme dei punti che soddisfano i vincoli che si vogliono decomporre è finito allora ogni punto può essere ottenuto da una combinazione convessa binaria degli altri punti

$$X = \{\underline{x}^1, \underline{x}^2, \dots, \underline{x}^Q\} = \left\{ \underline{x} \in \mathbb{N}^n \mid \underline{x} = \sum_{t=1}^Q \lambda^t \underline{x}^t, \sum_{t=1}^Q \lambda^t = 1, \lambda^t \in \{0, 1\} \right\}$$

# Decomposizione di Dantzig-Wolfe per PLI: una sola famiglia di vincoli

---

I punti ottenuti da questa combinazioni di punti di  $X$  possono essere inseriti nel modello eliminando i vincoli decomposti:

$$\begin{aligned} MP : z &= \min \underline{c}^T \sum_{t=1}^Q \lambda^t \underline{x}^t \\ A \sum_{t=1}^Q \lambda^t \underline{x}^t &\geq \underline{b} \\ \sum_{t=1}^Q \lambda^t &= 1 \\ \lambda^t &\in \{0, 1\} \end{aligned}$$

In questo modo ottengo un nuovo problema (nelle variabili lambda) con meno vincoli.

## PROBLEMA:

In questo modo ho aumentato il numero di variabili: ho una variabile per ogni elemento dell'insieme  $X$ , che potrebbe avere una cardinalità enorme.

# Decomposizione di Dantzig-Wolfe per PLI: più famiglie di vincoli

Il ragionamento fatto per una famiglia di vincoli può essere generalizzato se i dati del problema presentano una struttura “diagonale”:

$$\begin{aligned}
 P : z &= \min \underline{c}^T \underline{x} \\
 M \underline{x} &\geq \underline{n} \\
 \underline{x} &\in \mathbb{N}^n
 \end{aligned}$$

$$M = \begin{bmatrix} A_1 & A_2 & \dots & A_k \\ L_1 & 0 & \dots & 0 \\ 0 & L_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & L_k \end{bmatrix} = \begin{bmatrix} A \\ D_1 \\ D_2 \\ \vdots \\ D_k \end{bmatrix} \quad \underline{n} = \begin{bmatrix} \underline{b} \\ \underline{d}_1 \\ \underline{d}_2 \\ \vdots \\ \underline{d}_k \end{bmatrix} \quad \underline{c} = \begin{bmatrix} \underline{c}_1 \\ \underline{c}_2 \\ \underline{c}_3 \\ \vdots \\ \underline{c}_k \end{bmatrix}$$

Con una matrice così strutturata ogni famiglia di vincoli del tipo  $D_p \underline{x}_p \geq \underline{d}_p$  agisce solo su una parte delle componenti del vettore  $\underline{x}$ . Gli unici vincoli che “legano” le varie parti del problema sono quelli individuati dalla matrice  $A$ .

# Decomposizione di Dantzig-Wolfe per PLI: più sottoproblemi

Data questa interindipendenza fra i vincoli della diagonale, è possibile risolvere uno alla volta i  $k$  problemi derivati dai vincoli sulla diagonale:

$$X_p = \{ \underline{x}_p^1, \underline{x}_p^2, \dots, \underline{x}_p^Q, \dots \} = \{ \underline{x}_p \in \mathbb{N}^n \mid D_p \underline{x}_p \geq \underline{d}_p \}, \quad p = 1, \dots, k$$

Anche in questo caso, se i vari sottoproblemi ammettono un insieme finito di soluzioni posso ottenere tutte le soluzioni ammissibile con una combinazione lineare di queste:

$$X_p = \{ \underline{x}_p^1, \underline{x}_p^2, \dots, \underline{x}_p^Q \} = \left\{ \underline{x} \in \mathbb{N}^n \mid \underline{x} = \sum_{t=1}^Q \lambda^t \underline{x}_p^t, \sum_{t=1}^Q \lambda^t = 1, \lambda^t \in \{0, 1\} \right\}$$

Grazie alla struttura diagonale dei vincoli, l'insieme dei punti che appartengono a tutti gli insiemi  $X$  si ottengono sommando le soluzioni dei sottoproblemi:

$$\bigcap_{p=1}^k X_p = \left\{ \sum_{p=1}^k \underline{x}_p \mid \underline{x}_p \in X_p \right\} = \left\{ \sum_{p=1}^k \sum_{t=1}^Q \lambda_p^t \underline{x}_p^t \mid \sum_{t=1}^Q \lambda_p^t = 1, \lambda_p^t \in \{0, 1\}, p = 1, \dots, k \right\}$$

# Decomposizione di Dantzig-Wolfe per PLI: più sottoproblemi

---

Individuata quindi la forma delle soluzioni ammissibili per tutti i problemi decomposti, la si può inserire nel problema originale eliminando i vincoli diagonali:

$$\begin{aligned} MP : z &= \min \underline{c}^T \sum_{p=1}^k \sum_{t=1}^Q \lambda_p^t \underline{x}_p^t \\ A \sum_{p=1}^k \sum_{t=1}^Q \lambda_p^t \underline{x}_p^t &\geq \underline{b} \\ \sum_{t=1}^Q \lambda_p^t &= 1 \quad \forall p \\ \lambda_p^t &\in \{0, 1\} \quad \forall t, p \end{aligned}$$

# Decomposizione di Dantzig-Wolfe per PLI: più sottoproblemi

Se le varie parti della matrice diagonale sono identiche e quindi il problema ha una struttura del tipo:

$$M = \begin{bmatrix} \hat{A} & \hat{A} & \dots & \hat{A} \\ \hat{L} & 0 & \dots & 0 \\ 0 & \hat{L} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{L} \end{bmatrix} \quad \underline{n} = \begin{bmatrix} \underline{b} \\ \underline{\hat{d}} \\ \underline{\hat{d}} \\ \vdots \\ \underline{\hat{d}} \end{bmatrix} \quad \underline{c} = \begin{bmatrix} \underline{\hat{c}} \\ \underline{\hat{c}} \\ \underline{\hat{c}} \\ \vdots \\ \underline{\hat{c}} \end{bmatrix} \quad \begin{array}{l} \hat{A} \in \mathbb{Q}^{z \times w} \\ \hat{L} \in \mathbb{Q}^{m \times w} \\ \underline{\hat{d}} \in \mathbb{Q}^m \\ \underline{\hat{c}} \in \mathbb{Q}^w \end{array}$$

E' necessario risolvere soltanto un sottoproblema:

$$\hat{X} = \{ \hat{x}^1, \hat{x}^2, \dots, \hat{x}^Q \} = \left\{ \underline{\hat{x}} \in \mathbb{N}^w \mid \hat{L} \underline{\hat{x}} \geq \underline{\hat{d}} \right\}$$

**Nota:** il sottoproblema viene risolto nella dimensione  $w < n$  ( $n=w*k$ )

# Decomposizione di Dantzig-Wolfe per PLI: più sottoproblemi

Un vettore  $\underline{x}$  che è ammissibile per tutti i vincoli decomposti avrà la forma seguente:

$$\underline{x} = [\underline{\hat{x}}_1, \underline{\hat{x}}_2, \dots, \underline{\hat{x}}_k]^T \quad \underline{\hat{x}}_p \in \hat{X}$$

Dal momento che ogni  $\underline{\hat{x}}_p$  si ottiene dalla combinazione degli elementi di  $\hat{X}$ , posso riscrivere la funzione obiettivo e il vincolo legante del problema nel seguente modo:

$$\hat{c}^T \sum_{p=1}^k \underline{\hat{x}}_p = \hat{c}^T \sum_{p=1}^k \sum_{t=1}^Q \lambda_p^t \underline{\hat{x}}^t \qquad \hat{A} \sum_{p=1}^k \underline{\hat{x}}_p = \hat{A} \sum_{p=1}^k \sum_{t=1}^Q \lambda_p^t \underline{\hat{x}}^t \geq \underline{b}$$

A questo punto se si invertono fra loro le 2 sommatorie e per ogni elemento di  $\hat{X}$  si considera un unico coefficiente aggregato  $\hat{\lambda}^t = \sum_{p=1}^k \lambda_p^t$ , si ottiene:

$$\hat{c}^T \sum_{t=1}^Q \underline{\hat{x}}^t \sum_{p=1}^k \lambda_p^t = \hat{c}^T \sum_{t=1}^Q \underline{\hat{x}}^t \hat{\lambda}^t \qquad \hat{A} \sum_{t=1}^Q \underline{\hat{x}}^t \sum_{p=1}^k \lambda_p^t = \hat{A} \sum_{t=1}^Q \underline{\hat{x}}^t \hat{\lambda}^t \geq \underline{b}$$

# Decomposizione di Dantzig-Wolfe per PLI: più sottoproblemi

---

Se si eliminano dal problema iniziale i vincoli decomposti, si modificano la f.o. e il vincolo legante come fatto precedentemente e i  $p$  vincoli di convessità sui coefficienti  $\lambda$  vengono sostituiti da un vincolo aggregato è possibile ottenere una forma estesa del problema:

$$\begin{aligned} MP : z &= \min \underline{\hat{c}}^T \sum_{t=1}^Q \hat{\lambda}^t \underline{\hat{x}}^t \\ \hat{A} \sum_{t=1}^Q \hat{\lambda}^t \underline{\hat{x}}^t &\geq \underline{b} \\ \sum_{t=1}^Q \hat{\lambda}^t &= k \\ \hat{\lambda}^t &\in \mathbb{N}_0 \quad t = 1, \dots, Q \end{aligned}$$

- Anche in questo caso sono stati eliminati molti vincoli ma il numero di variabili è aumentato.
- Il vincolo di convessità può essere eliminato se il numero di sottoproblemi non è un parametro del problema.
- Il rilassamento lineare ottenuto da questa formulazione sarà intuitivamente migliore perchè il vincolo di integrità sui problema decomposto sono in parte già soddisfatti..

# Decomposizione di Dantzig-Wolfe per PLI: applicazione a SSTDMA

---

Il problema SSTDMA presenta una evidente struttura diagonale:

- Ogni elemento della diagonale impone che uno dei frame sia ammissibile
- I vincoli di copertura sono i vincoli leganti
- Il numero dei frame non è un parametro fissato del problema

Di conseguenza il sottoproblema derivato dai vincoli sui frame è:

$$\hat{F} = \left\{ (z \in \mathbb{R}^+, \underline{x} \in \{0, 1\}^{n^2}) \mid \sum_{i=1}^n x_{\rho(i,j)} \leq 1, \sum_{j=1}^n x_{\rho(i,j)} \leq 1, \sum_{i=1}^n \sum_{j=1}^n x_{\rho(i,j)} \leq l, d_{ij} x_{\rho(i,j)} \leq z \right\}$$
$$= \left\{ (z^1, \underline{x}^1), (z^2, \underline{x}^2), \dots, (z^Q, \underline{x}^Q) \right\}$$

Dove è stata definita la funzione  $\rho(i, j) = i + n(j - 1)$  per “rimappare” la matrice dei frame in una colonna in modo da gestire i dati in maniera più semplice

# Decomposizione di Dantzig-Wolfe per PLI: applicazione a SSTDMA

Applicando la decomposizione di Dantzig-Wolfe illustrata precedentemente si ottiene:

$$\begin{aligned} MP : z &= \min \sum_{t=1}^Q \hat{\lambda}^t z^t \\ \sum_{t=1}^Q \hat{\lambda}^t \underline{x}^t &= \underline{1} \\ \sum_{t=1}^Q \hat{\lambda}^t &= q \\ \hat{\lambda}^t &\in \mathbb{N}_0 \quad t = 1, \dots, Q \end{aligned}$$

Nella formulazione iniziale del problema SSTDMA il numero  $q$  di frame non era fissato e quindi è possibile eliminare il vincolo di convessità (l'ottimizzazione della f.o. porterà alla scelta migliore di  $q$ ). A causa del vincolo di copertura e visto che il problema è di minimo, soluzioni in cui vengono presi più volte determinati frame non possono essere ottime: posso limitare il dominio delle  $\lambda$  a  $\{0,1\}$

Tenendo conto di queste osservazioni, si ottiene una formulazione estesa del problema

# Formulazione estesa: set partitioning

$$\begin{aligned}MP : z &= \min \underline{c}^T \underline{\lambda} \\ A\underline{\lambda} &= 1 \\ \lambda_i &\in \{0, 1\}\end{aligned}$$

A=

1
0
0
0
1
0
0
0
0

10		
	30	

# Formulazione estesa: set partitioning

$$\begin{aligned}MP : z &= \min \underline{c}^T \underline{\lambda} \\ A\underline{\lambda} &= 1 \\ \lambda_i &\in \{0, 1\}\end{aligned}$$

A=

1
0
0
0
1
0
0
0
0
0

10		
	30	

# Formulazione estesa: set partitioning

$$\begin{aligned}MP : z &= \min \underline{c}^T \underline{\lambda} \\ A\underline{\lambda} &= 1 \\ \lambda_i &\in \{0, 1\}\end{aligned}$$

A=

1
0
0
0
1
0
0
0
0

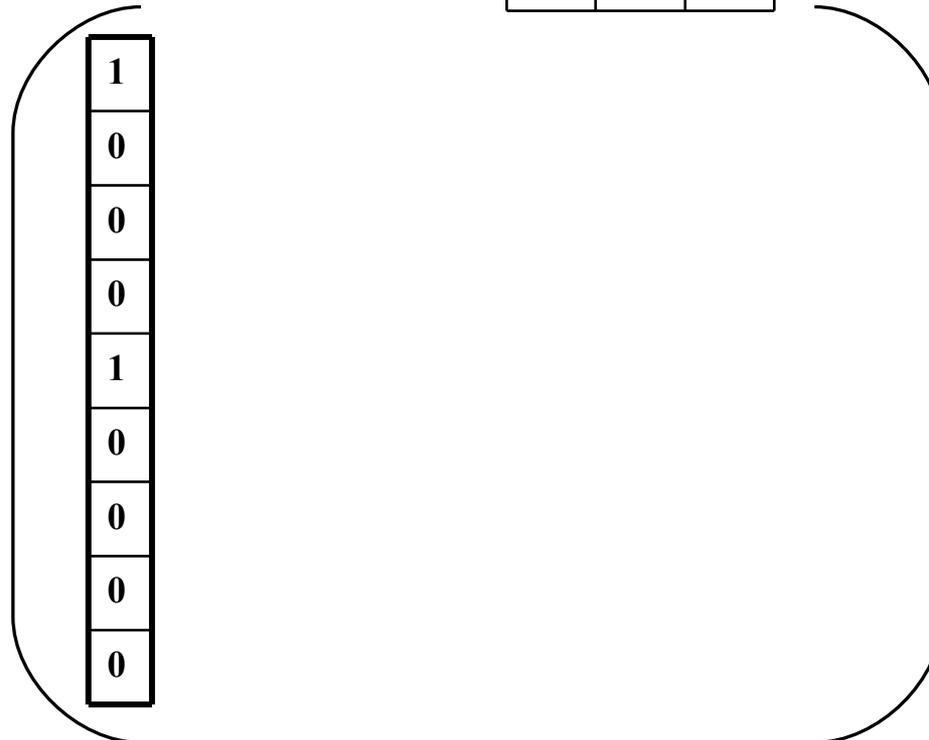
10		
	30	

# Formulazione estesa: set partitioning

$$\begin{aligned}
 MP : z &= \min \underline{c}^T \underline{\lambda} \\
 A\underline{\lambda} &= 1 \quad c^T = (30, \dots) \\
 \lambda_i &\in \{0, 1\}
 \end{aligned}$$

10		
	30	

A=



# Formulazione estesa: set partitioning

$$\begin{aligned}
 MP : z &= \min \underline{c}^T \underline{\lambda} \\
 A\underline{\lambda} &= 1 \\
 \lambda_i &\in \{0, 1\}
 \end{aligned}$$

$$c^T = (30, 35, \dots)$$

	25	
		35

A=

1	0
0	0
0	0
0	1
1	0
0	0
0	0
0	0
0	1

# Formulazione estesa: set partitioning

$$\begin{aligned}
 MP : z &= \min \underline{c}^T \underline{\lambda} \\
 A\underline{\lambda} &= 1 \\
 \lambda_i &\in \{0, 1\}
 \end{aligned}$$

$$c^T = (30, 35, \dots, 12)$$

$$A = \left( \begin{array}{cc|cc}
 1 & 0 & & \\
 0 & 0 & & \\
 0 & 0 & & \\
 0 & 1 & & \\
 1 & 0 & & \\
 0 & 0 & & \\
 0 & 0 & & \\
 0 & 0 & & \\
 0 & 1 & & \\
 \hline
 & & \dots & \\
 & & & \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{array}
 \end{array} \right)$$

# Formulazione estesa: set covering

---

Dal momento che la funzione obiettivo è di minimo e visto che togliendo un elemento ad un frame si può al più diminuire il costo di trasmissione, allora è possibile trasformare il problema in un problema di set covering. Se nella soluzione ottima ci sono frame che non rispettano il vincolo di uguaglianza, sarà sufficiente togliere da questi gli elementi in eccesso.

$$\begin{aligned} MP : z &= \min \underline{c}^T \underline{\lambda} \\ A\underline{\lambda} &\geq 1 \\ \lambda_i &\in \{0, 1\} \end{aligned}$$

La formulazione come set covering presenta principalmente due vantaggi:

- Le variabili duali del problema sono vincolate in segno
- La generazione di una soluzione ammissibile è molto più semplice

# Column generation: rilassamento lineare

$$MP : z = \min \underline{c}^T \underline{\lambda}$$

$$A \underline{\lambda} \geq 1$$

$$\underline{c}^T = (30, 35, \dots$$

...

$$, 12)$$

~~$$\lambda_i \in \{0, 1\}$$~~

$$\lambda_i \in [0, 1]$$

A =

1	0
0	0
0	0
0	1
1	0
0	0
0	0
0	0
0	0
0	1

...

0
0
0
0
0
0
0
1
0
0

# Column generation: rilassamento lineare del problema ridotto

---

Per affrontare il problema derivato dall'alto numero di colonne della matrice della formulazione estesa (sarebbe difficile soltanto enumerare tutti i frame ammissibili), si utilizza la tecnica di generazione di colonne. Si parte risolvendo un problema con una matrice ridotta (contenente solo le colonne necessarie a garantire una soluzione ammissibile al problema) e quindi un vettore dei costi ridotto.

$$\begin{aligned}MP : z &= \min \underline{c}_r^T \underline{\lambda} \\ A_r \underline{\lambda} &\geq 1 \\ \lambda_i &\in [0, 1]\end{aligned}$$

Una volta risolto il problema ridotto si deve verificare se la soluzione ottenuta è ottima anche per il problema completo. Per fare questo è necessario dimostrare che tutte le variabili fuori base (anche quelle non presenti nella matrice ridotta) hanno un coefficiente di costo ridotto non negativo

# Il problema di pricing

$$\begin{aligned}
 PR : w &= \min z - \sum_{i=1}^n \sum_{j=1}^n p_{ij} x_{ij} \\
 d_{ij} x_{ij} &\leq z \\
 \sum_{i=1}^n x_{ij} &\leq 1 \quad \forall j \\
 \sum_{j=1}^n x_{ij} &\leq 1 \quad \forall i \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij} &\leq l \\
 x_{ij} &\in \{0, 1\} \quad \forall i, j \dots n \\
 z &\geq 0
 \end{aligned}$$

Per testare i coefficienti di costo ridotto delle colonne non presenti nella matrice ridotta, si può risolvere il cosiddetto problema di pricing: fra tutte le colonne ammissibili cerco quella col coefficiente di costo ridotto minimo. Se la soluzione ottima del problema di pricing è non negativa allora la soluzione ottima del problema ridotto è ottima per il problema completo, altrimenti si inserisce la colonna trovata nella matrice ridotta e si risolve iterativamente il nuovo problema così ottenuto. Per il problema SSTDMA il problema di pricing diventa il problema di trovare il frame ammissibile con coefficiente di costo ridotto minimo.

Dato il vettore  $\underline{u}$  delle variabili duali del problema ridotto si definisce  $p_{ij} = u_{i+(j-1)n}$

# Il problema di pricing: tecnica di soluzione

$$\begin{array}{l}
 PR : w = \min z - \sum_{i=1}^n \sum_{j=1}^n p_{ij} x_{ij} \\
 d_{ij} x_{ij} \leq z \\
 \sum_{i=1}^n x_{ij} \leq 1 \quad \forall j \\
 \sum_{j=1}^n x_{ij} \leq 1 \quad \forall i \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij} \leq l \\
 x_{ij} \in \{0, 1\} \quad \forall i, j \dots n \\
 z \geq 0
 \end{array}
 \quad
 \begin{array}{c}
 \text{PR}_{hk} : (d_{hk} - p_{hk} - \bar{w}) \\
 \text{FPR} : \bar{w} = \max \sum_{i=1}^n \sum_{j=1}^n p_{ij}^{(h,k)} x_{ij} \\
 \sum_{i=1}^n x_{ij} \leq 1 \quad j = 1..n \\
 \sum_{j=1}^n x_{ij} \leq 1 \quad i = 1..n \\
 \sum_{i=1}^n \sum_{j=1}^n x_{ij} = l - 1 \\
 x_{ij} \in \{0, 1\} \quad i, j = 1..n
 \end{array}$$



Vincolo  
l'elemento  
(h,k) ad essere  
l'elemento  
massimo

La matrice  $P^{(h,k)}$  è definita in modo che sia possibile evitare di prendere elementi incompatibili con (h,k)

# Il problema di pricing

---

Ogni problema  $PR_{hk}$  si riduce a un problema di tipo  $k$ -cardinality Assignment Problem con  $k=l-1$  ( $(l-1)$ -LAP)

M.Dell'Amico and S. Martello, *The  $k$ -cardinality Assignment Problem*, D.A.M., Vol 76, 103-121 (1997)

A. Volgenant, *Solving the  $k$ -cardinality assignment problem by transformation*, E.J.O.R., Vol.157, 322-331(2004)

Il problema KLAP si riduce a un problema di assegnamento lineare di dimensione  $(2n-k)$ .

$$\begin{aligned} FPR : \bar{w} &= \max \sum_{i=1}^n \sum_{j=1}^n p_{ij}^{(h,k)} x_{ij} \\ \sum_{i=1}^n x_{ij} &\leq 1 \quad j = 1..n \\ \sum_{j=1}^n x_{ij} &\leq 1 \quad i = 1..n \\ \sum_{i=1}^n \sum_{j=1}^n x_{ij} &= l - 1 \\ x_{ij} &\in \{0, 1\} \quad i, j = 1..n \end{aligned}$$

# Il problema di pricing

---

Si può evitare di risolvere un determinato problema  $P_{hk}$  se si conosce il valore di una soluzione ammissibile  $v$ , un UB del problema e vale la relazione  $v \leq d_{hk} - UB$ .

Possibili upper bound:

- Rilassamento dei vincoli (a) o (b)  $O(n^2)$
- Risolvo il problema sulla matrice  $P$  invece che sulla matrice  $P^{(h,k)}$ . Il calcolo viene effettuato una volta per tutti gli elementi  $(h,k)$

$$FPR : \bar{w} = \max \sum_{i=1}^n \sum_{j=1}^n p_{ij}^{(h,k)} x_{ij}$$

$$\sum_{i=1}^n x_{ij} \leq 1 \quad j = 1..n \quad (a)$$

$$\sum_{j=1}^n x_{ij} \leq 1 \quad i = 1..n \quad (b)$$

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} = l - 1$$

$$x_{ij} \in \{0, 1\} \quad i, j = 1..n$$

# Gestione delle colonne

---

Il *pool* delle colonne è gestito attraverso tecniche di *hashing*

Si mantengono attive al più 5000 colonne

Tra le colonne attive, se possibile, vengono inserite tutte quelle con un costo ridotto negativo

E' stata applicata, senza buoni risultati, la tecnica di stabilizzazione:

L.M. Rousseau, M.Gendreau, D. Feillet *Interior Point Stabilization for Column Generation*,  
Operations Research Letters, 35, 660-668 (2007)

E' stato applicato un criterio di dominanza fra le colonne

# Branch and price

---

Una volta terminata la generazione delle colonne non ho ancora ottenuto la soluzione ottima (intera) del problema, ma solo un lower bound. Per ottenere la soluzione ottima si può utilizzare la tecnica di branch and bound utilizzando il lower bound fornito dal metodo di column generation. Questo tipo particolare di branch and bound viene chiamato branch and price.

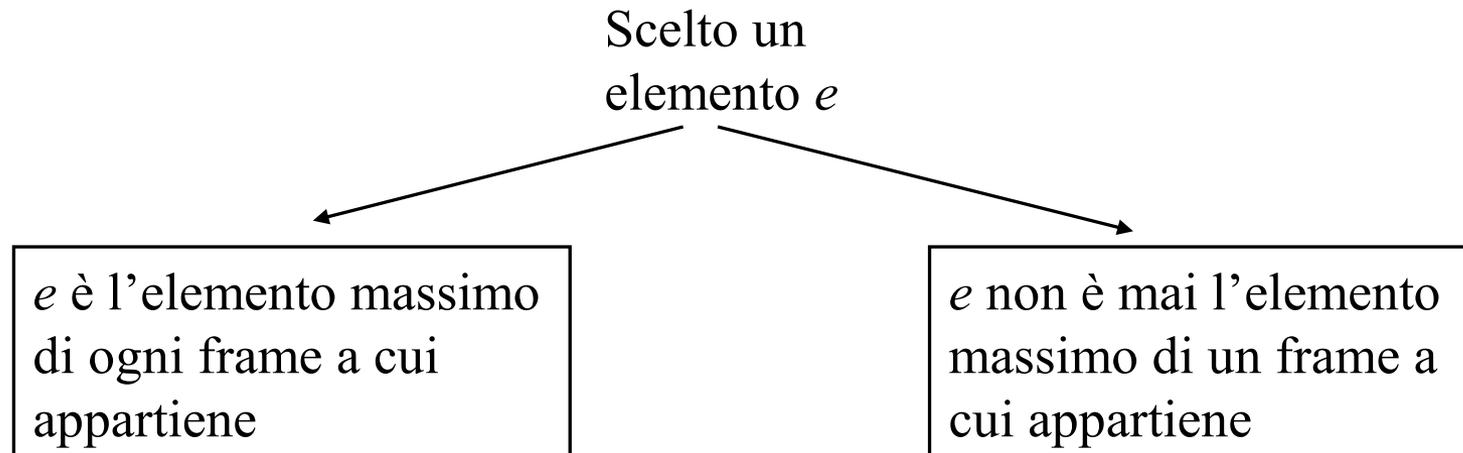
Dal momento che per il problema SSTDMA il problema di pricing può essere risolto attraverso un algoritmo ad-hoc è importante trovare una regola di branching che non stravolga troppo la struttura del sottoproblema e permetta di utilizzare l'algoritmo mostrato precedentemente con piccole variazioni.

# Le regole di branching

## Due fasi di branching binario

Prima fase

A. Bettinelli, A. Ceselli, G. Righini, *A branch-and-price algorithm for the bi-dimensional level strip packing problem*, AIROWinter, (2005)



$$e = \operatorname{argmin}_{s=1, \dots, n^2} \left\{ \left| \frac{1}{2} - \sum_{t \in T(s)} \lambda_t \right| \right\}$$

$T(s)$  è l'insieme delle colonne in cui  $s$  l'elemento massimo

# Le regole di branching

## Due fasi di branching binario

### Seconda fase

A. Bettinelli, A. Ceselli, G. Righini, *A branch-and-price algorithm for the bi-dimensional level strip packing problem*, AIROWinter, (2005)

Alla fine della prima fase di branching l'insieme degli elementi è diviso in due partizioni

$M = \{e \text{ è un massimo}\}$      $N = \{e \text{ non può essere massimo}\}$

$p \in M$  e  $q \in N$   
appartengono allo  
stesso frame.

$p \in M$  e  $q \in N$   
appartengono a  
frame differenti.

$$(p, q) = \operatorname{argmin}_{\sigma \in P, \tau \in Q} \left\{ \left| \frac{1}{2} - \sum_{c \in C(\sigma, \tau)} \lambda_c \right| \right\}$$

$C(\sigma, \tau)$  è l'insieme delle colonne  
in cui  $\sigma, \tau$  sono  
contemporaneamente inseriti

# Risultati sperimentali

---

- **Tecnica di esplorazione: Best Bound first**
  - **Soluzione: quella di costo minore tra le due ottenute dalla tecnica first fit e dalla risoluzione di  $q$  problemi K-LAP**
  - **Macchina: Intel Core2 Quad-core, 2.66Ghz, 4 Gb Ram**
- Randomly generated instances**

**Risultati esatti ottenuti da altri studi: 42 istanze con  $l=n$**

<b>ordine</b>	3	4	5	6	7	8	9	10	11	12	13	14	15
<b>numero istanze</b>	2	3	4	4	5	5	4	3	3	3	3	2	1
<b>densità media</b>	0,78	0,83	0,89	0,65	0,69	0,6	0,47	0,52	0,42	0,6	0,59	0,51	0,61
<b>media nodi esplorati</b>	1	1	1	1,25	1,8	2	2,5	3,33	5,33	8,67	6,67	7,5	7

C.C. Ribeiro, M. Minoux and M.C. Penna, *An optimal column-generation-with-ranking algorithm for very large scale set partitioning problems in traffic assignment*, E.J.O.R., Vol. 41, 232-239 (1989).

# Risultati sperimentali

Media su 5 istanze

$n^2$  elementi in

U[10,100]

Limite di tempo 7200  
sec.

$l = n$

n	lbCG	lbCO	UB	NG	NS	T	R	TO	MO	I
2	0	0	0	1	0	0	5	0	0	5
3	0	0	0	1	0	0	5	0	0	5
4	0	1,77	0,31	1	0	0	5	0	0	5
5	0	0,94	2,84	1	0	0	5	0	0	5
6	0	0,86	3,89	6,2	5,2	0	5	0	0	4
7	0	0,58	3,92	42,6	42	0,4	5	0	0	1
8	0	0,03	6,07	76,6	76	1,2	5	0	0	0
9	0	0,18	3,64	82,2	81,6	3	5	0	0	0
10	0	0,46	5,34	120,2	119,8	8,2	5	0	0	1
11	0	0,33	6,71	221	220,2	17,4	5	0	0	0
12	0,02	0,32	4,57	205,4	204,8	32,2	5	0	0	0
13	0,02	0,06	5,51	337,4	336,8	95	5	0	0	0
14	0	0,2	6,04	383,8	382,8	160,6	5	0	0	0
15	0	0,2	4,39	439,4	438,6	263,8	5	0	0	0
16	0	0,11	4,93	577,8	577,2	781,8	5	0	0	0
17	0,02	0,15	5	690,2	689,2	1409,8	5	0	0	0
18	0	0,06	4,95	795	794,8	2380,6	5	0	0	0
19	0	0,15	3,93	807,5	807	2804,5	4	1	0	0
20	0	0,06	4,63	981,8	981,2	5052,6	5	0	0	0

# Risultati sperimentali

Media su 5 istanze

$0,75 * n^2$  elementi in  
U[10,100]

Tempo limite 7200  
sec.

$l = n$

n	lbCG	lbCO	UB	NG	NS	T	R	TO	MO	I
2	0	0	0	1	0	0	5	0	0	5
3	0	0,5	1,2	1	0	0	5	0	0	5
4	0	0,66	1,79	1	0	0	5	0	0	5
5	0	0,69	1,97	1,4	0,4	0	5	0	0	4
6	0	0,7	2,64	6,2	5,2	0	5	0	0	4
7	0	0,48	3,2	21	20,4	0,2	5	0	0	2
8	0	0,69	4,38	59,8	59	0,8	5	0	0	1
9	0	0,11	3,5	58,6	58	1	5	0	0	2
10	0	0,09	2,77	165,4	164,6	4,6	5	0	0	0
11	0,03	0,29	3,65	212,2	212	8,4	5	0	0	0
12	0	0,17	4,81	203	202,2	12,2	5	0	0	0
13	0	0,1	2,45	258,6	258	18,2	5	0	0	1
14	0	0	1,22	337,8	337	26	5	0	0	1
15	0	0,08	0,25	294,2	293,4	34,8	5	0	0	1
16	0	0,1	1,19	476,2	475,4	107	5	0	0	0
17	0	0,04	1,6	579	578,4	158,6	5	0	0	0
18	0	0,02	1,37	549,8	549	287,2	5	0	0	1
19	0	0	1,15	609,4	609	392,4	5	0	0	0
20	0	0,02	1,14	733,8	733,4	564,4	5	0	0	0

# Risultati sperimentali

Media su 5 istanze

$n^2$  elementi in  
U[10,100]

Tempo limite 7200  
sec.

$$l = 0.9 * n$$

n	k	lbCG	lbCO	UB	NG	NS	T	R	TO	MO	I
3	2	3,28	1,3	3,2	4,2	0,4	0	5	0	0	2
4	3	1,52	3,98	3,32	11,4	1	0	5	0	0	0
5	4	1,71	2,77	3,46	67	24,4	0,4	5	0	0	0
6	5	0,39	1,67	6,77	31,8	26,2	0,4	5	0	0	1
7	6	0,92	1,88	6,11	705,4	704,8	6,2	5	0	0	0
8	7	0,73	1,14	5,56	1214,5	1212,25	19,75	4	0	1	0
9	8	0,69	1,09	4,67	24476,2	24453,6	726,2	5	0	0	0
10	9	0,5	1,51	4,71	2072,33	2071,33	260,67	3	2	0	0
11	9	0,23	1,97	6,35	3845,8	3845,4	775,2	5	0	0	0
12	10	0,11	2,1	6,3	3997	3996	2091,5	2	3	0	0
13	11	0,1	1,55	8,99	413	412	333	1	4	0	0

# Risultati sperimentali

Media su 5 istanze

$0,75 * n^2$  elementi in  
U[10,100]

Tempo limite 7200  
sec.

$l = 0,9 * n$

n	k	lbCG	lbCO	UB	NG	NS	T	R	TO	MO	I
3	2	1,29	4,39	1,89	1,4	0,6	0	5	0	0	4
4	3	0	4,29	3,3	1	0	0	5	0	0	5
5	4	0,78	2,68	0,89	3	2	0	5	0	0	3
6	5	0,05	1,52	5,88	1,8	0,8	0	5	0	0	3
7	6	0	0,09	4,74	35,4	34,8	0,2	5	0	0	0
8	7	0	0,94	4,36	56,2	55,2	0,8	5	0	0	2
9	8	0	0	2,3	105	104,2	1,4	5	0	0	1
10	9	0	0,03	1,19	61	60,2	2	5	0	0	3
11	9	0	0,18	3,78	220,6	219,6	8,8	5	0	0	0
12	10	0,05	0,34	6,49	282,6	282	28,4	5	0	0	0
13	11	0,02	0,26	4,41	326,2	325,2	29	5	0	0	0
14	12	0	0,02	5,47	396,2	395,2	54,2	5	0	0	0
15	13	0	0,02	3,01	463,8	463,4	66,2	5	0	0	0
16	14	0	0,12	2,01	375,8	374,8	100,8	5	0	0	1
17	15	0	0,11	2,82	591,8	591	169,6	5	0	0	0
18	16	0	0	1,39	703,8	703	282,2	5	0	0	0
19	17	0	0	0,76	712,6	711,8	396,2	5	0	0	1
20	18	0	0,02	1,15	632,6	631,8	602	5	0	0	1

# Risultati sperimentali

---

Media su 5 istanze

$n^2$  elementi in  
U[10,100]

Tempo limite 7200  
sec.

$l = 0.75 * n$

n	k	lbCG	lbCO	UB	NG	NS	T	R	TO	MO	I
5	3	2,83	2,73	2,05	655	289,8	3	5	0	0	0
6	4	0,64	3,58	6,39	97,8	55,4	0,8	5	0	0	0
7	5	0,12	2,53	6,49	87,4	86,6	1,4	5	0	0	0
8	6	0,31	2,59	4,7	179,4	164,4	5	5	0	0	0
9	6	0,67	2,98	6,06	69932,5	29381,2	3818,25	4	1	0	0
10	7	0,11	4,71	6,8	45	44	9	1	4	0	0
11	8	0,31	2,45	7,26	1001	1000	117	1	4	0	0
12	9	0,1	2,74	5,07	3317	3317	1451	1	4	0	0
13	9	0,09	1,89	7,04	673	672	386	1	4	0	0

# Risultati sperimentali

---

Media su 5 istanze

$0,75 * n^2$  elementi in  
U[10,100]

Tempo limite 7200  
sec.

$l = 0.75 * n$

n	k	lbCG	lbCO	UB	NG	NS	T	R	TO	MO	I
5	3	0,52	2,62	3,7	16,2	15,2	0,2	5	0	0	4
6	4	0,5	2,43	5,21	42,2	35,6	0,2	5	0	0	2
7	5	1,07	1,79	5,55	119,4	42,6	1,2	5	0	0	0
8	6	0,07	1,57	3,53	65,4	64,8	0,8	5	0	0	0
9	6	0,16	1,99	5,49	199,4	198,8	5,6	5	0	0	0
10	7	0,61	2,35	5,84	1357,5	1357	64,75	4	1	0	0
11	8	0,31	1,59	6,47	639	638,33	39,33	3	2	0	0
12	9	0,41	2,62	6,38	9907,5	9906,75	2211,25	4	1	0	0
13	9	0,18	2,02	4,95	520	520	239,5	2	3	0	0
14	10	0	0	6,82	513	513	195	1	4	0	0

# Sviluppi futuri

---

- Usare un upper bound migliore (es. Quello, basato sul metodo tabu search, descritto in “Dell'Amico et al. '98”, abbiamo già il codice fortran è necessario un lavoro di integrazione)
- Implementare il fissaggio di colonne attraverso i valori di costo ridotto
- Fare sperimentazioni su un altro schema di branching
- Cercare di migliorare l'algoritmo solutore del problema di pricing
- Fare ulteriori sperimentazioni sulla stabilizzazione del column generation
- Sperimentare una diversa gestione delle colonne
- Risolvere il problema con un algoritmo *ad-hoc* una volta che tutti i vincoli della prima fase sono stati aggiunti