Programmazione Lineare Intera: Introduzione

Daniele Vigo
D.E.I. – Università di Bologna
daniele.vigo@unibo.it

rev. 1.0 – 2023

Programmazione Lineare Intera

$$(P) \quad z_P = \min c^T x$$
$$A\,x \geq d$$
$$x \geq 0, \text{ intere}$$

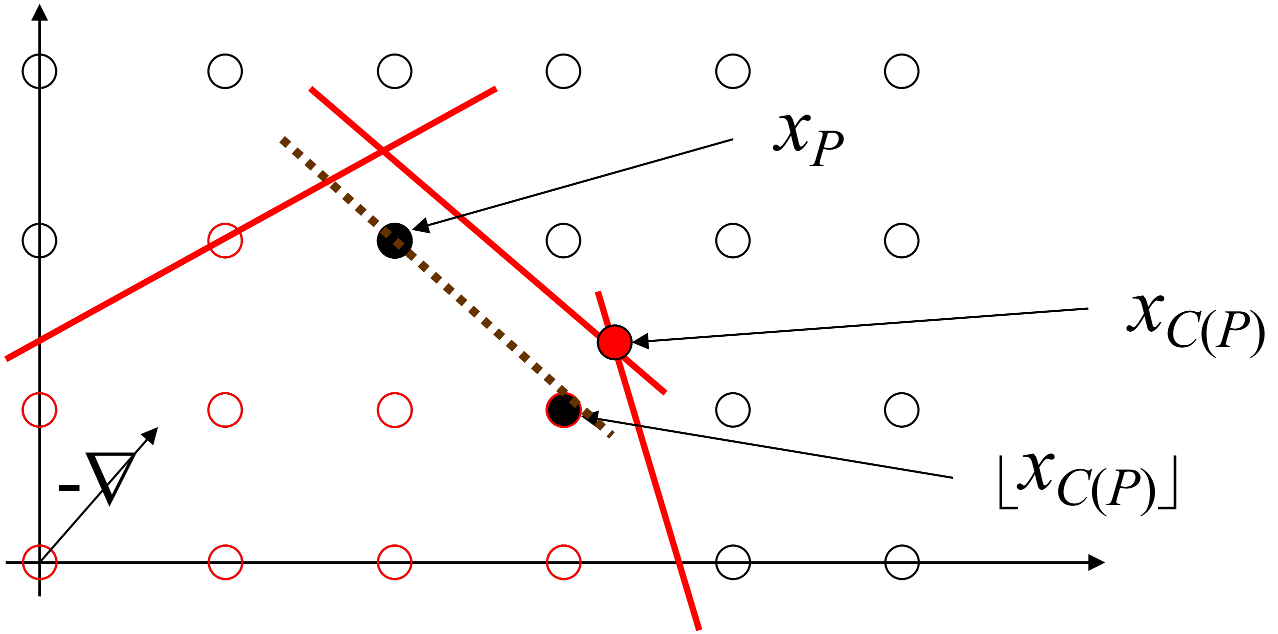
- vincoli di interezza: non lineari
  - x intera $\Leftrightarrow \sin \pi x = 0$
  - x binaria $\Leftrightarrow x\,(x - 1) = x^2 - x = 0$
- PLI $\approx$ NLP
- in realtà la non linearità del problema è "concentrata" nella prescrizione di interezza

Rilassamento continuo di PLI

- rimuovendo il vincolo di interezza:
  rilassamento continuo $C(P)$ "associato" a P

$$z_{C(P)} \leq z_P$$

Dim.: si cerca il minimo in un insieme più ampio



Rilassamento continuo (2)

Th.: *se la soluzione del rilassamento continuo è ammissibile per P (= è intera), allora è ottima per P*

Dim.:

1) $z_{C(P)} \leq z_P$

2) $x_{C(P)}$ è ammissibile per P

$\Rightarrow \quad z_{C(P)} = c^T x_{C(P)} \geq z_P \quad \Rightarrow \quad z_{C(P)} = z_P$

Problemi ed algoritmi

- Algoritmo esatto: determina la soluzione ottima
  - Se il problema è "difficile" il tempo di calcolo necessario ad un algoritmo esatto cresce molto rapidamente (= esponenzialmente) con la dimensione del problema
  - Si risolvono in modo esatto problemi "piccoli"
  - Molti problemi reali sono "difficili" e "grandi"
- Algoritmo euristico o approssimato:

  determina in tempo ragionevole una soluzione ammissibile di "buona" qualità
  - Si risolvono problemi "grandi"
  - In alcuni casi è possibile dare garanzie sulla qualità della soluzione ottenuta (Es. al più il 1.5 volte la soluzione ottima)

Algoritmo (euristico) per PLI

begin

determina con simplesso la soluzione x di $C(P)$

if $C(P)$ impossibile then STOP (P impossibile)

else

if $C(P)$ illimitato then STOP
(P illimitato, salvo casi particolari)

else

if x intero then STOP (x sol. ottima di P)

else

"arrotonda" ogni x_j frazionaria all'intero più vicino

end

Che soluzioni produce questo algoritmo ?

CASO 1: soluzioni utili

- Problemi per cui i valori delle variabili della soluzione ottima sono molto elevati
- Es. pezzi da produrre (elevata quantità)

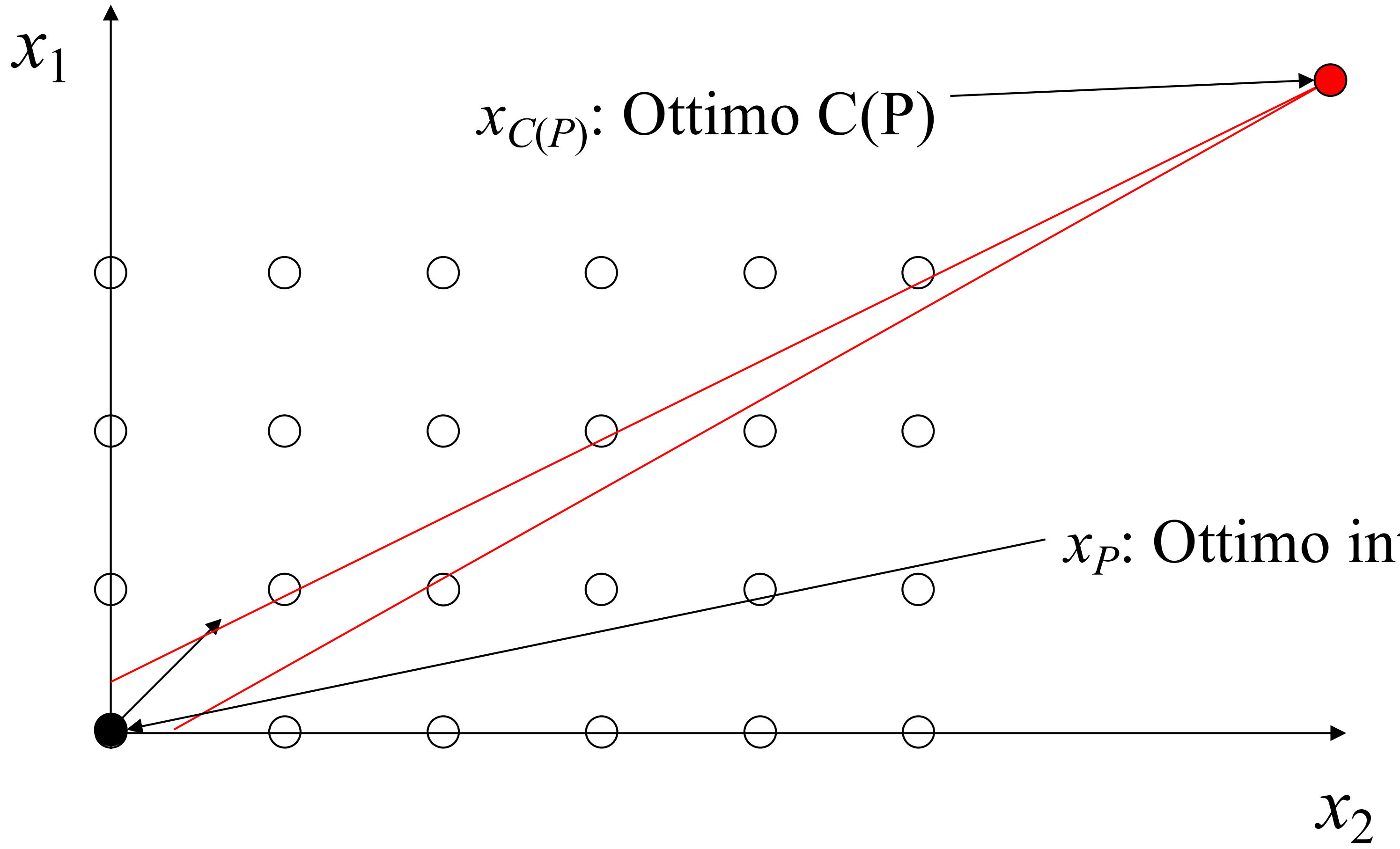
	$C(P)$	$C(P)$ arrotondato
$x_1 =$	2449.51	2450
$x_2 =$	14301.1	14301
$x_3 =$	7800.92	7801
max $x_1+x_2+x_3$	24551.53	24552
$3x_1+x_2 \leq 21650$	21649.63	21651

CASO 2: soluzioni inutili

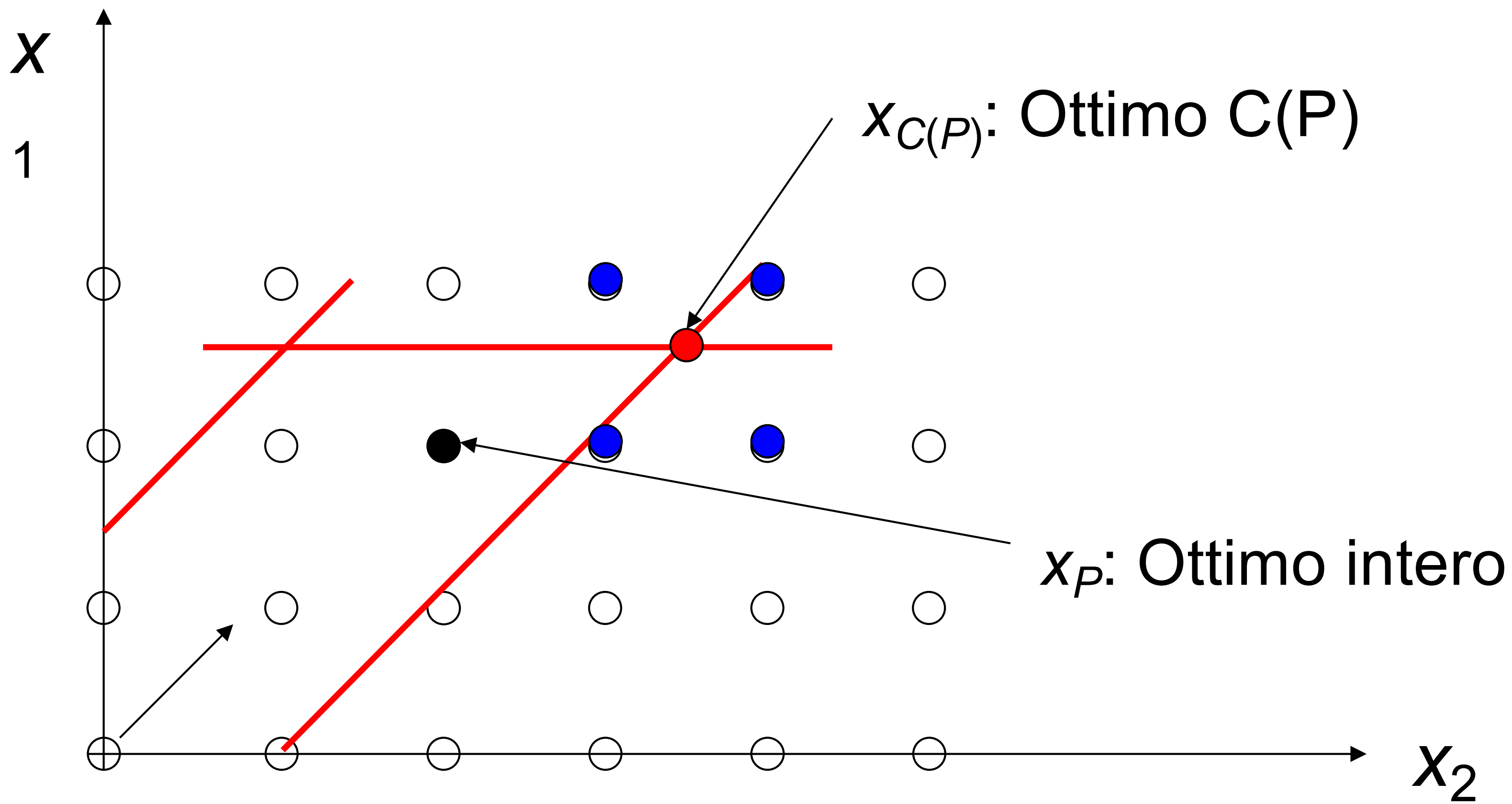
- Problemi in cui i valori delle variabili decisionali all'ottimo sono molto piccoli:
  - Numero di edifici da realizzare
  - Numero di veicoli da assegnare ad un servizio
  - Opportunità di una scelta
  - uso o meno di un tratto di strada in un percorso (sì/no)
  - ....
- La parte frazionaria non è trascurabile e l'arrotondamento può produrre facilmente soluzioni non ammissibili

CASO 2: soluzioni inutili (2)

- Soluzione intera e continua possono essere molto “lontane”



CASO 3: soluzioni non ammissibili

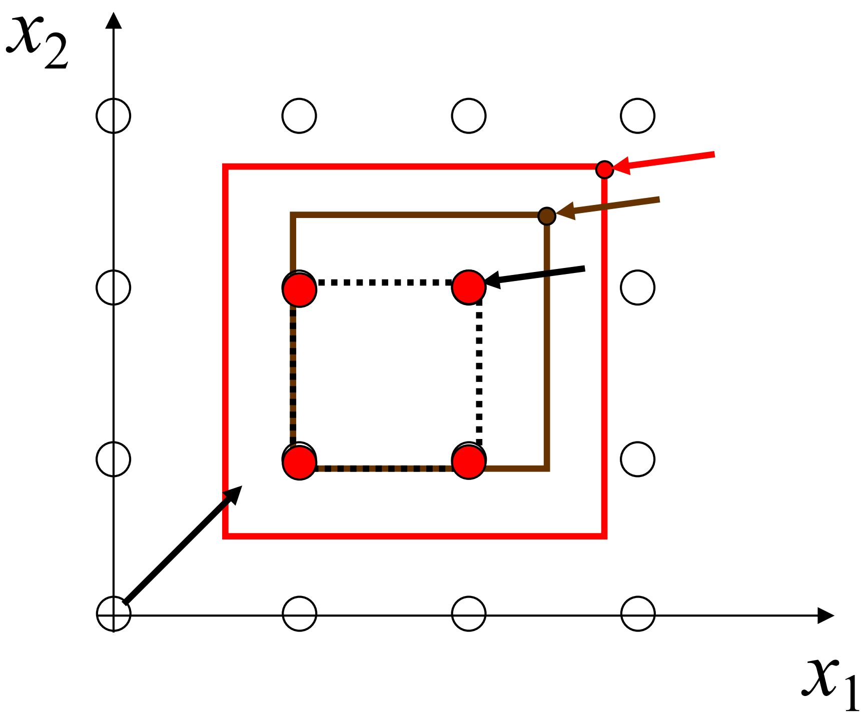


- Nessuno dei quattro punti interi attorno a $x_{C(P)}$ è ammissibile per P

Formulazioni equivalenti

- dato $z_P = min\ \{c^T x : x \in X\}$ esistono molte formulazioni equivalenti:

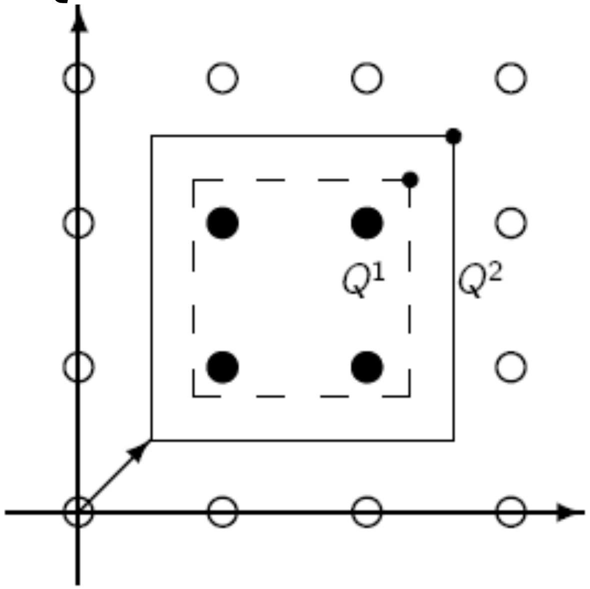
$$z_P = min\ \{c^T x : Ax \geq d,\ x \geq 0,\ x \text{ intero}\}$$



- i corrispondenti rilassamenti continui non sono però equivalenti !

Confronto di formulazioni

- Esistono formulazioni migliori di altre ?
- Una formulazione $Q^1 = \{A^1x = d^1, x \geq 0\}$ valida per P è migliore di una formulazione $Q^2 = \{A^2x = d^2, x \geq 0\}$ se $Q^1 \subset Q^2$



Se Q^1 e Q^2 sono due formulazioni di un problema di min con $Q^1 \subset Q^2$, allora $z_{C(Q1)} \geq z_{C(Q2)}$

Esempio: Knapsack 0-1

- *Dati n oggetti, ciascuno con peso w_j e profitto p_j, $j=1,\dots,n$ ed un contenitore di capacità W, determinare il sottoinsieme S di oggetti di profitto* ***massimo inseribili nel contenitore.***
- Esempio: dato $W = 6$ e $w = \{4, 3, 2\}$ *si ha:*

$X = \{(0,0,0),(1,0,0),(1,0,1),(0,1,0),(0,1,1),(0,0,1)\}$

$Q^1 = \{x \in R^3 : 0 \leq x \leq 1,\ 4x_1 + 3x_2 + 2x_3 \leq 6\}$

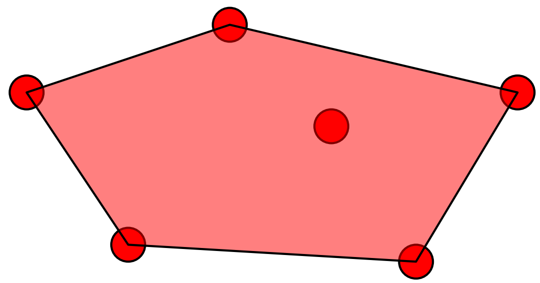
$Q^2 = \{x \in R^3 : 0 \leq x \leq 1,\ 4x_1 + 3x_2 + 2x_3 \leq 6,\ x_1 + x_2 \leq 1\}$

- è facile verificare che $Q^2 \subset Q^1$

Formulazione "ideale" di PLI

- Esiste una formulazione "ideale" di PLI ?

  Def.: *Dato un insieme* $S \subseteq R^n$ *si dice* convex hull (*guscio convesso*) *di S il più piccolo insieme convesso* conv(*S*) *che contiene S*



- Se *X* è un insieme di punti interi, conv(*X*) è un politopo $\tilde{P}$ i cui vertici sono tutti punti *interi*

Formulazione "ideale" di PLI (2)

- conv(X): politopo $\tilde{P}$ i cui vertici sono tutti punti *interi*
- $\exists\ \tilde{A}, \tilde{d}$, tali che $\tilde{P} = \{x \in R^n : \tilde{A}\, x \geq \tilde{d},\ x \geq 0\} = \text{conv}(X) \quad \Rightarrow$ $\min\{\ c^T x : x \in X\} = \min\{\ c^T x : \tilde{\tilde{A}}\, x \geq \tilde{d},\ x \geq 0\}$
- PLI può essere risolto con l' algoritmo del simplesso
- Purtroppo determinare conv(X) è molto difficile:
  - il sistema $\tilde{A}\, x \geq \tilde{d}$ ha generalmente un numero molto elevato di vincoli (esponenziale nella dimensione di P)
- Esistono casi in cui una formulazione naturale di PLI coincide con la formulazione ideale ?

Unimodularità

Def. 1: una matrice quadrata B ($n \times n$) intera è **unimodulare** (UM) se $\det(B) = \pm 1$

Def. 2: una matrice rettangolare A ($m \times n$) intera è **totalmente unimodulare** (TUM) se ogni sua sottomatrice quadrata non singolare è UM

Unimodularità (2)

Th.1: A TUM $\Rightarrow$ vertici di $\{x: Ax = d,\ x \geq 0\}$ interi $\forall\ d$ intero

Dim:

- B = matrice corr. ad una base qualsiasi di A
- Soluzione base :

$$x = B^{-1}d = \frac{B^a}{\det(B)}d$$

dove B^a = aggiunta di B : $b_{ij}^a = (-1)^{i+j} \cdot$minore (b_{ij})

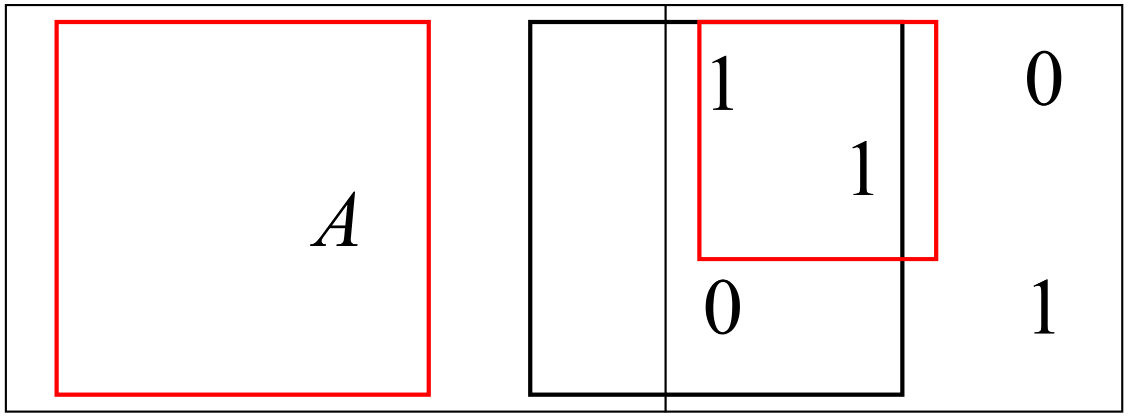
- A è TUM $\Rightarrow B$ è UM $\Rightarrow x$ è intero

Unimodularità (3)

Th. 2: A TUM $\Rightarrow$ vertici di $\{x : Ax \leq d,\ x \geq 0\}$ interi
$\forall$ d intero

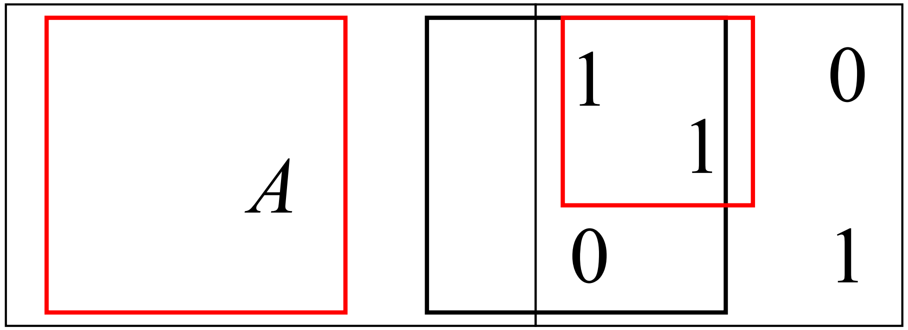
Dim: dimostriamo che se A è *TUM,* $(A|I\ )$ è *TUM.*

C = sottomatrice quadrata non singolare di $(A|I\ )$:



OK   OK ?   OK

Unimodularità (4)



permutiamo la righe di $C \Rightarrow \tilde{C} = \begin{bmatrix} B & 0 \\ D & I \end{bmatrix}$

$$\det(\tilde{C}) = \det(B) \Rightarrow \det(C) = \pm \det(\tilde{C}) = \pm 1$$

- Quindi : se *A* è *TUM*, l' *ILP* si può risolvere col simplesso
- $\exists$ condizioni sufficienti per verificare se *A* è *TUM*

Algoritmi generali per PLI

- Metodi esatti tradizionali (anni 60-oggi):
  - Metodo dei piani di taglio (cutting planes)
  - Branch-and-Bound
  - Programmazione Dinamica
- …
- Metodi esatti più avanzati (anni 90-oggi):
  - Branch-and-Bound + Cutting planes = Branch-and-Cut
  - Branch-and-Price/Column generation