Programmazione Lineare Intera: Introduzione

Alessandro Hill

Basato sul materiale di

Daniele Vigo (D.E.I.)

rev. 1.1(AH) – 2024

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

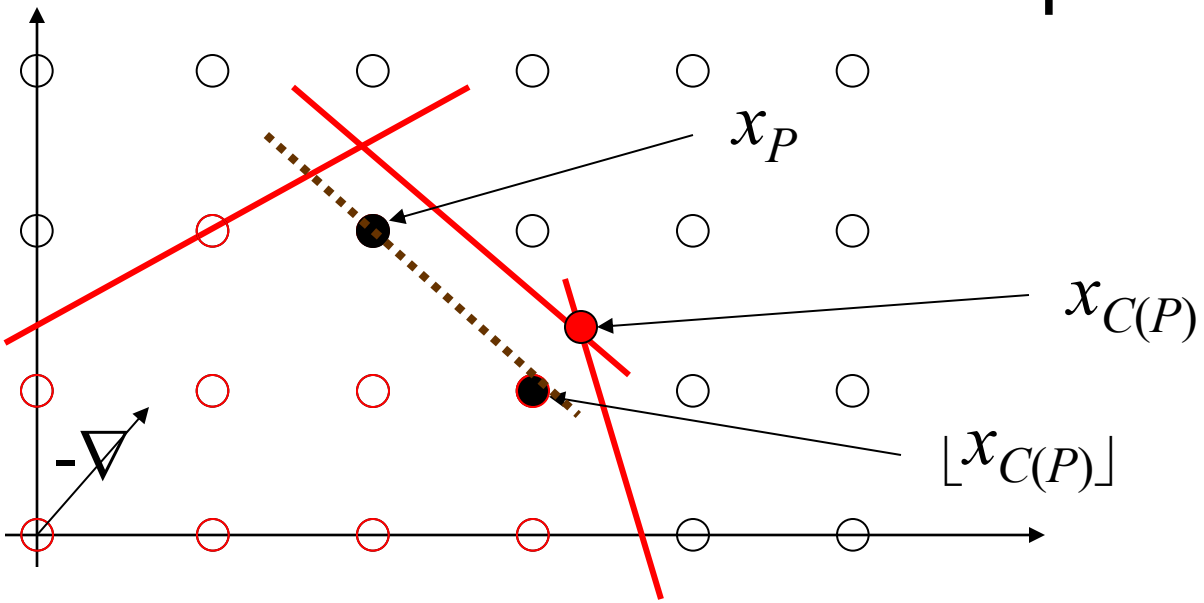
Rilassamento continuo di PLI

- 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)

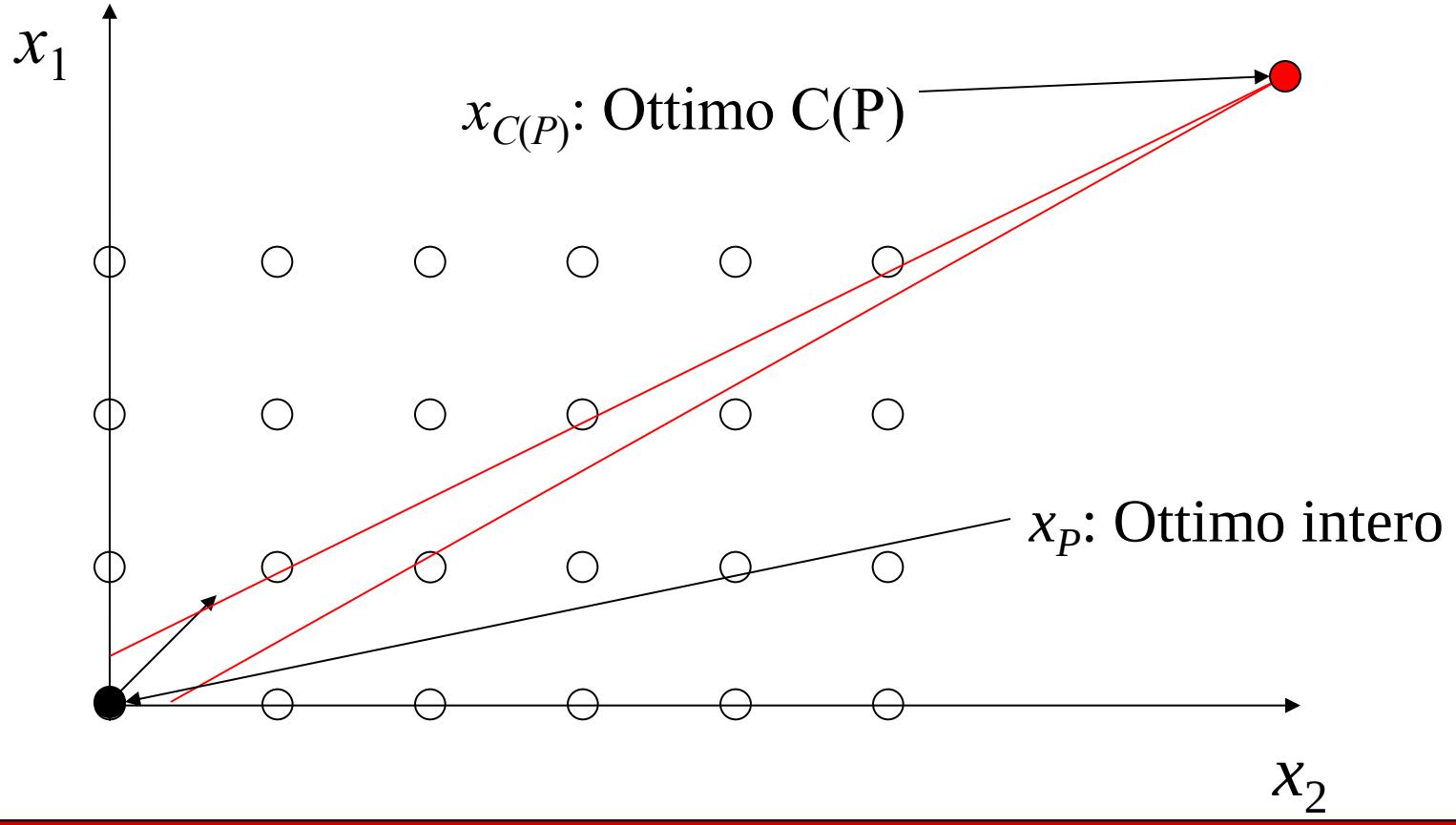
	$C(P)$	$C(P)$ arrotondato
$x_1 =$	2449.51	2450
$x_2 =$	14301.1	14301
$x_3 =$	7800.92	7801
$\max x_1+x_2+x_3$	24551.53	24552
$3x_1+x_2 \leq 21650$	21649.63	21651

CASO 2: soluzioni inutili

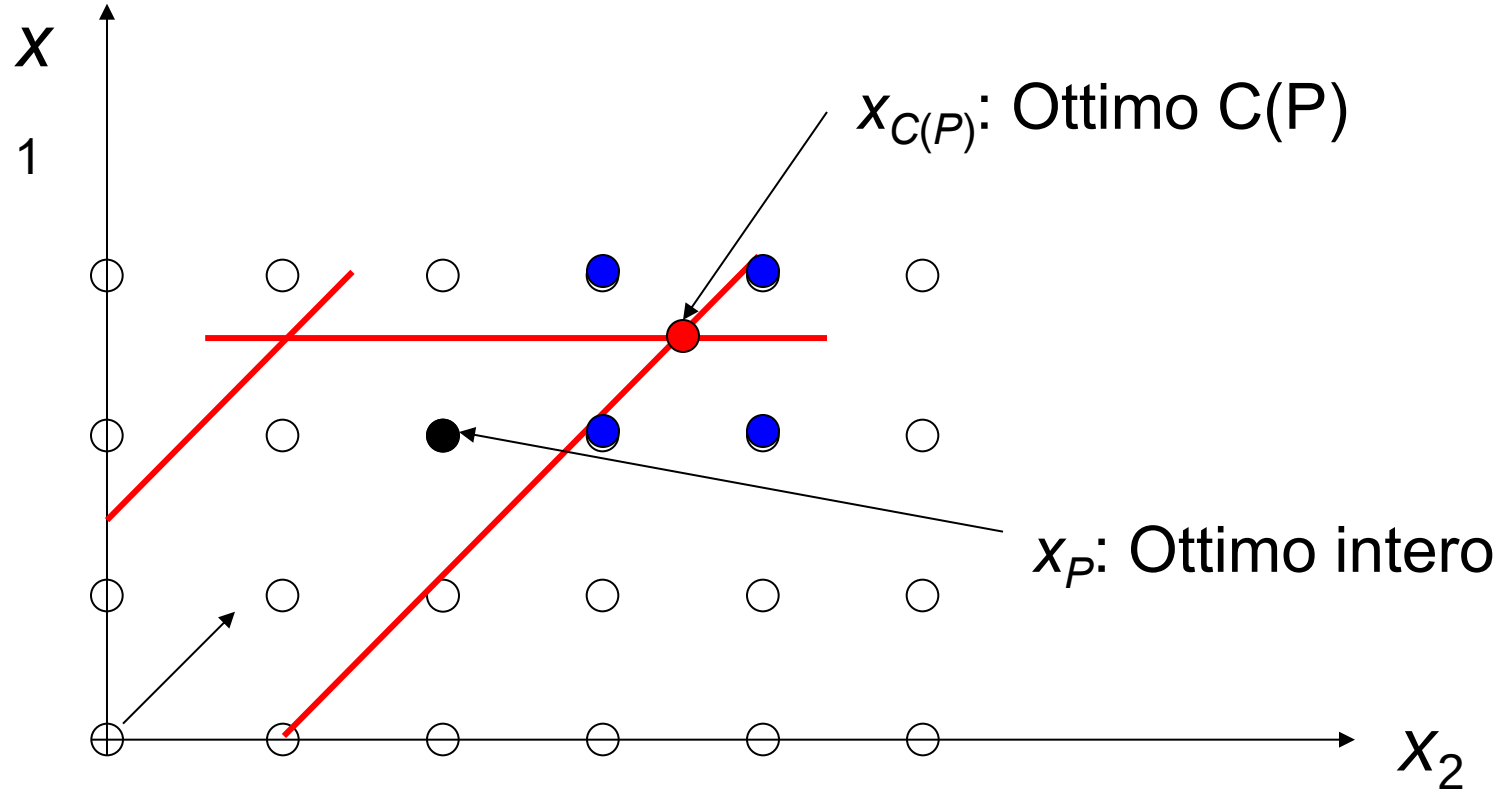
- Problemi in cui i valori delle variabili decisionali all'ottimo sono molto piccoli:
  - Numero di edifici da realizzare
  - Numero di veicoli da assegnare ad un servizio
  - Opportunità di una scelta
  - uso o meno di un tratto di strada in un percorso (sì/no)
  - .....
- La parte frazionaria non è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

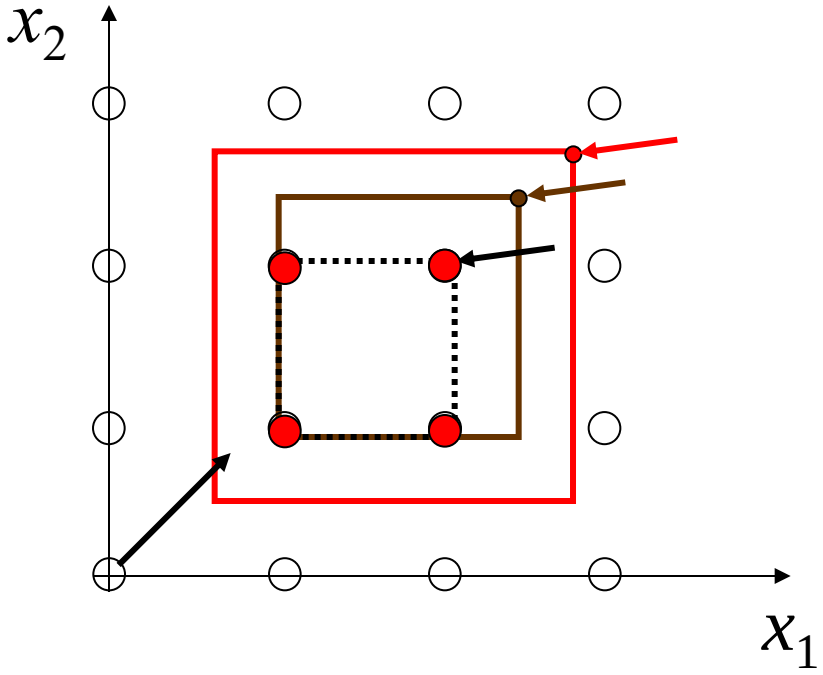


- Nessuno dei quattro punti interi attorno a $x_{C(P)}$ è ammissibile per *P*

Formulazioni equivalenti

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

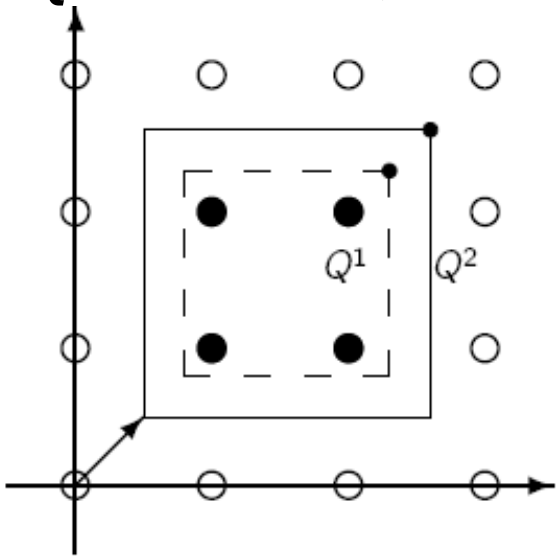
$$z_P = min \{c^T x : Ax \geq d,\ x \geq 0,\ x \text{ intero}\}$$



- 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 $Q^1 \subset Q^2$

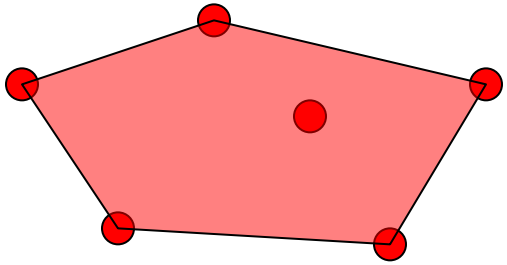


Se Q^1 *e* Q^2 sono due formulazioni di un problema di min con $Q^1 \subset Q^2$, allora $z_{C(Q1)} \geq z_{C(Q2)}$

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*

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