Ottimizzazione Combinatoria

Corso di Laurea in Informatica

Terza Parte: Programmazione Lineare

Ugo Dal Lago

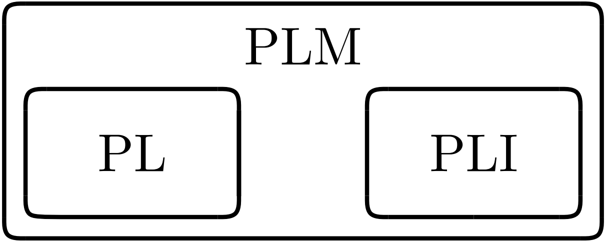
ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Inria informatiques mathématiques

Anno Accademico 2021-2022

Sezione 1

Geometria Della Programmazione Lineare

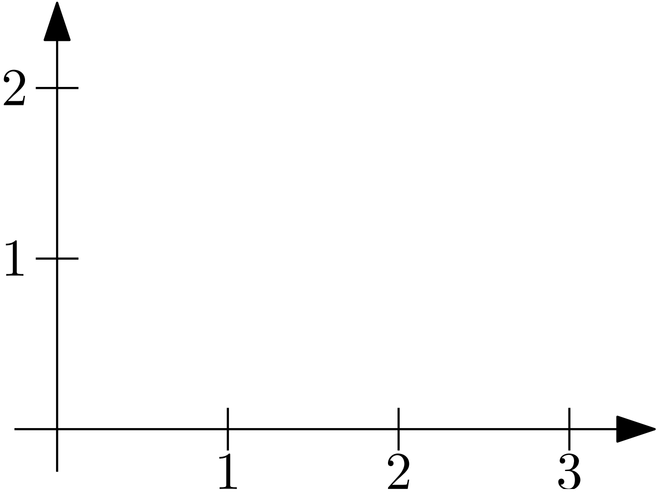

PLM
PL
PLI

PL

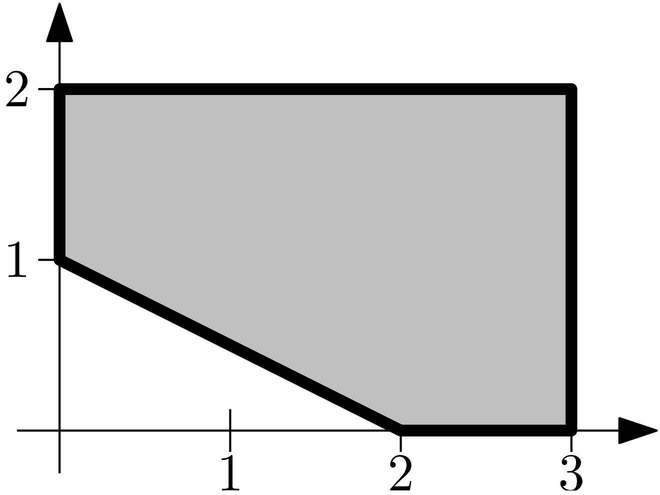
Un Esempio

$$
\begin{aligned}
\max x + y \\
x &\geq 0 \\
x &\leq 3 \\
x + 2y &\geq 2 \\
y &\leq 2 \\
y &\geq 0
\end{aligned}
$$

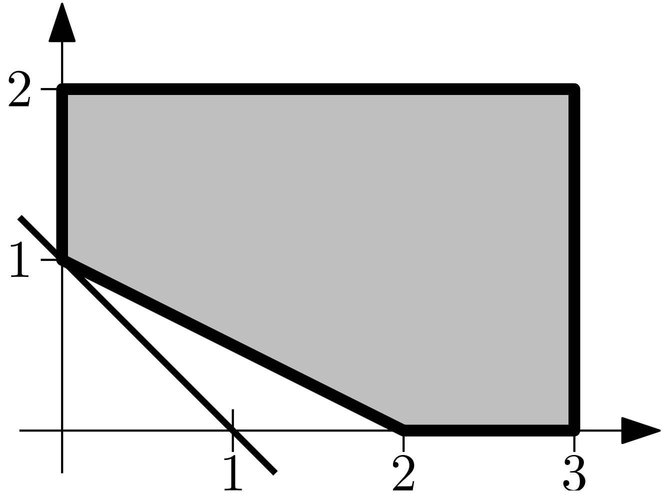
Un Esempio



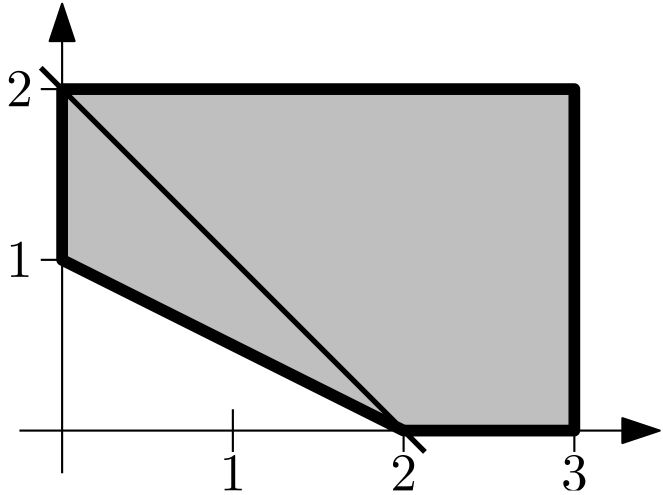
Un Esempio



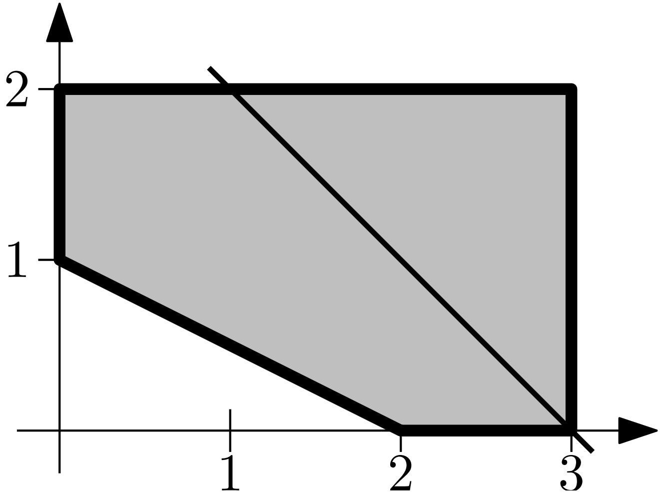
Un Esempio



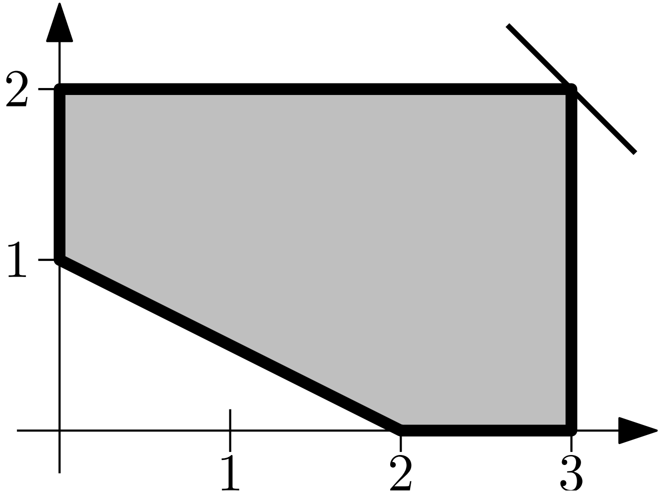
Un Esempio



Un Esempio



Un Esempio



Restringere lo Spazio di Ricerca

- Lo spazio di ricerca, nei problemi PL, è in linea di principio **infinito**.
  - Ha addirittura la cardinalità del continuo.
- L'esempio precedente ci mostra, però, che lo spazio di ricerca può essere *in qualche caso* ridotto ad un insieme **finito**, ossia l'insieme dei vertici del poliedro che definisce la regione ammissibile.
  - Il ragionamento sottostante ha natura essenzialmente geometrica, intuitiva.

Restringere lo Spazio di Ricerca

- Lo spazio di ricerca, nei problemi PL, è in linea di principio **infinito**.
  - Ha addirittura la cardinalità del continuo.
- L'esempio precedente ci mostra, però, che lo spazio di ricerca può essere *in qualche caso* ridotto ad un insieme **finito**, ossia l'insieme dei vertici del poliedro che definisce la regione ammissibile.
  - Il ragionamento sottostante ha natura essenzialmente geometrica, intuitiva.
- La domanda cui cercheremo di dare una risposta in questa prima sezione è la seguente: è possibile **generalizzare** quest'argomento al caso di problemi in $n > 2$ variabili.
  - Sarà necessario utilizzare l'algebra lineare in modo non triviale.

Nozioni Preliminari — I

- **Iperpiano**
  - Insieme $\{x \in \mathbb{R}^n \mid ax = b\}$ delle soluzioni dell'equazione lineare $ax = b$, dove $a \in \mathbb{R}^n$ e $b \in \mathbb{R}$.
- **Semispazio**
  - Insieme $\{x \in \mathbb{R}^n \mid ax \leq b\}$ delle soluzioni dell'equazione lineare $ax \leq b$, dove $a \in \mathbb{R}^n$ e $b \in \mathbb{R}$;
  - Un iperpiano è il "confine" del corrispondente semispazio.
- **Poliedro**
  - Intersezione P di un numero finito m di semispazi.
  - Devono esistere una matrice $A \in \mathbb{R}^{m \times n}$ e un vettore $b \in \mathbb{R}^m$ tali che $P = \{x \mid Ax \leq b\}$.
- **Insieme Convesso**
  - Un insieme $C \subseteq \mathbb{R}^n$ tale che tutti i punti che connettono $x, y \in C$ sono anch'essi in C, ossia

$$\forall x, y \in C \quad \forall \alpha \in [0,1] \qquad \alpha x + (1-\alpha)y \in C.$$

  - Semispazi e poliedri sono insiemi convessi.

Nozioni Preliminari — II

- Se consideriamo il poliedro $P = \{x \mid Ax \leq b\}$ (dove $A \in \mathbb{R}^{m \times n}$) e fissiamo un qualunque sottoinsieme I di $\{1, \dots, m\}$, indichiamo:
  - Con $\overline{I}$ il complementare $\{1, \dots, m\} - I$ di I.
  - Con A_I la sottomatrice di A ottenuta considerando solo le *righe* con indice in I
  - Con P_I il poliedro definito come seque:

$$\{x \mid A_I x = b_I \wedge A_{\overline{I}} x \leq b_{\overline{I}}\}.$$

- **Faccia**
  - Se I è tale che P_I non è vuoto, chiamiamo il poliedro P_I *faccia di* P.
  - Il numero di facce distinte di un poliedro $\{x \mid Ax \leq b\}$, dove $A \in \mathbb{R}^{m \times n}$, è al più pari a 2^m.
  - P stesso è una faccia, ovvero $P_\emptyset$.
  - Le facce proprie (cioè non banali) e massimali sono dette **faccette**.
  - La **dimensione** di una faccia è la dimensione del più piccolo sottospazio che la contenga.

Vertici

- Una faccia determinata da una matrice A_I di rango k ha dimensione $n - k$ o inferiore.
  - Può essere inferiore a causa delle equazioni in $A_{\bar{I}}$, che possono contenere un'equazione implicita.
- Le facce determinate da matrici A_I di rango n hanno quindi, necessariamente, dimensione 0 e sono dette **vertici**.
  - Chiaramente, per l'ipotesi sul rango di A_I, l'equazione $A_I x = b_I$ ammette una e una sola soluzione.
  - D'altra parte, le facce sono sempre non-vuote.
- Le facce individuate da sottomatrici A_I di rango $n - 1$ hanno dimensione al più 1 e sono dette **spigoli**.

Soluzioni di Base

- Supponiamo che B sia tale che A_B sia matrice quadrata e invertibile. Allora:
  - B è detta **base**;
  - A_B è detta **matrice di base**;
  - $x_B = A_B^{-1} b_B$ è detta **soluzione di base**
- Una soluzione di base x_B tale che $x_B \in P$ è detta **ammissibile**, altrimenti **non ammissibile**.
- È facile rendersi conto che i vertici di P sono tutte e sole le sue soluzioni di base ammissibili.

Vincoli Attivi

- Se $x \in P$, allora i vincoli che vengono soddisfatti *come uguaglianze*, sono detti **attivi** in x.
- Indichiamo con $I(x)$ l'insieme degli indici dei vincoli attivi in x:
$$I(x) = \{i \mid A_i x = b_i\}.$$
- Per ogni $J \subseteq I(x)$, l'insieme P_J è una faccia di P, e $P_{I(x)}$ è la faccia minimale tra esse.

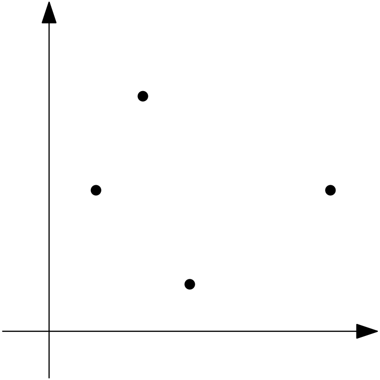
Inviluppi Convessi

- I poliedri possono essere rappresentati *per facce*, come abbiamo fatto fin'ora, ma anche *per punti* ossia facendo leva sull'insieme dei vertici.
- Dato un insieme di punti $X = \{x_1, \dots, x_s\} \subseteq \mathbb{R}^n$, l'**inviluppo convesso** di X è definito come l'insieme

$$\mathsf{conv}(X) = \left\{ x = \sum_{i=1}^{s} \lambda_i x_i \mid \sum_{i=1}^{s} \lambda_i = 1 \land \lambda_i \geq 0 \right\}$$

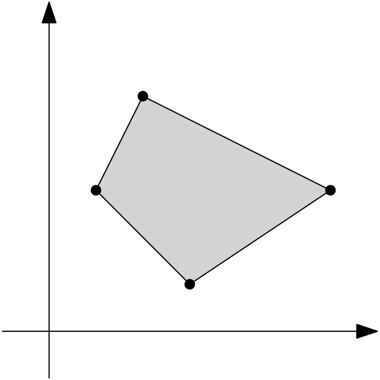
- Si può dimostrare che $\mathsf{conv}(X)$ è il più piccolo insieme convesso che contiene tutti i punti di X.
- $\mathsf{conv}(X)$ è un **politopo**, ossia un poliedro limitato, i cui vertici sono tutti in X.
  - Non tutti i poliedri sono politopi, perché i poliedri possono essere *illimitati*.

Inviluppi Convessi — Esempio



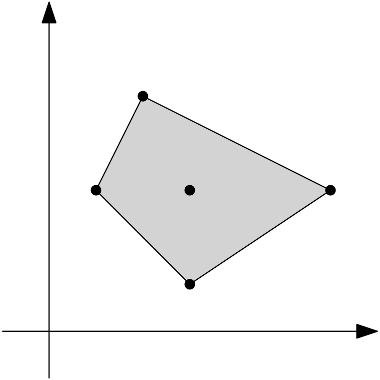
$$X = \{(1,3),(3,1),(2,5),(6,3)\}$$

Inviluppi Convessi — Esempio



$\mathsf{conv}(X)$

Inviluppi Convessi — Esempio



$\mathsf{conv}(X \cup \{(3,3)\})$

Coni Convessi

- Un insieme $C \subseteq \mathbb{R}^n$ è detto **cono** sse per ogni $x \in C$ e per ogni $\alpha \in \mathbb{R}^+$ vale che $\alpha x \in C$.
- I coni che siano anche insiemi convessi (detti **coni convessi** sono caratterizzabili equivalentemente come gli insiemi C tali che

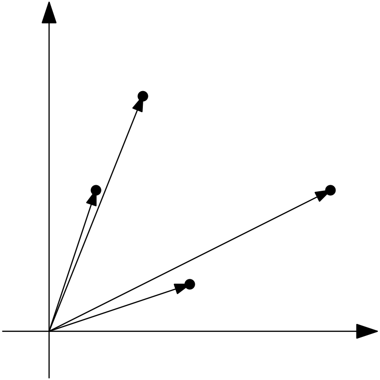
$$x, y \in C \wedge \lambda, \mu \in \mathbb{R} \implies \lambda x + \mu y \in C.$$

- Anche per i coni convessi esiste una rappresentazione basata sulle direzioni: dato un insieme $V = \{v_1, \ldots, v_t\} \subset \mathbb{R}^n$, il **cono finitamente generato** da V è

$$\mathsf{cono}(V) = \{v = \sum_{i=1}^{t} \nu_i v_i \mid \nu_i \in \mathbb{R}^+\}$$

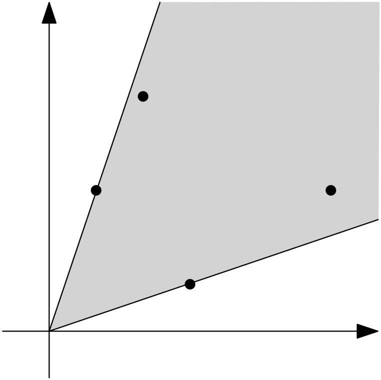
  - Si può dimostrare che $\mathsf{cono}(V)$ è il più piccolo cono convesso che contiene tutti i vettori di V.

Coni Convessi — Esempio



$$X = \{(1,3), (3,1), (2,5), (6,3)\}$$

Coni Convessi — Esempio



cono(X)

Il Teorema di Motzkin

- Dati $X, Y \subseteq \mathbb{R}^n$, indichiamo con $X + Y$ il sottoinsieme di $\mathbb{R}^n$ definito ponendo

$$X + Y = \{x + y \mid x \in X \land y \in Y\}$$

Il Teorema di Motzkin

- Dati $X, Y \subseteq \mathbb{R}^n$, indichiamo con $X + Y$ il sottoinsieme di $\mathbb{R}^n$ definito ponendo

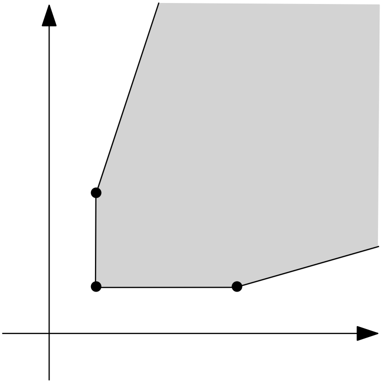
$$X + Y = \{x + y \mid x \in X \wedge y \in Y\}$$

Teorema (Motzkin)

$P \subseteq \mathbb{R}^n$ è un poliedro sse esistono X, V finiti tali che $P = \mathsf{conv}(X) + \mathsf{cono}(V)$

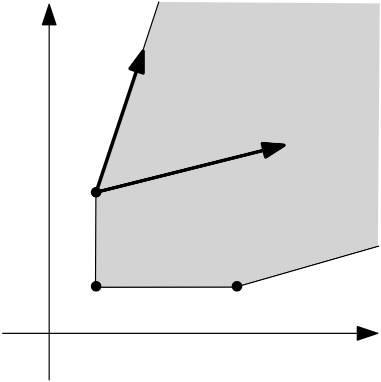
- Nel contesto del Teorema di Motzkin, diremo che P è **generato** dai punti in X e dalla direzioni in V.
- Se P è poliedro generato dai punti di X e X è minimale, allora i suoi elementi sono tutti e soli i vertici di P.
- Analogamente: se P è poliedro generato dalle direzioni in V e V è minimale, allora i suoi elementi sono detti **raggi esterni** e corrispondono alle direzioni degli spigoli illimitati.

Teorema di Motzin — Esempio



$$x \geq 1$$
$$y \geq 1$$
$$y \leq 3x$$
$$4y \geq x$$

Teorema di Motzin — Esempio



$\mathsf{conv}(X) + \mathsf{cono}(V)$

$X = \{(1,1),(1,3),(4,1)\}$

$V = \{(1,3),(4,1)\}$

Due Rappresentazioni

- Due rappresentazioni:
  1. Poliedri come **intersezioni di semispazi**.
  2. Poliedri come **somma di un politopo e di un cono**.
- Le due rappresentazioni sono equivalenti (grazie al teorema di Motzkin) ma **non hanno** la stessa dimensione.
  - Prendiamo come controesempio il poliedro definito dall'insieme di vincoli

$$0 \leq x_1 \leq 1 \qquad 0 \leq x_2 \leq 1 \qquad \cdots \qquad 0 \leq x_n \leq 1$$

  - I semispazi coinvolti sono $2n$.
  - I vertici sono invece 2^n; per rendersene conto basta osservare che i poliedri definiti sono gli *ipercubi* in $\mathbb{R}^n$ di lato pari a 1 e con un vertice nell'origine. Tali ipercubi hanno effettivamente 2^n vertici.

Teorema

Sia $P = \{x \mid Ax \leq b\}$ *e siano* $x_1, \ldots, x_s, v_1, \ldots, v_t \in \mathbb{R}^n$ *tali che*

$$P = \mathsf{conv}(\{x_1, \ldots, x_s\}) + \mathsf{cono}(\{v_1, \ldots, v_t\})$$

Allora il problema $\max\{cx \mid Ax \leq b\}$ *ha ottimo finito sse* $cv_j \leq 0$ *per ogni* $j \in \{1, \ldots, t\}$. *In tal caso esiste inoltre un* $k \in \{1, \ldots, s\}$ *tale che* x_k *è una soluzione ottima.*

Dimostrazione.

Per il Teorema di Decomposizione abbiamo che il problema $\max\{cx \mid Ax \leq b\}$ è equivalente al seguente problema sulle variabili $\lambda_1, \ldots, \lambda_s$ e $\nu_1, \ldots, \nu_t$:

$$\max c \left(\sum_{i=1}^{s} \lambda_i x_i + \sum_{j=1}^{t} \nu_j v_j \right) = \max \sum_{i=1}^{s} \lambda_i (cx_i) + \sum_{j=1}^{t} \nu_j (cv_j)$$

$$\sum_{i=1}^{s} \lambda_i = 1; \qquad \lambda_i \geq 0; \qquad \nu_j \geq 0.$$

Tale problema ha ottimo finito sse $cv_j \leq 0$ per ogni $j \in \{1, \ldots, t\}$. Infatti:

$\Longrightarrow$ Se fosse $cv_j > 0$ per qualche $j \in \{1, \ldots, t\}$, allora si potrebbe pompare ν_j facendo crescere a piacimento la funzione obbiettivo.

$\Longleftarrow$ Supponiamo che $c\nu_j \leq 0$ per ogni $j \in \{1, \ldots, t\}$, e prendiamo un $y \in P$. Abbiamo che, se λ_i e ν_j sono i corrispondenti coefficienti del teorema di decomposizione,

$$\begin{aligned} cy &= \sum_{i=1}^{s} \lambda_i (cx_i) + \sum_{j=1}^{t} \nu_j (cv_j) \\ &\leq \sum_{i=1}^{s} \lambda_i (cx_i) \leq \sum_{i=1}^{s} \lambda_i (cx_k) = cx_k \end{aligned}$$

dove x_k è il vettore tale che $x_k = \max\{cx_i \mid i = 1, \ldots, s\}$. Quindi x_k è una soluzione ottima finita.

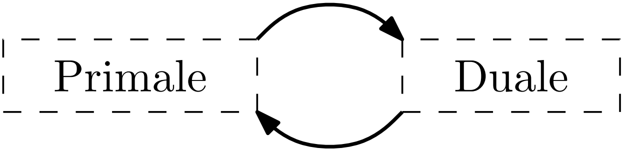
$\square$

Sezione 2

Dualità, Direzioni Ammissibili e di Crescita

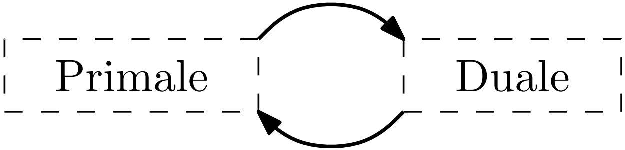
Perché la Dualità?

- La **teoria della dualità** è una branca dell'algebra lineare che risulta estremamente utile nella costruzione degli algoritmi per PL.
- In questa parte del corso, daremo solo uno *sguardo* alla teoria della dualità, senza addentrarci troppo nei dettagli.
- La teoria della dualità si basa sulla definizione di un'involuzione (ossia di una funzione inversa di sé stessa) che mappa ogni problema PL nel suo **duale**:



Perché la Dualità?

- La **teoria della dualità** è una branca dell'algebra lineare che risulta estremamente utile nella costruzione degli algoritmi per PL.
- In questa parte del corso, daremo solo uno *sguardo* alla teoria della dualità, senza addentrarci troppo nei dettagli.
- La teoria della dualità si basa sulla definizione di un'involuzione (ossia di una funzione inversa di sé stessa) che mappa ogni problema PL nel suo **duale**:



- **Esempio**: Problema di Trasporto.

Primale e Duale

- Lavoreremo con **coppie asimmetriche**:
  - *Primale*: $\max\{cx \mid Ax \leq b\}$;
  - *Duale*: $\min\{yb \mid (yA = c) \wedge (y \geq 0)\}$.
- Esiste anche il concetto di **coppie simmetriche**:
  - *Primale*: $\max\{cx \mid (Ax \leq b) \wedge (x \geq 0)\}$;
  - *Duale*: $\min\{yb \mid (yA \geq c) \wedge (y \geq 0)\}$.
- È abbastanza facile dimostrare che il duale del duale è il primale.
  - Ad esempio, nel caso di coppia simmetrica, possiamo esprimere il duale come

$$\begin{aligned} & -\max\{y(-b) \mid (yA \geq c) \wedge (y \geq 0)\} \\ = & -\max\{(-b^T)y \mid ((-A^T)y \leq -c) \wedge (y \leq 0)\} \end{aligned}$$

il cui duale è

$$\begin{aligned} & -\min\{-cx \mid ((x(-A^T) \geq (-b)) \wedge (x \geq 0)\} \\ = & \max\{cx \mid Ax \leq b) \wedge (x \geq 0)\} \end{aligned}$$

Teorema Debole di Dualità

Teorema

Se $\overline{x}$ e $\overline{y}$ sono soluzioni ammissibili per il primale e il duale, rispettivamente, allora $c\overline{x} \leq \overline{y}b$.

Dimostrazione.

Dimostriamo il teorema nel caso della coppia asimmetrica:

$$\left.\begin{array}{r} A\overline{x} \leq b \\ \overline{y}A = c, \overline{y} \geq 0 \end{array}\right\} \Longrightarrow \left.\begin{array}{c} \overline{y}A\overline{x} \leq \overline{y}b \\ \overline{y}A\overline{x} = c\overline{x} \end{array}\right\} \Longrightarrow c\overline{x} \leq \overline{y}b$$

e nel caso della coppia simmetrica:

$$\left.\begin{array}{r} A\overline{x} \leq b, \overline{x} \geq 0 \\ \overline{y}A \geq c, \overline{y} \geq 0 \end{array}\right\} \Longrightarrow \left.\begin{array}{c} \overline{y}A\overline{x} \leq \overline{y}b \\ \overline{y}A\overline{x} \geq c\overline{x} \end{array}\right\} \Longrightarrow c\overline{x} \leq \overline{y}b$$

$\square$

Corollari del Teorema Debole di Dualità

Corollario

Se il primale è illimitato, allora il duale è vuoto.

Dimostrazione.

Se il primale è illimitato, allora per ogni $M \in \mathbb{R}$ esiste una soluzione ammissibile x per il primale con $cx > M$. Ma quindi, se per assurdo ci fosse y ammissibile per il duale, troveremmo x ammissibile per il primale con $cx > yb$, in contrasto con il TdD. $\square$

Corollario

Se $\overline{x}$ e $\overline{y}$ sono soluzioni ammissibili per il primale e il duale, rispettivamente, e $c\overline{x} = \overline{y}b$, allora $\overline{x}$ e $\overline{y}$ sono soluzioni ottime.

Dimostrazione.

Se $c\overline{x} = \overline{y}b$ e $\overline{x}$ non fosse ottima, troveremmo z ammissibile per il primale con $cz > c\overline{x}$ e quindi con $cz > \overline{y}b$, in contrasto con il TdD. $\square$

Direzioni Ammissibili — I

- Data una coppia asimmetrica, consideriamento una soluzione ammissibile $\overline{x}$ per il primale, e chiediamoci se spostandoci lungo una direzione dell'iperspazio a partire da $\overline{x}$, si resta o meno nella regione ammissibile.
- Un vettore $\xi \in \mathbb{R}^n$ è detto **Direzione Ammissibile** se *esiste* $\overline{\lambda} > 0$ tale che $x(\lambda) = \overline{x} + \lambda\xi$ è ammissibile nel primale per ogni $\lambda \in [0, \overline{\lambda}]$.

Direzioni Ammissibili — II

Lemma

Il vettore ξ è direzione ammissibile per $\overline{x}$ sse $A_{I(\overline{x})}\xi \leq 0$.

Dimostrazione.

- Un modo equivalente di definire ξ come direzione ammissibile è dire che per ogni $i \in \{1, \ldots, m\}$,

$$A_i x(\lambda) = A_i\overline{x} + \lambda A_i\xi \leq b_i$$

- Osserviamo però che:
  - Se $i \in I(\overline{x})$, allora $A_i\overline{x} = b_i$, e quindi l'equazione è verificata *se e solo se* $\lambda A_i\xi \leq 0$.
  - se $i \notin I(\overline{x})$, allora l'equazione è verificata da qualunque ξ, purché λ sia piccolo a sufficienza.

□

Direzioni di Crescita

- Una direzione $\xi \in \mathbb{R}^n$ è una **direzione di crescita** per $\overline{x}$ se uno spostamento λ lungo ξ fa crescere il valore della funzione obiettivo, ossia se:

$$cx(\lambda) = c\overline{x} + \lambda c\xi > c\overline{x} \Longleftrightarrow c\xi > 0$$

- La nozione di direzione di crescita, dunque, non dipende dal punto $\overline{x}$!
- Osserviamo che:
  - Se $c = 0$, allora la funzione obiettivo vale sempre 0 e quindi *tutte* le soluzioni ammissibili sono ottime.
  - Se $c \neq 0$, allora se esiste una direzione ammissibile per $\overline{x}$ che sia anche di crescita, allora $\overline{x}$ *non può essere* ottimo.

Sezione 3

L'Algoritmo del Simplesso

Algoritmo del Simplesso

- Prima di presentare l'algoritmo, conviene dare uno sguardo alla sua **struttura**.
- L'algoritmo procede **iterativamente**, visitando successivamente alcuni tra i vertici del poliedro che definisce l'insieme delle soluzioni ammissibili.
- Dato un vertice $\overline{x}$, si cerca prima di tutto di determinare se tale vertice sia o meno ad una **soluzione ottima**, cercando di determinare se esiste una soluzione $\overline{y}$ per il duale con lo stesso valore della funzione obiettivo.
- Nel caso in cui $\overline{x}$ non sia ottima, si cerca di spostarsi in un altro vertice, seguendo una **direzione di crescita**, che sia anche **ammissibile**.
- Se è possibile spostarsi indefinitamente lungo questa direzione di crescita, allora il problema è illimitato, altrimenti si incontra un'altro vertice, e **ci si sposta**.

$\textsc{SimplessoPrimale}(A, b, c, B)$

1. $N \leftarrow \{1, \ldots, m\} - B$;
2. $\overline{x} \leftarrow A_B^{-1} b_B$;
3. $\overline{y}_B \leftarrow c A_B^{-1}$;
4. $\overline{y}_N \leftarrow 0$;
5. Se $\overline{y}_B \geq 0$, allora termina con successo e restituisci $\overline{x}$ e $\overline{y}$;
6. $h \leftarrow \min\{i \in B \mid \overline{y}_i < 0\}$;
7. Sia ξ la colonna di indice h in $-(A_B^{-1})$;
8. Se $A_N \xi \leq 0$, allora termina e restituisci ξ: il problema è illimitato;
9. $k \leftarrow \arg\min\{\frac{b_i - A_i \overline{x}}{A_i \xi} \mid A_i \xi > 0 \wedge i \in N\}$;
10. $B \leftarrow B \cup \{k\} - \{h\}$;
11. Torna al punto 1.

Correttezza del Simplesso — I

- L'algoritmo lavora mantenendo i seguenti tre *invarianti*:
  - B è una **base ammissibile**;
  - $\overline{x}$ è **soluzione ammissibile** per il problema primale

    $$\max\{cx \mid Ax \leq b\};$$

    mentre $\overline{y}A = c$.
- Questo significa, tra l'altro, che $\overline{x}$ è sempre un vertice.
- Osserviamo che per $\overline{y}$ la condizione $\overline{y}A = c$ vale per come $\overline{y}_B$ e $\overline{y}_N$ vengono inizializzati.
- Di conseguenza, $\overline{y}$ è soluzione per il **duale**

  $$\max\{yb \mid yA = c, y \geq 0\}$$

  sse $\overline{y}_B \geq 0$.

Correttezza del Simplesso — II

- Se, quindi $\overline{y}_B \geq 0$, allora l'algoritmo correttamente **termina**, restituendo $\overline{x}$ e $\overline{y}$ che sono soluzioni ottime per il primale e per il duale (riga 5.)
- Se, invece, c'è un elemento di $\overline{y}$ strettamente negativo, allora non vale l'ottimalità. Cerchiamo quindi una direzione ammissibile e di crescita per $\overline{x}$.
  - ξ, per come definito in riga 7. è sempre **direzione di crescita**, perché

    $$c\xi = c(-A_B^{-1}u_h) = -(cA_B^{-1})u_h = -\overline{y}u_h = -\overline{y}_h > 0$$

    dove u_h è un vettore ovunque nullo, tranne nella componente corrispondente a i, in cui vale 1.
  - ξ, però, potrebbe non essere **direzione ammissibile**, e soprattutto, non sappiamo quale sia il "primo" vertice lungo ξ.

Correttezza del Simplesso — III

- Il vettore $A_B\xi$ è una delle colonne della matrice identica, cambiata di segno, e quindi $A_B\xi \leq 0$.
- Se $i \in N$ e $A_i\xi \leq 0$, allora $\overline{x}(\lambda)$ soddisfa l'i-esimo vincolo per ogni valore non-negativo di λ.
- Se $i \in N$ e $A_i\xi > 0$, allora

$$A_i\overline{x}(\lambda) = A_i\overline{x} + \lambda A_i\xi \leq b_i \quad \Leftrightarrow \quad \lambda \leq (b_i - A_i\overline{x})/A_i\xi$$

- Scegliamo l'indice i che rende tale λ minimo. Chiamiamolo k.
- Sia $\overline{\lambda}$ il valore $\min\{\lambda_i \mid i \in N\}$.

Correttezza del Simplesso — IV

1. Se $\overline{\lambda} = +\infty$, ossia se $A_N\xi < 0$, allora il problema è **illimitato**.
   - Questo caso è gestito dalla riga 8. dell'Algoritmo.
2. Se $0 < \overline{\lambda} < +\infty$, allora $x(\lambda)$ è ammissibile per ogni $\lambda \in [0, \overline{\lambda}]$ e non ammissibile altrimenti. Possiamo quindi spostarci da B a $B \cup \{k\} - \{h\}$, che corrisponde ad un **altro vertice**.
   - Questo caso è gestito dalle righe 9.-10. dell'Algoritmo.
3. Se $\overline{\lambda} = 0$, allora la direzione **non è ammissibile**, ma possiamo comunque effettuare un cambio di base verso $B \cup \{k\} - \{h\}$, che ci fa restare sullo stesso vertice.
   - Questo caso è gestito dalle righe 9.-10. dell'Algoritmo.

Complessità del Simplesso

- Si può dimostrare che ogni base ammissibile viene trattata **al più** una volta durante l'esecuzione dell'Algoritmo.
- Di conseguenza, vi saranno al più $\binom{m}{n}$ iterazioni, ovvero un numero che può divenire **esponenziale** in n.
- Detto questo:
  - Da un punto di vista *teorico*, la complessità dell'Algoritmo nel caso **medio** è polinomiale.
  - Da un punto di vista *pratico*, si osserva come il Simplesso sia l'algoritmo più efficiente, e che si comporti **meglio** di altri algoritmi (alcuni dei quali si possono dimostrare essere polinomiali in tempo).

Sezione 4

La Tecnica Branch-and-Bound

Ricerca dell'Ottimo in PLI

- Finora ci siamo occupati della ricerca dell'ottimo in programmi lineari in cui **tutte** le variabili siano reali.
  - L'Algoritmo del Simplesso si basa in modo essenziale su quest'assunzione.
  - Ciò ci permette di costruire tecniche risolutive che, almeno nel caso medio, lavorano in tempo polinomiale.

Ricerca dell'Ottimo in PLI

- Finora ci siamo occupati della ricerca dell'ottimo in programmi lineari in cui **tutte** le variabili siano reali.
  - L'Algoritmo del Simplesso si basa in modo essenziale su quest'assunzione.
  - Ciò ci permette di costruire tecniche risolutive che, almeno nel caso medio, lavorano in tempo polinomiale.
- Si può fare la stessa cosa in PLI?

Ricerca dell'Ottimo in PLI

- Finora ci siamo occupati della ricerca dell'ottimo in programmi lineari in cui **tutte** le variabili siano reali.
  - L'Algoritmo del Simplesso si basa in modo essenziale su quest'assunzione.
  - Ciò ci permette di costruire tecniche risolutive che, almeno nel caso medio, lavorano in tempo polinomiale.
- Si può fare la stessa cosa in PLI?
  - La risposta è purtroppo **negativa**.
  - Trovare l'ottimo di un programma lineare intero è, come sappiamo, un problema **NP**-completo. È quindi improbabile che vi siano soluzioni efficienti.

Ricerca dell'Ottimo in PLI

- Finora ci siamo occupati della ricerca dell'ottimo in programmi lineari in cui **tutte** le variabili siano reali.
  - L'Algoritmo del Simplesso si basa in modo essenziale su quest'assunzione.
  - Ciò ci permette di costruire tecniche risolutive che, almeno nel caso medio, lavorano in tempo polinomiale.
- Si può fare la stessa cosa in PLI?
  - La risposta è purtroppo **negativa**.
  - Trovare l'ottimo di un programma lineare intero è, come sappiamo, un problema **NP**-completo. È quindi improbabile che vi siano soluzioni efficienti.
- Alla PLI si applica però una tecnica, detta **branch-and-bound** che, anche se esponenziale in tempo nel caso peggiore, permette in molti casi di *evitare* l'enumerazione esaustiva.

Due Prerequisiti - I

- Presenteremo la tecnica del branch-and-bound *senza* fare esplicito riferimento alla PLI, ma tenendo bene in mente che questa è il caso di studio che abbiamo in mente.
- Perché la tecnica del branch-and-bound sia applicabile ad una certa classe di problemi di ottimizzazione (che assumiamo di minimo), devono valere due requisiti
  1. **Rilassamento**
     - Deve essere possibile, in ogni momento, passare da un problema $\mathbb{P}$ ad un suo rilassamento $\mathbb{T} = \mathsf{RELAX}(\mathbb{P})$, tipicamente più semplice da risolvere da un punto di vista computazionale.
     - In questo modo si possono facilmente calcolare limitazioni *inferiori* al valore ottimo di $\mathbb{P}$.
  2. **Branching**
     - Deve esistere un modo per *partizionare* l'insieme delle soluzioni ammissibili di $\mathbb{P}$ ottenendo due sottoproblemi $\mathsf{PARTITION}(\mathbb{P}) = (\mathbb{T}, \mathbb{Q})$.
     - In questo modo decomponendo un problema complesso in due problemi più semplici.

Due Prerequisiti - II

- In PLI, questi due prerequisiti sono entrambi soddisfatti.

Due Prerequisiti - II

- In PLI, questi due prerequisiti sono entrambi soddisfatti.
- Il **rilassamento** di un PLI si ottiene semplicemente *non considerando* i vincoli di interezza e dà luogo ad un PL:

$$\mathsf{RELAX}(\min\{cx \mid Ax \leq b \wedge x \in \mathbb{Z}^n\}) = \min\{cx \mid Ax \leq b\}$$

- Il **branching** di un PLI si ottiene scegliendo una variabile x_i e un bound n:

$$\begin{aligned}\mathsf{PARTITION}(\min\{cx \mid Ax \leq b \wedge x \in \mathbb{Z}^n\}) = \\ (\min\{cx \mid Ax \leq b \wedge x_i \leq n \wedge x \in \mathbb{Z}^n\}, \\ \min\{cx \mid Ax \leq b \wedge x_i \geq n+1 \wedge x \in \mathbb{Z}^n\})\end{aligned}$$

  In questo senso $\mathsf{PARTITION}(\cdot)$ non è una vera e propria *funzione*, ma ogni strategia di scelta per x_i e n va bene.

BranchAndBound($\mathbb{P}$)

1. $S \leftarrow \{\mathbb{P}\}$; $v^* \leftarrow \infty$;
2. Se $S = \emptyset$, allora termina e restituisci la soluzione ottima x^* se definita;
3. Scegli un problema $\mathbb{T}$ in S; $S \leftarrow S - \{\mathbb{T}\}$;
4. Se $\mathsf{RELAX}(\mathbb{T})$ è vuoto, allora ritorna al punto 2;
5. Se $\mathsf{RELAX}(\mathbb{T})$ è illimitato, allora $S \leftarrow S \cup \{\mathbb{Q}, \mathbb{S}\}$ dove $(\mathbb{Q}, \mathbb{S}) = \mathsf{PARTITION}(\mathbb{T})$ e ritorna al punto 2;
6. Siano x e v la soluzione e il valore ottimo di $\mathsf{RELAX}(\mathbb{T})$
7. Se $v \geq v^*$, allora torna al punto 2;
8. Se x è soluzione ammissibile per $\mathbb{T}$ e $v < v^*$, allora $v^* \leftarrow v$ e $x^* \leftarrow x$; torna al punto 2;
9. Se x non è soluzione ammissibile per $\mathbb{T}$ e $v < v^*$, allora $S \leftarrow S \cup \{\mathbb{Q}, \mathbb{S}\}$ dove $(\mathbb{Q}, \mathbb{S}) = \mathsf{PARTITION}(\mathbb{T})$ e ritorna al punto 2;

Sulla Correttezza

- La correttezza dell'algoritmo di BranchAndBound si basa sulle seguenti osservazioni:
  1. Se il test dell'istruzione 7 dà esito positivo, allora esiste una soluzione ottima di $\mathbb{P}$ che non sta in $\mathbb{T}$.
  2. Nell'ambito dell'istruzione 9, ogni soluzione ammissibile per $\mathbb{T}$ si trova in $\mathbb{Q}$ oppure in $\mathbb{S}$.
  3. Ogniqualvolta viene eseguita l'istruzione 2, l'ottimo di $\mathbb{P}$ è in x^* (se definita) oppure è uno tra gli ottimi dei problemi in S.
     - Certamente questa condizione vale all'inizio.
     - Se tale condizione vale e $S \neq \emptyset$, allora si può far vedere che tale condizione continuerà a valere la prossima volta che si torna in linea 2: per convincerci di questa cosa basta una semplice analisi per casi sulla natura di $\mathbb{T}$.

Sulla Complessità

- **Non si può dire molto** sulla complessità di BranchAndBound in senso astratto.
  - Se il problema di partenza è illimitato, la procedura può anche *divergere*!

Sulla Complessità

- **Non si può dire molto** sulla complessità di BranchAndBound in senso astratto.
  - Se il problema di partenza è illimitato, la procedura può anche *divergere*!
- Le procedure $\mathsf{RELAX}(\cdot)$ e $\mathsf{PARTITION}(\cdot)$ hanno molti gradi di libertà. Inoltre la scelta del problema in S è nondeterministica.
  - Risolvere tale nondeterminismo in un modo piuttosto che in un altro può avere un impatto enorme sulle performance dell'algoritmo.
  - Sperimentalmente, si osserva che trattare S come uno **stack** (ossia visitare il sottostante albero depth-first) può portare ad un miglioramento delle prestazioni.
  - Sempre sperimentalmente, si osserva che fare branch in modo da portare v^* a diminuire *più* possibile è una strategia che paga.