

# Capitolo 4

## Programmazione Lineare Intera

(Versione in  $\LaTeX$  a cura di Lorenzo Pace, [l.pace4@studenti.unipi.it](mailto:l.pace4@studenti.unipi.it))

### 4.1 Problemi di PLI

Un problema di Programmazione Lineare Intera è un problema di ottimizzazione della stessa forma dei problemi di  $PL$ , ma con l'aggiunta di un *vincolo di interezza* su tutte le variabili:

$$\begin{array}{ll} \max c \cdot x & \min y \cdot b \\ (P_I) \quad Ax \leq b & (D_I) \quad yA = c \\ & x \in \mathbb{Z}^n & y \in \mathbb{Z}_+^m \end{array}$$

Chiamiamo  $P$  e  $Q$  i poliedri individuati rispettivamente dai vincoli di  $(P_I)$  e  $(D_I)$ , e descriviamo le regioni ammissibili  $S$  e  $T$  dei due problemi:

$$\begin{array}{ll} P = \{x \in \mathbb{R}^n : Ax \leq b\} & Q = \{y \in \mathbb{R}^m : yA = c, y \geq 0\} \\ S = P \cap \mathbb{Z}^n & T = Q \cap \mathbb{Z}^m \end{array}$$

#### Regione ammissibile

La regione ammissibile  $S$  del problema primale non è quindi un poliedro, come nei problemi di  $PL$ , ma un *reticolo intero* composto da tutti i punti del poliedro  $P$  a componenti intere.

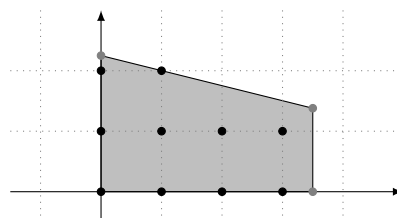


Figura 4.1

#### Rilassamento continuo

Se si priva un problema  $(P_I)$  di  $PLI$  dei suoi vincoli di interezza si ottiene un problema di  $PL$  che chiamiamo *rilassamento continuo* di  $(P_I)$ :

$$\begin{array}{ll} \max c \cdot x & \min y \cdot b \\ (RC_P) \quad Ax \leq b & (RC_D) \quad yA = c \\ & y \geq 0 \end{array}$$

Se  $\bar{x} \in P$  soluzione ottima di  $(RC_P)$  è a componenti intere (ossia se  $x \in S$ ) allora si ha che  $\bar{x}$  è soluzione ottima di  $(P_I)$  (analogamente con  $(RC_D)$ ).

Questa situazione accade sicuramente qualora tutti i vertici di  $P$  siano a componenti intere, ossia quando ogni vertice è soluzione di base ammissibile (*proprietà dell'interezza*).

**Esempio 4.1:**

Sia  $A$  la matrice di incidenza nodi/archi di un grafo, con  $c$  a componenti intere. Allora il poliedro  $Q = \{y \in \mathbb{R}^m : yA = c, y \geq 0\}$  ha la proprietà dell'interezza (in quanto gli indici dei nodi sono ovviamente interi).

**Arrotondamento intero**

Si potrebbe pensare che, data una soluzione ottima  $\bar{x} \in P$  di  $RC_P$ , allora un opportuno arrotondamento intero delle componenti frazionarie fornisca una soluzione ottima di  $(P)$ . Questo è falso, come illustra il seguente esempio.

**Esempio 4.2:**

$$\begin{aligned} \max \quad & x_1 + x_2 \\ & x_1 \leq 5/2 \\ & x_1 + 3x_2 \leq 7 \\ & x_1, x_2 \geq 0 \quad x_1, x_2 \in \mathbb{Z} \end{aligned}$$

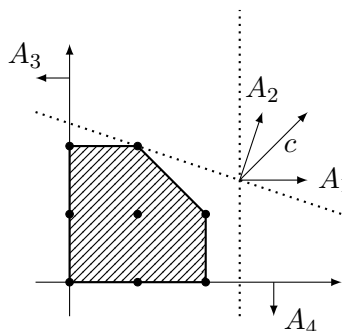


Figura 4.2

Notiamo che la soluzione del rilassamento continuo è  $(5/2, 3/2)$ , mentre la soluzione ottima è  $(1, 2)$ . Dato che  $1 \notin \{[5/2], \lceil 5/2 \rceil\}$ , si ha un controesempio dell'arrotondamento intero. Nella figura è evidenziato il poliedro che corrisponde all'involucro convesso del reticolo intero  $S$ . Notiamo che questo poliedro contiene *tutte le soluzioni* del problema e gode della *proprietà dell'interezza*: ne consegue che se utilizziamo tale poliedro come regione ammissibile di un problema di  $PL$  con la stessa funzione obiettivo del problema originale otteniamo un problema di  $PL$  con *le stesse soluzioni ottime* del problema di  $PLI$  originale (vedremo un caso più generale nel prossimo paragrafo).

**Convessificazione reticolo intero**

Consideriamo l'involucro (o *inviluppo*) convesso di  $S$ :

$$\text{conv } S = \left\{ \sum_{i=1}^{n+1} \lambda_i x^i : \lambda_i \geq 0, \sum_{i=1}^{n+1} \lambda_i = 1, x^i \in S \right\} = \bigcap_{\substack{A \supseteq S \\ A \text{ convesso}}} A$$

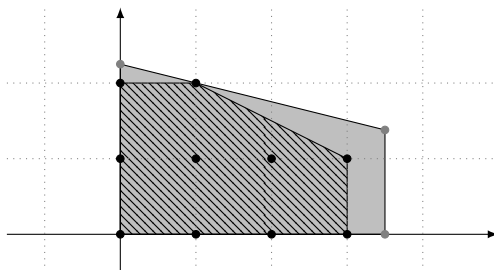


Figura 4.3: Inviluppo convesso di  $S$ .

L'involucro convesso di  $S$  non è sempre un poliedro, come si vede dal seguente esempio.

**Esempio 4.3:**

$P = \{x \in \mathbb{R}_+^2 : x_2 \geq \sqrt{2}x_1\}$ . Si ha che  $\text{conv } S$  coincide con il cono  $C = \{x \in \mathbb{R}_+^2 : x_2 > \sqrt{2}x_1\}$ , che non è un poliedro poiché non è chiuso.

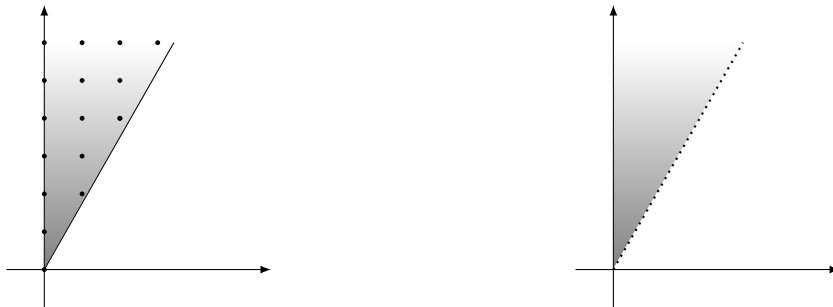


Figura 4.4:  $P$  e  $\text{conv } S$ .

(Intuizione: ci sono punti a componenti intere arbitrariamente vicini al bordo mancante, ma nessun punto sul bordo ha componenti intere.)

*Dimostrazione.*

- Si dimostra prima che  $\text{conv } S \subseteq C$ : se  $x \in S$ , allora  $x \in P$ , cioè  $0 \leq \sqrt{2}x_1 \leq x_2$ . poiché  $\sqrt{2}$  è irrazionale e  $x$  ha componenti intere si ha  $0 \leq \sqrt{2}x_1 < x_2$ , cioè  $x \in C$ , quindi  $S \subseteq C$ . Poiché  $C$  è convesso si ha anche  $\text{conv } S \subseteq C$ .
- Dimostriamo ora che  $C \subseteq \text{conv } S$ : consideriamo  $y \in C$ . Se  $y = (0, 0)$ , allora  $y \in S \subseteq \text{conv } S$ . In caso contrario si ha:

$$0 < \frac{y_1}{y_2} < \frac{1}{\sqrt{2}}$$

Poiché  $\mathbb{Q}$  è denso in  $\mathbb{R}$ , è possibile trovare un razionale  $a/b$  tale che:

$$0 < \frac{y_1}{y_2} < \frac{a}{b} < \frac{1}{\sqrt{2}}$$

Quindi il vettore  $x = (a, b)$  ha componenti intere e  $0 < \sqrt{2}x_1 < x_2$ , perciò  $x \in S$ . Analogamente tutti i vettori del tipo  $(na, nb)$ ,  $n \in \mathbb{N}_+$  appartengono ad  $S$ , e quindi la semiretta:

$$r_1 = \left\{ x \in \mathbb{R}_+^2 : x_1 = \frac{a}{b}x_2 \right\}$$

è contenuta in  $\text{conv } S$ . Inoltre è evidente che la semiretta:

$$r_2 = \{x \in \mathbb{R}_+^2 : x_1 = 0\}$$

è contenuta in  $\text{conv } S$ ; allora questo contiene anche il cono generato dalle due semirette, cioè:

$$\left\{ x \in \mathbb{R}_+^2 : 0 \leq x_1 \leq \frac{a}{b}x_2 \right\} \subseteq \text{conv } S$$

Dato che  $0 < y_1 < \frac{a}{b}y_2$ , vale  $y \in \text{cono}(r_1, r_2) \subseteq \text{conv } S$

■

Se  $S$  è finito oppure  $A$  e  $b$  sono a componenti intere si ha che  $\text{conv } S$  è un poliedro con la proprietà dell'interrezza, pertanto:

$$\max_{x \in S} c \cdot x \quad \equiv \quad \max_{x \in \text{conv } S} c \cdot x$$

ma caratterizzare  $\text{conv } S$  tramite vincoli espliciti è estremamente difficile (la caratterizzazione è nota in pochi casi). Vi sono alcuni metodi per trovare iterativamente l'involucro convesso di  $S$  "tagliando" porzioni della regione ammissibile, detti *metodi dei piani di taglio*.

## 4.2 Metodi dei piani di taglio

### Disuguaglianze valide

Siano  $d \in \mathbb{R}^n$  e  $\gamma \in \mathbb{R}$ .

$d \cdot x \geq \gamma$  si dice *disuguaglianza valida* per  $S$  se  $d \cdot x \geq \gamma, \forall x \in S$ , ovvero

$$\{x \in \mathbb{R}^n : Ax \leq b, d \cdot x \geq \gamma\} \cap \mathbb{Z}^n = S$$

### Piani di taglio

Sia  $\bar{x} \in S$  soluzione ottima del rilassamento continuo ( $RC_P$ ).

Una disuguaglianza valida è detta *piano di taglio* se:

$$d \cdot \bar{x} < \gamma$$

Ovvero se  $\{x \in \mathbb{R}^n : Ax \leq b, d \cdot x \geq \gamma\} \supseteq S$  approssima  $S$  meglio di  $P$  e “taglia” fuori la soluzione ottima del relativo rilassamento.

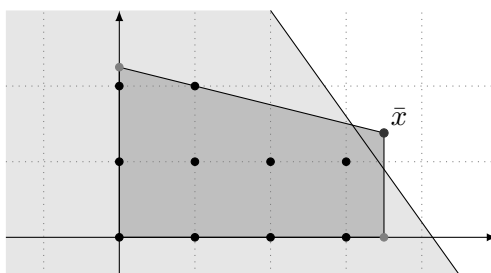


Figura 4.5: Piano di taglio.

### Piani di taglio di Gomory

Consideriamo un problema in forma duale ed il suo rilassamento continuo:

$$(D_I) \quad \begin{array}{l} \min y \cdot b \\ yA = c \\ y \in \mathbb{Z}^m \end{array} \quad \rightsquigarrow \quad (RC_D) \quad \begin{array}{l} \min y \cdot b \\ yA = c \\ y \geq 0 \end{array}$$

Con  $A \in \mathbb{R}^{m \times n}$

Siano  $\bar{y} = (cA_B^{-1}, 0)$  soluzione ottima di base per  $(RC_D)$ , e supponiamo  $\bar{y}_r \notin \mathbb{Z}$  (se tale  $r$  non esistesse  $\bar{y}$  sarebbe soluzione ottima di  $D_I$ ).

Sia  $\tilde{A} = A_N A_B^{-1} \in \mathbb{R}^{(m-n) \times n}$

(si numerino le righe con gli indici di  $N$  e le colonne con quelli di  $B$ )

Vale la seguente proposizione:

$$\sum_{j \in N} \{\tilde{A}_{jr}\} y_j \geq \{\bar{y}_r\} \text{ è un piano di taglio.}$$

(dove  $\{\cdot\}$  denota la parte frazionaria.<sup>1</sup>)

<sup>1</sup>Si ricordi che, ad esempio:

$\{3.7\} = 3.7 - \lfloor 3.7 \rfloor = 0.7$ , mentre  $\{-3.7\} = \lceil 3.7 \rceil - 3.7 = 0.3$

*Dimostrazione.*

- Si dimostra anzitutto che  $\bar{y}$  non soddisfa la disuguaglianza ( $\bar{y}_j = 0$  per  $j \in N$ ).

$$\sum_{j \in N} \{\tilde{A}_{jr}\} \bar{y}_j = 0 < \{\bar{y}_r\} \quad (\bar{y}_r \notin \mathbb{Z} \implies \{\bar{y}_r\} > 0)$$

- Si dimostra quindi che la disuguaglianza è valida:

Sia  $y \in Q \cap \mathbb{Z}^m$ :

$$\begin{aligned} y_B A_B + y_N A_N = c &\implies y_B = (c - y_N A_N) A_B^{-1} \\ &= c A_B^{-1} - y_N A_N A_B^{-1} \\ &= \bar{y}_B - y_N \underbrace{A_N A_B^{-1}}_A \end{aligned}$$

$$\text{Da cui } y_r = \bar{y}_r - \sum_{j \in N} \{\tilde{A}_{jr}\} y_j = [\bar{y}_r] + \{\bar{y}_r\} - \sum_{j \in N} ([\tilde{A}_{jr}] + \{\tilde{A}_{jr}\}) y_j$$

(dove  $[\cdot]$  denota la parte intera)

$$-\{\bar{y}_r\} + \sum_{j \in N} \{\tilde{A}_{jr}\} y_j = [\bar{y}_r] - \underbrace{y_r}_{\in \mathbb{Z}} - \sum_{j \in N} [\tilde{A}_{jr}] \underbrace{y_j}_{\in \mathbb{Z}} \in \mathbb{Z}$$

$$\{\tilde{A}_{jr}\}, y_j \geq 0 \implies -\{\bar{y}_r\} + \sum_{j \in N} \{\tilde{A}_{jr}\} y_j \geq -\{\bar{y}_r\} > -1$$

Poiché  $-\{\bar{y}_r\} + \sum_{j \in N} \{\tilde{A}_{jr}\} y_j \in \mathbb{Z}$ , risulta  $-\{\bar{y}_r\} + \sum_{j \in N} \{\tilde{A}_{jr}\} y_j \geq 0$

Ovvero  $\sum_{j \in N} \{\tilde{A}_{jr}\} y_j \geq \{\bar{y}_r\}$  e  $y$  soddisfa la disuguaglianza. ■

Se  $\bar{y} \cdot b \notin \mathbb{Z}$ , si può dimostrare analogamente che:

$$\sum_{j \in N} \{b_j - A_j A_B^{-1} b_B\} y_j \geq \{\bar{y} \cdot b\} \text{ è un piano di taglio.}$$

(si utilizzi  $y \cdot b = y_B b_B + y_N b_N = \bar{y} \cdot b + y_N (b_N - A_N A_B^{-1} b_B)$ )

### Algoritmo di Gomory

Un algoritmo basato sui piani di taglio è l'algoritmo di Gomory:

1. Calcolare  $\bar{y}$  soluzione ottima di base di  $\min\{y \cdot b : y \in Q\}$
2. Se  $y \in \mathbb{Z}^m$ : STOP
3. Costruire piano di taglio  $d \cdot x \geq \gamma$  relativo a  $\bar{y}$
4.  $Q := Q \cap \{x \in \mathbb{R}^n : d \cdot x \geq \gamma\}$ , ritornare a (1)

L'algoritmo termina in un numero finito di passi se i piani sono scelti con opportune regole.

**Esempio 4.4:**

$$\begin{aligned}
 \max \quad & x_2 \\
 & + 3x_1 + 2x_2 \leq 6 \\
 & - 3x_1 + 2x_2 \leq 0 \\
 & x_1, x_2 \geq 0 \quad x_1, x_2 \in \mathbb{Z}
 \end{aligned}$$

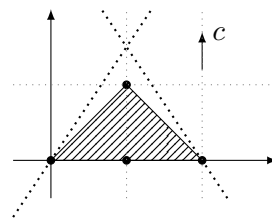


Figura 4.6

Per individuare un piano di taglio di Gomory bisogna portare il problema in forma duale:

$$\begin{aligned}
 \min \quad & -y_2 \\
 & + 3y_1 + 2y_2 + y_3 \leq 6 \\
 & - 3y_1 + 2y_2 + y_4 \leq 0 \\
 & y_1, y_2, y_3, y_4 \in \mathbb{Z}_+
 \end{aligned}
 \quad
 A = \begin{pmatrix} 3 & -3 \\ 2 & 2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}
 \quad
 b = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \end{pmatrix}$$

$$c = (6, 0)$$

Data la base  $B = \{1, 2\}$  una soluzione ottima di base del rilassamento continuo è  $\bar{y} = (1, 3/2, 0, 0)$ , quindi  $r = 2$ . Calcolo  $\tilde{A}$ :

$$A_B = \begin{pmatrix} 3 & -3 \\ 2 & 2 \end{pmatrix} \quad A_B^{-1} = \frac{1}{12} \begin{pmatrix} 2 & 3 \\ -2 & 3 \end{pmatrix}$$

$$\tilde{A} = A_N A_B^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1/6 & 1/4 \\ -1/6 & 1/4 \end{pmatrix} = \begin{pmatrix} 1/6 & 1/4 \\ -1/6 & 1/4 \end{pmatrix} \begin{matrix} 3 \\ 4 \end{matrix}$$

Piano di taglio di Gomory:  $\frac{1}{4}y_3 + \frac{1}{4}y_4 \geq \left\lceil \frac{3}{2} \right\rceil = \frac{1}{2} \rightarrow y_3 + y_4 \geq 2$

E dato che:

$$y_3 = 6 - 3y_1 - 2y_2 = 6 - 3x_1 - 2x_2 \quad y_4 = 3y_1 - 2y_2 = 3x_1 - 2y_2$$

Possiamo riscrivere il taglio di Gomory come:

$$\begin{aligned}
 y_3 + y_4 \geq 2 & \iff 6 - 4x_2 \geq 2 \\
 & \iff x_2 \leq 1
 \end{aligned}$$

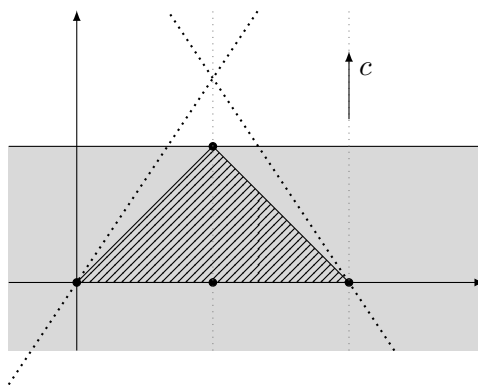


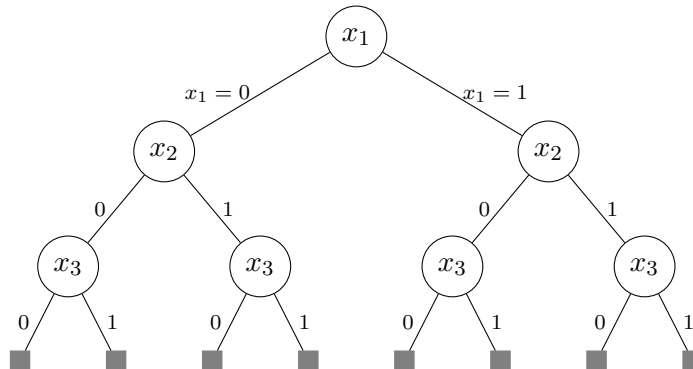
Figura 4.7

## 4.3 Metodi enumerativi

I problemi di *PLI* con un numero finito di soluzioni, possono essere risolti *per enumerazione*. In particolare considereremo problemi con variabili  $x \in \{0, 1\}^n$

### Albero di enumerazione totale

I metodi enumerativi si basano sull'*albero di enumerazione totale*, un albero radicato in cui i nodi di un dato livello identificano una variabile e gli archi che portano al livello successivo i possibili valori della variabile. Le foglie dell'albero individuano univocamente tutte le possibili soluzioni (incluse quelle non ammissibili):



Ogni nodo individua un sottoproblema (=sottoalbero) in cui i valori di alcune variabili sono stati fissati.

### 4.3.1 Branch and Bound

Talvolta enumerare e controllare tutte le possibili soluzioni non è fattibile: esistono dei metodi che ci permettono di escludere molte delle soluzioni ammissibili, controllando soltanto alcune delle soluzioni del problema.

L'idea di base dei metodi *branch and bound* è esplorare l'albero, chiudendo/potando quei rami che conducono a foglie che individuano soluzioni non ammissibili o a soluzioni ammissibili non migliori della migliore soluzione ammissibile nota (detta *soluzione corrente*). Tramite l'esplorazione diretta o indiretta (tramite la potatura dei rami) di tutte le possibili soluzioni si individua quella ottima.

Poiché ciascun sottoproblema è analogo a quello di partenza (ma con meno variabili), serve conoscere una stima delle qualità delle soluzioni del sottoproblema che sia più facilmente calcolabile. A questo scopo si considerano opportuni *rilassamenti* del problema, cioè problemi che contengono tutte le soluzioni ammissibili del (sotto)problema più altre, e la cui soluzione ottima sia ottenibile tramite algoritmi noti.

Uno specifico metodo branch and bound richiede:

1. Una soluzione ammissibile di partenza, da costruire con una tecnica euristica ad hoc
2. Un rilassamento del problema
3. Regole di ramificazione (che mi servono a costruire l'albero dinamicamente)
4. Regole di potatura

Posso notare quando:

- (a) Il sottoalbero non contiene soluzioni ammissibili per il nostro problema;
- (b) La soluzione ottima individuata per il rilassamento del sottoproblema è ammissibile per il problema di partenza  
(Sol. ottima individuata migliore della sol. amm. corrente  $\rightarrow$  aggiornare la sol. amm. corrente)
- (c) Il valore ottimo del rilassamento del sottoproblema è *non migliore* del valore della soluzione ammissibile corrente;

Il caso (a) è ovvio. Nel caso (b) la soluzione ottima del rilassamento è la migliore soluzione ammissibile per il problema originale in tutto il sottoalbero considerato, quindi è inutile proseguire l'esplorazione di quel sottoalbero. Qualora fosse anche migliore della soluzione ammissibile corrente, si sostituisce quest'ultima con la soluzione trovata. Nel caso (c) nessuna soluzione può essere migliore della soluzione corrente, quindi l'esplorazione del sottoalbero può terminare.

Si osservi che problemi di massimizzazione (minimizzazione) il valore della soluzione corrente costituisce una valutazione inferiore (superiore) del valore ottimo e la soluzione ottima di un rilassamento (completata con le variabili di valore fissato) costituisce una valutazione superiore (inferiore).

### Problema dello zaino

Ho  $n$  oggetti, a ciascuno dei quali sono associati un peso  $a_i$  ed un valore  $c_i$ . Lo zaino ha capacità  $b > 0$ .

$$\begin{aligned} \max c \cdot x \\ ax \leq b \\ x \in \{0, 1\}^n \end{aligned}$$

Utilizziamo come rilassamento il *rilassamento continuo* del problema, mantenendo come vincolo  $x \geq 0$ , e consideriamolo nella coppia simmetrica col suo duale:

### Rilassamento continuo I

$$\begin{array}{ll} \max c \cdot x & \min y \cdot b \\ ax \leq b & y \geq c_i/a_i \\ x \geq 0 & i = 0 \dots n \\ & y \geq 0 \end{array}$$

Il valore ottimo del duale è  $\bar{y} = \max \left\{ \frac{c_i}{a_i}, i = 0 \dots n \right\}$ , quindi (per gli scarti complementari in forma simmetrica) la soluzione ottima del rilassamento è:

$$\bar{x}_i = \begin{cases} b/a_i & \text{se } i = k \\ 0 & \text{se } i \neq k \end{cases}$$

In effetti  $b/a_i$  è il *beneficio unitario* (o **rendimento**) di un oggetto.



### Tecnica euristica dei rendimenti decrescenti

Posso ordinare gli elementi per beneficio unitario decrescente ed inserirli in ordine escludendo quelli che causerebbero uno *sforamento* della capacità. La soluzione ammissibile risultante  $\hat{x} \in \{0, 1\}^n$  è caratterizzata dalle seguenti proprietà:

$$(RD) \quad \begin{aligned} \hat{x}_1 &= 1 \\ \hat{x}_j = 1 &\iff b - \sum_{i=1}^{j-1} a_i \hat{x}_i \geq a_j \quad j \geq 2 \end{aligned}$$

Analoghe tecniche euristiche di tipo “greedy” sui benefici decrescenti oppure sui pesi crescenti  $\rightarrow$  non sono legate a rilassamenti ma in alcuni casi possono fornire soluzioni migliori.

### Rilassamento continuo II

$$\begin{aligned} \max c \cdot x \\ ax \leq b \\ 0 \leq x \leq 1 \end{aligned}$$

Costituisce un “miglior” rilassamento continuo (in quanto la regione ammissibile è più piccola).

Supponendo  $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \dots \geq \frac{c_n}{a_n}$ , la soluzione ottima è data da  $\bar{x}$  con :

$$\bar{x}_j = \begin{cases} 1 & \text{se } j < h \\ \frac{b - \sum_{i=1}^h a_i}{a_h} & \text{se } j = h + 1 \\ 0 & \text{se } j > h + 1 \end{cases}$$

Dove  $h \in \{1, \dots, n - 1\}$  è tale che  $\sum_{i=1}^h a_i \leq b$ ,  $\sum_{i=1}^{h+1} a_i > b$  (ossia è il numero di elementi che inseriamo).

*Dimostrazione.* Proviamo l’ottimalità di  $\bar{x}$ :

Posto  $\bar{z}_i = a_i x_i$  si ottiene:

$$\bar{z}_j = \begin{cases} a_i & \text{se } j < h \\ b - \sum_{i=1}^h a_i & \text{se } j = h + 1 \\ 0 & \text{se } j > h + 1 \end{cases}$$

e il rilassamento continuo assume la forma:

$$\begin{array}{ll} \max (c_1/a_1)z_1 + \dots + (c_n/a_n)z_n & \text{forma} \\ z_1 + \dots + z_n \leq b & \text{matriciale} \\ 0 \leq z_i \leq a_i, i = 1 \dots n & \end{array} \quad \begin{array}{l} \max \hat{c} \cdot z \\ \hat{A}z \leq b \\ z \geq 0 \end{array}$$

con  $\hat{c}_i = \frac{c_i}{a_i}$ ,  $\hat{A} = \begin{pmatrix} 1 & \dots & 1 \\ & I & \end{pmatrix}$ ,  $\hat{b} = \begin{pmatrix} b \\ a_1 \\ \vdots \\ a_n \end{pmatrix}$

Il cui duale (simmetrico) risulta essere:

$$\begin{array}{ll} \max w \cdot \hat{b} & \min yb + \sum_{i=1}^n a_i v_i \\ w \hat{A} \geq \hat{c} & y + v_i \geq \frac{c_i}{a_i} \quad i = 1 \dots n \\ w \geq 0 & y \geq 0, v_i \geq 0, \quad i = 1 \dots n \end{array} \quad w = (y, v) \in R \times R^n$$

Le condizioni degli scarti complementari (in forma simmetrica) sono pertanto:

$$\begin{aligned} w(\hat{b} - \hat{A}z) = 0 & \iff y(z_1 + \dots + z_n - b) = 0 & (1) \\ (w\hat{A} - \hat{c})z = 0 & \iff v_i(a_i - z_i) = 0 & (2) \\ & \iff (y + v_i - c_i/a_i)z_i = 0 & (3) \end{aligned}$$

Per dimostrare l'ottimalità di  $\bar{z}$  (e quindi di  $\bar{x}$ ) bisogna trovare  $\bar{w} = (\bar{y}, \bar{v})$  ammissibile per il problema duale tale che  $\bar{z}$  e  $\bar{w}$  soddisfino (1), (2), (3):

- $\bar{z}_1 + \dots + \bar{z}_n = b \implies$  (1) è verificata
- $\bar{z}_i < a_i, i \geq h+1 \implies$  (2) è verificata se  $\bar{v}_{h+1} = \dots = \bar{v}_n = 0$
- $\bar{z}_i > 0, i \leq h \implies$  (3) è verificata se  $\bar{v}_i = c_i/a_i - \bar{y}, i = 1 \dots h$

Un tale  $(\bar{y}, \bar{v})$  è ammissibile se e solo se  $\frac{c_{h+1}}{a_{h+1}} \leq \bar{y} \leq \frac{c_h}{a_h}$

Infatti:

$$\begin{aligned} \bar{y} \leq \frac{c_h}{a_h} & \implies \bar{y} \leq \frac{c_i}{a_i}, i \leq h \implies v_i \geq 0, i \leq h \\ \bar{y} \geq \frac{c_{h+1}}{a_{h+1}} & \implies \bar{y} + \bar{v}_i - \frac{c_i}{a_i} = \bar{y} - \frac{c_i}{a_i} \geq \frac{c_{h+1}}{a_{h+1}} - \frac{c_i}{a_i} \geq 0, i \geq h+1 \end{aligned}$$

mentre  $\bar{y} + \bar{v}_i - \frac{c_i}{a_i} = 0$  per  $i = 1 \dots h$ . ■

### Branch And Bound per Zaino

Usiamo la tecnica euristica dei rendimenti decrescenti per individuare una soluzione ammissibile iniziale (in alternativa possiamo utilizzare altre tecniche "greedy"), ed il *Rilassamento II* come rilassamento per il metodo Branch and Bound.

Come possibili *regole di ramificazione* possiamo scegliere di ramificare sulla variabile con:

- Valore frazionario nella soluzione ottima del rilassamento continuo
- Miglior rendimento
- Peggior rendimento

#### Esempio 4.5:

Si ha il problema dello zaino con valori:  $c = (11, 23, 18, 6)$ , pesi  $a = (7, 6, 3, 2)$  e capacità  $b = 8$ :

$$\begin{aligned} \max \quad & 11x_1 + 23x_2 + 18x_3 + 6x_4 \\ & 7x_1 + 6x_2 + 3x_3 + 2x_4 \leq 8 \\ & x_1, x_2, x_3, x_4 \in \{0, 1\} \end{aligned}$$

Calcolo i rendimenti:  $(11/7, 23/6, 6, 3)$  ed ordino gli elementi per rendimento decrescente:  $(x_3, x_2, x_4, x_1)$ . Cerco una soluzione ammissibile:

- Aggiungo l'elemento 3, il cui peso è 3.
- Provo ad aggiungere l'elemento 2, ma non posso perché  $3 + 6 > 8$
- Aggiungo l'elemento 4, peso raggiunto 5
- Non posso aggiungere l'elemento 1,  $5 + 7 > 8$

→ La soluzione ammissibile trovata è  $(0, 0, 1, 1)$ , di valore 24, **stima inferiore** della sol. ottima.

Cerchiamo quindi una soluzione ottima al rilassamento II, per averne anche una stima superiore:

- Inseriamo nell'ordine di prima: Inserisco l'elemento 3, che entra per intero nello zaino (nuova capacità  $8 - 3 = 5$ ).
- Cerco di inserire l'elemento 2, che non entra per intero nello zaino, quindi ne prendo una frazione:  $x_2 = 5/6$
- Soluzione ottima:  $(0, 5/6, 1, 0)$ , di valore  $37 + 1/6$ . Possiamo considerare 37 come **stima superiore** perché il valore ottimo del problema deve essere intero.

Abbiamo perciò una *valutazione inferiore*  $V_I = 24$  ed una *valutazione superiore*  $\lfloor V_S \rfloor = 37$

Dobbiamo definire una regola di ramificazione: possiamo scegliere, ad esempio, di ramificare in ordine di rendimento crescente.

Ramifico perciò in base al valore di  $x_3$ , ottenendo i sottoproblemi  $P_1$  e  $P_2$ .

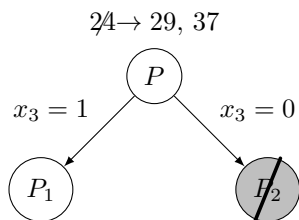


Figura 4.8: Poto  $P_2$  per ammissibilità

- La soluzione ottima del rilassamento di  $P_1$  è sempre  $(0, 5/6, 1, 0)$ ,  $\lfloor \text{valore} \rfloor = 37$
- La soluzione ottima del rilassamento di  $P_2$  è  $(0, 1, 0, 1)$  (ammissibile!), valore = 29 > 24  
 → Poto  $P_2$  (ammissibilità)  
 → soluzione ammiss. corrente:= 29

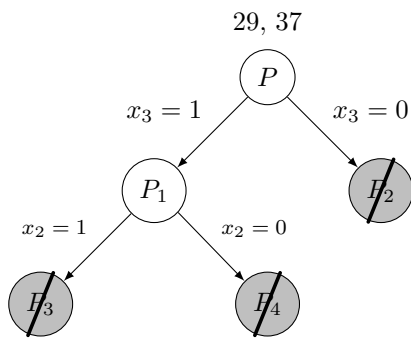


Figura 4.9: Poto  $P_3$  per inammissibilità e  $P_4$  per valore non migliore

- $P_3$  non contiene soluzioni ammissibili per P: poto
- La soluzione ottima del rilassamento di  $P_4$  è  $(3/7, 0, 1, 1)$ , valore = 28 < 29, poto  
 → **ho potato tutto.**  
 ⇒ soluzione ottima P:  $\hat{x} = (0, 1, 0, 1)$

## 4.4 Problema del commesso viaggiatore

Un commesso viaggiatore vuole visitare  $n$  città (tornando alla fine alla città di partenza) minimizzando il percorso complessivo. Se le distanze tra le città soddisfano la disuguaglianza triangolare, la soluzione ottima passa una sola volta per ogni città

Matematicamente si può astrarre il seguente problema: dato un grafo non orientato con un costo su ciascun arco, trovare il *ciclo hamiltoniano* di costo minimo (un ciclo hamiltoniano è un ciclo che passa per ogni nodo esattamente una volta – toccando tutti i nodi del grafo).

### K-alberi

Scelto un nodo  $k$ , un ciclo hamiltoniano è un albero di copertura per il grafo privato di  $k$  e di tutti gli archi incidenti in  $k$  unito a due archi incidenti in  $k$ .

Un grafo del genere (albero di copertura di costo minimo + due archi incidenti in  $k$  si dice *k-albero*).

Si noti che ogni ciclo hamiltoniano è un  $k$ -albero, ma non viceversa.

### Esempio 4.6:

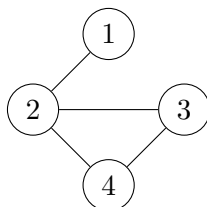


Figura 4.10: 3-albero che non è un ciclo hamiltoniano

**Prop:** Un ciclo hamiltoniano è un  $k$ -albero in cui ogni nodo ha grado 2 (cioè due archi incidenti)

### Branch and Bound per TSP simmetrico

Supponiamo che il grafo sia completo (TSP simmetrico), cioè esista l'arco  $(i, j)$  per ogni coppia di nodi  $i$  e  $j$ .

Poiché un ciclo hamiltoniano è un insieme di archi, si possono considerare le variabili binarie:

$$x_{ij} = \begin{cases} 1 & \text{se } (i, j) \text{ fa parte del ciclo} \\ 0 & \text{altrimenti} \end{cases}$$

per costruire l'albero di enumerazione totale.

### Tecnica euristica del “nodo più vicino”

Poiché il grafo è completo, ogni permutazione dei nodi individua un ciclo hamiltoniano. A partire da un qualsiasi nodo dato, una permutazione si ottiene visitando i nodi in una sequenza tale che il nodo che viene visitato sia, tra quelli non ancora visitati, quello a minor distanza dall'ultimo nodo visitato. Sia  $V = \{i_1, \dots, i_k\}$  l'insieme dei nodi già visitati:

$$i_{k+1} \text{ soddisfa: } \begin{cases} i_{k+1} \notin V \\ c_{i_k i_{k+1}} = \min\{c_{i_k j} : j \notin V\} \end{cases}$$

**Rilassamento**  $k$ -albero di costo minimo.

Questo può essere individuato risolvendo il problema dell'albero di copertura di costo minimo sul sottografo ottenuto rimuovendo  $k$  e gli archi in esso incidenti ed aggiungendo a tale albero i due archi di costo minimo incidenti in  $k$ .

**Esempio 4.7:**

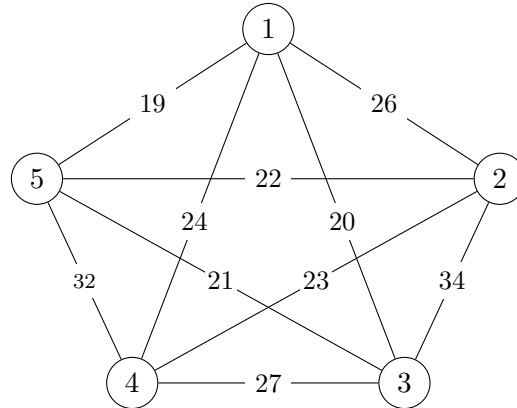


Figura 4.11

*Soluzione ammissibile iniziale:*

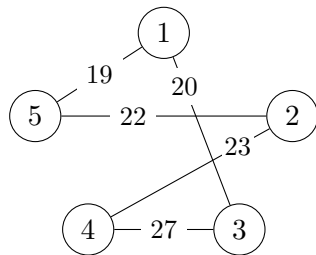


Figura 4.12

*Rilassamento:*

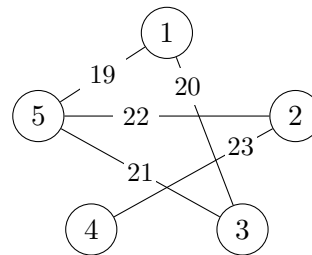


Figura 4.13

(algoritmo nodo più vicino a partire da 2)

(3-albero di costo minimo)

Il costo della soluzione è 111; questa è una valutazione superiore del valore ottimo.

Il costo del rilassamento è 105; questa è una valutazione inferiore del valore ottimo.

Ramifichiamo su  $x_{15}$  (arco di costo minore):

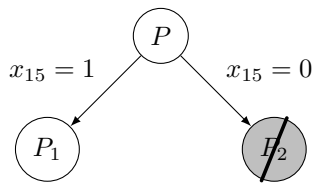


Figura 4.14

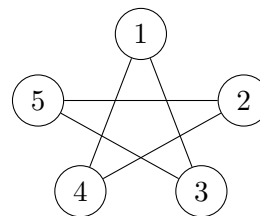


Figura 4.15: Ramo  $x_{15} = 0$

- Poto il ramo  $x_{15} = 0$  poiché il 3-albero di costo minimo con  $x_{15} = 0$  è anche un ciclo hamiltoniano.
- Nel caso  $x_{15} = 1$  si ha un 3-albero di costo minimo di costo 105 (lo stesso del rilassamento iniziale)

Ramifico su  $x_{13}$ :

- $x_{13} = 0$ : Il 3-albero di costo minimo è:

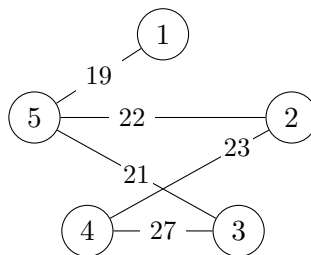


Figura 4.16

Poiché  $112 > V_S$ , nel sottoalbero non ci possono essere cicli hamiltoniani migliori della soluzione corrente, perciò potiamo.

- $x_{13} = 1$ : Il 3-albero di costo minimo è lo stesso dei passi precedenti, di costo  $105 < V_S$

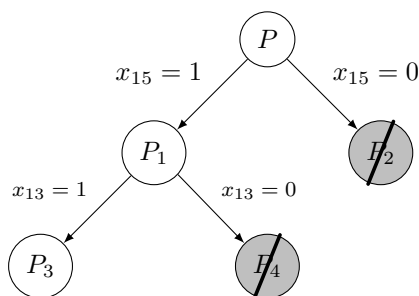


Figura 4.17: Poto  $P_3$  per inammissibilità e  $P_4$  per valore non migliore

Ramifichiamo su  $x_{35}$ :

- $x_{35} = 1$ : Nel sottoalbero non ci sono cicli hamiltoniani: potiamo
- $x_{35} = 0$ : Ciclo hamiltoniano di costo  $111 > V_S$ , si può potare e l'albero è chiuso. La soluzione ammissibile corrente è il ciclo hamiltoniano di costo minimo.

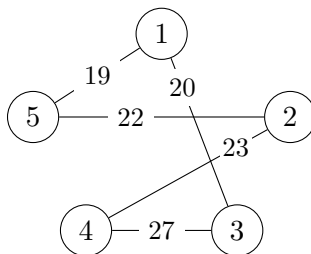


Figura 4.18:  $x_{35} = 0$

## Bibliografia

- Note manoscritte del professor Bigi:  
<http://pages.di.unipi.it/bigidi/dida/rob/2122/appunti/capitolo4.pdf>
- Adatt. da Pappalardo, *Passacantando – “Ricerca Operativa”*: dimostrazione esempio (4.3)

Versione aggiornata al 13/09/2022

Il codice L<sup>A</sup>T<sub>E</sub>X di questa trascrizione è reperibile al seguente indirizzo:

<https://github.com/mell-o-tron/RO-Capitolo4>

In seguito alle inevitabili segnalazioni di errori allo stesso indirizzo si troverà un errata corrige.