Programmazione Lineare Intera

Mauro Passacantando

Dipartimento di Informatica, Università di Pisa
mauro.passacantando@unipi.it

Corso di Ricerca Operativa A
Laurea in Informatica - Università di Pisa - a.a. 2019/20

Relazioni tra PLI e PL

Consideriamo un problema di Programmazione Lineare Intera nella forma

$$\begin{cases} \max\ c^{\mathsf{T}}x \\ Ax \leq b \\ x \in \mathbb{Z}^n \end{cases} \tag{P}$$

dove i dati A, b, c sono a componenti intere e la regione ammissibile Ω è limitata.

Teorema

Il problema (P) è NP-hard.

Relazioni tra PLI e PL

Consideriamo un problema di Programmazione Lineare Intera nella forma

$$\begin{cases} \max\ c^{\mathsf{T}}x \\ Ax \leq b \\ x \in \mathbb{Z}^n \end{cases} \tag{P}$$

dove i dati A, b, c sono a componenti intere e la regione ammissibile Ω è limitata.

Teorema
Il problema (P) è *NP*-hard.

Definizione
Il problema di Programmazione Lineare

$$\begin{cases} \max\ c^{\mathsf{T}}x \\ Ax \leq b \end{cases} \tag{RC}$$

è detto rilassamento continuo di (P).

Relazioni tra PLI e PL

Che relazione c'è tra (P) e (RC)?

Teorema

- Il valore ottimo di (RC) è maggiore o uguale del valore ottimo di (P).

Relazioni tra PLI e PL

Che relazione c'è tra (P) e (RC)?

Teorema

- Il valore ottimo di (RC) è maggiore o uguale del valore ottimo di (P).
- Se la soluzione ottima di (RC) è ammissibile per (P), allora è ottima anche per (P).

Relazioni tra PLI e PL

Che relazione c'è tra (P) e (RC)?

Teorema

- Il valore ottimo di (RC) è maggiore o uguale del valore ottimo di (P).
- Se la soluzione ottima di (RC) è ammissibile per (P), allora è ottima anche per (P).

Spesso la soluzione ottima di (RC) non è ammissibile per (P)...

Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata?

Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

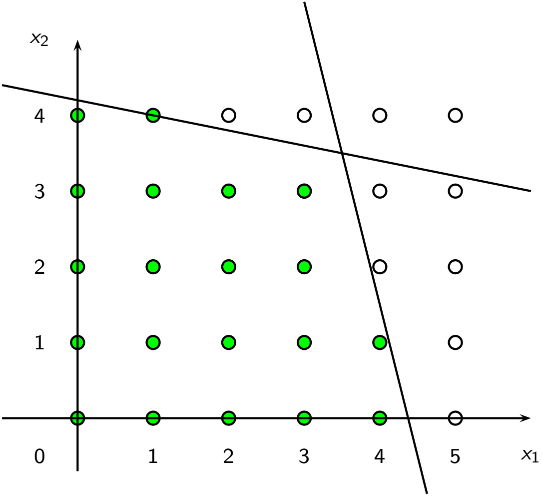
$$
\begin{cases}
\max\ x_1 + 3\,x_2 \\
x_1 + 5\,x_2 \leq 21 \\
8\,x_1 + 2\,x_2 \leq 35 \\
x \geq 0 \\
x \in \mathbb{Z}^2
\end{cases}
$$

Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

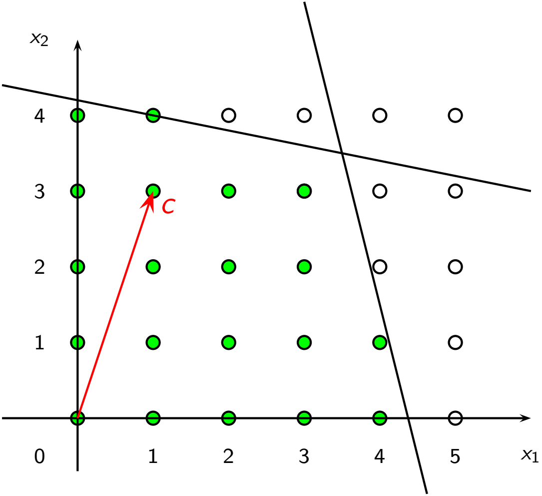


Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

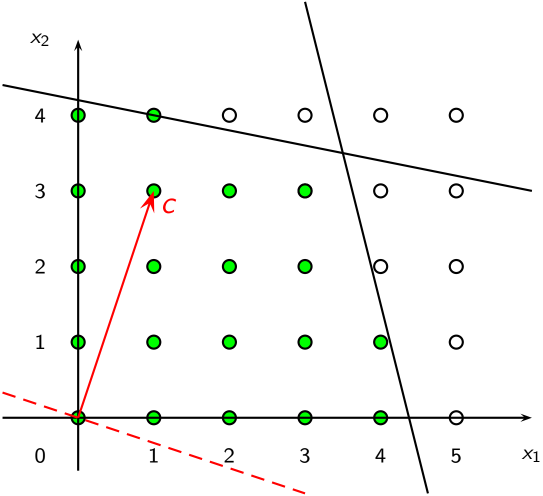


Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

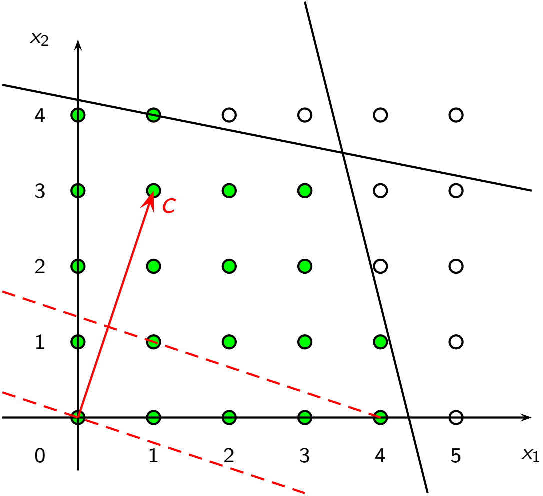


Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

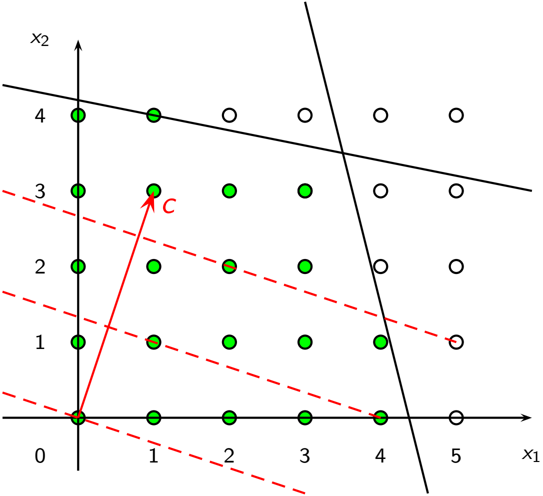


Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

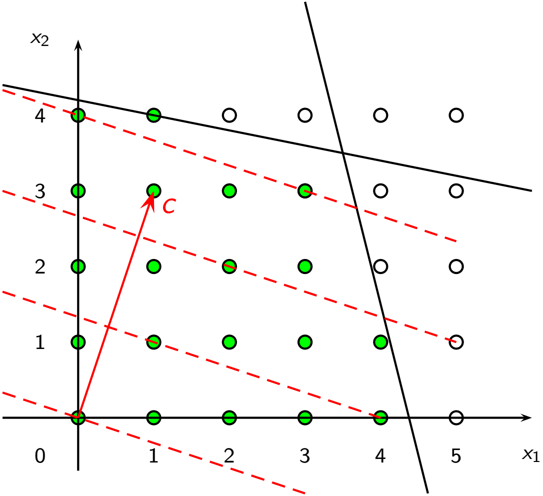


Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$



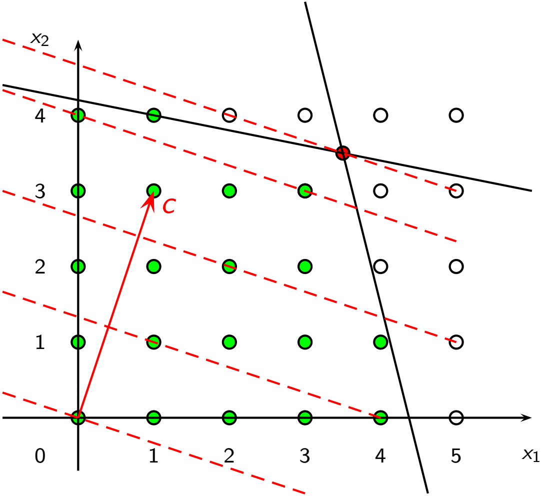
Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max \ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

$\left(\frac{7}{2}, \frac{7}{2}\right)$ ottimo rilas. continuo



Relazioni tra PLI e PL

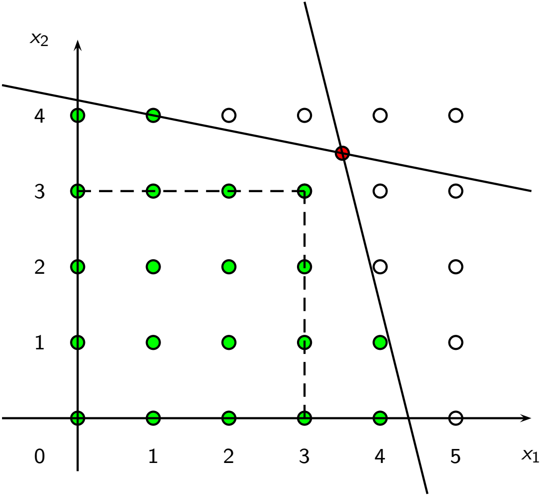
Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

$\left(\frac{7}{2}, \frac{7}{2}\right)$ ottimo rilas. continuo

arrotondamento $\rightarrow (3, 3)$



Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

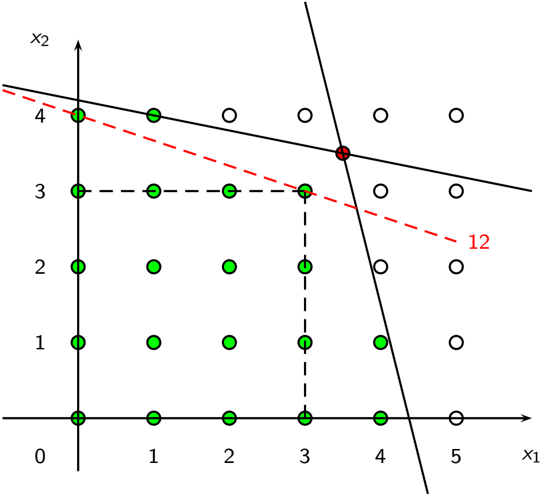
Esempio

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

$\left(\frac{7}{2}, \frac{7}{2}\right)$ ottimo rilas. continuo

arrotondamento $\rightarrow (3,3)$

$(3,3)$ non è ottima



Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e arrotondare la soluzione trovata? NO

Esempio

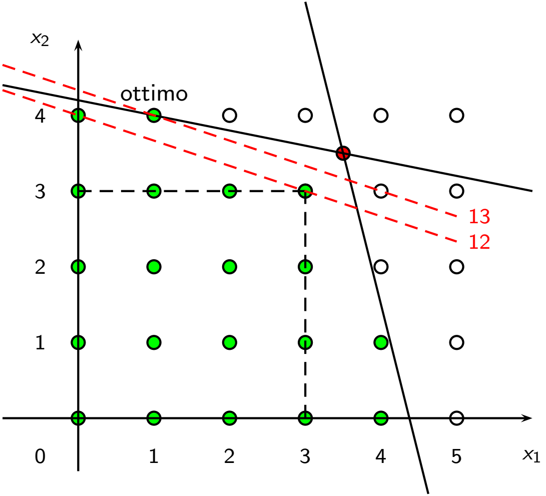
$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

$\left(\frac{7}{2}, \frac{7}{2}\right)$ ottimo rilas. continuo

arrotondamento $\rightarrow (3, 3)$

$(3, 3)$ non è ottima

la sol. ottima è $(1, 4)$



Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e scegliere la soluzione ammissibile più vicina rispetto alla distanza euclidea?

Relazioni tra PLI e PL

Per risolvere (P) è sufficiente risolvere il suo rilassamento continuo e scegliere la soluzione ammissibile più vicina rispetto alla distanza euclidea? NO

Esempio

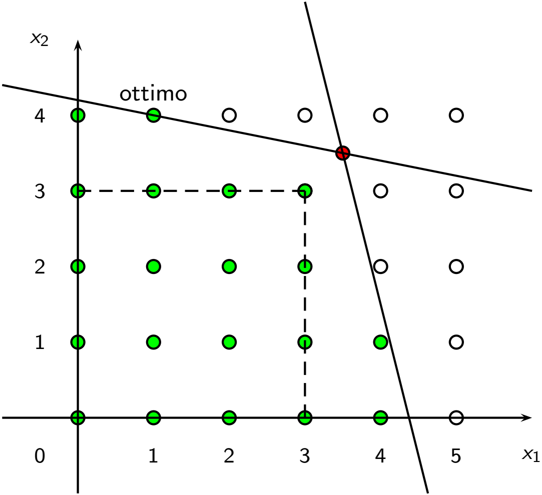
$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

$\left(\frac{7}{2}, \frac{7}{2}\right)$ ottimo ril. cont.

sol. più vicina è $(3, 3)$

ma $(3, 3)$ non è ottima

la sol. ottima è $(1, 4)$



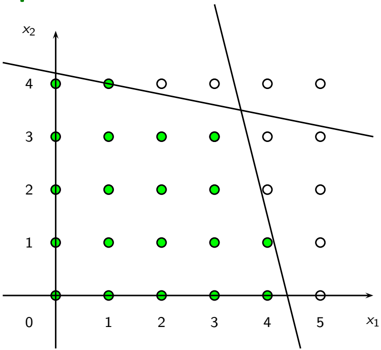
Relazioni tra PLI e PL

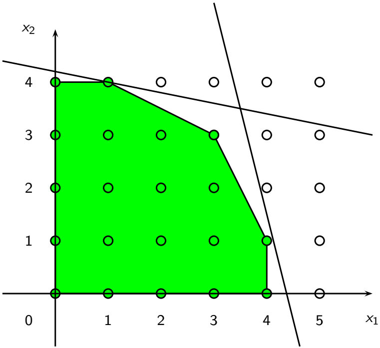
Consideriamo i problemi:

$$\begin{cases} \max c^T x \\ x \in \Omega \end{cases} \qquad \begin{cases} \max c^T x \\ x \in \operatorname{conv}(\Omega) \end{cases}$$

dove $\operatorname{conv}(\Omega)$ è l'involucro convesso delle soluzioni ammissibili, cioè il più piccolo insieme convesso che contiene Ω.

Esempio





Relazioni tra PLI e PL

Teorema

- conv(Ω) è un poliedro

Relazioni tra PLI e PL

Teorema

- $\text{conv}(\Omega)$ è un poliedro
- I vertici di $\text{conv}(\Omega)$ appartengono a Ω, cioè sono a componenti intere

Relazioni tra PLI e PL

Teorema

- $\text{conv}(\Omega)$ è un poliedro
- I vertici di $\text{conv}(\Omega)$ appartengono a Ω, cioè sono a componenti intere
- I problemi

$$\begin{cases} \max c^{\mathsf{T}} x \\ x \in \Omega \end{cases} \qquad\qquad \begin{cases} \max c^{\mathsf{T}} x \\ x \in \text{conv}(\Omega) \end{cases}$$

hanno lo stesso valore ottimo e almeno una soluzione ottima comune

Relazioni tra PLI e PL

Teorema

- conv(Ω) è un poliedro
- I vertici di conv(Ω) appartengono a Ω, cioè sono a componenti intere
- I problemi

$$\begin{cases} \max c^{\mathsf{T}} x \\ x \in \Omega \end{cases} \qquad \begin{cases} \max c^{\mathsf{T}} x \\ x \in \text{conv}(\Omega) \end{cases}$$

  hanno lo stesso valore ottimo e almeno una soluzione ottima comune

Corollario

Il problema di PLI

$$\max_{x \in \Omega} c^{\mathsf{T}} x$$

è equivalente al problema di PL

$$\max_{x \in \text{conv}(\Omega)} c^{\mathsf{T}} x$$

Relazioni tra PLI e PL

Teorema

- $\text{conv}(\Omega)$ è un poliedro
- I vertici di $\text{conv}(\Omega)$ appartengono a Ω, cioè sono a componenti intere
- I problemi

$$\begin{cases} \max c^{\mathsf{T}} x \\ x \in \Omega \end{cases} \qquad \begin{cases} \max c^{\mathsf{T}} x \\ x \in \text{conv}(\Omega) \end{cases}$$

  hanno lo stesso valore ottimo e almeno una soluzione ottima comune

Corollario

Il problema di PLI

$$\max_{x \in \Omega} c^{\mathsf{T}} x$$

è equivalente al problema di PL

$$\max_{x \in \text{conv}(\Omega)} c^{\mathsf{T}} x$$

In generale è difficile trovare i vincoli che definiscono $\text{conv}(\Omega)$ ...

Matrici totalmente unimodulari

Per alcuni particolari problemi si riesce a caratterizzare conv(Ω).

Teorema

Sia $\Omega = \{x \in \mathbb{Z}^n : \; Ax \leq b\}$. Se A è una matrice totalmente unimodulare (cioè il determinante di ogni sua sottomatrice quadrata è 0 oppure 1 oppure -1), allora

$$\text{conv}(\Omega) = \{x \in \mathbb{R}^n : \; Ax \leq b\},$$

cioè conv(Ω) coincide con la regione ammissibile del rilassamento continuo.

Matrici totalmente unimodulari

Per alcuni particolari problemi si riesce a caratterizzare conv(Ω).

Teorema

Sia $\Omega = \{x \in \mathbb{Z}^n : \ Ax \leq b\}$. Se A è una matrice totalmente unimodulare (cioè il determinante di ogni sua sottomatrice quadrata è 0 oppure 1 oppure -1), allora

$$\text{conv}(\Omega) = \{x \in \mathbb{R}^n : \ Ax \leq b\},$$

cioè conv(Ω) coincide con la regione ammissibile del rilassamento continuo.

Esempi

In un problema di flusso di costo minimo con variabili intere:

$$\begin{cases} \min \ c^{\mathsf{T}} x \\ E\,x = b \\ 0 \leq x \leq u \\ x \in \mathbb{Z}^n \end{cases}$$

la matrice dei vincoli è totalmente unimodulare. Quindi per risolverlo basta trovare un vertice ottimo del suo rilassamento continuo.

Lo stesso vale per il problema del cammino minimo.

Disuguaglianze valide e piani di taglio

In generale è difficile caratterizzare conv(Ω).
Si aggiungono vincoli alla regione ammissibile del rilassamento continuo in modo da approssimare conv(Ω).

Definizioni

La disequazione $p^{\mathsf{T}}x \le p_0$ è detta disuguaglianza valida (DV) per l'insieme Ω se

$$p^{\mathsf{T}}x \le p_0 \qquad \forall\, x \in \Omega.$$

Un piano di taglio è una disuguaglianza valida $p^{\mathsf{T}}x \le p_0$ per Ω tale che $p^{\mathsf{T}}\bar{x} > p_0$, dove $\bar{x}$ è l'ottimo del rilassamento continuo.

Disuguaglianze valide e piani di taglio

In generale è difficile caratterizzare conv(Ω).
Si aggiungono vincoli alla regione ammissibile del rilassamento continuo in modo da approssimare conv(Ω).

Definizioni

La disequazione $p^{\mathsf{T}}x \leq p_0$ è detta disuguaglianza valida (DV) per l'insieme Ω se

$$p^{\mathsf{T}}x \leq p_0 \qquad \forall\ x \in \Omega.$$

Un piano di taglio è una disuguaglianza valida $p^{\mathsf{T}}x \leq p_0$ per Ω tale che $p^{\mathsf{T}}\bar{x} > p_0$, dove $\bar{x}$ è l'ottimo del rilassamento continuo.

Idea alla base del metodo dei piani di taglio:
se P è la regione ammissibile del rilassamento continuo e la soluzione ottima $\bar{x}$ di $\max\limits_{x \in P} c^{\mathsf{T}}x$ appartiene ad Ω, allora $\bar{x}$ è ottima anche per $\max\limits_{x \in \Omega} c^{\mathsf{T}}x$;
altrimenti si costruisce un piano di taglio $p^{\mathsf{T}}x \leq p_0$ in modo da tagliare fuori $\bar{x}$ e poi risolvere il nuovo problema di PL:

$$\begin{cases} \max\ c^{\mathsf{T}}x \\ x \in P \\ p^{\mathsf{T}}x \leq p_0 \end{cases}$$

Piani di taglio di Gomory

Supponiamo che il problema di PLI sia nella forma

$$\begin{cases} \max\ c^{\mathsf{T}} x \\ A x = b \\ x \geq 0 \\ x \in \mathbb{Z}^n \end{cases} \qquad \text{(P)}$$

e che $\bar{x}$ sia una soluzione di base (relativa alla base B) ottima del rilassamento continuo di (P). Poniamo:

$$A = (A_B\ A_N) \qquad x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} \qquad \widetilde{A} = A_B^{-1} A_N \qquad \widetilde{b} = \bar{x}_B.$$

La parte frazionaria di un numero reale z è $\{z\} := z - \lfloor z \rfloor$, dove $\lfloor z \rfloor$ è la parte intera inferiore di z (o arrotondamento per difetto all'intero più vicino).
Esempio: $\{2.3\} = 2.3 - 2 = 0.3, \qquad \{-1.4\} = -1.4 - (-2) = 0.6.$

Piani di taglio di Gomory

Supponiamo che il problema di PLI sia nella forma

$$\begin{cases} \max\ c^{\mathsf{T}}x \\ A\,x = b \\ x \geq 0 \\ x \in \mathbb{Z}^n \end{cases} \tag{P}$$

e che $\bar{x}$ sia una soluzione di base (relativa alla base B) ottima del rilassamento continuo di (P). Poniamo:

$$A = (A_B\ A_N) \qquad x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} \qquad \widetilde{A} = A_B^{-1}A_N \qquad \widetilde{b} = \bar{x}_B.$$

La parte frazionaria di un numero reale z è $\{z\} := z - \lfloor z \rfloor$, dove $\lfloor z \rfloor$ è la parte intera inferiore di z (o arrotondamento per difetto all'intero più vicino).
Esempio: $\{2.3\} = 2.3 - 2 = 0.3, \qquad \{-1.4\} = -1.4 - (-2) = 0.6.$

Teorema

Se esiste un indice $r \in B$ tale che $\widetilde{b}_r \notin \mathbb{Z}$, allora

$$\sum_{j \in N} \{\widetilde{a}_{rj}\} x_j \geq \{\widetilde{b}_r\}$$

è un piano di taglio (detto di Gomory) per il problema (P).

Piani di taglio di Gomory

Dimostrazione

Fissiamo $x \in \Omega$. Allora

$$A\,x = A_B\,x_B + A_N\,x_N = b,$$

quindi

$$x_B = A_B^{-1}b - A_B^{-1}\,A_N\,x_N = \widetilde{b} - \widetilde{A}\,x_N.$$

Definiamo il vettore $\widetilde{x} = x_B$. Quindi:

$$\begin{aligned}
\widetilde{x}_r &= \widetilde{b}_r - \sum_{j \in N} \widetilde{a}_{rj} x_j \\
&= \lfloor \widetilde{b}_r \rfloor + \{\widetilde{b}_r\} - \sum_{j \in N} \left(\lfloor \widetilde{a}_{rj} \rfloor + \{\widetilde{a}_{rj}\} \right) x_j \\
&= \lfloor \widetilde{b}_r \rfloor + \{\widetilde{b}_r\} - \sum_{j \in N} \lfloor \widetilde{a}_{rj} \rfloor x_j - \sum_{j \in N} \{\widetilde{a}_{rj}\} x_j,
\end{aligned}$$

di conseguenza

Piani di taglio di Gomory

Dimostrazione (segue)

$$\sum_{j \in N} \{\widetilde{a}_{rj}\} x_j - \{\widetilde{b}_r\} = \lfloor \widetilde{b}_r \rfloor - \sum_{j \in N} \lfloor \widetilde{a}_{rj} \rfloor x_j - \widetilde{x}_r \in \mathbb{Z}. \tag{1}$$

Inoltre si ha:

$$\sum_{j \in N} \{\widetilde{a}_{rj}\} x_j - \{\widetilde{b}_r\} \geq -\{\widetilde{b}_r\} > -1. \tag{2}$$

Dalle relazioni (1) e (2), otteniamo che:

$$\sum_{j \in N} \{\widetilde{a}_{rj}\} x_j \geq \{\widetilde{b}_r\}. \tag{3}$$

quindi la (3) è una DV per Ω. Inoltre la (3) non è soddisfatta da $\bar{x}$, infatti:

$$0 = \sum_{j \in N} \{\widetilde{a}_{rj}\} \bar{x}_j < \{\widetilde{b}_r\},$$

quindi la (3) è un piano di taglio. □

Piani di taglio di Gomory

Esempio

Consideriamo di nuovo il problema

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbb{Z}^2 \end{cases}$$

Trasformiamo i vincoli di $\leq$ in vincoli di $=$ aggiungendo variabili di scarto x_3, x_4:

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 + x_3 = 21 \\ 8\,x_1 + 2\,x_2 + x_4 = 35 \\ x \geq 0 \\ x \in \mathbb{Z}^4 \end{cases}$$

$$A = \begin{pmatrix} 1 & 5 & 1 & 0 \\ 8 & 2 & 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 21 \\ 35 \end{pmatrix}, \qquad c^{\mathsf{T}} = (1, 3, 0, 0).$$

Piani di taglio di Gomory

Esempio

La soluzione ottima del rilassamento continuo è $\bar{x} = \left(\frac{7}{2}, \frac{7}{2}, 0, 0\right)$.

Quindi la base ottima è $B = \{1, 2\}$,

$$A_B = \begin{pmatrix} 1 & 5 \\ 8 & 2 \end{pmatrix} \qquad A_B^{-1} = \begin{pmatrix} -\frac{1}{19} & \frac{5}{38} \\ \frac{4}{19} & -\frac{1}{38} \end{pmatrix}, \qquad \widetilde{A} = \begin{pmatrix} -\frac{1}{19} & \frac{5}{38} \\ \frac{4}{19} & -\frac{1}{38} \end{pmatrix}.$$

Poiché $\widetilde{b} = \left(\frac{7}{2}, \frac{7}{2}\right)$ ha entrambe le componenti non intere, esistono due tagli di Gomory.

Piani di taglio di Gomory

Esempio

Se $r = 1$, allora il piano di taglio di Gomory è

$$\left\{-\frac{1}{19}\right\} x_3 + \left\{\frac{5}{38}\right\} x_4 \geq \left\{\frac{7}{2}\right\},$$

cioè

$$\frac{18}{19} x_3 + \frac{5}{38} x_4 \geq \frac{1}{2},$$

ossia

$$36 x_3 + 5 x_4 \geq 19,$$

che nelle variabili (x_1, x_2) equivale a

$$36(21 - x_1 - 5x_2) + 5(35 - 8x_1 - 2x_2) \geq 19$$

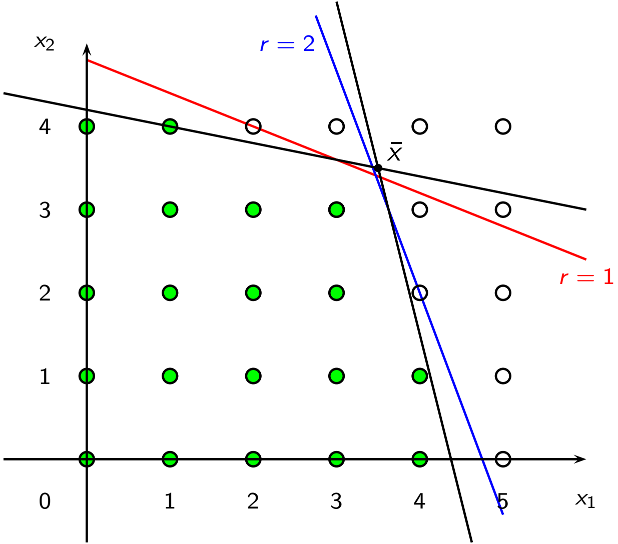
cioè

$$2x_1 + 5x_2 \leq 24.$$

Analogamente, per $r = 2$ si ottiene il piano di taglio $8x_1 + 3x_2 \leq 38$.

Piani di taglio di Gomory

Esempio



Esercizio 1

Si consideri il seguente problema di programmazione lineare intera:

$$\begin{cases} \max\ 12\ x_1 + 5\ x_2 \\ 8\ x_1 + 15\ x_2 \leq 45 \\ 15\ x_1 + 11\ x_2 \leq 63 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z} \end{cases}$$

a) Disegnare l'insieme conv(Ω), dove Ω è la regione ammissibile del problema.

b) Risolvere graficamente il rilassamento continuo del problema.

c) Calcolare un taglio di Gomory.

Esercizio 2

Si consideri il seguente problema di programmazione lineare intera:

$$\begin{cases} \min\ 6\,x_1 + 13\,x_2 \\ 17\,x_1 + 7\,x_2 \geq 63 \\ 7\,x_1 + 6\,x_2 \geq 30 \\ x_1, x_2 \geq 0 \\ x_1, x_2 \in \mathbb{Z} \end{cases}$$

a) Disegnare l'insieme conv(Ω), dove Ω è la regione ammissibile del problema.

b) Risolvere graficamente il rilassamento continuo del problema.

c) Calcolare un taglio di Gomory.

Enumerazione esplicita

Consideriamo il problema di PLI

$$\begin{cases} \max\ 5\,x_1 + 6\,x_2 \\ 3\,x_1 + 4\,x_2 \leq 7 \\ x1 \geq 0,\ x_2 \geq 0 \\ x \in \mathbb{Z}^2 \end{cases} \qquad (P)$$

Enumerazione esplicita

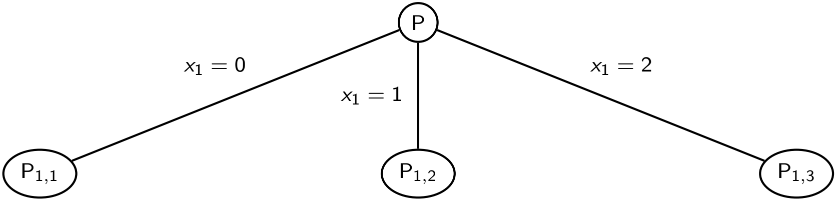
Consideriamo il problema di PLI

$$\begin{cases} \max\ 5\,x_1 + 6\,x_2 \\ 3\,x_1 + 4\,x_2 \leq 7 \\ x1 \geq 0,\ \ x_2 \geq 0 \\ x \in \mathbb{Z}^2 \end{cases} \qquad \text{(P)}$$

I vincoli implicano che $x_1 = 0$ oppure 1 oppure 2.
Facciamo una partizione della regione ammissibile Ω in tre sottoinsiemi:

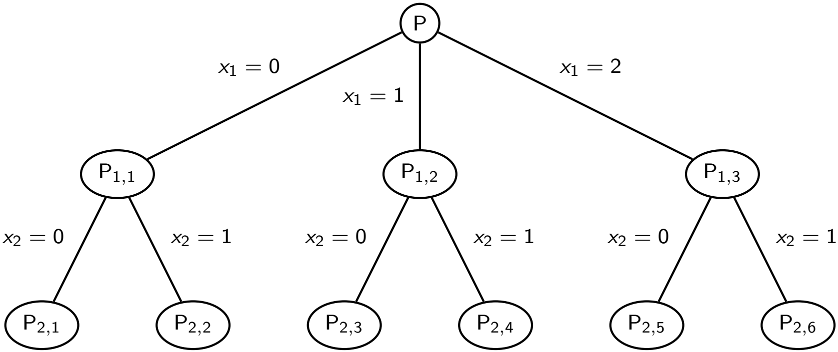
$$\Omega = (\Omega \cap \{x_1 = 0\}) \cup (\Omega \cap \{x_1 = 1\}) \cup (\Omega \cap \{x_1 = 2\})$$

che corrispondono al primo livello dell'albero decisionale:



Enumerazione esplicita

Analogamente, $x_2 = 0$ oppure 1. Pertanto, l'albero decisionale completo è



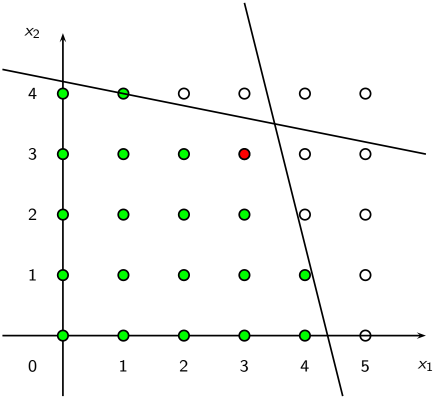
I nodi $P_{2,1}, \ldots, P_{2,5}$ corrispondono a soluzioni ammissibili di $(\mathcal{P})$, mentre il nodo $P_{2,6}$ corrisponde alla soluzione $x = (2, 1)$ che non è ammissibile.
I valori della funzione obiettivo in corrispondenza dei nodi $P_{2,1}, \ldots, P_{2,5}$ sono rispettivamente 0, 6, 5, 11, 10. Pertanto, la soluzione ottima è ottenuta nel nodo $P_{2,4}$, cioè $x^* = (1, 1)$.

Enumerazione implicita

- L'enumerazione esplicita è molto costosa dal punto di vista computazionale
- Possiamo però esaminare (e scartare) le soluzioni a gruppi, anziché singolarmente $\longrightarrow$ enumerazione implicita

Enumerazione implicita

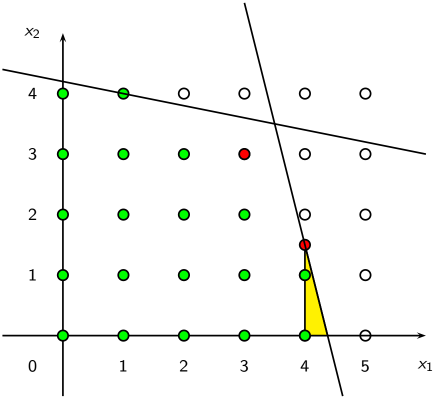
$$\begin{cases} \max \ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0, \ x \in \mathbb{Z}^2 \end{cases}$$



- Sappiamo che il punto $(3, 3)$ è ammissibile con valore 12.

Enumerazione implicita

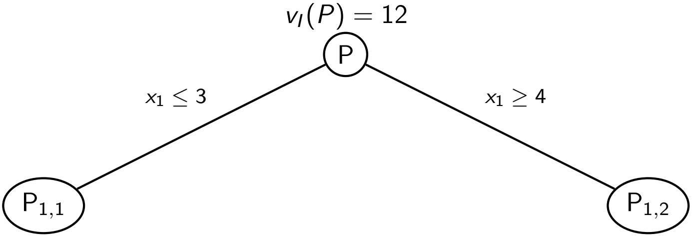
$$\begin{cases} \max \ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0, \ x \in \mathbb{Z}^2 \end{cases}$$



- Sappiamo che il punto $(3,3)$ è ammissibile con valore 12.
- Consideriamo il sottoproblema in cui aggiungiamo il vincolo $x_1 \geq 4$: la soluzione ottima del suo rilassamento continuo è $(4,3/2)$ con valore 8.5, quindi tutte le soluzioni ammissibili del sottoproblema sono peggiori di $(3,3)$ $\rightarrow$ enumerazione implicita.

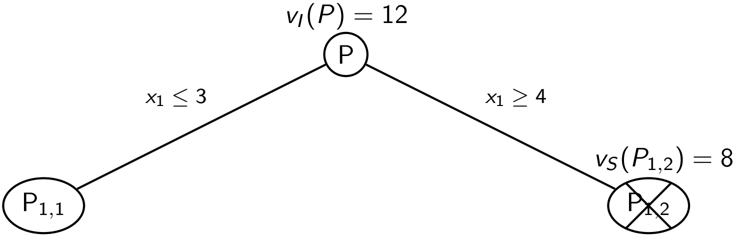
Albero decisionale ridotto

Indichiamo con
$v_I(P)$ una valutazione inferiore (stima per difetto) del valore ottimo di P
$v_S(P)$ una valutazione superiore (stima per eccesso) del valore ottimo di P



Albero decisionale ridotto

Indichiamo con
$v_I(P)$ una valutazione inferiore (stima per difetto) del valore ottimo di P
$v_S(P)$ una valutazione superiore (stima per eccesso) del valore ottimo di P



Metodo Branch and Bound

Componenti principali

Metodo Branch and Bound

Componenti principali

- *Branch (ramificazione)*: partizionare la regione ammissibile in sotto-regioni

Metodo Branch and Bound

Componenti principali

- *Branch (ramificazione)*: partizionare la regione ammissibile in sotto-regioni
- *Bound (valutazione)*: stimare il valore ottimo di ogni sotto-problema

Metodo Branch and Bound

Componenti principali

- *Branch (ramificazione)*: partizionare la regione ammissibile in sotto-regioni
- *Bound (valutazione)*: stimare il valore ottimo di ogni sotto-problema
- *Potatura*: scartare le sotto-regioni che non contengono soluzioni migliori di quella corrente (o chiudere i corrispondenti nodi nell'albero decisionale)

Metodo Branch and Bound

Componenti principali

- *Branch (ramificazione)*: partizionare la regione ammissibile in sotto-regioni
- *Bound (valutazione)*: stimare il valore ottimo di ogni sotto-problema
- *Potatura*: scartare le sotto-regioni che non contengono soluzioni migliori di quella corrente (o chiudere i corrispondenti nodi nell'albero decisionale)
- *Visita*: in quale ordine visitare i nodi nell'albero decisionale

Regole di Branch

Come partizionare la regione ammissibile?

- Le sotto-regioni si generano aggiungendo vincoli

Regole di Branch

Come partizionare la regione ammissibile?

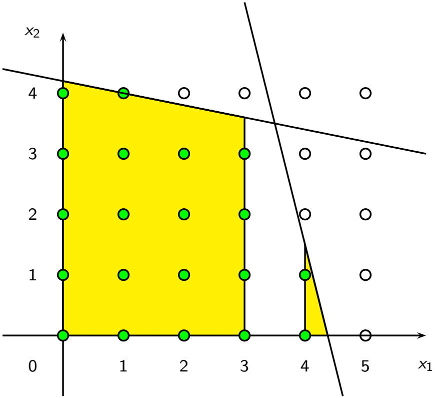
- Le sotto-regioni si generano aggiungendo vincoli
- Il risultato deve essere una partizione della regione ammissibile Ω, per non perdere nessuna soluzione ammissibile (che potrebbe essere ottima)

Branch binario

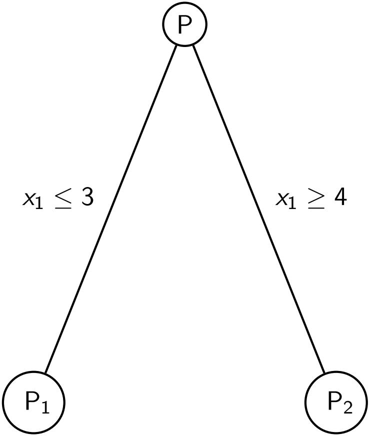
Branch binario

Ogni regione viene partizionata in due sotto-regioni, aggiungendo due vincoli

Esempio: $x_1 \leq 3$ e $x_1 \geq 4$

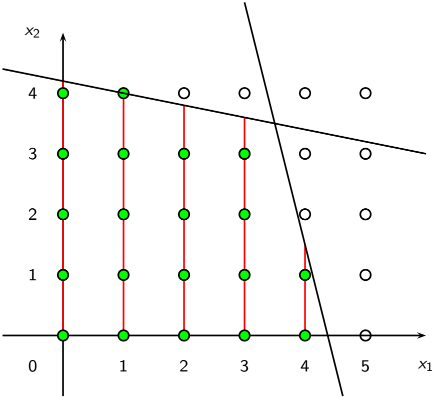


Branch binario

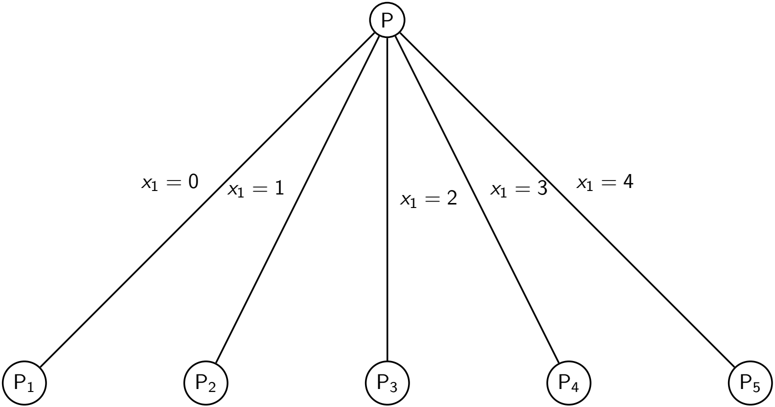


Branch non binario

Ogni regione si partiziona in k sotto-regioni, aggiungendo k vincoli.
Esempio: $x_1 = 0$, $x_1 = 1$, $x_1 = 2$, $x_1 = 3$ e $x_1 = 4$



Branch non binario



Bound

Una valutazione inferiore del valore ottimo di P è data dal valore della funzione obiettivo in una qualunque soluzione ammissibile.

Una valutazione superiore del valore ottimo di ogni sotto-problema è data dal valore ottimo di un suo *rilassamento*:

- Rilassamento continuo (eliminare i vincoli di interezza sulle variabili)

  $x \in \{0,1\} \Rightarrow 0 \leq x \leq 1$

  $x \in \mathbb{Z}_+ \Rightarrow x \geq 0$
- Eliminazione di vincoli
- Somma di due o più vincoli
- ...

Criteri di potatura dell'albero decisionale

Un nodo P_i dell'albero decisionale può essere chiuso se vale una qualunque delle seguenti condizioni:

- il sotto-problema P_i non ha soluzioni ammissibili

Criteri di potatura dell'albero decisionale

Un nodo P_i dell'albero decisionale può essere chiuso se vale una qualunque delle seguenti condizioni:

- il sotto-problema P_i non ha soluzioni ammissibili
- $v_S(P_i) \leq v_I(P)$, cioè le soluzioni ammissibili di P_i non sono migliori della soluzione ammissibile corrente

Criteri di potatura dell'albero decisionale

Un nodo P_i dell'albero decisionale può essere chiuso se vale una qualunque delle seguenti condizioni:

- il sotto-problema P_i non ha soluzioni ammissibili
- $v_S(P_i) \leq v_I(P)$, cioè le soluzioni ammissibili di P_i non sono migliori della soluzione ammissibile corrente
- $v_S(P_i) > v_I(P)$ e la soluzione ottima di P_i è ammissibile per P. In tal caso si aggiorna $v_I(P) = v_S(P_i)$.

Visita dell'albero decisionale

In quale ordine visitiamo i nodi dell'albero?

Depth first (in profondità)

- Il prossimo nodo da visitare è uno dei figli del nodo attualmente visitato (se rimasto aperto)
- questa strategia trova rapidamente una soluzione ammissibile, occupa poca memoria, ma non tiene conto della qualità della soluzione trovata

Visita dell'albero decisionale

In quale ordine visitiamo i nodi dell'albero?

Depth first (in profondità)

- Il prossimo nodo da visitare è uno dei figli del nodo attualmente visitato (se rimasto aperto)
- questa strategia trova rapidamente una soluzione ammissibile, occupa poca memoria, ma non tiene conto della qualità della soluzione trovata

Best first

- Il prossimo nodo da visitare è il più promettente, cioè quello con il massimo valore di v_S
- trova rapidamente una soluzione ammissibile, occupa molto memoria, ma tiene conto della qualità della soluzione trovata

Visita dell'albero decisionale

In quale ordine visitiamo i nodi dell'albero?

Depth first (in profondità)

- Il prossimo nodo da visitare è uno dei figli del nodo attualmente visitato (se rimasto aperto)
- questa strategia trova rapidamente una soluzione ammissibile, occupa poca memoria, ma non tiene conto della qualità della soluzione trovata

Best first

- Il prossimo nodo da visitare è il più promettente, cioè quello con il massimo valore di v_S
- trova rapidamente una soluzione ammissibile, occupa molto memoria, ma tiene conto della qualità della soluzione trovata

Breadth first (in ampiezza)

- Si esplorano prima tutti i nodi dello stesso livello
- in generale non fornisce buone prestazioni dal punto di vista computazionale

Metodo Branch and bound

1. Genera il nodo radice P, trova una soluzione ammissibile per P e calcola una $v_I(P)$.

Metodo Branch and bound

1. Genera il nodo radice P, trova una soluzione ammissibile per P e calcola una $v_I(P)$.
2. Se tutti i nodi sono stati visitati, allora STOP (la soluzione ammissibile corrente è ottima).

Metodo Branch and bound

1. Genera il nodo radice P, trova una soluzione ammissibile per P e calcola una $v_I(P)$.
2. Se tutti i nodi sono stati visitati, allora STOP (la soluzione ammissibile corrente è ottima).
3. Seleziona un nodo P_i da visitare.

Metodo Branch and bound

1. Genera il nodo radice P, trova una soluzione ammissibile per P e calcola una $v_I(P)$.
2. Se tutti i nodi sono stati visitati, allora STOP (la soluzione ammissibile corrente è ottima).
3. Seleziona un nodo P_i da visitare.
4. Se P_i non contiene soluzioni ammissibili, allora chiudi il nodo P_i e torna al passo 2.

Metodo Branch and bound

1. Genera il nodo radice P, trova una soluzione ammissibile per P e calcola una $v_I(P)$.
2. Se tutti i nodi sono stati visitati, allora STOP (la soluzione ammissibile corrente è ottima).
3. Seleziona un nodo P$_i$ da visitare.
4. Se P$_i$ non contiene soluzioni ammissibili, allora chiudi il nodo P$_i$ e torna al passo 2.
5. (Bound) risolvi un rilassamento di P$_i$ e calcola $v_S(P_i)$.

Metodo Branch and bound

1. Genera il nodo radice P, trova una soluzione ammissibile per P e calcola una $v_I(P)$.
2. Se tutti i nodi sono stati visitati, allora STOP (la soluzione ammissibile corrente è ottima).
3. Seleziona un nodo P_i da visitare.
4. Se P_i non contiene soluzioni ammissibili, allora chiudi il nodo P_i e torna al passo 2.
5. (Bound) risolvi un rilassamento di P_i e calcola $v_S(P_i)$.
   - se $v_S(P_i) \leq v_I(P)$, allora chiudi il nodo P_i e torna al passo 2.
   - se $v_S(P_i) > v_I(P)$ e la soluzione ottima del rilassamento di P_i è ammissibile per P, allora chiudi il nodo P_i, aggiorna la soluzione ammissibile, poni $v_I(P) = v_S(P_i)$ e torna al passo 2.
6. (Branch) Partiziona la regione ammissibile di P_i in sotto-regioni e genera nuovi nodi da visitare. Torna al passo 2.

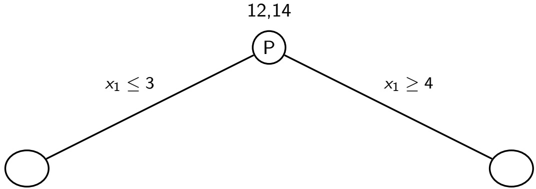
Esempio

Applichiamo il metodo Branch and Bound per risolvere il problema di PLI

$$\begin{cases} \max\ x_1 + 3\,x_2 \\ x_1 + 5\,x_2 \leq 21 \\ 8\,x_1 + 2\,x_2 \leq 35 \\ x \geq 0,\ x \in \mathbb{Z}^2 \end{cases} \qquad (P)$$

In ogni nodo dell'albero decisionale risolviamo il rilassamento continuo, usiamo un *branch* binario basato sulla prima componente non intera della soluzione ottima del rilassamento. Visitiamo l'albero decisionale in profondità.

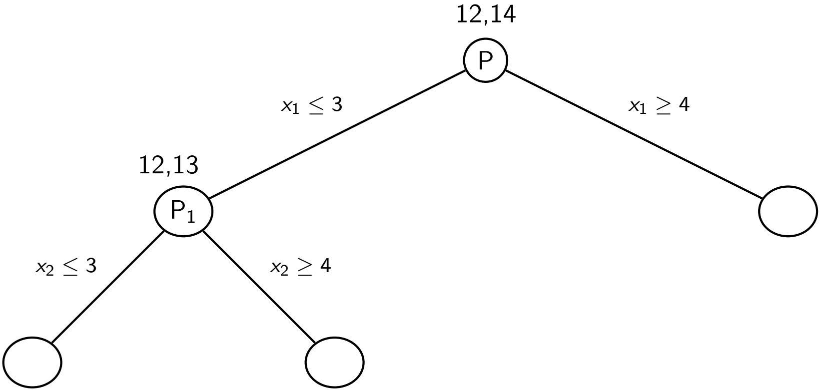
Sappiamo che $(3,3)$ è una soluzione ammissibile, quindi $v_I(P) = 12$.
Sappiamo che la soluzione ottima del rilassamento continuo è $(7/2, 7/2)$, quindi $v_S(P) = 14$. Il nodo P rimane aperto. Branch: $x_1 \leq \lfloor 7/2 \rfloor = 3$ oppure $x_1 \geq \lceil 7/2 \rceil = 4$.



Esempio

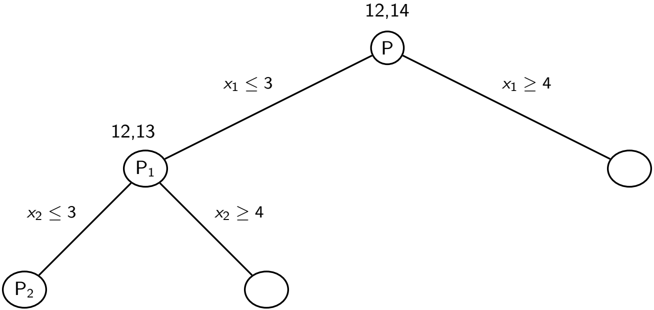
La soluzione ottima del rilassamento continuo di P_1 è $(3, 18/5)$, non è ammissibile ed ha valore 13.8, quindi $v_S(P_1) = 13 > 12 = v_I(P)$. Pertanto il nodo P_1 rimane aperto.

Dal nodo P_1 facciamo *branch* binario sulla variabile x_2: $x_2 \leq 3$ oppure $x_2 \geq 4$.



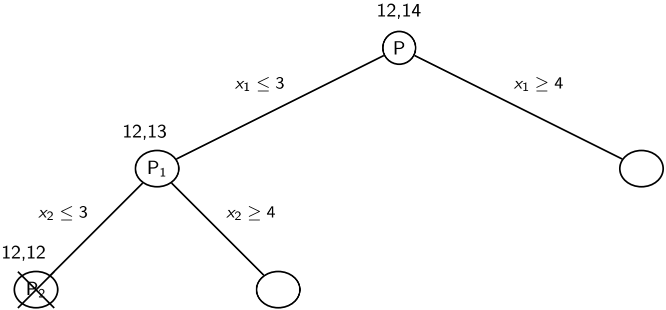
Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3,3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.



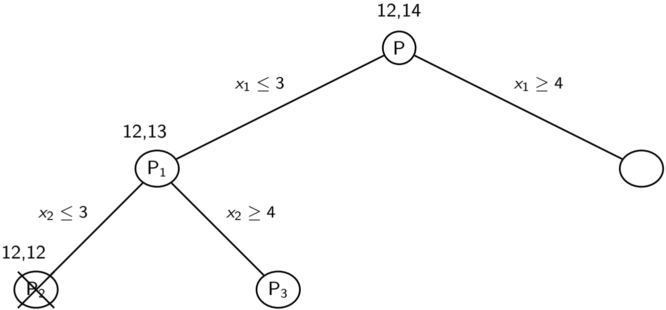
Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3,3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.



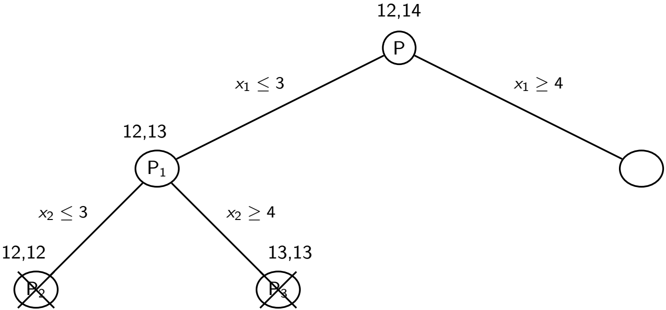
Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3, 3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.
Visitiamo ora l'altro nodo figlio di P_1. La soluzione ottima del rilassamento continuo di P_3 è $(1, 4)$ e $v_S(P_3) = 13 > 12 = v_I(P)$. Poiché $(1, 4)$ è ammissibile per P, aggiorniamo $v_I(P) = 13$ e chiudiamo il nodo P_3.



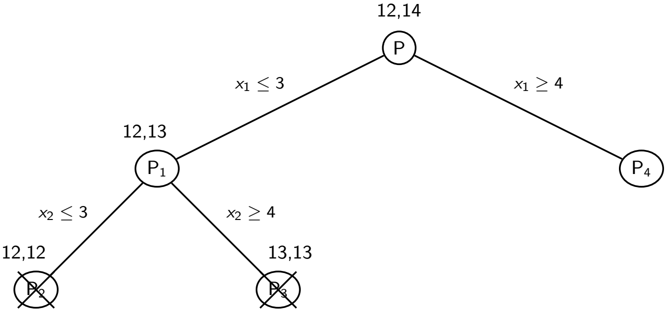
Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3, 3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.
Visitiamo ora l'altro nodo figlio di P_1. La soluzione ottima del rilassamento continuo di P_3 è $(1, 4)$ e $v_S(P_3) = 13 > 12 = v_I(P)$. Poiché $(1, 4)$ è ammissibile per P, aggiorniamo $v_I(P) = 13$ e chiudiamo il nodo P_3.



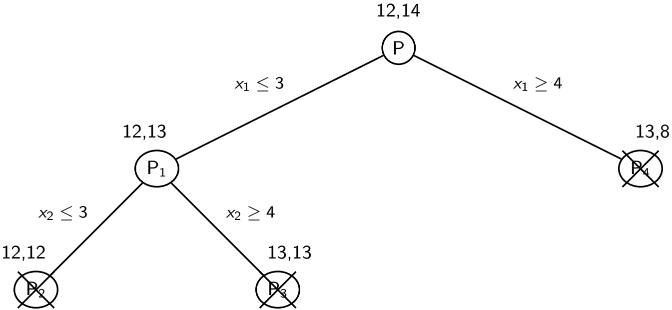
Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3,3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.
Visitiamo ora l'altro nodo figlio di P_1. La soluzione ottima del rilassamento continuo di P_3 è $(1,4)$ e $v_S(P_3) = 13 > 12 = v_I(P)$. Poiché $(1,4)$ è ammissibile per P, aggiorniamo $v_I(P) = 13$ e chiudiamo il nodo P_3.
È rimasto da visitare l'altro figlio del nodo P. La soluzione ottima del rilassamento continuo di P_4 è $(4, 3/2)$ con valore 8.5, quindi $v_S(P_4) = 8 < 13 = v_I(P)$. Pertanto, chiudiamo il nodo P_4.



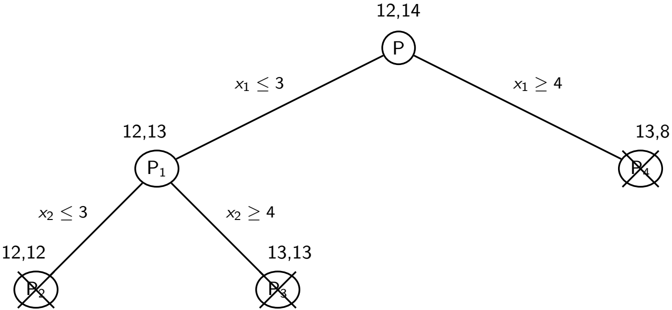
Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3,3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.
Visitiamo ora l'altro nodo figlio di P_1. La soluzione ottima del rilassamento continuo di P_3 è $(1,4)$ e $v_S(P_3) = 13 > 12 = v_I(P)$. Poiché $(1,4)$ è ammissibile per P, aggiorniamo $v_I(P) = 13$ e chiudiamo il nodo P_3.
È rimasto da visitare l'altro figlio del nodo P. La soluzione ottima del rilassamento continuo di P_4 è $(4, 3/2)$ con valore 8.5, quindi $v_S(P_4) = 8 < 13 = v_I(P)$. Pertanto, chiudiamo il nodo P_4.



Esempio

La soluzione ottima del rilassamento continuo di P_2 è $(3,3)$, quindi $v_S(P_2) = 12 = v_I(P)$, quindi chiudiamo il nodo P_2.
Visitiamo ora l'altro nodo figlio di P_1. La soluzione ottima del rilassamento continuo di P_3 è $(1,4)$ e $v_S(P_3) = 13 > 12 = v_I(P)$. Poiché $(1,4)$ è ammissibile per P, aggiorniamo $v_I(P) = 13$ e chiudiamo il nodo P_3.
È rimasto da visitare l'altro figlio del nodo P. La soluzione ottima del rilassamento continuo di P_4 è $(4, 3/2)$ con valore 8.5, quindi $v_S(P_4) = 8 < 13 = v_I(P)$. Pertanto, chiudiamo il nodo P_4.



Poiché tutti i nodi dell'albero sono stati visitati, la soluzione ottima di P è $(1,4)$ ed il valore ottimo è 13.

Problema dello zaino

Dati n oggetti di valore $v_1, \ldots, v_n$ e peso $p_1, \ldots, p_n$, ed un contenitore di capacità C, quali oggetti inserisco nel contenitore, rispettando la sua capacità, in modo da massimizzare il valore totale degli oggetti inseriti?

Variabili: $$x_i = \begin{cases} 1 & \text{se oggetto } i \text{ viene inserito,} \\ 0 & \text{altrimenti.} \end{cases}$$

Modello di PLI:

$$\begin{cases} \max \ \sum_{i=1}^{n} v_i \, x_i \\ \sum_{i=1}^{n} p_i \, x_i \leq C \\ x_i \in \{0, 1\}, \qquad i = 1, \ldots, n \end{cases}$$

Problema dello zaino: rilassamento continuo

Il rilassamento continuo è

$$\begin{cases} \max \sum_{i=1}^{n} v_i\, x_i \\ \sum_{i=1}^{n} p_i\, x_i \leq C \\ 0 \leq x_i \leq 1 \qquad i = 1, \dots, n \end{cases}$$

Le condizioni degli scarti complementari garantiscono che la soluzione ottima $\bar{x}$ del rilassamento continuo si può calcolare nel modo seguente:

1. Ordina gli oggetti in ordine decrescente di valore per unità di peso (rendimento) $\dfrac{v_i}{p_i}$
2. Trova l'indice h tale che $\sum_{i=1}^{h} p_i \leq C$ e $\sum_{i=1}^{h+1} p_i > C$,
3. Poni $\bar{x}_1 = 1, \dots, \bar{x}_h = 1,\ \bar{x}_{h+1} = \dfrac{C - \sum_{i=1}^{h} p_i}{p_{h+1}},\ \bar{x}_{h+2} = 0, \dots, \bar{x}_n = 0.$

Problema dello zaino: rilassamento continuo

Esempio.

$$\begin{cases} \max\ 11\,x_1 + 23\,x_2 + 18\,x_3 + 6\,x_4 \\ 7\,x_1 + 6\,x_2 + 3\,x_3 + 2\,x_4 \leq 8 \\ x_i \in \{0,1\} \qquad i = 1,\dots,4 \end{cases}$$

Ordiniamo gli oggetti in ordine decrescente di rendimento:

i	3	2	4	1
v_i/p_i	6	3.8	3	1.6

La soluzione ottima del rilassamento continuo è $\bar{x}_3 = 1$, $\bar{x}_2 = 5/6$, $\bar{x}_4 = \bar{x}_1 = 0$, cioè $\bar{x} = \left(0, \frac{5}{6}, 1, 0\right)$ di valore 37.2, quindi $v_S(P) = 37$.

Problema dello zaino: algoritmo euristico

Una soluzione ammissibile può essere ottenuta utilizzando il seguente algoritmo euristico:

1. Ordina gli oggetti in ordine decrescente di rendimento $\dfrac{v_i}{p_i}$
2. Poni $\bar{C} = C$ $\quad$ ($\bar{C}$ rappresenta la capacità residua del contenitore)
3. **for** $i = 1, \ldots, n$ **do**
   if $p_i \leq \bar{C}$
   then $x_i = 1,\ \bar{C} = \bar{C} - p_i$
   else $x_i = 0$
   end

Problema dello zaino: algoritmo euristico

Una soluzione ammissibile può essere ottenuta utilizzando il seguente algoritmo euristico:

1. Ordina gli oggetti in ordine decrescente di rendimento $\frac{v_i}{p_i}$
2. Poni $\bar{C} = C$ ($\bar{C}$ rappresenta la capacità residua del contenitore)
3. **for** $i = 1, \ldots, n$ **do**
   - **if** $p_i \leq \bar{C}$
     - **then** $x_i = 1$, $\bar{C} = \bar{C} - p_i$
   - **else** $x_i = 0$

   end

Esempio (continua).

$$\begin{cases} \max\ 11\,x_1 + 23\,x_2 + 18\,x_3 + 6\,x_4 \\ 7\,x_1 + 6\,x_2 + 3\,x_3 + 2\,x_4 \leq 8 \\ x_i \in \{0,1\} \qquad i = 1,\ldots,4 \end{cases}$$

L'algoritmo euristico fornisce la soluzione ammissibile $x_3 = 1$, $x_2 = 0$, $x_4 = 1$, $x_1 = 0$, quindi $x = (0,0,1,1)$ e $v_I(P) = 24$.

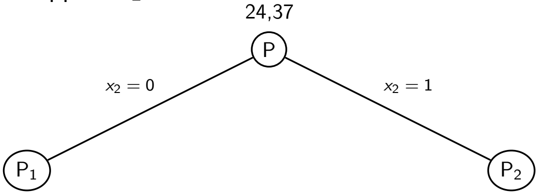
Problema dello zaino: metodo Branch and Bound

Utilizziamo il metodo Branch and Bound per trovare una soluzione ottima del problema

$$\begin{cases} \max\ 11\,x_1 + 23\,x_2 + 18\,x_3 + 6\,x_4 \\ 7\,x_1 + 6\,x_2 + 3\,x_3 + 2\,x_4 \leq 8 \\ x_i \in \{0,1\} \qquad i = 1,\ldots,4 \end{cases}$$

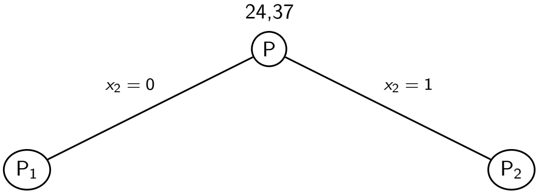
Bound: rilassamento continuo.
Branch binario: istanziamo l'eventuale variabile frazionaria della soluzione ottima del rilassamento continuo.
Visita dell'albero decisionale: in ampiezza.

La soluzione ottima del rilassamento continuo di P è $\left(0, \frac{5}{6}, 1, 0\right)$, istanziamo la variabile x_2: $x_2 = 0$ oppure $x_2 = 1$.



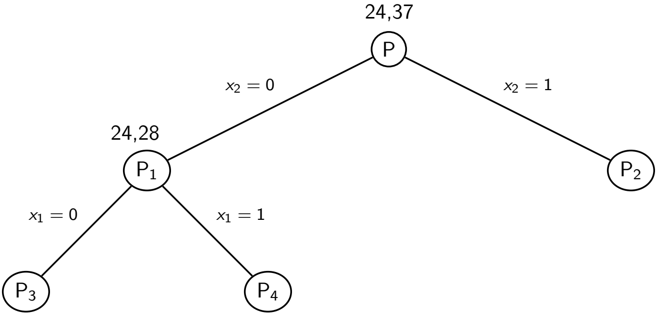
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_1 è $(3/7, 0, 1, 1)$ di valore $28 = v_S(P_1) > 24 = v_I(P)$, quindi P_1 rimane aperto.



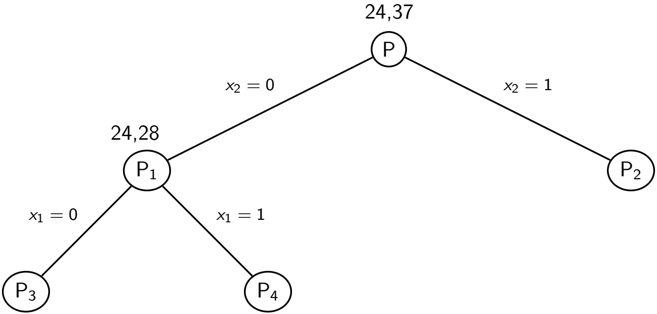
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_1 è $(3/7, 0, 1, 1)$ di valore $28 = v_S(P_1) > 24 = v_I(P)$, quindi P_1 rimane aperto. Generiamo due nodi figli P_3 e P_4 istanziando la variabile x_1.



Problema dello zaino: metodo Branch and Bound

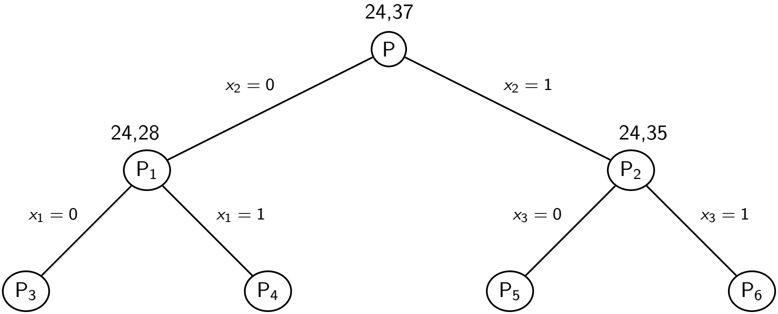
La soluzione ottima del rilassamento continuo di P_1 è $(3/7, 0, 1, 1)$ di valore $28 = v_S(P_1) > 24 = v_I(P)$, quindi P_1 rimane aperto. Generiamo due nodi figli P_3 e P_4 istanziando la variabile x_1.
La soluzione ottima del rilassamento continuo di P_2 è $(0, 1, 2/3, 0)$ di valore $35 = v_S(P_2) > 24 = v_I(P)$, quindi P_2 rimane aperto.



Problema dello zaino: metodo Branch and Bound

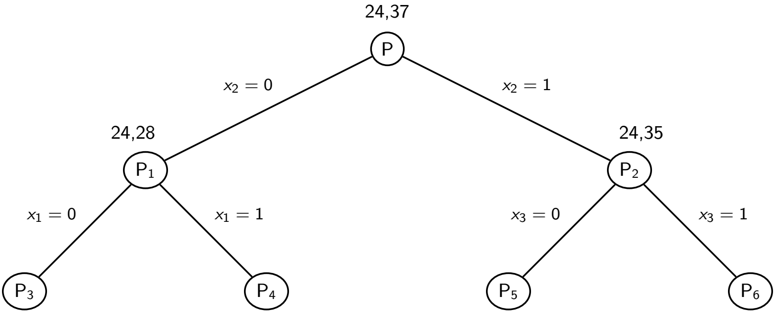
La soluzione ottima del rilassamento continuo di P_1 è $(3/7, 0, 1, 1)$ di valore $28 = v_S(P_1) > 24 = v_I(P)$, quindi P_1 rimane aperto. Generiamo due nodi figli P_3 e P_4 istanziando la variabile x_1.

La soluzione ottima del rilassamento continuo di P_2 è $(0, 1, 2/3, 0)$ di valore $35 = v_S(P_2) > 24 = v_I(P)$, quindi P_2 rimane aperto. Generiamo due nodi figli P_5 e P_6 istanziando la variabile x_3.



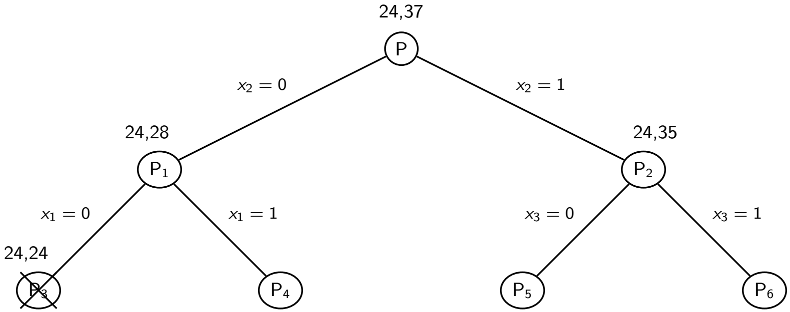
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.



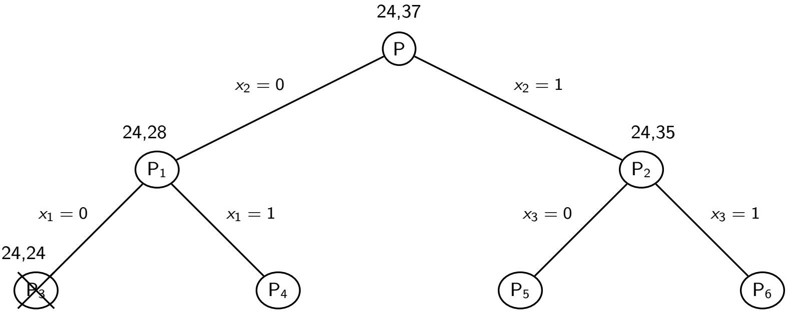
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.



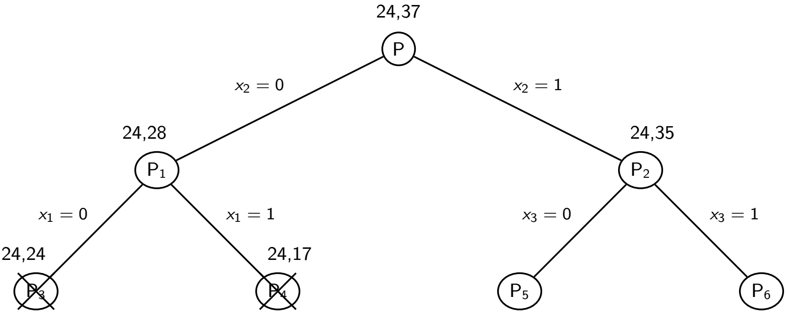
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.
La soluzione ottima del rilassamento continuo di P_4 è $(1, 0, 1/3, 0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P_4.



Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.
La soluzione ottima del rilassamento continuo di P_4 è $(1, 0, 1/3, 0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P_4.

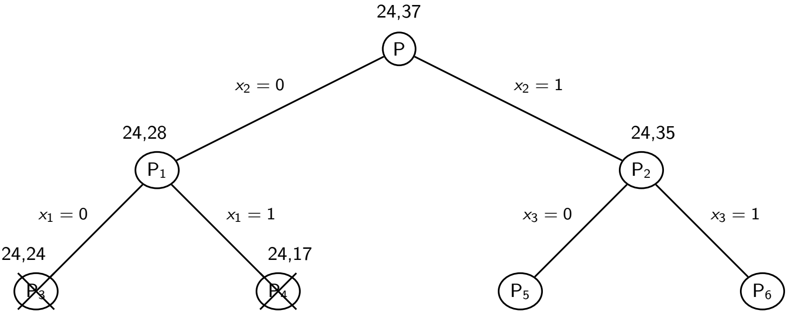


Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0,0,1,1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.

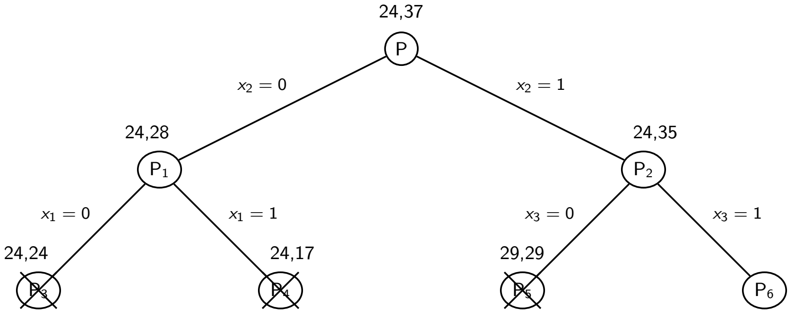
La soluzione ottima del rilassamento continuo di P_4 è $(1,0,1/3,0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P_4.

La soluzione ottima del rilassamento continuo di P_5 è $(0,1,0,1)$ di valore $29 = v_S(P_5) > 24 = v_I(P)$. Poiché tale soluzione è ammissibile per P, aggiorniamo $v_I(P) = 29$ e chiudiamo P_5.



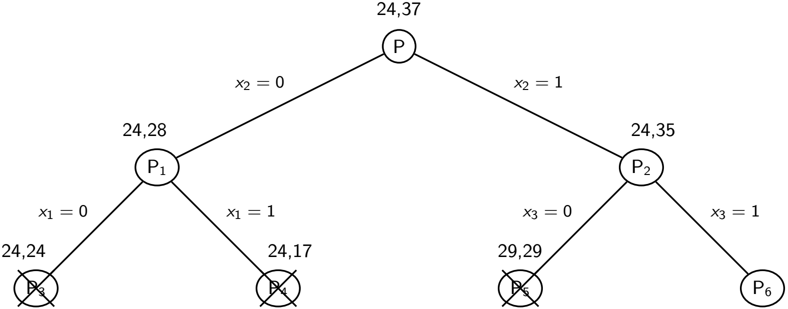
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P$_3$ è $(0,0,1,1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P$_3$.
La soluzione ottima del rilassamento continuo di P$_4$ è $(1,0,1/3,0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P$_4$.
La soluzione ottima del rilassamento continuo di P$_5$ è $(0,1,0,1)$ di valore $29 = v_S(P_5) > 24 = v_I(P)$. Poiché tale soluzione è ammissibile per P, aggiorniamo $v_I(P) = 29$ e chiudiamo P$_5$.



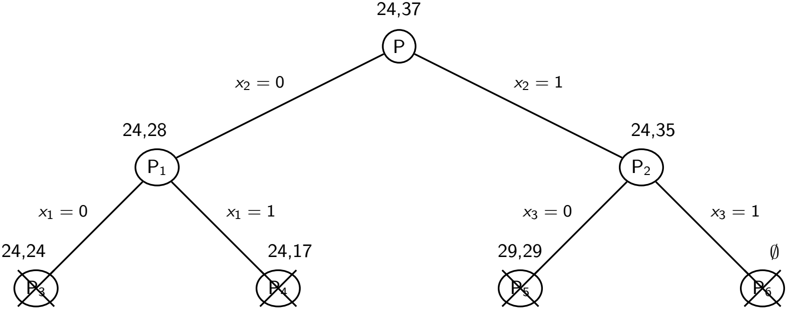
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.
La soluzione ottima del rilassamento continuo di P_4 è $(1, 0, 1/3, 0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P_4.
La soluzione ottima del rilassamento continuo di P_5 è $(0, 1, 0, 1)$ di valore $29 = v_S(P_5) > 24 = v_I(P)$. Poiché tale soluzione è ammissibile per P, aggiorniamo $v_I(P) = 29$ e chiudiamo P_5.
P_6 non ammette soluzioni ammissibili, quindi chiudiamo P_6.



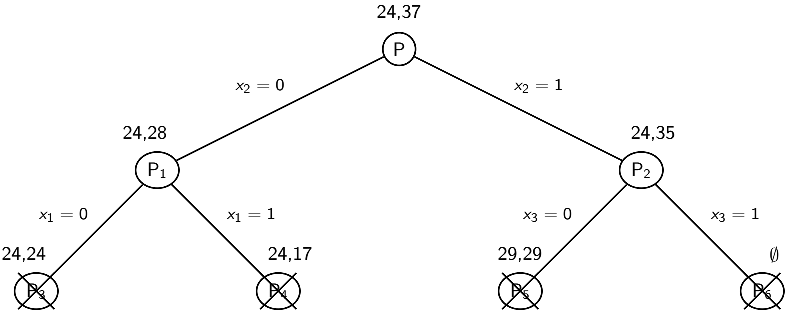
Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.
La soluzione ottima del rilassamento continuo di P_4 è $(1, 0, 1/3, 0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P_4.
La soluzione ottima del rilassamento continuo di P_5 è $(0, 1, 0, 1)$ di valore $29 = v_S(P_5) > 24 = v_I(P)$. Poiché tale soluzione è ammissibile per P, aggiorniamo $v_I(P) = 29$ e chiudiamo P_5.
P_6 non ammette soluzioni ammissibili, quindi chiudiamo P_6.



Problema dello zaino: metodo Branch and Bound

La soluzione ottima del rilassamento continuo di P_3 è $(0, 0, 1, 1)$ di valore $24 = v_S(P_3) = v_I(P)$, quindi chiudiamo P_3.
La soluzione ottima del rilassamento continuo di P_4 è $(1, 0, 1/3, 0)$ di valore $17 = v_S(P_4) < 24 = v_I(P)$, quindi chiudiamo P_4.
La soluzione ottima del rilassamento continuo di P_5 è $(0, 1, 0, 1)$ di valore $29 = v_S(P_5) > 24 = v_I(P)$. Poiché tale soluzione è ammissibile per P, aggiorniamo $v_I(P) = 29$ e chiudiamo P_5.
P_6 non ammette soluzioni ammissibili, quindi chiudiamo P_6.



Poiché tutti i nodi dell'albero sono stati visitati, la soluzione ottima di P è $(0, 1, 0, 1)$ ed il valore ottimo è 29.

Problema del commesso viaggiatore (TSP)

Problema

Dato un grafo (N, A) completo, in cui c_{ij} è costo dell'arco (i, j), trovare un ciclo che passi su tutti i nodi una ed una sola volta (ciclo hamiltoniano) di costo totale minimo.

Se la matrice dei costi è simmetrica, cioè $c_{ij} = c_{ji}$ per ogni arco (i, j), il problema è detto simmetrico; altrimenti è detto asimmetrico.
Nel seguito studieremo solo il caso simmetrico, quindi supponiamo che il grafo (N, A) sia non orientato.

Teorema

Questo problema è NP-hard.

Esercizio. Quanti cicli hamiltoniani esistono in un grafo completo con n nodi?

Problema del commesso viaggiatore (TSP)

Applicazioni

- trasporti, logistica: (N', A') rete stradale. $S \subseteq N'$, cerco ciclo di costo minimo che passi su tutti i nodi di S. Il problema è un TSP sul grafo (N, A), dove $N = S$, $A = S \times S$, c_{ij} = costo cammino minimo da i a j sul grafo (N', A').

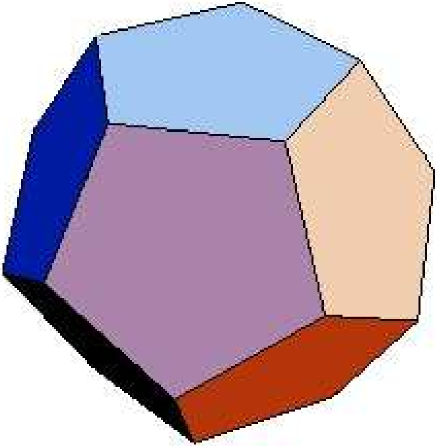
Problema del commesso viaggiatore (TSP)

Applicazioni

- trasporti, logistica: (N', A') rete stradale. $S \subseteq N'$, cerco ciclo di costo minimo che passi su tutti i nodi di S. Il problema è un TSP sul grafo (N, A), dove $N = S$, $A = S \times S$, $c_{ij} =$ costo cammino minimo da i a j sul grafo (N', A').
- scheduling
- produzione di circuiti integrati
- data analysis
- sequenze DNA
- ... (vedi http://www.tsp.gatech.edu/)

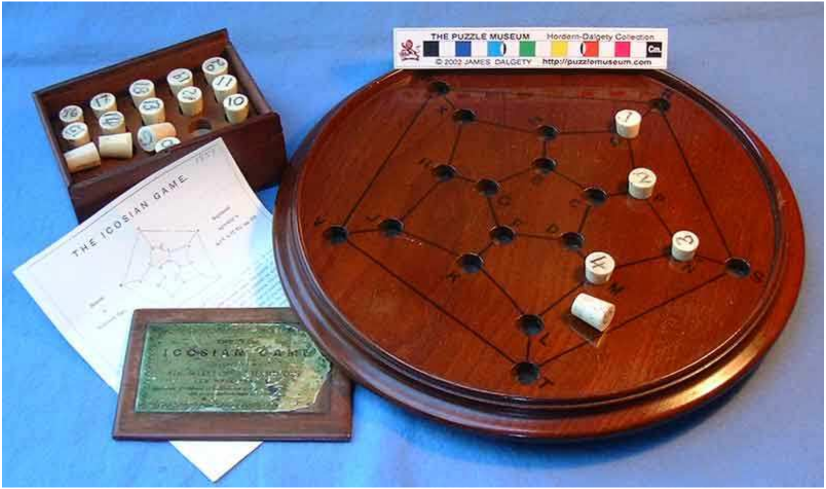
Perché ciclo hamiltoniano?

William Rowan Hamilton (1805-1865): in un dodecaedro regolare è possibile partire da un vertice e, passando sugli spigoli, toccare tutti i vertici una ed una sola volta e tornare al vertice di partenza?



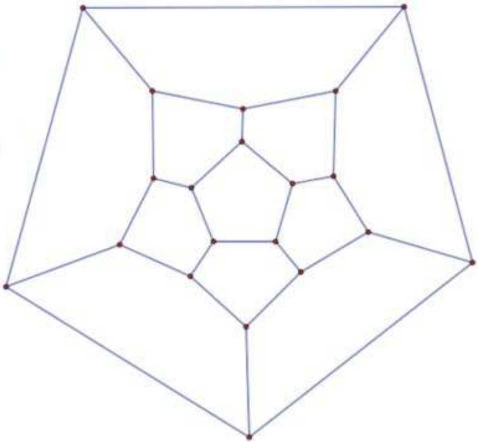
Perché ciclo hamiltoniano?

Icosian game: in un dodecaedro regolare è possibile partire da un vertice e, passando sugli spigoli, toccare tutti i vertici una ed una sola volta e tornare al vertice di partenza?



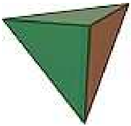
Perché ciclo hamiltoniano?

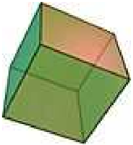
Esercizio. Risolvere l'icosian game:

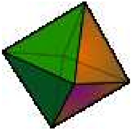


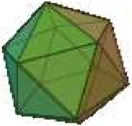
Perché ciclo hamiltoniano?

Esercizio. Per ognuno degli altri 4 solidi regolari









costruire un grafo in cui i nodi rappresentano i vertici del poliedro e gli archi rappresentano gli spigoli e cercare un ciclo hamiltoniano.

Modello 1

Variabili: $x_{ij} = \begin{cases} 1 & \text{se } \{i,j\} \in \text{ciclo hamiltoniano}, \\ 0 & \text{altrimenti}, \end{cases}$ con $i < j$.

$$\min \sum_{i \in N} \sum_{\substack{j \in N \\ j > i}} c_{ij}\, x_{ij}$$

$$\sum_{\substack{j \in N \\ j > i}} x_{ij} + \sum_{\substack{j \in N \\ j < i}} x_{ji} = 2 \qquad \forall\, i \in N \tag{4}$$

$$\sum_{i \in S} \sum_{\substack{j \notin S \\ j > i}} x_{ij} + \sum_{i \notin S} \sum_{\substack{j \in S \\ j > i}} x_{ij} \geq 1 \quad \forall\, S \subset N, \quad |S| \geq 1 \tag{5}$$

$$x_{ij} \in \{0,1\} \quad \forall\, i,j \in N, \quad i < j$$

(4): due archi incidenti su ogni nodo (cioè ogni nodo ha grado 2).
(5): eliminazione di sottocicli.

Modello 2

Variabili: $x_{ij} = \begin{cases} 1 & \text{se } \{i,j\} \in \text{ciclo hamiltoniano,} \\ 0 & \text{altrimenti,} \end{cases}$ con $i < j$.

$$
\begin{aligned}
\min \; & \sum_{i \in N} \sum_{\substack{j \in N \\ j > i}} c_{ij}\, x_{ij} \\
& \sum_{\substack{j \in N \\ j > i}} x_{ij} + \sum_{\substack{j \in N \\ j < i}} x_{ji} = 2 \qquad \forall\ i \in N && (6) \\
& \sum_{i \in S} \sum_{\substack{j \in S \\ j > i}} x_{ij} \le |S| - 1 \quad \forall\ S \subset N, \quad |S| \ge 3 && (7) \\
& x_{ij} \in \{0,1\} \quad \forall\ i,j \in N, \quad i < j
\end{aligned}
$$

(6): ogni nodo ha grado 2.
(7): eliminazione di sottocicli.

r-albero

Fissiamo un nodo r nel grafo. Un r–albero è un insieme di n archi di cui

- 2 sono incidenti sul nodo r
- $n - 2$ formano un albero di copertura sui nodi diversi da r

Ogni ciclo hamiltoniano è un r–albero.

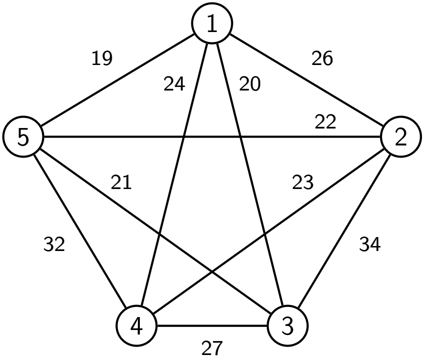
Il problema dell'r–albero di costo minimo è un rilassamento del TSP simmetrico. Inoltre è facile da risolvere, perché basta trovare

- 2 archi di costo minimo incidenti sul nodo r (ovvio)
- un albero di copertura di costo minimo sui nodi diversi da r (alg. Kruskal/Prim)

Rilassamento: r-albero di costo minimo

Esempio

Il 3–albero di costo minimo sul grafo

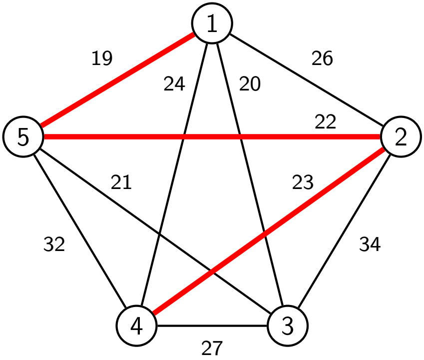


è formato dagli archi:

Rilassamento: r-albero di costo minimo

Esempio

Il 3–albero di costo minimo sul grafo

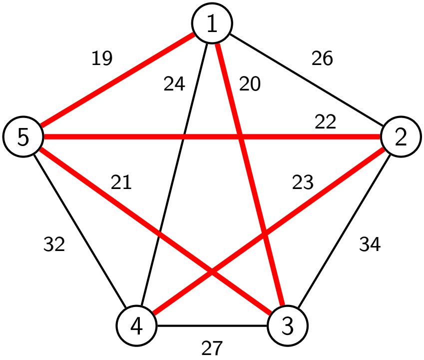


è formato dagli archi:
$\{1, 5\}$, $\{2, 5\}$, $\{2, 4\}$ (albero di copertura sui nodi diversi da 3).

Rilassamento: r-albero di costo minimo

Esempio

Il 3–albero di costo minimo sul grafo



è formato dagli archi:

$\{1,5\}$, $\{2,5\}$, $\{2,4\}$ (albero di copertura sui nodi diversi da 3).

$\{1,3\}$, $\{3,5\}$ (incidenti sul nodo 3)

e ha costo $105 = v_I(P)$.

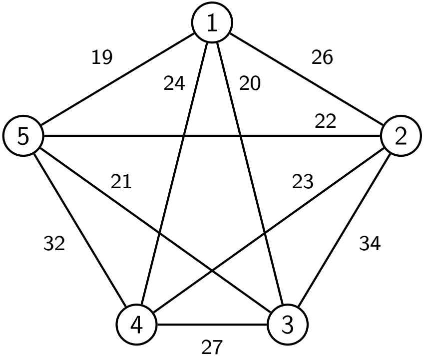
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo

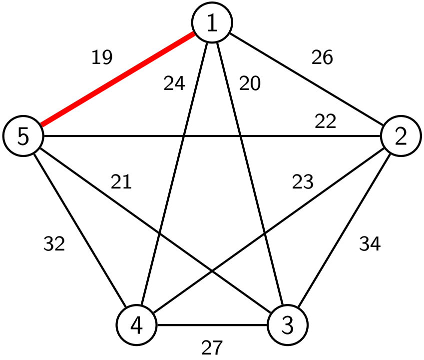
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.

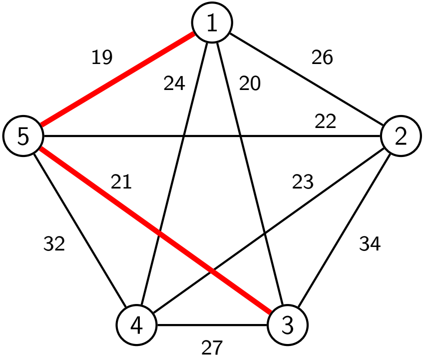
3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5

Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).
1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.
2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \; c_{ki}\}$.
3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3

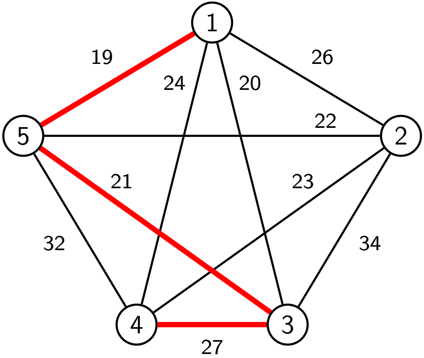
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \; c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4

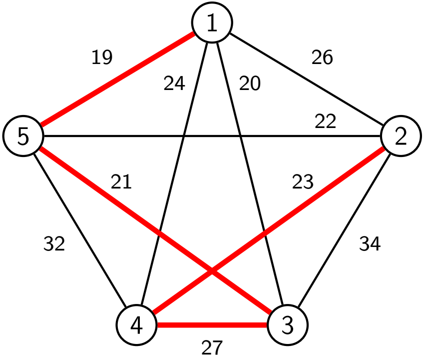
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \; c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2

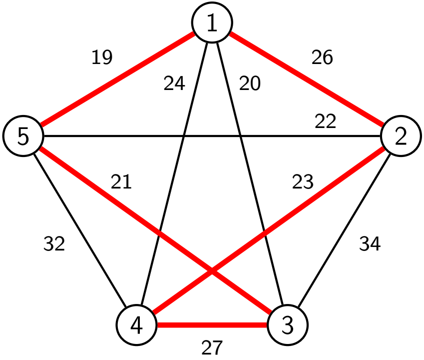
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.

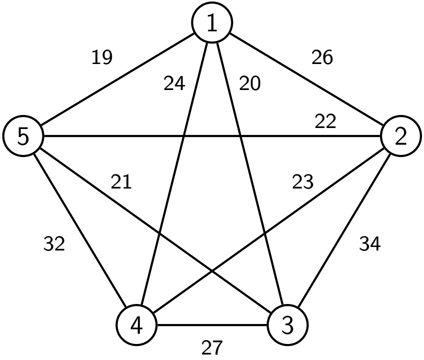
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \; c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.
Partendo dal nodo 2 si ottiene il ciclo

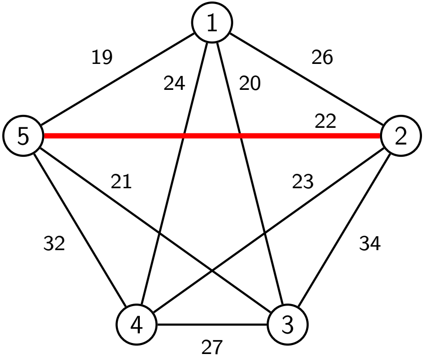
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.
Partendo dal nodo 2 si ottiene il ciclo 2–5

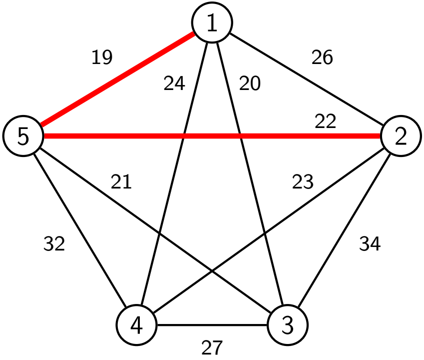
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \; c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.
Partendo dal nodo 2 si ottiene il ciclo 2–5–1

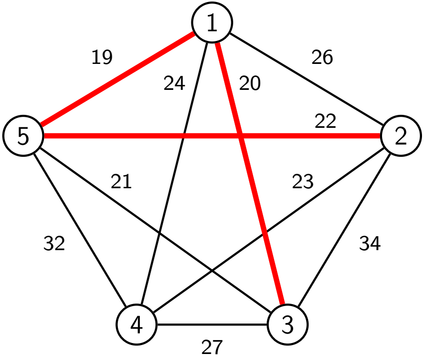
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.

3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.
Partendo dal nodo 2 si ottiene il ciclo 2–5–1–3

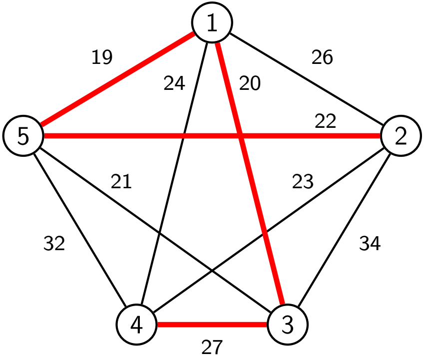
Metodo euristico: algoritmo del nodo più vicino

0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).

1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.

2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.

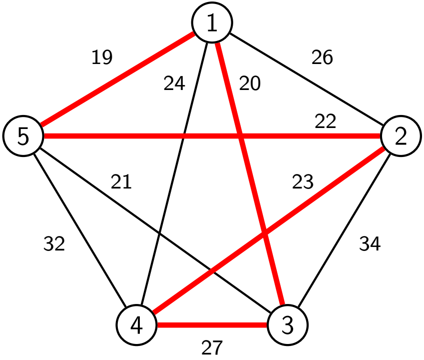
3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.
Partendo dal nodo 2 si ottiene il ciclo 2–5–1–3–4

Metodo euristico: algoritmo del nodo più vicino

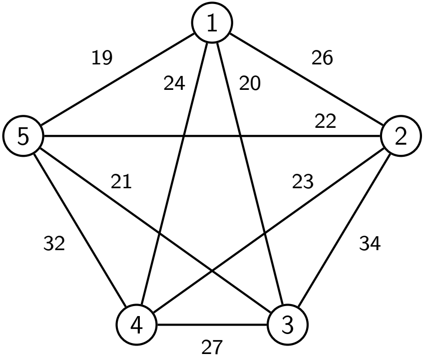
0. Scegli un nodo qualunque i, poni $k = i$ (nodo corrente), $U = N \setminus \{i\}$ (nodi non visitati), $C = i$ (ciclo - sequenza di nodi).
1. Se $U = \emptyset$, allora aggiungi i in fondo a C e STOP.
2. Scegli il nodo di U più vicino a k, cioè $j = \arg\min\{i \in U : \ c_{ki}\}$.
3. Aggiungi j in fondo a C, poni $U = U \setminus \{j\}$ e $k = j$. Torna al passo 1.



Partendo dal nodo 1 si ottiene il ciclo 1–5–3–4–2–1 di costo $116 = v_S(P)$.
Partendo dal nodo 2 si ottiene il ciclo 2–5–1–3–4–2 di costo $111 = v_S(P)$.

Metodo Branch and Bound

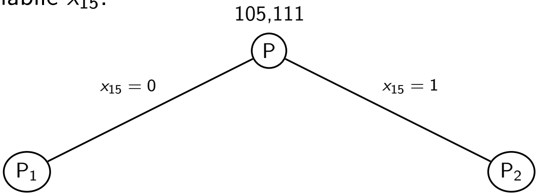
Utilizziamo il metodo Branch and Bound per trovare un ciclo hamiltoniano di costo minimo sul seguente grafo:



Bound: 3-albero di costo minimo.
Branch binario: istanziamo nell'ordine le variabili x_{15}, x_{35}, x_{13}.
Visita dell'albero decisionale: in ampiezza.

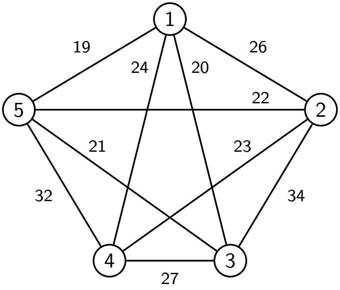
Metodo Branch and Bound

L'algoritmo del nodo più vicino fornisce $v_S(P) = 111$, mentre il 3-albero di costo minimo continuo fornisce $v_I(P) = 105$.
Istanziamo la variabile x_{15}.



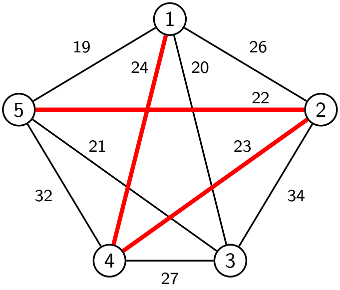
Metodo Branch and Bound

Rilassamento di P_1 ($x_{15} = 0$): il 3-albero di costo minimo contiene gli archi



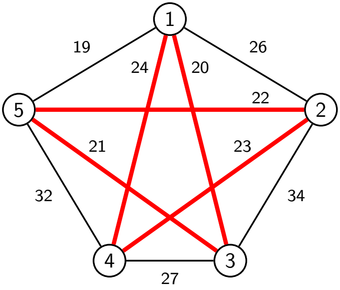
Metodo Branch and Bound

Rilassamento di P_1 ($x_{15} = 0$): il 3-albero di costo minimo contiene gli archi $\{2, 5\}$, $\{2, 4\}$, $\{1, 4\}$ (albero di copertura sui nodi diversi da 3) e



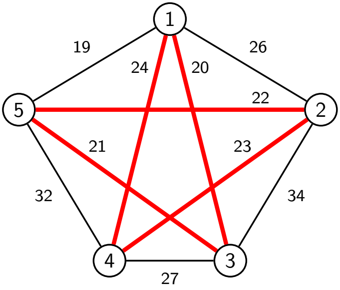
Metodo Branch and Bound

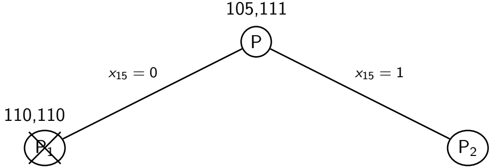
Rilassamento di P_1 ($x_{15} = 0$): il 3-albero di costo minimo contiene gli archi $\{2,5\}$, $\{2,4\}$, $\{1,4\}$ (albero di copertura sui nodi diversi da 3) e $\{1,3\}$, $\{3,5\}$ (incidenti sul nodo 3) e costa $110 = v_I(P_1)$.



Metodo Branch and Bound

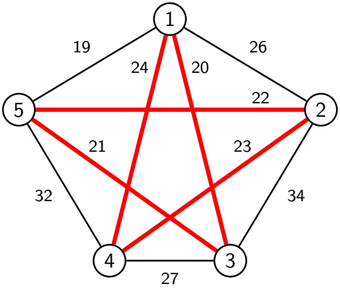
Rilassamento di P_1 ($x_{15} = 0$): il 3-albero di costo minimo contiene gli archi $\{2, 5\}$, $\{2, 4\}$, $\{1, 4\}$ (albero di copertura sui nodi diversi da 3) e $\{1, 3\}$, $\{3, 5\}$ (incidenti sul nodo 3) e costa $110 = v_I(P_1)$. Questi archi formano il ciclo hamiltoniano 1-3-5-2-4-1, quindi chiudiamo P_1 e aggiorniamo $v_S(P) = 110$.

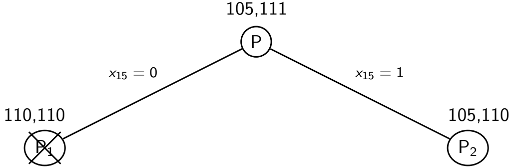




Metodo Branch and Bound

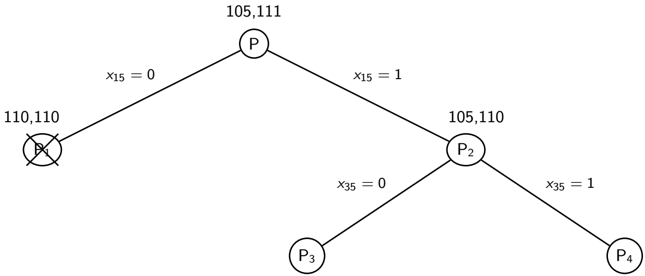
Rilassamento di P$_1$ ($x_{15} = 0$): il 3-albero di costo minimo contiene gli archi $\{2,5\}$, $\{2,4\}$, $\{1,4\}$ (albero di copertura sui nodi diversi da 3) e $\{1,3\}$, $\{3,5\}$ (incidenti sul nodo 3) e costa $110 = v_I(P_1)$. Questi archi formano il ciclo hamiltoniano 1-3-5-2-4-1, quindi chiudiamo P$_1$ e aggiorniamo $v_S(P) = 110$.



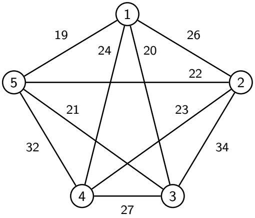


Rilassamento di P$_2$ ($x_{15} = 1$): il 3-albero di costo minimo coincide con quello calcolato nel nodo P perché quest'ultimo contiene l'arco (1,5), quindi $v_I(P_2) = 105 < 110 = v_S(P)$ e P$_2$ rimane aperto.

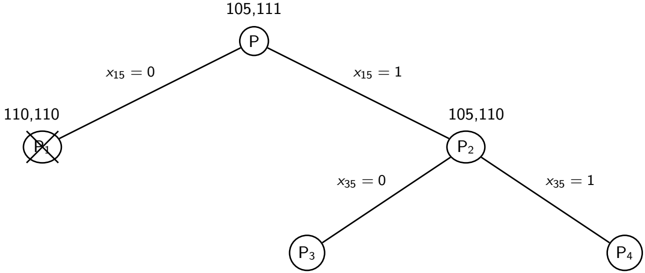
Metodo Branch and Bound



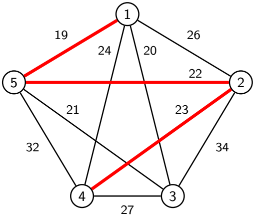
Rilassamento di P_3: il 3-albero di costo minimo contiene gli archi



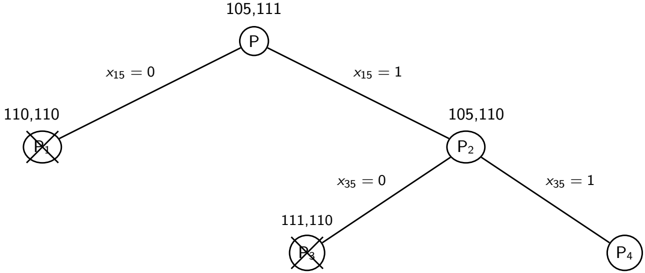
Metodo Branch and Bound



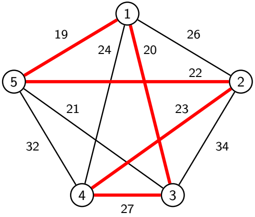
Rilassamento di P_3: il 3-albero di costo minimo contiene gli archi $\{1,5\}$, $\{2,4\}$, $\{2,5\}$ (albero di copertura sui nodi diversi da 3) e



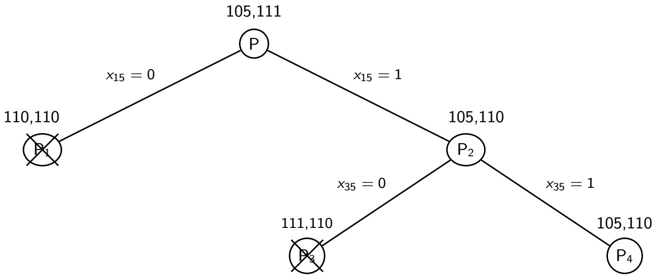
Metodo Branch and Bound



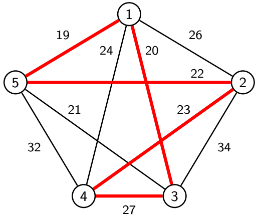
Rilassamento di P_3: il 3-albero di costo minimo contiene gli archi $\{1,5\}$, $\{2,4\}$, $\{2,5\}$ (albero di copertura sui nodi diversi da 3) e $\{1,3\}$, $\{3,4\}$ (incidenti sul nodo 3) e costa $111 = v_I(P_3) > 110 = v_S(P)$, quindi chiudiamo P_3.



Metodo Branch and Bound

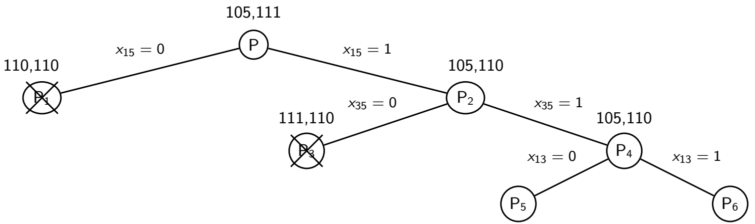


Rilassamento di P_3: il 3-albero di costo minimo contiene gli archi $\{1,5\}$, $\{2,4\}$, $\{2,5\}$ (albero di copertura sui nodi diversi da 3) e $\{1,3\}$, $\{3,4\}$ (incidenti sul nodo 3) e costa $111 = v_I(P_3) > 110 = v_S(P)$, quindi chiudiamo P_3.

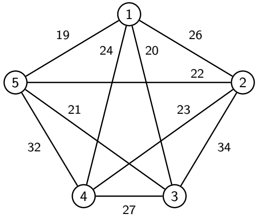


Rilassamento di P_4: il 3-albero di costo minimo coincide con quello calcolato nel nodo P perché quest'ultimo contiene gli archi (1,5) e (3,5), quindi $v_I(P_4) = 105 < 110 = v_S(P)$ e P_4 rimane aperto.

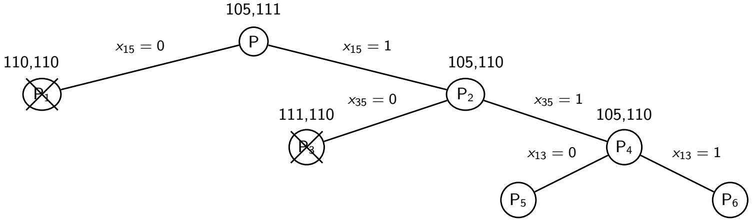
Metodo Branch and Bound



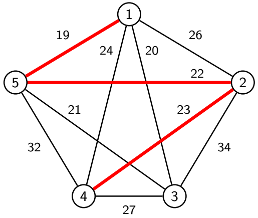
Rilassamento di P_5: il 3-albero di costo minimo contiene gli archi



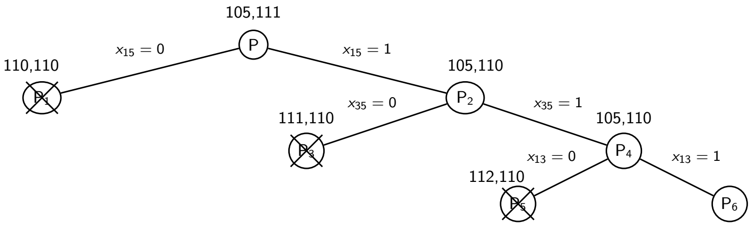
Metodo Branch and Bound



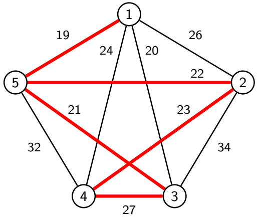
Rilassamento di P_5: il 3-albero di costo minimo contiene gli archi $\{1,5\}$, $\{2,5\}$, $\{2,4\}$ (albero di copertura sui nodi diversi da 3) e



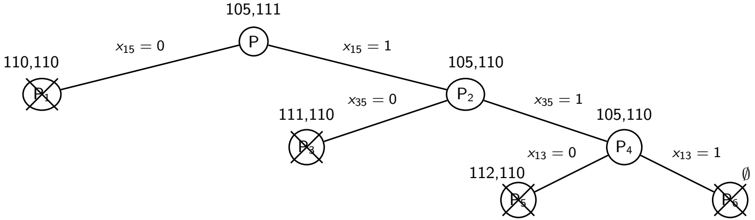
Metodo Branch and Bound



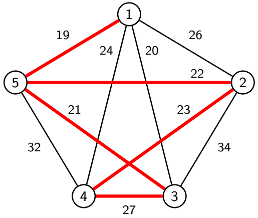
Rilassamento di P_5: il 3-albero di costo minimo contiene gli archi $\{1,5\}$, $\{2,5\}$, $\{2,4\}$ (albero di copertura sui nodi diversi da 3) e $\{3,5\}$, $\{3,4\}$ (incidenti sul nodo 3) e costa $112 = v_I(P_5) > 110 = v_S(P)$, quindi chiudiamo P_5.



Metodo Branch and Bound



Rilassamento di P_5: il 3-albero di costo minimo contiene gli archi $\{1,5\}$, $\{2,5\}$, $\{2,4\}$ (albero di copertura sui nodi diversi da 3) e $\{3,5\}$, $\{3,4\}$ (incidenti sul nodo 3) e costa $112 = v_I(P_5) > 110 = v_S(P)$, quindi chiudiamo P_5.



P_6 non contiene cicli hamiltoniani perché è presente il sottociclo 1-5-3, quindi chiudiamo anche P_6.
Pertanto il ciclo hamiltoniano di costo minimo è 1-3-5-2-4-1 di costo 110.