Capitolo 4
Programmazione Lineare Intera

note a cura di Francesco Straullu

Indice

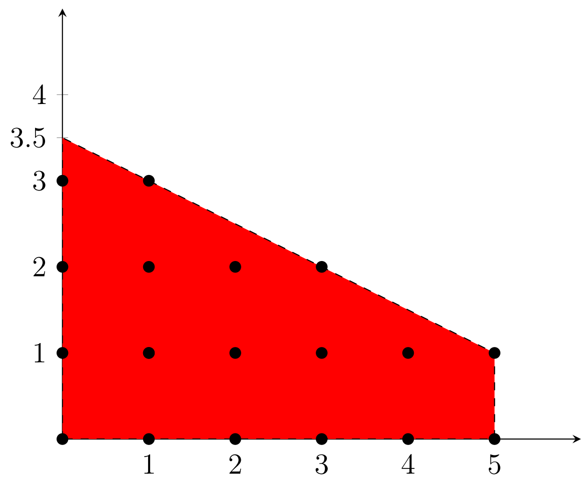
1 Programmazione Lineare Intera

La Programmazione Lineare Intera studia problemi di ottimizzazione analoghi ai problemi di PL in cui tutte le relazioni tra le variabili e la funzione obbiettivo sono lineari, ma ai quali si chiede che le soluzioni siano a componenti intere. La coppia di problemi assomiglia quindi alla coppia asimmetrica tipica della programmazione lineare.
Questi problemi si possono modellare con variabili che prendono solo valori in $\{0,1\}$, ma sono un sottocaso del caso in $\mathbb{Z}^n$ in cui il **vincolo di interezza** $x \in \mathbb{Z}^n$ diventa semplicemente $x \in \{0,1\}$ Possiamo quindi definire i problemi **($\mathbf{P}_I$)** e **($\mathbf{D}_I$)** come:

$$(P_I) \quad \begin{array}{l} \max\, c \cdot x \\ Ax \leq b \\ x \in \mathbb{Z}^n \end{array} \qquad\qquad (D_I) \quad \begin{array}{l} \min\, y \cdot b \\ yA = c \\ y \in \mathbb{Z}^m_+ \end{array}$$

Il problema **($\mathbf{P}_I$)** è quindi il problema in forma primale, il cui poliedro può essere descritto come nella programmazione lineare da disuguaglianze lineari, ma la cui regione ammissibile è l'intersezione tra il poliedro e i punti che appartengono a $\mathbb{Z}^n$, e viene anche detto **reticolo intero**.

$$\begin{array}{rll} \mathrm{P} & = \{x \in \mathbb{R}^n : Ax \leq b\}, & \textbf{Poliedro} \\ \mathrm{S} & = P \cap \mathbb{Z}^n, & \textbf{Regione ammissibile} \end{array}$$



La regione in rosso è il poliedro P (rilassamento continuo del problema intero) mentre i punti in nero sono il reticolo intero del problema. I vertici del poliedro possono avere componenti non intere come ad esempio $(0, 3.5)$.

Analogamente per il problema duale, si definiscono il poliedro e la regione ammissibile.

$$\begin{array}{rll} \mathrm{Q} & = \{y \in \mathbb{R}^m_+ : yA = c, y \geq 0\}, & \textbf{Poliedro} \\ \mathrm{T} & = Q \cap \mathbb{Z}^m, & \textbf{Regione ammissibile} \end{array}$$

Tra questi due problemi, continua a valere la dualità debole ma non la dualità forte. Questo comporta una difficoltà maggiore nel risolvere i problemi di PLI e, quando possibile, a costruire algoritmi di complessità polinomiale.

1.1 Rilassamenti continui

Per provare a risolvere i problemi di PLI, un buon punto di partenza è lo studio dei rilassamenti continui associati. Nel problema $(\mathbf{P}_I)$ significa studiare il primale senza il vincolo di interezza, mentre nel $(\mathbf{D}_I)$ si impone la positività in quanto $y \in \mathbb{Z}_+^m$.

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

Il rilassamento continuo ha più soluzioni del corrispettivo problema intero, per cui l'insieme delle soluzioni di $(\mathbf{RC}_P)$ è più grande di quello delle soluzioni di $(\mathbf{P}_I)$ e quindi tutte le soluzioni ammissibili del problema intero sono anche ammissibili per il rilassamento continuo.
Supponendo quindi di risolvere il rilassamento continuo, se la soluzione ottima trovata $\bar{x} \in P$ è anche a componenti intere ($\bar{x} \in S$), questa è anche soluzione del problema intero di partenza (P_I). Vale discorso analogo per il duale ($\bar{y} \in Q$ sol.ne ottima di (RC_D) e $\bar{y} \in T, \rightarrow \bar{y}$ sol.ne ottima di (D_I)). Questo, accade sicuramente qualora tutti i vertici (ovvero le soluzioni di base ammissibili) di P siano a componenti intere (**proprietà dell'interezza**). Per vederlo, basta notare che essendo $\bar{x} \in P$ ha il massimo tra tutti i punti in P, ed essendo $S \subset P$, allora $\bar{x}$ è anche il massimo di S, ed essendo a componenti intere, è soluzione ammissibile e quindi, ottima.
Si potrebbe pensare che utilizzando una **tecnica dell'arrotondamento intero** si potrebbe trovare la soluzione del problema intero arrotondando (per difetto o per eccesso) la soluzione del rilassamento. In generale, però, questo è falso.

1.2 Convessificazione reticolo intero

Si dice involucro (o inviluppo) convesso di S l'insieme:

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

N.B. conv S può non essere un poliedro

$$P = \left\{ x \in \mathbb{R}_+^2 : x_2 \geq \sqrt{2} x_1 \right\} \quad \text{conv S} = \left\{ x \in \mathbb{R}_+^2 : x_2 > \sqrt{2} x_1 \right\} \subsetneq P$$

Quando **S** è finito o **A** e **b** sono a componenti intere, **conv S** è un poliedro con la proprietà dell'interezza e per cui risulta:

$$\begin{array}{c} \max\ c \cdot x \\ x \in S \end{array} \equiv \begin{array}{c} \max\ c \cdot x \\ x \in \text{conv S} \end{array}$$

Ma caratterizzare **conv S** tramite vincoli espliciti è estremamente difficile (la caratterizzazione è nota solo in pochi casi).

2 Disuguaglianze Valide e Piani di taglio

Dati $d \in \mathbb{R}^n,\ \gamma \in \mathbb{R},\ d \cdot x \geq \gamma$ si dice **disuguaglianza valida** per S se $d \cdot x \geq \gamma \forall x \in S$. Ovvero:

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

Sia $\bar{x} \notin S$ una sol.ne ottima del rilassamento continuo (RC_P). Una disuguaglianza valida $d \cdot x \geq \gamma$ si dice **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 sol.ne ottima del relativo rilassamento continuo.*

Piani di Taglio di Gomory

I piani di taglio di Gomory si costruiscono per problemi in *forma duale* ovvero nella forma

$$(D_I)\ \begin{array}{l} \min\ y \cdot b \\ yA = c \\ y \in \mathbb{Z}^m_+ \end{array}$$

Data quindi una soluzione ottima (di base) $\bar{y} == (cA_B^{-1}, 0)$ del rilassamento continuo (RC_D) e si suppone $\bar{y}_r \notin \mathbb{Z}$ (dove $r \in B$ indice di base)[1].
Si definisce una matrice $\tilde{A} = A_N A_B^{-1} \in \mathbb{R}^{(m-n)\times n}$ dove le righe sono numerate secondo gli indici di N e le colonne secondo gli indici di B.

Prop: $\displaystyle\sum_{j \in N} \{\tilde{A}_{jr}\} y_j \geq \{\bar{y}_r\}$ *è un piano di taglio (per* $\bar{y}$*)*

Con $\{\cdot\}$ si indica la parte frazionaria[2] di un numero definita come:

$$\{x\} = x - [x]$$

[1] Se tale r non esistesse, $\bar{y}$ sarebbe sol.ne ottima di (D_I)

[2] es. $\{3.7\} = 0.7$, $\{-3.7\} = 0.3$

dove $[\cdot]$ denota la parte intera, ossia il più grande intero più piccolo del numero stesso.

Dimostrazione
$\bar{y}$ non soddisfa la disuguaglianza ($\bar{y}_j = 0, j \in N$):

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

La disuguaglianza è valida: sia $y \in Q \cap \mathbb{Z}^m$

$y_B A_B + y_N A_N = c$

$y_B = (c - y_N A_N)A_B^{-1} = cA_B^{-1} - y_N A_N A_B^{-1} = \bar{y}_B - y_N \underline{A_N A_B^{-1}}$

Essendo la parte sottolineata esattamente $\tilde{A}$, si ha che:

$y_r = \bar{y}_r - \sum_{j \in N} \tilde{A}_{jr} y_j =$

$= [\bar{y}_r] + \{\bar{y}_r\} - \sum_{j \in N} ([\tilde{A}_{jr}] + \{\tilde{A}_j r\}) y_j$

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

La parte frazionaria e le y_i sono maggiori o uguali a 0 risulterà che

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

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

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

e y soddisfa la disuguaglianza □

Analogamente, si può dimostrare se $\bar{y} \notin \mathbb{Z}$ che:

$$\sum_{j \in N} \{b_j - A_j A_B^{-1} b_B\} y_j \geq \{\bar{y} \cdot b\}$$

è un piano di taglio.[3]

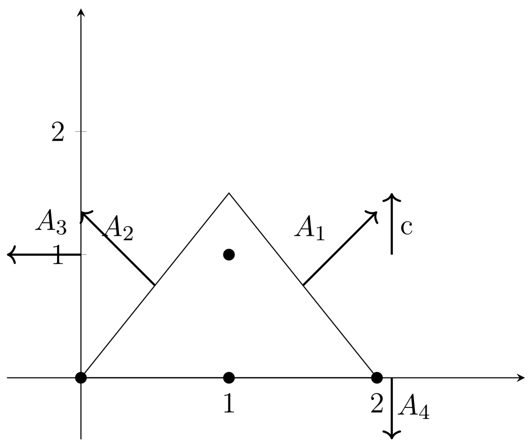
[3] si utilizza $y \cdot b = y_B b_B + y_N b_N = \bar{y} \cdot b + y_N (b_N - A_N A_B^{-1} b_B)$

In definitiva, per trovare una soluzione a componenti intere in un problema in forma duale, si può utilizzare il seguente algoritmo (**di Gomory**)[4].

1. Calcolare $\bar{y}$ sol.ne ottima di base di $\min\{y \cdot b : y \in Q\}$
2. Se $\bar{y} \in \mathbb{Z}^m$, STOP
3. Costruire <u>un</u> pinao di taglio $d \cdot x \geq \gamma$ relativo a $\bar{y}$
4. $Q = Q \cap \{x \in \mathbb{R}^n : d \cdot x \geq \gamma\}$ e si ritorna al punto 1.

Esempio:
Considero il rilassamento continuo del problema primale di partenza:

$$(P_I)\ \begin{array}{c} \max x_2 \\ 3x_1 + 2x_2 \leq 6 \\ -3x_1 + 2x_2 \leq 0 \\ x_1, x_2 \in \mathbb{Z}_+ \end{array} \qquad (RC_P)\ \begin{array}{c} \max x_2 \\ 3x_1 + 2x_2 \leq 6 \\ -3x_1 + 2x_2 \leq 0 \\ -x_1 \leq 0 \\ -x_2 \leq 0 \end{array}$$



Risolvo (RC_P) ottentendo $B = \{1, 2\}$ e $\bar{x} = (1, 3/2)$
Per individuare il taglio di Gomory bisogna portare il problema in forma duale: $y_1 = x_1,\ y_2 = x_2$ e aggiungo $s_1,\ s_2$ variabili di scarto. $(\bar{y} = (\bar{x_1}, \bar{x_2}, \bar{s_1}, \bar{s_2}))$

$$\begin{array}{c} \text{-min } -y_2 \\ 3y_1 + 2y_2 + s_1 = 6 \\ -3y_1 + 2y_2 + s_2 = 0 \\ y_1, y_2, s_1, s_2 \in \mathbb{Z}_+ \end{array}$$

e ricavo:

$$A = \begin{pmatrix} 3 & -3 \\ 2 & 2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \quad c = (6, 0)$$

[4] l'algoritmo termina in un numero finito di iterazioni se i piani sono scelti con opportune regole

la base risulta essere $B = \{1, 2\}$[5] e soluzione ottima di base $\bar{y} = (1, 3/2, 0, 0)$.
Dalla soluzione di base, ho che $r = 2$. Trovo le matrici $A_B,\ \tilde{A}$:

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

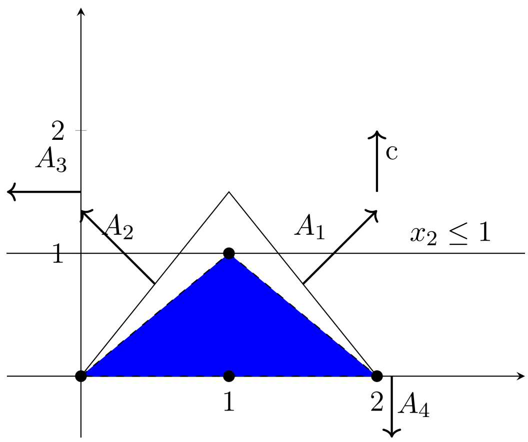
$$\tilde{A} = A_N A_B^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1/6 & 1/4 \\ -1/6 & 1/4 \end{pmatrix} = \begin{pmatrix} 1/6 & 1/4 \\ -1/6 & 1/4 \end{pmatrix}$$

ricordando che le colonne di $\tilde{A}$ sono (1,2) e le righe (3,4).
Applicando la formula di Gomory $\sum_{j \in \{3,4\}} \{\tilde{A}_{jr}\} y_j \geq \bar{y}_r,\ con\ r = 2$

$$\frac{1}{4}y_3 + \frac{1}{4}y_4 \geq \left\{\frac{3}{2}\right\} = \frac{1}{2}$$

da cui essendo $y_3 = s_1,\ y_4 = s_2 \Rightarrow s_1 + s_2 \geq 2$
Dai vincoli, riscrivo le variabili di scarto in funzione di $x_1,\ x_2$

$$\begin{array}{l} s_1 = 6 - 3y_1 - 2y_2 = 6 - 3x_1 - 2x_2 \\ s_2 = 3y_1 - 2y_2 = 3x_1 - 2x_2 \end{array} \quad \rightarrow \quad \begin{array}{rcr} s_1 + s_2 \geq 2 & \Leftrightarrow & 6 - 4x_2 \geq 2 \\ & \Leftrightarrow & x_2 \leq 1 \end{array}$$

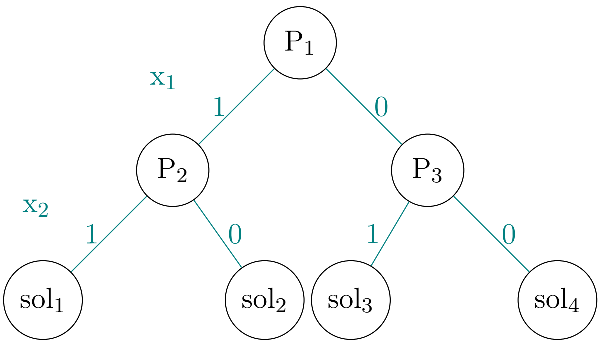


la parte in blu è il conv S

[5] La base non è per forza uguale a quella trovata in partenza; può cambiare per dimensione e indici

3 I metodi Enumerativi

I metodi enumerativi sono una famiglia di metodi risolutivi che si applica ai problemi di PLI con un numero finito di (possibili) soluzioni. Sono particolarmente idonei per i problemi combinatori con variabili binarie.[6]
I metodi enumerativi sfruttano **l'albero di enumerazione totale** per analizzare il problema. È un albero radicato in una ω_i, dove i nodi di un dato livello identificano una variabile e gli archi che portano al livello successivo i possibili valori della suddetta. Le foglie, invece, indicano tutte le possibili soluzioni (comprese quelle non ammissibili.) Si possono quindi, in maniera del tutto analoga, fissare i sottoproblemi nei nodi (il problema originale nella radice) e le variabili con i relativi valori sugli archi, come mostrato di seguito.



Facile esempio di albero di enumerazione.

Ogni nodo individua quindi un sottoalbero, che identifica un sottoproblema (in figura P_2 o P_3) del problema originale (P_1), in cui i valori di alcune variabili sono stati fissati. Per esempio (P_2) si ottiene fissando $x_1 = 1$ nel problema originale.

3.1 Branch and Bound

Abbiamo quindi visto come ciascun sottoproblema sia analogo a quello originale ma con un numero inferiore di variabili. Serve quindi conoscere una stima della qualità delle soluzioni del sottoproblema che sia più facilmente calcolabile. Si considera quindi un **rilassamento:** problema che contiene tutte le soluzioni ammissibili del (sotto) problema più altre, e la cui soluzione ottima è ottenibile tramite algoritmi noti.
L'algoritmo "Branch and Bound" utilizza i rilassamenti come valutazioni superiori o inferiori della soluzione a seconda che il problema sia di massimizzazione o minimizzazione.

[6]Binario: $x \in \{0, 1\}^n$

Gli elementi necessari per un algorimto **B&B** sono:

1) una soluzione ammissibile (di partenza)

2) un rilassamento del problema

3) regole di ramificazione

4) regole di potatura

  4a) il sottoalbero non contiene sol.ni ammissibili di P

  4b) il valore ottimo del rilassamento del sottoproblema è peggiore/non migliore del valore della sol.ne ammissibile corrente

  4c) la sol.ne ottima individuata per il rilassamento del sottoproblema è **ammissibile** per P[7]

La soluzione di partenza, viene calcolata tramite un algoritmo euristico ad-hoc, che varia a seconda del tipo di problema che stiamo analizzando.
Ad ogn iterazione, abbiamo una soluzione corrente, ammissibile ma che non sappiamo se sia ottima. Tramite l'albero di enumerazione valutiamo caso per caso se effettivamente non esistono delle soluzioni migliori. Per cui, se alla fine della visita dell'albero la nostra soluzione non è cambiata, è la soluzione ottima del problema. Nel caso però in cui una delle soluzioni individuate tramite l'algoritmo B&B (punto **4c**) è ammissibile **e migliore della nostra soluzione corrente** diventa la nostra soluzione corrente.
A seconda del tipo di problema, soluzione e rilassamento assumono significati diversi:

- **Massimizzazione:** in un problema di max
  - la soluzione ammissibile fornisce una valutazione inferiore del valore ottimo
  - il rilassamento (di un sottoprob.) fornisce la valutazione superiore del valore ottimo (nel sottoprob.)
- **Minimizzazione:** in un problema di min
  - la soluzione ammissibile fornisce una valutazione superiore del valore ottimo
  - il rilassamento (di un sottoprob.) fornisce la valutazione inferiore del valore ottimo (nel sottoprob.)

[7] N.B. non importa che sia migliore! Se una soluzione è ammissibile posso potare sempre.

Detto altrimenti, in un problema di massimizzazione vogliamo rendere la nostra soluzione più grande possibile, quindi dalla nostra soluzione ammissibile di partenza cerchiamo una soluzione *verso cui crescere* tramite i rilassamenti. Analogamente per il problema di min.

3.2 PB Zaino

Dato uno zaino di capacità b, vogliamo inserire un certo numero di oggetti rispettando il limite di peso e massimizzando il beneficio.
Il problema dello zaino, già visto come modello, in cui si hanno **n oggetti**, di beneficio $c_i > 0$ e peso $a_i > 0$, e uno **zaino** di capacità $b > 0$ e che viene modellato come:

$$\begin{aligned} \max\ & c_1x_1 + \ldots + c_nx_n \\ & a_1x_1 + \ldots + a_nx_n \leq b \\ & x_i \in \{0,1\} \end{aligned}$$

è risolubile tramite un metodo B&B.

3.2.1 Rilassamento continuo I

Per il rilassamento continuo, consideriamo il problema di partenza, senza il vincolo sulle x di essere un valore binario, e il corrispondente problema duale:

$$\begin{aligned} \max\ & c_1x_1 + \ldots + c_nx_n \\ & a_1x_1 + \ldots + a_nx_n \leq b \\ & x_i \geq 0 \end{aligned} \qquad\qquad \begin{aligned} \min\ & y \cdot b \\ & y \geq \tfrac{c_i}{a_i} \qquad i = 1, \ldots, p \\ & y \geq 0 \end{aligned}$$

La soluzione ottima del duale risulta:

$$\bar{y} = \max\left\{\frac{c_i}{a_i}, i = 1 \ldots n\right\} = \frac{c_k}{a_k}$$

da cui $\bar{x}$ con

$$\bar{x_i} = \begin{cases} \frac{b}{a_i} & \text{se } i = k \\ 0 & \text{altrimenti} \end{cases}$$

è la soluzione ottima del rilassamento continuo.
Detto $\frac{c_i}{a_i}$ beneficio unitario (rendimento) dell'oggetto i-esimo, definiamo la **tecnica euristica dei rendimenti decrescenti**: *riordinando le variabili secondo i benefici unitari, in ordine descrescente, inseriremo un oggetto nello zaino sse il suo peso è non maggiore della capacità residua.* Questa sarà la nostra soluzione di partenza nel problema dello zaino.
Supponiamo quindi un riordinamento tra le variabili:

$$\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$$

Il primo oggetto viene messo nello zaino[8], mentre l'inserimento degli altri si valuta caso per caso, fino a quando non è più possibile (zaino pieno o i pesi sono tutti maggiori della capacità residua).

$$\hat{x} = 1, \qquad \hat{x}_j = 1 \Leftrightarrow a_j \leq b - \sum_{i=1}^{j-1} a_i \hat{x}_i, \qquad \text{per } j = 2, \ldots, n$$

3.2.2 Rilassamento continuo II

Un secondo esempio di possibile rilassamento continuo si può ottenere ammettendo soluzioni con valori frazionari, ovvero:

$$\begin{gathered} \max c_1 x_1 + \ldots + c_n x_n \\ a_1 x_1 + \ldots + a_n x_n \leq b \\ 0 \leq x_i \leq 1 \end{gathered}$$

In questo caso, riutilizzando il riordinamento $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$, definisco $h \in \{1, \ldots, n-1\}$ t.c.

$$\sum_{i=1}^{h} a_i \leq b \text{ e } \sum_{i=1}^{h+1} a_i > b$$

ovvero h è l'ultimo oggetto inseribile nello zaino. La soluzione ottima allora si ottiene da:

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

Ovviamente si ha che $a_1 \bar{x_1} + \ldots + a_n \bar{x_n} = b$.

3.2.3 Esempio di esercizio

Considero il seguente problema dello zaino:

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

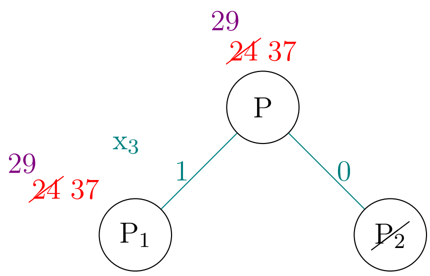
Calcoliamo i rendimenti per poter applicare la tecnica euristica dei rendimenti decrescenti e riordiniamo le variabili. A questo punto calcoliamo la soluzione ammissibile di partenza, ottenuta seguendo il riordinamento e **a componenti intere**. Essendo questo un problema di massimo, la soluzione sarà la valutazione inferiore. Calcoliamo il rilassamento come indicato nel

[8]Siamo sotto l'ipotesi che i singoli oggetti abbiano $c_i < b$

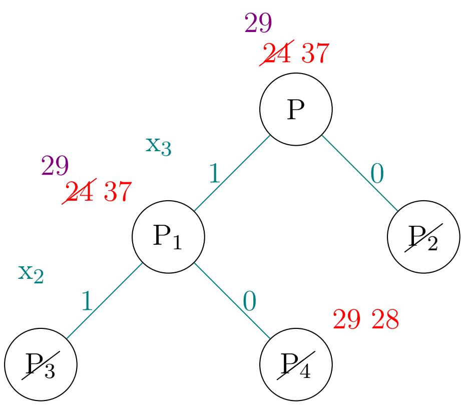
punto **3.2.2** che sarà la valutazione superiore. A questo punto, si esplora l'albero di enumerazione seguendo l'ordine dell'ordinamento (spesso si ramifica a partire dalla variabile frazionaria)

rendimenti: 11/7 23/6 6 3
ordinamento: x_3 x_2 x_4 x_1
sol.ne ammissibile: $(0, 0, 1, 1)$, $V_I = 24$
sol.ne rilassamento: $(0, 5/6, 1, 0)$, $V_S = 37 + 1/6$

A questo punto calcolo le soluzioni dei rilassamenti nei sottoproblemi e applicando le regole di potatura esploro l'albero e poto quando possibile. Quando il valore non è intero, si prende la parte intera inferiore del valore. $\lfloor V \rfloor = \lfloor (37 + 1/6) \rfloor = 37$.



P1 $(x_3 = 1)$**:** sol. rilassamento $(0, 5/6, 1, 0)$, $\lfloor V \rfloor = 37$. Non posso potare.
P2 $(x_3 = 0)$**:** sol. rilassamento $(0, 1, 0, 1)$, $V = 29$, posso potare perché la soluzione è ammissibile. Inoltre il valore è migliore della soluzione corrente, quindi aggiorno la soluzione. $V_I = 29$. Aggiorno l'albero di enumerazione. Passo a x_2.



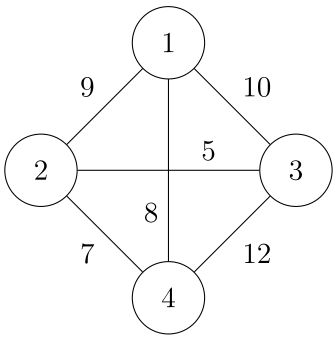
P3 $(x_3 = 1, x_2 = 1)$**:** non contiene soluzioni ammissibili per P (ottengo un peso di 9), posso potare.
P4 $(x_3 = 1, x_2 = 0)$**:** sol. rilassamento $(3/7, 0, 1, 1)$, $V = 24 + 11 \cdot 3/7 =$

$28 + 5/7$, posso potare in quanto $\lfloor V \rfloor = V_S = 28 < V_I = 29$.
L'albero è totalmente potato, la soluzione corrente $(0, 1, 0, 1)$ è ottima.

3.3 PB commesso viaggiatore (TSP = Traveling Salesman Problem)

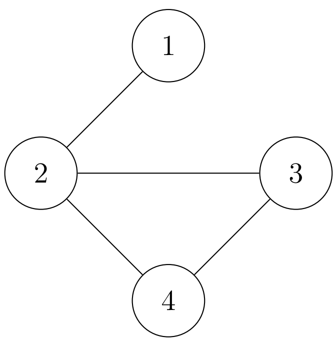
Un commesso viaggiatore vuole visitare n città (tornando infine alla città di partenza) minimizzando il percorso complessivo.
Il secondo tipo di problema su cui si applicano i metodi del Branch and Bound, è il TSP. Definiamo il **ciclo Hamiltoniano** come una sequenza di archi che inizia e finisce in uno stesso nodo, passando negli altri nodi solo una volta.
Si può quindi astrarre matematicament il TSP con il seguente problema:

> Dato un grafo non orientato con un costo (*distanza* tra i nodi incidenti $i, j : d_{ij}$) su ciascun arco, trovare il ciclo hamiltoniano di costo minimo.

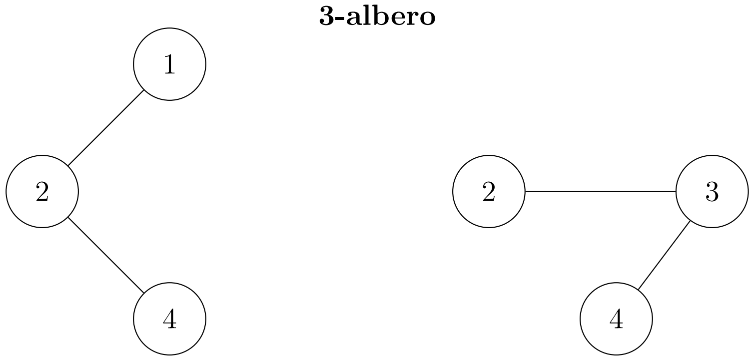


Esempio di grafo

Proprietà *Scelto un nodo k, un ciclo Hamiltoniano individua un albero di copertura per il grafo privato di k e di tutti gli archi incidenti in k.*
Inoltre si parla di **grado di k** come il numero di archi incidenti nel nodo k. Si parla poi di **k-albero**: *albero di copertura per il sottografo ottenuto rimuovendo k e gli archi incidenti in k, reinserendo poi k con 2 archi incidenti in k.*
Esempio utilizzando il grafo di prima:



3-albero



Un ciclo Hamiltoninao è un k-albero, ma non tutti i k-alberi sono cicli Hamiltoniani, come si vede nell'esempio sopra.
Prop: Fissato $\forall k \in N$:

$$\text{ciclo Hamilstoniano} \equiv \text{k- albero in cui ogni nodo ha grado 2}$$

Il k-albero di costo minimo è un rilassamento di TSP. Ricapitolando:

$$x_{ij} = \begin{cases} 1 & \text{se } (i,j) \in A \text{ appartiene al ciclo} \\ 0 & \text{altrimenti} \end{cases}$$

$$\min \sum_{a \in A_i} c_a x_a \qquad \text{ove } A_i = \{\text{archi incidenti in i}\}$$

$$\sum_{a \in A_i} x_a = 2 \qquad i \in N$$

$$\sum_{a \in A(N',N'')} x_a \geq 2 \quad \forall\ (N', N'')\ \textit{taglio}$$

3.3.1 Tecnica Euristica del nodo più vicino

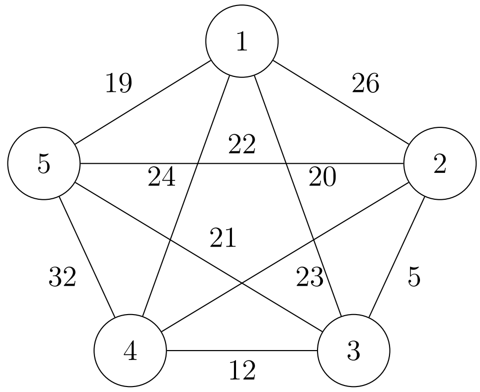
Tecnica utile per trovare la soluzione iniziale in un problema TSP. Partendo da un nodo k, si procede aggiungendo via via tutti i nodi non ancora nel grafo, di costo minimo. In formule, se

$$V = \{i_1, i_2, \ldots, i_t\} \text{ con } i_j \text{ tutti distinti}$$

individua l'insieme dei nodi già visitati,

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

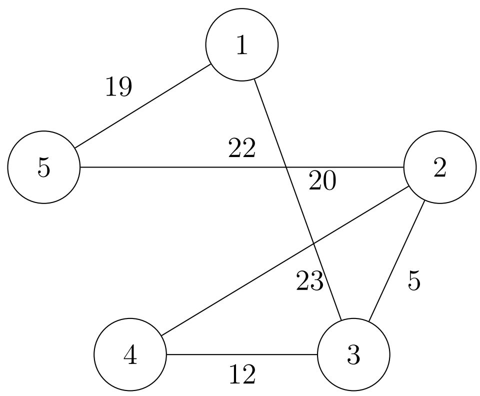
3.3.2 Esempio di esercizio



Rilassamento: 3-albero di costo minimo
Sol.ne iniziale: nodo più vicino a partire da **2**
Ramificazione: arco di costo minimo

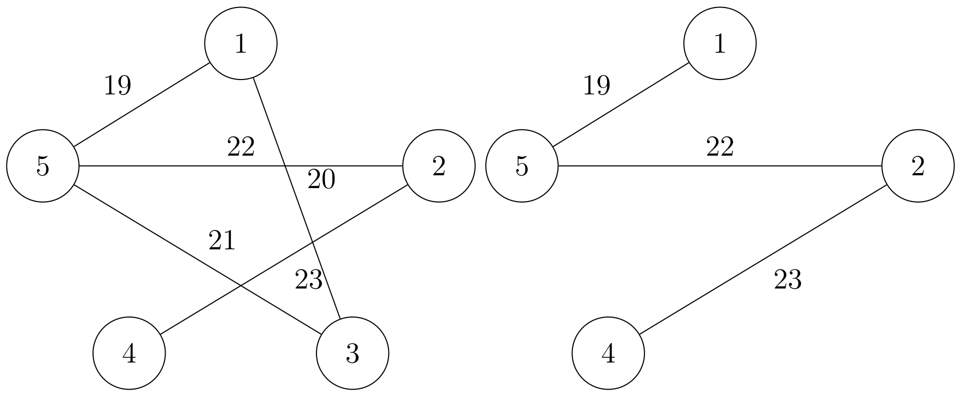
Si individua la soluzione utilizzando la tecnica euristica partendo da 2.

1. parto da due e prendo l'arco (2,5)
2. da 5 prendo l'arco (5,1)
3. da 1, il secondo di costo minimo è (1,3)
4. da 3, si prende (3,4)
5. da 4, si chiude il ciclo



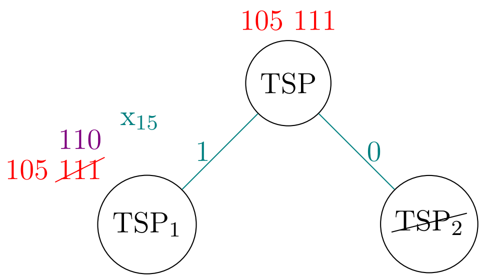
$V_S = 111$

A questo punto, si fa il 3-albero di costo minimo.

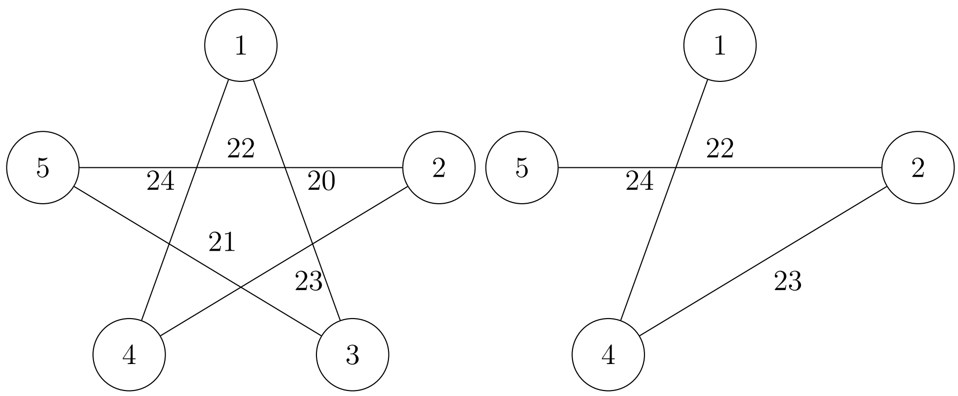


3-albero di costo minimo; $\mathbf{V}_I = \mathbf{105}$ (l'albero di copertura senza il nodo 3)

Si fa il branch and bound:

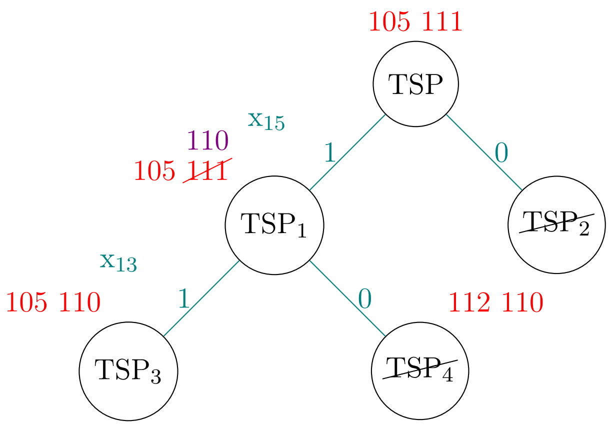


TSP$_1$: Analizzando il problema , si ottiene lo stesso 3 albero di prima. Non posso chiudere il branch.
TSP$_2$: Cambia il 2 albero:

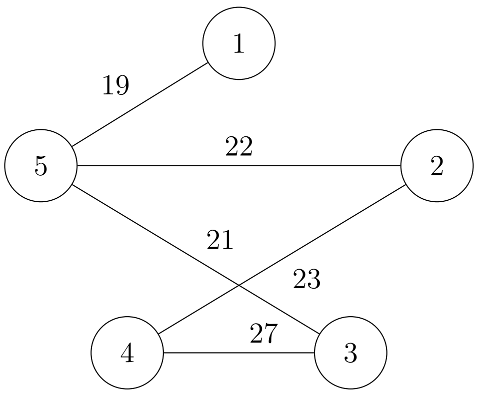


3-albero di costo minimo TSP$_2$

Il costo risulta essere $V_I = 110$. L'albero è un ciclo Hamiltoniano, la prendiamo come nuova soluzione ottima con $V_S = 110$. Si chiude il branch.
Si continua quindi nel nodo ancora aperto, analizzando il ramo $x_1 3$.

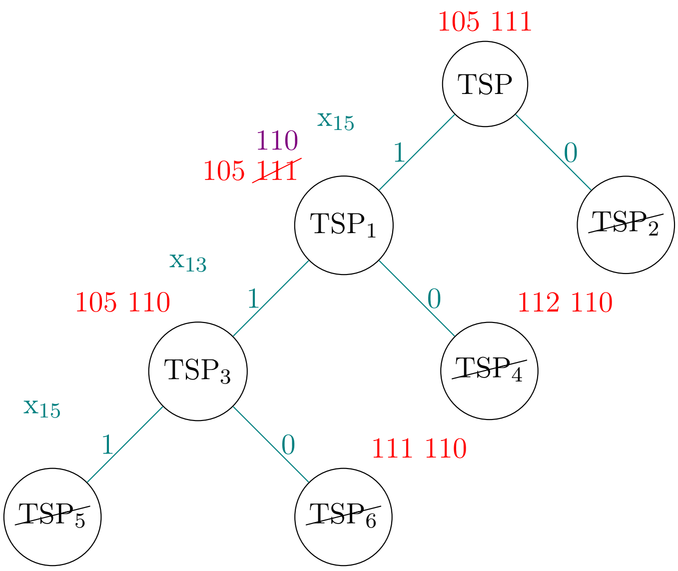


$\mathbf{TSP}_3$: il 3-albero è lo stesso della radice. Non si può chiudere.
$\mathbf{TSP}_4$: Studio il 3-albero di costo minimo. Avendo ottenuto però un valore $V_I = 112$, maggiore del V_S corrente, si chiude.

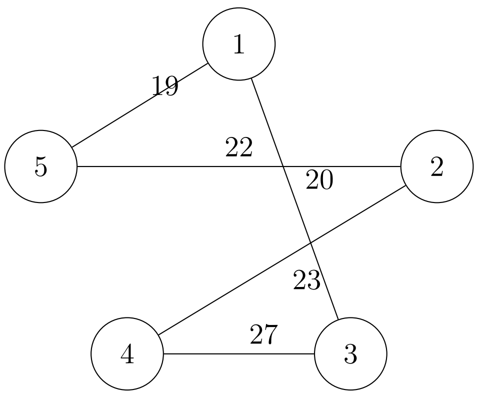


3-albero di costo minimo TSP_4, $V_I = 112$

Si analizza l'arco x_{35}, e si chiudono i restanti branch.



TSP_5: non ha cicli Hamiltoniniani, quindi si chiude per inammissibilità. **TSP_6:** il 3-albero è un ciclo Hamiltoniano. Il costo però risulta essere V_I = 111, che è maggiore del V_S corrente. Si chiude per ammissibilità e si mantiene la souzione corrente.



3-albero di costo minimo TSP_4, $V_I = 111$. È un ciclo Hamiltoniano

Avendo chiudo tutti i branch, l'algoritmo finisce. La coluzione corrente è il ciclo Hamiltoniano di costo minimo (c= 110).