Programmazione Lineare Intera

1 Introduzione

Come si è visto in precedenza, la programmazione lineare è uno strumento estremamente potente per modellare numerosi problemi decisionali che si presentano nel campo della produzione, allocazione di risorse, trasporto, assegnamento e molti altri. Sfortunatamente queste numerose applicazioni non sono sufficienti a coprire la maggior parte dei problemi di pianificazione del mondo reale, ed in particolar modo restano esclusi proprio i problemi più difficili da risolvere. Un vincolo presente in molti problemi reali è infatti quello dell'interezza di alcune variabili decisionali. È ovvio che questo vincolo non può essere imposto solo con l'uso di disuguaglianze lineari, non importa quante (perchè ?). Il vincolo di interezza è fondamentale e, come vedremo, non può essere trascurato nella speranza che una soluzione frazionaria arrotondata all'intero sia sempre ottima o comunque buona. Si consideri ad esempio la produzione annua di un determinato modello di automobili. In questo caso, per quanto non abbia senso produrre mezza automobile, una soluzione ottima al problema senza vincoli di interezza che richieda di produrre 324623.5 auto potrà verosimilmente essere arrotondata a 324624 senza un grande degrado nel valore della soluzione. Ben diversa è la situazione di un problema in cui si decida, insieme ad altre cose, dove costruire dei punti vendita di una catena commerciale, al costo di svariati miliardi l'uno. La soluzione ottima senza vincoli di interezza potrebbe suggerire di costruire 3/4 di un negozio in una città C_1 e l'altro quarto nella città C_2. Ciò è chiaramente impossibile. È allora vantaggioso costruirlo in C_1 o in C_2? Dipende dai dati (costi e/o vincoli. Potrebbe perfino accadere che per una soluzione intera la soluzione ottima localizzi il negozio in questione in una terza città).

1.1 Tipi di PLI

Il modello più generale di problema di programmazione lineare intera (PLI) consiste nell'ottimizzare una funzione obiettivo lineare, in presenza di vincoli lineari, e richiedendo che k delle n variabili decisionali possano assumere soltanto valori interi. Si denoti con $(\mathbf{x}, \mathbf{y})'$ una soluzione del problema, in cui $\mathbf{x} = (x_1, \ldots, x_k)' \in Z^k$, è il vettore delle variabili che devono risultare intere e $\mathbf{y} = (y_1, \ldots, y_{n-k})' \in R^{n-k}$ è il vettore delle variabili che possono assumere valori frazionari. Sia A una matrice $m \times k$ e B una matrice $m \times (n-k)$, $\mathbf{b}$ un vettore a m componenti, e $\mathbf{c}$ un vettore a n componenti. $\mathbf{c}$ è il vettore dei coefficienti di costo, e sia ripartito come $(\mathbf{c}_x, \mathbf{c}_y)'$, dove $\mathbf{c}_x = (c_1, \ldots, c_k)'$ e $\mathbf{c}_y = (c_{k+1}, \ldots, c_n)'$. Il problema di *Programmazione Lineare Intera Mista* (PLIM), si definisce come:

PLIM: $\min \mathbf{c}'_x\mathbf{x} + \mathbf{c}'_y\mathbf{y}$

(M1) $A\mathbf{x} + B\mathbf{y} \geq \mathbf{b}$
(M2) $\mathbf{x} \in Z^k$
(M3) $\mathbf{y} \in R^{n-k}$
(M4) $\mathbf{x}, \mathbf{y} \geq \mathbf{0}$

Esempio 1: Si consideri il caso di una compagnia di trasporti con vetture di diverso tipo $V_1, \ldots, V_p$, che richiedono diverse quantità di risorse materiali (carburante, olio...), nonchè umane (autistii). Una soluzione potrà decidere di assegnare 50.73 litri di benzina ai mezzi di tipo V_1, ma certamente non 4.5 autisti!

Un caso particolare di programmazione lineare intera mista si ha quando tutte le variabili sono vincolate ad essere intere (ossia $k = n$ e il vettore $\mathbf{y}$ nel modello precedente scompare). Si parla in questo caso di *Programmazione Lineare Intera Pura* (PLIP):

PLIP: $\min \mathbf{c}'\mathbf{x}$

(P1) $A\mathbf{x} \geq \mathbf{b}$
(P2) $\mathbf{x} \in Z^n$
(P3) $\mathbf{x} \geq \mathbf{0}$

Esempio 2: Si consideri un problema aziendale in cui di un prodotto esistano vari (n) modelli e le variabili di decisione rappresentino il numero di esemplari da produrre per ciascun modello.

Abbiamo infine un caso particolare di programmazione lineare intera, nel quale le variabili intere sono vincolate ad assumere soltanto i valori 0 o 1. Parliamo in questo caso di *Programmazione Lineare 0-1*, o di *Programmazione Lineare Intera Binaria* (PLIB). Nel caso puro, il problema diventa:

PLIB: $\min \mathbf{c}'\mathbf{x}$

(B1) $A\mathbf{x} \geq \mathbf{b}$
(B2) $\mathbf{x} \in \{0,1\}^n$

Esempio 3: Tipicamente, le variabili binarie vengono usate per rappresentare decisioni di intraprendere oppure no una certa azione, come "produrre il prodotto i", o "accendere il generatore j". Un esempio di PLIB è dato dal problema di decidere a quali incroci piazzare dei poliziotti, in modo tale che tutte le strade siano pattugliate. Un poliziotto controlla tutte le strade che si intersecano ad un dato incrocio i. La variabile x_i verrà usata per decidere se piazzare un poliziotto all'incrocio i (nel qual caso $x_i = 1$) oppure no ($x_i = 0$). La funzione obiettivo potrebbe essere minimizzare il numero di poliziotti utilizzati (ossia $\sum_{i=1}^{n} x_i$).

Come nel caso della PL, anche per la programmazione intera esistono diverse formulazioni, in cui le disuguaglianze possono essere invertite e/o sostituite con uguaglianze. Per esempio nell' ipotesi che A e $\mathbf{b}$ siano a valori interi, si possono aggiungere variabili $\mathbf{s}$ slack/surplus ai vincoli di disuguaglianza, in modo da trasformarli in equazioni, e i valori di $\mathbf{s}$ saranno anch'essi interi. Perciò ad esempio il problema $PLIP$ può essere trasformato in

$\min \mathbf{c}'\mathbf{x}^*$

(P1) $A^*\mathbf{x}^* = \mathbf{b}$
(P2) $\mathbf{x}^* \in Z^{n^*}$
(P3) $\mathbf{x}^* \geq \mathbf{0}$

dove $A^* = (A \ -I)$, $\mathbf{x}^* = (\mathbf{x}, \mathbf{s})$ e $n^* = n + m$.

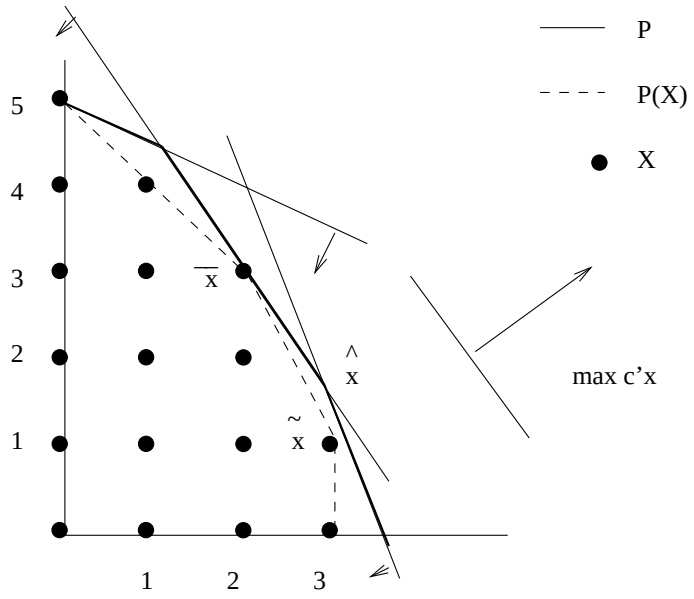


Figure 1: Visualizzazione geometrica.

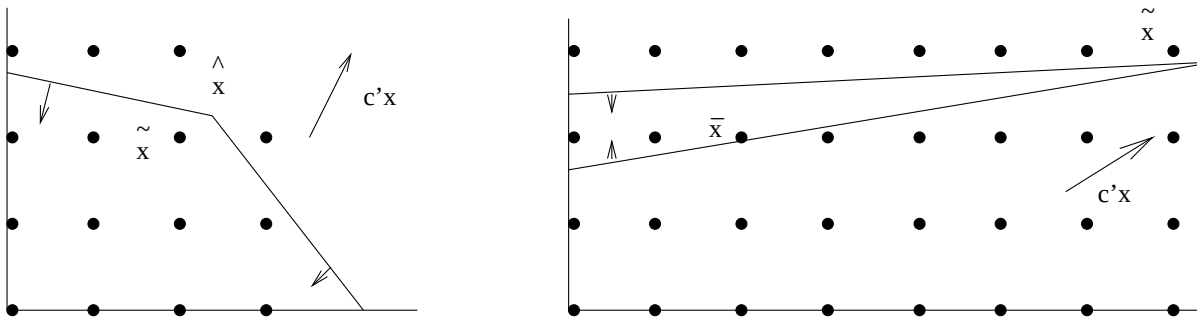
2 Geometria della PLI

2.1 Introduzione

In questa sezione discuteremo alcune proprietà geometriche della programmazione lineare intera, e in particolare le relazioni tra l'insieme delle soluzioni intere e quello delle soluzioni al problema in cui il vincolo di interezza viene tralasciato (*rilassamento continuo*). Il materiale viene qui trattato in modo puramente introduttivo, ma le idee illustrate sono di fondamentale importanza per capire il funzionamento dei metodi risolutivi descritti nelle sezioni successive. Inoltre esiste tutta una branca della ricerca operativa, nota come *Combinatorica poliedrale* (polyhedral combinatorics) che studia con successo la risoluzione di complessi problemi combinatorici tramite le proprietà geometriche dei poliedri associati agli insiemi di soluzioni. Nel seguito faremo principalmente riferimento al problema PLIP, vale a dire un problema di minimizzazione, con vincoli in forma canonica. Il lettore è invitato a riformulare i concetti qui espressi per adattarli al caso di un problema di massimizzazione e/o vincoli in forma standard.

Dato un problema di PLIP, si denoterà con P l'insieme delle soluzioni al rilassamento continuo, cioè $P = \{\mathbf{x} : A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, e con $\hat{\mathbf{x}}$ la soluzione ottima al problema di PL associato a P, ossia $[\min \mathbf{c}'\mathbf{x}, \mathbf{x} \in P]$. Più in generale, qualora ci si debba riferire ad un poliedro contenuto in P e ottenuto aggiungendo vincoli a P, esso sarà denotato da P^i, e la soluzione ottima del corrispondente PL sarà $\hat{\mathbf{x}}^i$. Per convenzione, poniamo anche $P^0 := P$.

L'insieme delle soluzioni intere sarà denotato con $X = P \cap Z^n$ (rispettivamente, $X^i = P^i \cap Z^n$). La soluzione ottima in X sarà denotata da $\bar{\mathbf{x}}$ (e, rispettivamente, $\bar{\mathbf{x}}^i$ sarà l'ottimo in X^i). Si denoterà infine con $P(X)$ il più piccolo poliedro contenente X, detto anche l'*involucro convesso* di X (analogamente si definisce $P(X^i)$). Gli insiemi P, X e $P(X)$ sono rappresentati in figura 1 nel caso del seguente problema a due dimensioni:



(a) (b)

Figure 2: Diversi casi di arrotondamento

$$\begin{array}{lrcrcl} \max & 20x_1 & + & 15x_2 & & \\ & & & & & \\ & 6x_1 & + & 4x_2 & \leq & 24 \\ & x_1 & + & 2x_2 & \leq & 10 \\ & 5x_1 & + & 2x_2 & \leq & 18 \end{array}$$

$\mathbf{x} \in Z_+^2$

2.2 Come risolvere un PLI

Per risolvere un problema a variabili intere, la soluzione più ovvia sembrerebbe quella di ignorare i vincoli di interezza e calcolare l'ottimo $\hat{\mathbf{x}}$ del rilassamento continuo. Se $\hat{\mathbf{x}} \in X$, cioè se tutte le componenti di $\hat{\mathbf{x}}$ sono intere, allora $\hat{\mathbf{x}}$ è anche l'ottimo cercato. Altrimenti, si potrebbe dire, basterà arrotondare le componenti non intere al valore intero, ottenendo la soluzione $\tilde{\mathbf{x}} \in X$. Con riferimento alla figura 1, $\hat{\mathbf{x}} = (3, \frac{3}{2})$. Ora, un possibile arrotondamento ci da $\tilde{\mathbf{x}} = (3, 1) \in X$. Tuttavia, si potrebbe anche arrotondare $\hat{\mathbf{x}}$ a $\tilde{\mathbf{x}} = (3, 2)$, punto che non è in X. Da questo esempio si evince che l'arrotondamento può portare in certi casi, a soluzioni non ammissibili. Come si vedrà nella sezione 4.1, la soluzione ottima al problema dell'esempio è $\bar{\mathbf{x}} = (2, 3)$. Ecco quindi che in questo problema, anche arrotondando a $(3, 1)$ si otterrebbe una soluzione ammissibile ma non ottima. In generale, arrotondando una soluzione frazionaria $\hat{\mathbf{x}}$ a una soluzione intera $\tilde{\mathbf{x}}$ si verifica pertanto una tra le seguenti possibilità :

1. La soluzione $\tilde{\mathbf{x}}$ è **ottima**. Tale situazione è descritta graficamente in figura 2(a)
2. La soluzione $\tilde{\mathbf{x}}$ è **ammissibile** ma **non ottima**. Si riconsideri l'esempio di figura 1.
3. La soluzione $\tilde{\mathbf{x}}$ è **non ammissibile**. Si veda ad esempio la figura 2(b)

Nelle figure forniamo rappresentazioni geometriche delle varie situazioni. Il lettore è invitato a ricavare i coefficienti di opportuni vincoli che diano luogo al verificarsi delle tre possibilità. Si noti anche che comparando il valore della soluzione arrotondata $\mathbf{c}'\tilde{\mathbf{x}}$ con quello della soluzione ottima $\mathbf{c}'\bar{\mathbf{x}}$, può accadere che i due siano abbastanza vicini (come nel caso delle figure 1 e 2(a)), o estremamente lontani (come avviene in figura 2(b)).

In virtù di quanto appena visto, si conclude che il metodo dell'arrotondamento non garantisce sempre la soluzione ottima, e pertanto è necessario un maggiore sforzo computazionale che non la semplice risoluzione di un PL. Risolvere un PL sarebbe però sufficiente se anzichè P avessimo

a disposizione i vincoli che definiscono $P(X)$. Siano ext(P) e ext$(P(X))$ gli insiemi dei vertici di P e $P(X)$ rispettivamente. Chiaramente, tutti i punti in ext$(P(X))$ sono interi per come $P(X)$ è definito. In generale invece, vi saranno vertici di P che non sono interi. L'importanza dei vertici di $P(X)$ è chiara qualora si consideri che

- la soluzione ottima sarà sempre ottenuta ad un vertice di $P(X)$
- avendo una rappresentazione esplicita dei vincoli che definiscono $P(X)$, si potrebbe usare la PL per trovare il vertice ottimo di $P(X)$

Purtroppo avviene che ricavare i vincoli che definiscono $P(X)$ esplicitamente è in generale estremamente difficile. Ciò che si può invece fare è utilizzare $P^0 := P$ come una prima approssimazione di $P(X)$, e via via migliorarla, ottenendo $P^1, P^2, \ldots, P(X)$. P^i può essere ottenuto da P^{i-1} con l'aggiunta di *disuguaglianze valide*. Una disuguaglianza $\mathbf{a}'\mathbf{x} \geq a_0$ è **valida** per X se $\mathbf{a}'\mathbf{x} \geq a_0$ per ogni $\mathbf{x} \in X$. Le disuguaglianze valide sono utili qualora l'inclusione dei poliedri sia propria, ossia venga *tagliata* parte di P^{i-1}. In particolare, nel metodo dei piani di taglio descritto in 4.3, $P^i \subset P^{i-1}$ e $\hat{\mathbf{x}}^i \neq \hat{\mathbf{x}}^{i-1}$. Le disuguaglianze valide più utili sono quelle che "toccano" $P(X)$, ossia per le quali l'insieme $F = \{\mathbf{x} : \mathbf{a}'\mathbf{x} \geq a_0\} \cap P(X)$ non è vuoto. F è detto una *faccia* di $P(X)$ ed è anch'esso un poliedro (ad esempio, si pensi ad uno spigolo, o un vertice, di un parallelepipedo). Se la dimensione di F è di uno inferiore a quella di $P(X)$, F si dirà una *faccetta* (facet) di $P(X)$ (ad esempio, ognuno dei sei rettangoli che delimitano un parallelepipedo). L'importanza delle faccette è data dal fatto che la rappresentazione esplicita di $P(X)$ è fornita dall'elenco dei vincoli che definiscono le sue faccette.

2.3 Problemi a soluzioni naturalmente intere

Supponiamo che tutti i coefficienti di A e $\mathbf{b}$ siano interi (questo si può fare senza perdita di generalità, in quanto, qualora non lo fossero, basterà moltiplicare ogni disuguaglianza per il minimo comune multiplo dei denominatori). Detta B la matrice di base associata alla soluzione $\hat{\mathbf{x}}$ ottima del rilassamento continuo, una condizione sufficiente affinchè $\hat{\mathbf{x}} = (B^{-1}\mathbf{b}, \mathbf{0})'$ sia intero è che tutte le componenti di B^{-1} siano intere.

Definizione. Una matrice A $m \times n$ si dice *totalmente unimodulare* se ogni sottomatrice quadrata Q di A ha determinante pari a 0, 1 o -1.

Si noti che dalla definizione discende che tutti gli elementi di A appartengono all'insieme $\{0, 1, -1\}$, in quanto sono anch'essi sottomatrici quadrate 1×1.

Teorema. Se $\mathbf{b}$ ha tutte le componenti intere, e A è totalmente unimodulare, allora $P(X) = P$, ossia tutti i vertici di P sono interi.

Dim. Sia $\mathbf{x}^*$ un vertice di P. Esiste allora una sottomatrice $m \times m$ di base B tale che $\mathbf{x}^* = (B^{-1}\mathbf{b}, \mathbf{0})$. Ma $B^{-1} = \frac{1}{\det(B)} \begin{pmatrix} \beta_{11} & \cdots & \beta_{1m} \\ & \cdots & \\ \beta_{m1} & \cdots & \beta_{mm} \end{pmatrix}'$, dove $\beta_{ij} = (-1)^{i+j}\det(\bar{B}_{ij})$ e $\bar{B}_{ij}$ è la matrice ottenuta da B eliminandone la riga i e la colonna j. Siccome B è nonsingolare e A è totalmente unimodulare, $|\frac{1}{\det(B)}| = 1$. Inoltre, siccome $\det(\bar{B}_{ij}) \in \{0, 1, -1\}$ per ogni i e j, si ottiene che le componenti di B^{-1} sono tutte a valori in $\{0, 1, -1\}$. Infine, è immediato dedurre che anche $B^{-1}\mathbf{b}$ risulterà intero. (QED)

Diventa pertanto importante caratterizzare le matrici totalmente unimodulari. A questo proposito, il seguente teorema fornisce una condizione sufficiente.

Teorema. Se $A \in \{0, 1, -1\}^{m \times n}$ è una matrice per cui

1. ogni colonna contiene al più due elementi nonnulli
2. esiste una ripartizione delle righe in due insiemi R_1 e R_2 tali che ogni colonna con due elementi non nulli li ha
   - entrambi in R_1 o in R_2 se di segno diverso
   - uno in R_1 e l'altro in R_2 se dello stesso segno,

allora A è totalmente unimodulare.

La dimostrazione di questo teorema è lasciata per esercizio, e può essere fatta per induzione. Tra i casi importanti delle matrici totalmente unimodulari troviamo le matrici di incidenza nodi/archi dei grafi orientati, e quelle dei grafi bipartiti non orientati. Tali matrici verranno definite più avanti nel corso. Per il momento basti dire che la loro unimodularità fa sì che i problemi di flusso su reti e del trasporto siano risolvibili in tempo polinomiale tramite la programmazione lineare.

3 Modellazione

La scelta di un buon modello per un problema di PLI può rivelarsi vitale, e fare la differenza tra il riuscire a risolvere il problema o meno. Disegnare buoni modelli è tanto una scienza quanto un'arte e richiede notevoli doti di inventiva e una grande esperienza. In particolare, è di grande importanza analizzare approfonditamente quanti più numerosi modelli possibile, mettendone in evidenza i punti deboli e i pregi. Inoltre, esistono numerosi casi standard per trattare diverse situazioni modellate con variabili intere e/o binarie. In questa sezione cercheremo di evidenziare le principali fra queste tecniche con alcuni esempi.

3.1 Limiti inferiori/superiori

In molti problemi, le variabili in una soluzione, qualora non nulle, dovranno avere un valore contenuto in un predeterminato intervallo. Ad esempio, sia x una variabile rappresentante la quantità di pezzi da produrre. Nei casi reali, attivare una produzione comporta dei costi fissi (vedi oltre). Esisterà pertanto un numero minimo N_m di pezzi sotto il quale non è redditizio neppure cominciare la produzione. Sia N_M il numero massimo di pezzi che si intende produrre. La variabile x assumerà pertanto valori nell' insieme $X = \{0\} \cup [N_m, N_M]$. Si noti che X non è convesso. Il vincolo $x \in X$ si può forzare introducendo una variabile binaria y, con il seguente significato:

$$y = \begin{cases} 1 & \text{se } x \in [N_m, N_M] \\ 0 & \text{se } x = 0 \end{cases}$$

e imponendo i vincoli $\begin{array}{ll} (1) & x \geq N_m y \\ (2) & x \leq N_M y \end{array}$

Si noti che $y = 1$ forza x nell'intervallo $[N_m, N_M]$, mentre se $y = 0$, a causa del vincolo (2), si avrà anche $x = 0$.

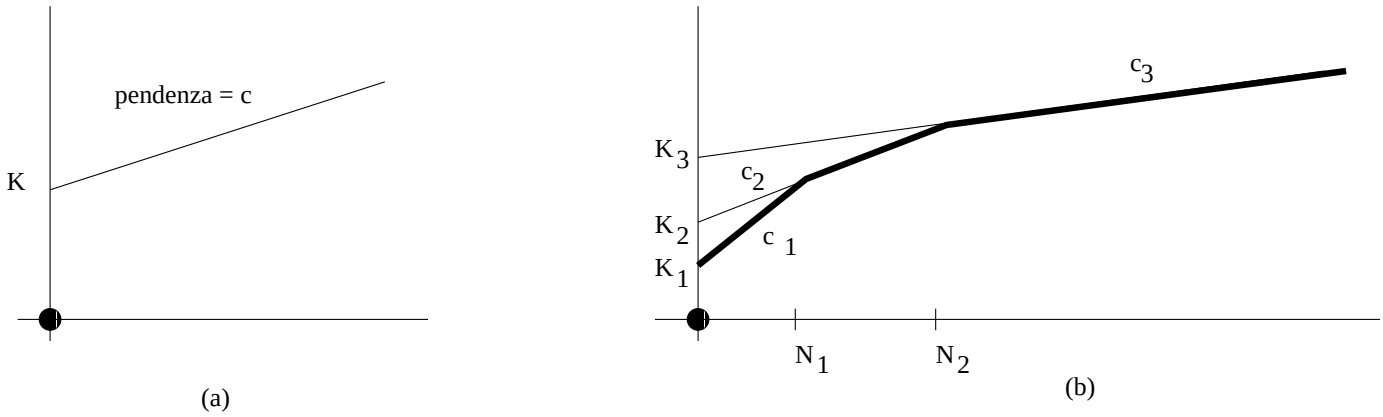


Figure 3: Generalizzazioni di funzioni di costo lineari.

3.2 Costi fissi di attivazione

In molte situazioni reali, il costo associato al produrre una quantita x di un prodotto, non è lineare, ma *affine*, ovvero ottenuto traslando un costo lineare. Infatti, per produrre una quantità anche infinitesimale di x, si incorre in un costo fisso K per attivare la produzione. Non produrre alcunchè, invece, ha ovviamente costo 0. La funzione di costo $f(x)$ è pertanto del tipo (vedi figura 3.a)

$$f(x) = \begin{cases} K + cx & \text{se } x > 0 \\ 0 & \text{se } x = 0 \end{cases}$$

Per rappresentare questo tipo di funzioni obiettivo, si introduce una variabile binaria y, con il significato:

$$y = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x = 0 \end{cases}$$

Nella riga del costo da minimizzare, si avrà allora

$\min Ky + cx + \ldots$, mentre nei vincoli bisognerà imporre $x \leq N_M y$, dove N_M è come prima un limite superiore ai valori ammissibili per x.

3.3 Economie di scala

Tipicamente, nella produzione industriale, più unità si producono di un certo prodotto, minore sarà il costo unitario (fenomeno noto come *economia di scala*). Esisteranno perciò valori $N_0 = 0, N_1, N_2, \ldots, N_n$ tali che la produzione di un numero $x \in (N_{i-1}, N_i]$ di esemplari avrà un costo c_i unitario tale che $c_1 \geq c_2 \geq \ldots \geq c_n \geq 0$ (si veda la figura 3.b). Sia K_1 il costo fisso per attivare la produzione. Per $i = 2, \ldots n$, si definisca $K_i = K_{i-1} + (c_{i-1} - c_i)N_i$. La funzione di costo sarà del tipo

$$f(u) = \begin{cases} 0 & \text{se} & & u & = 0 \\ K_1 + c_1 u & \text{se} & 0 < & u & \leq N_1 \\ K_2 + c_2 u & \text{se} & N_1 < & u & \leq N_2 \\ \cdots & & & & \\ K_n + c_n u & \text{se} & N_{n-1} < & u & \leq N_n \end{cases}$$

Questa situazione si può modellare con l'introduzione di n variabili reali x_i. Vogliamo che x_i sia diverso da 0 al più per un indice j, rappresentando il caso in cui $x_j = u \in (N_{j-1}, N_j]$. Il caso $x_i = 0$ per $i = 1, \ldots, n$ corrisponde invece a $u = 0$. Per forzare questa relazione tra le x_i e u, si introducono anche n variabili binarie y_i, con il significato

$$y_i = \begin{cases} 1 & \text{se } N_{i-1} < u \leq N_i \\ 0 & \text{altrimenti} \end{cases}$$

I vincoli che mettono in relazione $\mathbf{y}$ e $\mathbf{x}$ saranno

(1) $x_i \leq N_i y_i$ per $i = 1, \ldots, n$

(2) $y_1 + y_2 + \cdots y_n \leq 1$

mentre nella funzione obiettivo comparirà (assieme agli eventuali costi di altre variabili) l'espressione

$$\min K_1 y_1 + \cdots + K_n y_n + c_1 x_1 + \cdots c_n x_n + \cdots$$

Si noti che il vincolo (2) potrebbe essere omesso, visto che in una soluzione *ottima* del problema, non avrà senso avere più di una y_i al valore 1, dato che ogni tale y_i comporta un costo K_i non negativo. Per la stessa ragione, vincoli del tipo $x_i \geq N_{i-1} y_i$ non sono stati introdotti nel modello, perchè implicitamente soddisfatti da ogni soluzione ottima.

3.4 Selezione da un insieme finito

Il vincolo (2) dell'esempio precedente, è un tipico caso in di uso di variabili binarie come *selettori* di al più una tra n possibilità. Analogamente, l'equazione $y_1 + y_2 + \cdots y_n = 1$ con $y_i \in \{0, 1\}$ per $i = 1, \ldots, n$ è soddisfatta quando *esattamente* una delle y_i vale 1 e tutte le altre 0. Tale vincolo rappresenta il verificarsi di una tra n possibilità alternative e incompatibili.

Ad esempio, supponiamo di dover risolvere un problema in cui una variabile x sia vincolata ad assumere un valore nell'insieme $\{a_1, \ldots, a_n\}$ dove gli a_i sono numeri reali. Tale situazione si può rappresentare sostituendo a x dappertutto (nei vincoli e nell'obiettivo) l'espressione $a_1 y_1 + \cdots + a_n y_n$, e imponendo $y_1 + y_2 + \cdots y_n = 1$, dove le y_i sono variabili binarie.

3.5 Unione di vincoli

Esistono alcuni problemi in cui si richiede alle variabili di soddisfare (assieme ai soliti $A\mathbf{x} \leq \mathbf{b}$) almeno uno tra due determinati vincoli, ma ma non necessariamente entrambi. Siano $\mathbf{a}'\mathbf{x} \leq b$ e $\mathbf{d}'\mathbf{x} \leq e$ i vincoli in questione. L'insieme dei vettori che soddisfano ad almeno uno fra i due vincoli è l'*unione* di due semispazi, ossia $\{\mathbf{x} : \mathbf{a}'\mathbf{x} \leq b\} \cup \{\mathbf{x} : \mathbf{d}'\mathbf{x} \leq e\}$. Pur essendo un' unione di poliedri, tale insieme non è necessariamente un poliedro, nè tantomeno convesso. Per modellare questa situazione, si introduce una costante M "sufficientemente grande", tale cioè che $\mathbf{a}'\mathbf{x} \leq M$ e $\mathbf{d}'\mathbf{x} \leq M$ per ogni $\mathbf{x} : A\mathbf{x} \leq \mathbf{b}$. M è una specie di "infinito", nel senso che la presenza di M a destra di una disuguaglianza $\leq$ deve rendere sempre tale disuguaglianza vera. Per motivi pratici, è bene scegliere un M quanto più piccolo possibile tra quelli che soddisfano questa proprietà . Il valore di M dipenderà dai dati dello specifico problema. Sia ora y una variabile binaria con il significato

$$y_i = \begin{cases} 1 & \text{se } \mathbf{a}'\mathbf{x} \leq b \\ 0 & \text{se } \mathbf{d}'\mathbf{x} \leq e \end{cases}$$

Si aggiungano ora ai vincoli $A\mathbf{x} \leq \mathbf{b}$ i nuovi vincoli

(1) $\mathbf{a}'\mathbf{x} \leq b + M(1 - y)$

(2) $\mathbf{d}'\mathbf{x} \leq e + My$

Si noti che per $y \in \{0,1\}$ esattamente uno fra i vincoli (1) e (2) viene imposto, mentre l'altro risulterà vero per ogni $\mathbf{x}$.

Esempio: vincoli del tipo l'uno o l'altro (esclusivi). In alcuni casi i vincoli $\mathbf{a}'\mathbf{x} \leq b$ e $\mathbf{d}'\mathbf{x} \leq e$ sono incompatibili, cioè l'intersezione dei due semispazi è vuota. In tali casi, il trucco di cui sopra servirà ad imporre che *esattamente* uno tra i due vincoli debba essere soddisfatto. Come esempio, si pensi ad un problema di schedulazione in cui n lavori, ciascuno con una sua durata p_i debbano essere elaborati su una stessa macchina. Una volta cominciato, un lavoro deve essere portato a termine, e pertanto impegna la macchina per un tempo p_i. Si denotino con t_i i tempi (ossia le variabili) da determinare, rappresentanti l'istante d'inizio dell'elaborazione di ogni lavoro i. Il problema richiede, per ogni coppia di lavori i e j di decidere se i debba venire eseguito prima di j o viceversa. Nel primo caso, dovrà essere $t_j \geq t_i + p_i$; nel secondo $t_i \geq t_j + p_j$. Si introducano pertanto $n(n-1)/2$ variabili binarie y_{ij} con l'interpretazione

$$y_{ij} = \begin{cases} 1 & \text{se } i \text{ viene eseguito prima di } j \\ 0 & \text{se } j \text{ viene eseguito prima di } i \end{cases}$$

Avremo allora per ogni coppia di lavori i e j i vincoli $t_i \leq t_j - p_i + M(1 - y_{ij})$ e $t_j \leq t_i - p_j + M y_{ij}$.

3.6 Relazioni booleane

Nella quasi totalità dei casi, l'uso di variabili binarie non è dettato dal fatto che i valori numerici siano vincolati ad essere 1 o 0, quanto dalla loro interpretazione come "verificarsi o meno di un evento". In questo senso, le variabili binarie sono di fatto delle *variabili booleane*, ossia possiamo interpretare il valore 1 come `VERO` e 0 come `FALSO` . In questa interpretazione, diventa importante essere in grado di legare fra loro variabili booleane tramite relazioni booleane, come `NOT` , `AND` e `OR` . È ovvio che `NOT` x non è altro che $(1 - x)$. Leggermente più complesse sono le relazioni `AND` e `OR` . Si voglia imporre la relazione $z = x$ `AND` y. Basterà introdurre i vincoli $z \leq x$, $z \leq y$ e $z \geq x + y - 1$. Per realizzare la relazione $z = x$ `OR` y, si dovranno invece imporre i vincoli $z \geq x$, $z \geq y$ e $z \leq x + y$.

3.7 Esercizi

1. **Costi lineari a tratti.** Si modelli una funzione di costo $f(x)$ che sia lineare a tratti (non necess. convessa o concava). In particolare, sia c_i, per $i = 1, \ldots, n$ il coefficiente angolare nel tratto $N_{i-1} \leq x \leq N_i$. Sia inoltre $K_0 = f(0)$.

2. **Unione di poliedri.** Si modelli il caso in cui l'insieme delle soluzioni ammissibili è dato dall'unione di k poliedri $A^i\mathbf{x} \geq \mathbf{b}^i$ (generalizza 3.5).

3. Si modellino le seguenti relazioni tra le variabili binarie (booleane) x, y e z. (i) $z =$ x `XOR` y, (ii) $x \to y$ e (iii) $z \to (x$ `AND` $y)$ (dove $x \to y$ significa "x implica y").

4. **Vincoli di aciclicità** . Nell'esempio di 3.5 mancano dei vincoli sulle variabili che impongano condizioni di consistenza della soluzione. In particolare, se $y_{ij}, y_{jk}, y_{ku}, \ldots, y_{vz}$ valgono 1, allora anche y_{iz} deve valere 1. Questi vincoli impongono che le coppie (i,j) per cui $y_{ij} = 1$ definiscano un ordine totale su $\{1, \ldots, n\}$. Si scrivano tali vincoli (ce ne sono $O(2^n)$). Si ricavi un modo per imporre tali vincoli con l'uso di solo $O(n^3)$ disuguaglianze.

5. È ovvio che un problema di programmazione binaria (pura) è un caso speciale di programmazione intera (pura). Si dimostri che anche l'opposto è vero, ossia che ogni problema intero può essere ricondotto a un opportuno equivalente problema binario.

3.8 Un problema di modellazione

In questa sezione studieremo le varie fasi della modellazione per un esempio di problema reale (per quanto semplificato, per ovvie esigenze didattiche).

Un problema quotidiano della compagnia elettrica ELEN è quello di decidere quali generatori accendere, a seconda delle richieste di energia. In particolare, ci siano tre generatori A, B e C con le seguenti caratteristiche:

GENERATORE	COSTO FISSO DI ACCENSIONE	COSTO FISSO D'USO PER PERIODO	COSTO PER PERIODO PER MW	CAPACITÀ MASSIMA IN MW PER PERIODO
A	3000	700	5	2100
B	2000	800	4	1800
C	1000	900	7	3000

Un giorno è diviso in due periodi. Un generatore, qualora acceso nel primo periodo, può essere usato nel secondo periodo senza incorrere in costi di accensione aggiuntivi. Si assuma che, in un determinato giorno, il numero di megawatt richiesti nel primo periodo sia 2900, mentre il secondo periodo richieda 3900 megawatt.

a) Assumendo che i costi fissi siano zero, e quindi possano essere ignorati, quali sono le variabili di decisione? Qual'è la formulazione di PL in questo caso ?

Risposta: Si denotino con x_{ij} il numero di megawatt prodotti dal generatore $i \in \{A, B, C\}$ per essere usati nel periodo $j = 1, 2$. I vincoli di capacità dei generatori diventano

$$\begin{aligned} &(1)\ x_{A1} \leq 2100 \qquad x_{A2} \leq 2100 \\ &(2)\ x_{B1} \leq 1800 \qquad x_{B2} \leq 1800 \\ &(3)\ x_{C1} \leq 3000 \qquad x_{C2} \leq 3000 \end{aligned}$$

I vincoli relativi alle potenze richieste saranno

$$\begin{aligned} &(4)\ x_{A1} + x_{B1} + x_{C1} \geq 2900 \\ &(5)\ x_{A2} + x_{B2} + x_{C2} \geq 3900 \end{aligned}$$

Le variabili x_{ij} sono reali, ossia abbiamo il vincolo

$$(6)\ x_{ij} \geq 0, \qquad i \in \{A, B, C\} \qquad j = 1, 2$$

La funzione obiettivo consiste nel minimizzare i costi, ossia

$$\min 5(x_{A1} + x_{A2}) + 4(x_{B1} + x_{B2}) + 7(x_{C1} + x_{C2})$$

con i vincoli (1), (2), (3), (4), (5) e (6).

b) Si considerino ora nel modello i costi fissi di esercizio. Quali sono le variabili 0-1 che è necessario introdurre? Come cambia la funzione obiettivo? E i vincoli?

Risposta: Si definiscano le variabili 0-1 y_{ij} per $i \in \{A, B, C\}$ e $j = 1, 2$ con il seguente significato:

$$y_{ij} = \begin{cases} 1 & \text{se il generatore } i \text{ viene usato nel periodo } j \\ 0 & \text{altrimenti} \end{cases}$$

Come spiegato in precedenza, il legame fra le variabili x_{ij} e y_{ij} è del seguente tipo:

$$(1') \; x_{A1} \leq 2100 y_{A1} \qquad x_{A2} \leq 2100 y_{A2}$$
$$(2') \; x_{B1} \leq 1800 y_{B1} \qquad x_{B2} \leq 1800 y_{B2}$$
$$(3') \; x_{C1} \leq 3000 y_{C1} \qquad x_{C2} \leq 3000 y_{C2}$$

Inoltre, si definiscano delle variabili 0-1 z_i per $i \in \{A, B, C\}$, con il seguente significato:

$$z_i = \begin{cases} 1 & \text{se il generatore } i \text{ viene acceso} \\ 0 & \text{altrimenti} \end{cases}$$

Il generatore i viene acceso se viene usato nel primo o nel secondo periodo. In termini logici, la relazione tra z_i e y_{ij} è $z_i = (y_{i1}$ OR $y_{i2})$. Da 3.6 sappiamo che questa relazione può venire imposta dai vincoli

$$(7) \; z_A \geq y_{A1} \qquad z_A \geq y_{A2} \qquad z_A \leq y_{A1} + y_{A2}$$
$$(8) \; z_B \geq y_{B1} \qquad z_B \geq y_{B2} \qquad z_B \leq y_{B1} + y_{B2}$$
$$(9) \; z_C \geq y_{C1} \qquad z_C \geq y_{C2} \qquad z_C \leq y_{C1} + y_{C2}$$

Le variabili y_{ij} e z_i sono, come detto, binarie:

$$(10) \; y_{ij} \in \{0, 1\}, \qquad z_i \in \{0, 1\}, \text{ per } i \in \{A, B, C\}, j = 1, 2$$

La funzione obiettivo finale sarà

$$\begin{aligned} \min \quad & 3000 z_A + 2000 z_B + 1000 z_C + \\ & 700(y_{A1} + y_{A2}) + 800(y_{B1} + y_{B2}) + 900(y_{C1} + y_{C2}) + \\ & 5(x_{A1} + x_{A2}) + 4(x_{B1} + x_{B2}) + 7(x_{C1} + x_{C2}) \end{aligned}$$

con i vincoli (1'), (2'), (3'), (4), (5), (6). (7), (8), (9) e (10). Si rifletta infine sul fatto che i vincoli del tipo $z_i \leq y_{i1} + y_{i2}$ potrebbero essere omessi dal modello (perchè ?).

4 Metodi risolutivi

Si consideri l'esempio della sezione 3.8. In tale problema, il modello prevede 9 variabili 0-1. Ad ogni assegnamento di tali variabili a valori 0 e 1, corrisponde un problema di PL, ottenuto sostituendo i corrispondenti valori nei vincoli (1'), (2'), (3'). Si noti che per alcuni assegnamenti binari, tale PL può risultare non ammissibile. Un modo semplicistico di pensare di risolvere il problema potrebbe pertanto essere quello di enumerare tutti i $512 = 2^9$ assegnamenti binari, ricavare e risolvere altrettanti PL, e restituire come soluzione finale la soluzione migliore tra i vari PL risolti. Si veda la seguente tabella

y_{A1}	y_{A2}	y_{B1}	y_{B2}	y_{C1}	y_{C2}	Z_A	Z_B	Z_C	ottimo del PL
0	0	0	0	0	0	0	0	0	non ammiss.
0	0	0	0	0	0	0	0	1	non ammiss.
...									
1	1	1	1	0	0	1	1	0	38400
...									
1	1	1	1	1	1	1	1	1	41200

Il principio di provare tutti gli assegnamenti possibili è noto come *enumerazione completa*. È chiaro che tale principio risulta estremamente inefficiente, per non dire completamente impraticabile, per tutti i problemi a parte quelli di dimensioni molto ridotte. Infatti, n variabili booleane possono assumere 2^n valori diversi, e questo numero cresce astronomicamente all'aumentare di n. Si noti che anche nella programmazione lineare il numero di soluzioni di base possibili (vertici del poliedro delle soluzioni) può essere molto elevato, anche maggiore di 2^n. Tuttavia in tale problema, esistono metodi risolutivi più efficienti dell'enumerazione completa. Ad esempio, nella media, l'algoritmo del simplesso determina un vertice ottimo dopo aver esplorato solo un numero molto ridotto di vertici via via migliori. Purtroppo, nel caso della programmazione lineare *intera*, a tutt'oggi non esisono metodi che garantiscano di produrre una soluzione ottima, per ogni istanza, in un numero di passi sostanzialmente minore di quelli necessari all'enumerazione completa. Tale risultato negativo è dovuto al fatto che i problemi di programmazione lineare intera appartengono alla classe dei problemi più difficili in ottimizzazione combinatoria (noti come problemi *NP-hard*).

Nonostante questo risultato negativo, l'enumerazione può essere impiegata con successo, anche su problemi di notevoli dimensioni, qualora si adottino delle strategie per evitare di esplorare tutti i punti dello spazio delle soluzioni. In particolare, si cercherà di eliminare da ogni considerazione alcuni sottoinsiemi di soluzioni, senza averle neppure esaminate, perchè siamo certi che non possono contenere soluzioni ottime.

L'idea alla base di questi ragionamenti è quella di *ripartire* lo spazio delle soluzioni in sottoinsiemi disgiunti e via via di dimensioni più piccole. Nell'esempio precedente, l'enumerazione completa può essere vista come un caso limite di ripartizione, in sottoinsiemi di un elemento ciascuno. Ragionando sui dati, si possono però ottenere delle ripartizioni migliori. Ad esempio, si considerino i vincoli del tipo $z_A = y_{A1}$ `OR` y_{A2}. L'insieme di tutti i punti le cui proiezioni sulle componenti y_{A1}, y_{A2}, z_A non soddisfano tale relazione, non conterrà alcuna soluzione ammissibile (e quindi certamente neanche ottima) e può pertanto essere tralasciato senza enumerarne gli elementi. Si noti che possiamo perciò scartare in un sol colpo $64 = 2^6$ assegnamenti del tipo $(1, 1, \cdot, \cdot, \cdot, \cdot, 0, \cdot, \cdot)$, altri 64 del tipo $(0, 0, \cdot, \cdot, \cdot, \cdot, 1, \cdot, \cdot)$, e così via. In realtà , siccome le variabili z sono determinate non appena le y lo sono, si può affermare che il problema prevede al più $64 = 2^6$ soluzioni, ossia tante quanti gli assegnamenti alle variabili y. Si noti che si sono quindi evitati di enumerare 448 possibili assegnamenti in un sol colpo! Spingendo tali ragionamenti più in dettaglio, l'analisi dei dati fornisce vincoli impliciti che permettono di evitare l'enumerazione di ulteriori assegnamenti. Per esempio, siccome il fabbisogno di energia del secondo periodo è di 3900 MW, e nessun singolo generatore può fornire tale potenza, è ovvio che nel secondo periodo dovranno essere accesi perlomeno 2 generatori, e quindi tutti gli assegnamenti per cui

$$y_{A2} + y_{B2} + y_{C2} \leq 1$$

non potranno corrispondere a soluzioni ammissibili. Ad esempio gli assegnamenti del tipo $(\cdot, 0, \cdot, 0, \cdot, 0)$ possono essere scartati (8 possibilità). Analogamente si scartano altri $24 = 3 \times 8$

assegnamenti. Ecco che il nostro spazio di soluzioni si è ulteriormente ridotto a sole 32 possibilità . Nella sezione 4.1 si esaminerà una procedura nota come *branch and bound* che formalizza alcune idee qui espresse in un metodo generale per la risoluzione di problemi di programmazione lineare intera.

Dire che $y_{A2} + y_{B2} + y_{C2} \leq 1$ deve essere falso, è come dire che

$$y_{A2} + y_{B2} + y_{C2} \geq 2$$

deve essere vero. La disuguaglianza $y_{A2} + y_{B2} + y_{C2} \geq 2$ è pertanto un esempio di *disuguaglianza valida*, che si potrebbe aggiungere ai vincoli, senza alterare lo spazio delle soluzioni ammissibili. L'utilità dell'aggiunta di vincoli al problema si ha qualora si decida di utilizzare come metodo di risoluzione la approssimazione successiva tramite problemi di PL. In tale metodo, il primo passo consiste nel trascurare completamente i vincoli di interezza delle variabili e risolvere il problema risultante (rilassamento continuo) che è un normale PL. Qualora la soluzione ottenuta $\hat{\mathbf{x}}$ risulti intera, sarà senz'altro ottima. Altrimenti, si può pensare di aggiungere dei vincoli che eliminino $\hat{\mathbf{x}}$ dalle soluzioni ammissibili del rilassamento continuo, ma non eliminino alcuna soluzione ammissibile intera. Siccome $\hat{\mathbf{x}}$ viene "tagliata via" dalla presenza del nuovo vincolo, quest'ultimo si dirà un *piano di taglio (cutting plane)*. I piani di taglio sono perciò opportune disuguaglianze valide. Come esercizio, si ricavi una disuguaglianza valida per il problema dell'esempio, basata sul fatto che se nel periodo 1 il generatore C è spento, allora sia A che B devono essere accesi. Il metodo delle disuguaglianze valide e dei piani di taglio è esaminato in dettaglio nella sezione 4.3.

4.1 Branch–and–bound

Il *branch and bound* è una procedura per risolvere problemi di PLI, del tipo *divide et impera*, ossia in cui per risolvere un problema complesso, lo si riconduce a sottoproblemi più semplici (più vincolati, vale a dire con regioni ammissibili più piccole). Le idee alla base di tale metodo sono

- Ripartire lo spazio delle soluzioni ammissibili, ossia spezzare il problema in sottoproblemi più vincolati (e, si spera, più semplici) che si possano
  - risolvere all'ottimalità , o
  - dichiarare non ammissibili, o
  - spezzare ricorsivamente (*branching*)
- Usare tecniche che permettono di decidere quando un sottoproblema **non può** contenere una soluzione ottima, e pertanto non è necessario risolverlo (*bounding*).

Si consideri il problema di programmazione lineare intera pura, cioè . $[\min \mathbf{c}'\mathbf{x}, A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in Z^n]$. Si denoti con P^0 il poliedro ottenuto dal rilassamento continuo, ovvero $P^0 = \{\mathbf{x} : A\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$. Pur non essendo formalmente corretto, per semplicità nel seguito denoteremo con P^i tanto un poliedro quanto l'associato problema di PL, ossia $[\min \mathbf{c}'\mathbf{x}, \mathbf{x} \in P^i]$. Analogamente, definiremo con X^i tanto l'insieme dei punti interi contenuti in P^i (cioè $X^i := P^i \cap Z^n$), quanto il problema di programmazione lineare intera $[\min \mathbf{c}'\mathbf{x}, \mathbf{x} \in X^i]$.

Per risolvere il problema X^0, si può cominciare con il risolvere P^0. Se $\hat{\mathbf{x}}^0$ è intero, allora è anche ottimo per X^0. Altrimenti, sia j una componente tale che $\hat{x}^0_j$ è frazionario. Siano P^1 e P^2 i poliedri $P^1 = P^0 \cap \{\mathbf{x} : x_j \leq \lfloor \hat{x}^0_j \rfloor\}$ e $P^2 = P^0 \cap \{\mathbf{x} : x_j \geq \lceil \hat{x}^0_j \rceil\}$. Si noti che non esistono punti di X^0 per i quali $\lfloor \hat{x}^0_j \rfloor < x_j < \lceil \hat{x}^0_j \rceil$, perchè $\lfloor \hat{x}^0_j \rfloor$ e $\lceil \hat{x}^0_j \rceil$ sono due interi consecutivi.

La soluzione ottima di X^0 sarà allora necessariamente la migliore tra le soluzioni di X^1 e X^2. Siccome non è a priori possibile sapere quale tra X^1 e X^2 conterrà una soluzione ottima, diventa necessario risolverli entrambi. Ciò può essere fatto ricorsivamente, tramite i PL definiti da P^1 e P^2. Come si vede, per risolvere un problema complesso, lo si riconduce a sottoproblemi più vincolati. In alcuni casi, tali sottoproblemi possono dare luogo a soluzioni intere ottenute facilmente, nel qual caso la suddivisione non è più necessaria. Altre volte può accadere che un sottoproblema risulti non ammissibile. Un'ultima possibilità si ha poi quando è possibile affermare che un sottoproblema non può contenere una soluzione ottima di X^0, e pertanto non vale la pena risolverlo. Ad esempio, supponiamo di conoscere una soluzione ammissibile $\tilde{\mathbf{x}} \in X^0$. Sia inoltre X^i un sottoproblema ottenuto da X^0 tramite l'aggiunta di un numero arbitrario di vincoli. Siccome $X^i \subseteq P^i$, si avrà

$$\mathbf{c}'\hat{\mathbf{x}}^i = \min_{\mathbf{x} \in P^i} \mathbf{c}'\mathbf{x} \leq \min_{\mathbf{x} \in X^i} \mathbf{c}'\mathbf{x}$$

o, come si dice, $\mathbf{c}'\hat{\mathbf{x}}^i$ fornisce un *lower bound* al valore ottimo di X^i. Se ora avvenisse che $\mathbf{c}'\tilde{\mathbf{x}} \leq \mathbf{c}'\hat{\mathbf{x}}^i$, potremmo concludere che la soluzione ottima di X^i è irrilevante, in quanto non potrà mai essere migliore della soluzione $\tilde{\mathbf{x}}$ che già abbiamo. Il problema X^i può pertanto essere rimosso da ogni ulteriore considerazione, o, come si dice in gergo, "ucciso".

Le idee appena esposte trovano la loro formalizzazione nella procedura nota come *branch and bound*. Il termine *branch* fa riferimento al fatto che un problema può essere decomposto in sottoproblemi. Graficamente, associando ad ogni problema un nodo di un albero noto come *albero di ricerca*, le diramazioni (*branchings*) di questo albero corrispondono alle suddivisioni di problemi in problemi alternativi e più vincolati (si veda la figura 4). Il termine *bound* deriva invece dal fatto che ad ogni nodo dell'albero di ricerca si calcola un lower bound al valore del sottoproblema; qualora tale lower bound fosse peggiore (ossia maggiore, per un problema di minimizzazione) di un valore ammissibile già noto, il sottoproblema viene ucciso senza generare branching.

Formalmente, la procedura di branch and bound può essere riassunta dal seguente algoritmo:

0. $\tilde{\mathbf{x}} \leftarrow$ indefinito. $\tilde{v} \leftarrow +\infty$. $\mathcal{L} \leftarrow \{P^0\}$
1. **while** $\mathcal{L} \neq \emptyset$ **do**
2. Si scelga un problema $P^i \in \mathcal{L}$, e si ponga $\mathcal{L} \leftarrow \mathcal{L} \setminus P^i$
3. Si risolva P^i come PL
4. **if** P^i è ammissibile **then begin**
5. Sia $\hat{\mathbf{x}}^i$ la soluzione ottima di P^i
6. **if** $\mathbf{c}'\hat{\mathbf{x}}^i < \tilde{v}$ **then begin**
7. **if** $\hat{\mathbf{x}}^i$ è intero **then** $\tilde{\mathbf{x}} \leftarrow \hat{\mathbf{x}}^i$; $\tilde{v} \leftarrow \mathbf{c}'\hat{\mathbf{x}}^i$
8. **else begin**
9. Sia k una componente frazionaria di $\hat{\mathbf{x}}^i$.
   Si creino due problemi figli del problema P^i, ossia
   $P^{i_1} = P^i \cap \{\mathbf{x} : x_k \leq \lfloor \hat{x}^i_k \rfloor\}$ e $P^{i_2} = P^i \cap \{\mathbf{x} : x_k \geq \lceil \hat{x}^i_k \rceil\}$
10. Si aggiungano alla lista dei problemi da risolvere: $\mathcal{L} \leftarrow \mathcal{L} \cup \{P^{i_1}, P^{i_2}\}$
11. **end else**
12. **end if**
13. **end if**
14. **end while**

In questo algoritmo si è denotato con $\tilde{\mathbf{x}}$ la migliore soluzione intera corrente (alla fine sarà anche l'ottimo globale); con $\tilde{v}$ il valore di tale soluzione (detto anche l'*incombente*); con $\mathcal{L}$

la *lista dei problemi aperti*, ossia l'elenco dei sottoproblemi ancora da risolvere. La scelta di quale problema risolvere al passo 2. è arbitraria, e dipende solitamente da come si sceglie di implementare la lista $\mathcal{L}$. In particolare, si hanno tipicamente le seguenti tre possibilità :

1. **LIFO** (Last In First Out). In tale caso, il problema risolto ad ogni passo è quello che è stato generato più di recente. In tale caso, si dice che $\mathcal{L}$ agisce come una "pila" (stack), e che l'albero di ricerca è esplorato *in profondità* o *a scandaglio* (depth–first).

2. **FIFO** (First In First Out). In questo caso, il problema risolto ad ogni iterazione è quello che è rimasto nella lista dei problemi aperti per più tempo. Si dice allora che $\mathcal{L}$ agisce come una "coda" (queue), e che l'albero di ricerca è esplorato *in ampiezza* o *a ventaglio* (breadth–first).

3. **Best First**. In questo caso, si sceglie di risolvere per primo il sottoproblema aperto che ha il migliore lower bound, nella speranza che possa contenere la soluzione più buona.

Un ulteriore grado di libertà nell'esecuzione del precedente algoritmo si ha al passo 9, qualora ci sia più di una componente frazionaria nell'ottimo $\hat{\mathbf{x}}^i$ corrente. In tale caso, la scelta può essere fatta in modo completamente arbitrario. È comunque consuetudine l'usare come componente per il branching quella che più si discosta da un valore intero.

4.2 Esempio

Si consideri il seguente problema "giocattolo", cioè puramente illustrativo del metodo. In una copisteria si vogliono fotocopiare e rilegare dispense di due corsi. Si hanno a disposizione 240 grammi di colla, 1000 fogli di carta e 3 ore di tempo. Le dispense del primo corso vanno preparate a mano e hanno 100 pagine, richiedono 60 grammi di colla e 50 minuti di lavoro. Quelle del secondo corso sono fatte in automatico e hanno 200 pagine, 40 grammi di colla e 20 minuti di lavoro. Il profitto per ogni dispensa del primo tipo è di 1900 lire, mentre quelle del secondo tipo rendono 1000 lire. Si vuole massimizzare il profitto.

Si denotino con x_1 e x_2 il numero di dispense da produrre per ciascun corso. Il problema diventa pertanto

$$\begin{array}{lrcrcr} \max & 1900x_1 & + & 1000x_2 & & \\ & & & & & \\ & 60x_1 & + & 40x_2 & \leq & 240 \\ & 100x_1 & + & 200x_2 & \leq & 1000 \\ & 50x_1 & + & 20x_2 & \leq & 180 \end{array}$$

$\mathbf{x} \in Z_+^2$

L'insieme delle soluzioni ammissibili è rappresentato in figura 1. Denotiamo con $c(\mathbf{x})$ il valore di una soluzione $\mathbf{x}$. Da un esame della figura, si vede che i vertici di P^0 sono $O = (0,0)$, $A = (0,5)$, $B = (1, \frac{9}{2})$, $C = (3, \frac{3}{2})$ e $D = (\frac{18}{5}, 0)$, mentre i vertici di $P(X)$ sono O, A, $E = (2,3)$, $F = (3,1)$ e $G = (3,0)$. Un semplice calcolo in corrispondenza di tali punti ci porterebbe a concludere che C, per il quale $c(C) = 7200$ è l'ottimo di P, mentre E, per il quale $c(E) = 6800$ è l'ottimo intero cercato.

Supponiamo di applicare il metodo branch and bound, con esplorazione FIFO dei problemi aperti a questo esempio.

1. Il metodo comincia risolvendo P^0 e ottenendo $\hat{\mathbf{x}}^0 = C = (3, \frac{3}{2})$. Siccome $\hat{x}_2^0$ è frazionario, si creano i due vincoli $x_2 \leq 1$ e $x_2 \geq 2$ e due nuovi problemi $P^1 = P^0 \cap \{\mathbf{x} : x_2 \leq 1\}$ e $P^2 = P^0 \cap \{\mathbf{x} : x_2 \geq 2\}$. L'albero di ricerca è descritto nella figura 4.

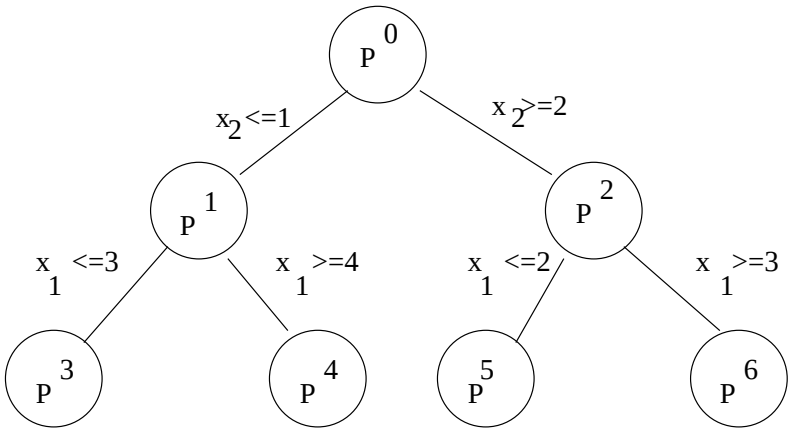


Figure 4: Albero di ricerca.

2. Si passa ora a risolvere P_1. In P^1 compare il nuovo vertice $H = (\frac{16}{5}, 1)$, per il quale $c(H) = 7080$. $\hat{\mathbf{x}}^1 = H$ è ottimo per P^1, e siccome $\frac{16}{5}$ è frazionario, aggiungiamo a P^1 in un caso il vincolo $x_1 \leq 3$, ottenendo P^3, e nell'altro il vincolo $x_1 \geq 4$, ottenendo P^4.

3. Si passa ora alla soluzione di P^3. P^3 è un poliedro a **vertici interi**! Nel caso di un P limitato (come noi assumiamo) questo deve succedere prima o poi. L'ottimo di P^3 è $\hat{\mathbf{x}}^3 = F$, di valore $c(F) = 6700$. Siccome $\hat{\mathbf{x}}^3$ è intero, **non c'è branching** da P^3. Inoltre, siccome abbiamo trovato la prima **soluzione intera**, la memorizziamo in $\tilde{\mathbf{x}} \leftarrow (3, 1)$ e il suo valore in $\tilde{v} \leftarrow 6700$.

4. È il turno di P^4, che risulta vuoto, ossia **non ammissibile**. P^4 viene ucciso e si conclude la sua esplorazione, e, di conseguenza, anche quella di P^1. Passiamo a P^2.

5. In P^2 compare il vertice $I = (\frac{8}{3}, 2)$, di valore $c(I) = 7066.66\ldots$. Tale vertice è ottimo per P^2, ossia $\hat{\mathbf{x}}^2 = I$. Confrontiamo il suo valore con l'incombente. Siccome $7066.66\ldots = \mathbf{c}'\hat{\mathbf{x}}^2 > \mathbf{c}'\tilde{\mathbf{x}} = 6700$, **non c'è bounding** (si ragioni sul fatto che ora stiamo risolvendo un problema di massimizzazione, per cui le disuguaglianze del bounding sono invertite). Da P^2 si generano i due nuovi problemi, $P^5 = P^2 \cap \{\mathbf{x} : x_1 \leq 2\}$ e $P^6 = P^2 \cap \{\mathbf{x} : x_1 \geq 3\}$

6. Si risolve ora P^5, a cui compare il nuovo vertice intero E, che è anche l'ottimo per P^5. Siccome $\hat{\mathbf{x}}^5$ è intero, da P^5 **non c'è branching**. Si confronta $\hat{\mathbf{x}}^5$ con $\tilde{\mathbf{x}}$. Siccome $6800 = \mathbf{c}'\hat{\mathbf{x}}^5 > \mathbf{c}'\tilde{\mathbf{x}} = 6700$, si **aggiorna l'ottimo**, ponendo $\tilde{\mathbf{x}} \leftarrow \hat{\mathbf{x}}^5 = (2, 3)$ e $\tilde{v} \leftarrow 6800$

7. Si passa infine a P^6, che risulta **non ammissibile**. Siccome la lista di problemi aperti è ora vuota, **l'esplorazione termina** e la soluzione ottima è ottenuta al punto $E = (2, 3)$

Si noti che in questo esempio, ad ogni risoluzione di un P^i sarebbe corrisposta la soluzione di un PL tramite il metodo del simplesso. Vista la semplicità dell'esempio abbiamo preferito evitare di usare tale metodo. Il lettore è però invitato a ripetere i passi dell'algoritmo utilizzando il simplesso per ottenere le varie soluzioni ottime $\hat{\mathbf{x}}^i$.

4.3 I piani di taglio

Si consideri il problema di programmazione lineare intera in forma standard: $\min\{\mathbf{c}'\mathbf{x}, \mathbf{x} \in X\}$ dove $X = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{x} \text{ intero}\}$. Sia, al solito, $P = \{\mathbf{x} : A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, e si risolva il PL associato a P, ottenendo la soluzione ottima $\hat{\mathbf{x}}$. Se $\hat{\mathbf{x}}$ è intero, è anche l'ottimo per il problema

originario. Altrimenti, si cercherà di aggiungere dei vincoli a P così da ottenere un poliedro P' con $X \subseteq P' \subset P$ e $\hat{\mathbf{x}}^0 \notin P'$. P' è ottenuto da P con l'aggiunta di disuguaglianze *valide* ma non soddisfatte da $\hat{\mathbf{x}}$. Si dirà perciò che l'ottimo frazionario è "tagliato" da tali disuguaglianze, o anche che tali disuguaglianze sono dei "tagli validi".

4.4 Disuguaglianze di Chvatal

Una procedura generale per generare disuguaglianze valide è la seguente. Sia $\mathbf{u} \in R^m$. Se $A\mathbf{x} = \mathbf{b}$ allora anche $\mathbf{u}'A\mathbf{x} = \mathbf{u}'\mathbf{b}$, ossia

$$\sum_{j=1}^{n}(\sum_{i=1}^{m} u_i A_{ij})x_j = \sum_{i=1}^{m} u_i b_i$$

Riducendo i coefficienti a sinistra, e in virtù della nonnegatività di $\mathbf{x}$, si ottiene

$$\sum_{j=1}^{n}\lfloor\sum_{i=1}^{m} u_i A_{ij}\rfloor x_j \leq \sum_{i=1}^{m} u_i b_i$$

Se $\mathbf{x} \in X$, la parte sinistra della disuguaglianza è un numero intero, e pertanto anche la parte destra si può arrotondare ad un intero, ottenendo una disuguaglianza valida per X:

$$\sum_{j=1}^{n}\lfloor\sum_{i=1}^{m} u_i A_{ij}\rfloor x_j \leq \lfloor\sum_{i=1}^{m} u_i b_i\rfloor \qquad (C1)$$

Tale disuguaglianza si dirà *disuguaglianza di Chvatal, di rango 1, generata da* $\mathbf{u}$.

Esempio: Si consideri il sistema

$$\begin{array}{rcrcrcrcl} 2x_1 & + & 5x_2 & + & x_3 & & & = & 30 \\ 4x_1 & - & 3x_2 & & & + & x_4 & = & 6 \end{array}$$

con $\mathbf{x} \in Z^4$. Sia $\mathbf{u} = (\frac{5}{6}, \frac{11}{12})$. Si ottiene, dapprima $\frac{16}{3}x_1 + \frac{17}{12}x_2 + \frac{5}{6}x_3 + \frac{11}{12}x_4 = 30\frac{1}{2}$. Arrotondando a sinistra si ricava $5x_1 + x_2 \leq 30\frac{1}{2}$. Infine, arrotondando a destra si ottiene la disuguaglianza di Chvatal $5x_1 + x_2 \leq 30$.

Le disuguaglianze di Chvatal sono sempre valide, ma non necessariamente dei tagli per un ottimo $\hat{\mathbf{x}}$ corrente. Per ottenere tali disuguaglianze, utilizziamo i cosiddetti *tagli di Gomory.*

4.5 I tagli di Gomory

Sia B la matrice di base associata alla soluzione ottima $\hat{\mathbf{x}}$ dell'LP corrente. Sia inoltre $\mathcal{F}$ l'insieme di indici delle variabili fuori base. Definiamo $A^* = B^{-1}A$ e $\mathbf{b}^* = B^{-1}\mathbf{b} = \hat{\mathbf{x}}$. Si noti che, a meno di una permutazione delle colonne, la matrice A^* ha la forma $A^* = (I \mid B^{-1}A^{\mathcal{F}})$. Sia $\hat{x}_l$ una componente frazionaria di $\hat{\mathbf{x}}$. Si consideri la riga l del sistema $A^*x = \mathbf{b}^*$, ossia

$$x_l + \sum_{j \in \mathcal{F}} A^*_{lj} x_j = b^*_l$$

dove $b^*_l = \hat{x}_l$ è , come detto, non intero. Come descritto in precedenza, si potrà arrotondare prima la parte sinistra

$$x_l + \sum_{j \in \mathcal{F}} \lfloor A^*_{lj}\rfloor x_j \leq b^*_l$$

e poi la destra, ottenendo la disuguaglianza valida

$$x_l + \sum_{j \in \mathcal{F}} \lfloor A^*_{lj} \rfloor x_j \leq \lfloor b^*_l \rfloor \qquad (G1)$$

Tale disuguaglianza è detta *taglio di Gomory*, in *forma intera*. Si rifletta sul fatto che (G1) altro non è se non la disuguaglianza di Chvatal generata da un $\mathbf{u}$ che è la l-ma riga di B^{-1}. Si noti che (G1)

1. è **valida** per X (essendo una disuguaglianza di Chvatal)
2. è **violata** da $\hat{\mathbf{x}}$. Infatti, per $\hat{\mathbf{x}}$ si ha $\hat{x}_l = b^*_l$ (frazionario) e $\hat{x}_j = 0$ per $j \in \mathcal{F}$. Sostituendo in (G1), si ottiene $b^*_l \leq \lfloor b^*_l \rfloor$, chiaramente falso per $b^*_l \notin Z$. Ecco quindi che $\hat{\mathbf{x}}$ è stato tagliato da (G1), che può essere aggiunta ai vincoli di P, ottenendo P'.

Esiste un'altra forma dei tagli di Gomory, nota coma *forma frazionaria*. Siano la parte intera e la parte frazionaria di un numero reale a definite da $[a]_I = \lfloor a \rfloor$ e $[a]_F = a - [a]_I$ (chiaramente $0 \leq [a]_F < 1$ per ogni a). Dal vincolo $x_l + \sum_{j \in \mathcal{F}} A^*_{lj} x_j = b^*_l$ si ottiene

$$x_l + \sum_{j \in \mathcal{F}} ([A^*_{lj}]_I + [A^*_{lj}]_F) x_j = [b^*_l]_I + [b^*_l]_F$$

ossia

$$\sum_{j \in \mathcal{F}} [A^*_{lj}]_F x_j = [b^*_l]_F + ([b^*_l]_I - \sum_{j \in \mathcal{F}} [A^*_{lj}]_I x_j - x_l) \qquad (G2)$$

Nel termine tra parentesi a destra, riconosciamo i termini della forma intera del vincolo di Gomory, il quale afferma che $[b^*_l]_I \geq x_l + \sum_{j \in \mathcal{F}} [A^*_{lj}]_I x_j$. Concludiamo perciò che

$$[b^*_l]_I - \sum_{j \in \mathcal{F}} [A^*_{lj}]_I x_j - x_l \geq 0$$

Usando questa relazione in (G2), si ricava

$$\sum_{j \in \mathcal{F}} [A^*_{lj}]_F x_j \geq [b^*_l]_F \qquad (G3)$$

(G3) è detto taglio di Gomory in forma frazionaria. Si noti ancora una volta che (G3) non è soddisfatto dalla base corrente, dato che $x_j = 0$, per $j \in \mathcal{F}$ nella base corrente, mentre b^*_l è frazionario.

4.6 L'algoritmo dei piani di taglio

Il metodo dei piani di taglio consiste nel partire risolvendo P^0, e iterare la seguente procedura

1. **while** $\hat{\mathbf{x}}^i$ non è intero **do**
2. Si generi un taglio di Gomory corrispondente a una componente frazionaria di $\hat{\mathbf{x}}^i$
3. Si aggiunga tale taglio ai vincoli di P^i, ottenendo P^{i+1}
4. $i \leftarrow i + 1$
5. Si risolva P^i, ottenendo $\hat{\mathbf{x}}^i$
6. **end while**

Si noti che al passo 5, la soluzione di P^i può essere ottenuta sfruttando il tableau della soluzione di P^{i-1}. Infatti, aggiungere un vincolo di disuguaglianza $\sum_{j \in \mathcal{F}} [A^*_{lj}]_F x_j \geq [b^*_l]_F$,

corrisponde a introdurre una nuova variabile ausiliaria x_{n+i} (ossia una nuova colonna nel tableau) e una nuova equazione (ossia una nuova riga) $\sum_{j \in \mathcal{F}} [A^*_{lj}]_F x_j - x_{n+i} = [b^*_l]_F$. Nella nuova colonna i coefficienti sono tutti 0, tranne un -1 in corrispondenza alla nuova riga. Inoltre, nella nuova riga, i coefficienti delle variabili correntemente in base sono nulli. Si ottiene pertanto un tableau che soddisfa i vincoli di ottimalità (costi ridotti non negativi), ma non quelli di ammissibilità (infatti, la componente x_{n+i} del nuovo tableau risulta negativa). In tale caso, basterà applicare il metodo duale del simplesso, per ottenere con un pivot, una nuova soluzione ottima per il tableau di P^i.

4.7 Esempio

Si consideri la produzione di due tipi di prodotto A e B tali che

- Ogni unità di A richiede 2 ore di lavoro e ogni unità di B ne richiede 5. Sono in tutto disponibili 30 ore di lavoro.
- Un'unità di A genera 4 pezzi di un sottoprodotto, mentre ogni unità di B consuma 3 pezzi di tale sottoprodotto. I pezzi rimanenti in eccesso di tale sottoprodotto possono essere depositati in magazzino. Il magazzino ha una capacità residua in grado di accogliere al più 6 nuovi pezzi di tale sottoprodotto
- Il profitto derivante dalla vendita del prodotto A è lo stesso che per il prodotto B.

Si denotino con x_1 e x_2 il numero di elementi di A e B prodotti. Il problema diventa pertanto

$$\begin{array}{lrcrcr} \max & x_1 & + & x_2 & & \\ & & & & & \\ & 2x_1 & + & 5x_2 & \leq & 30 \\ & 4x_1 & - & 3x_2 & \leq & 6 \end{array}$$

$\mathbf{x} \in Z^2_+$

Trasformando il problema in forma standard e trascurando inizialmente i vincoli di interezza, si ottiene il seguente problema

$$\begin{array}{lrcrcrcrcr} \min & z & & & & & & & & \\ & & & & & & & & & \\ & 2x_1 & + & 5x_2 & + & x_3 & & & = & 30 \\ & 4x_1 & - & 3x_2 & & & + & x_4 & = & 6 \\ & -x_1 & - & x_2 & & & & & = & z \end{array}$$

$\mathbf{x} \geq \mathbf{0}$

il cui tableau ottimo è dato da

$$\begin{array}{cccc|c} 0 & 1 & \frac{2}{13} & \frac{-1}{13} & \frac{54}{13} \\ & & & & \\ 1 & 0 & \frac{3}{26} & \frac{5}{26} & \frac{60}{13} \\ \hline & & & & \\ 0 & 0 & \frac{7}{26} & \frac{3}{26} & \frac{114}{13} \end{array}$$

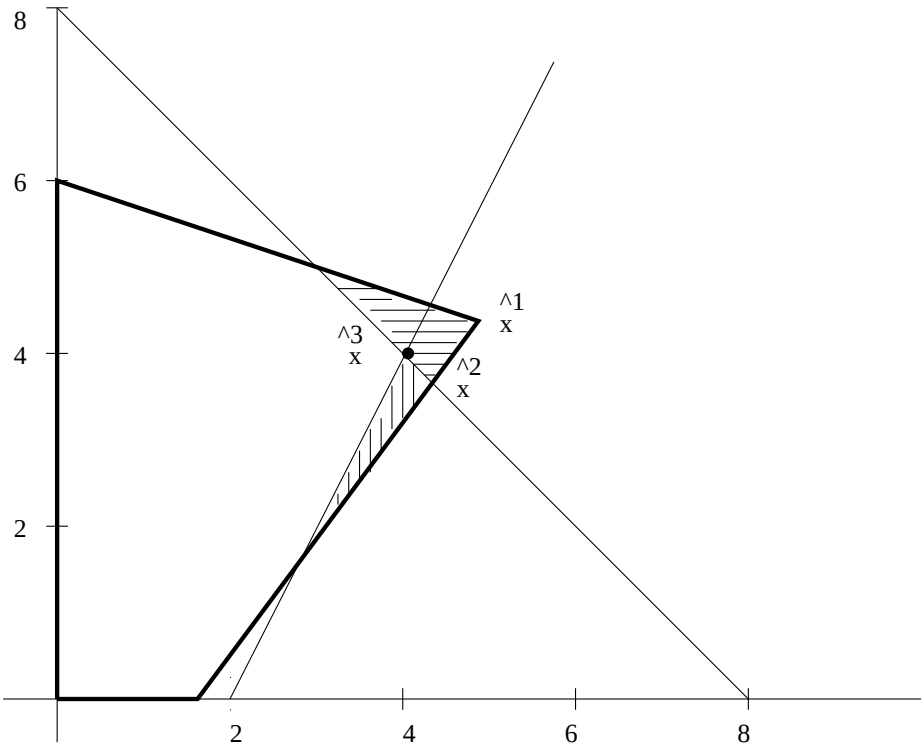


Figure 5: Tagli di Gomory per l'esempio

Siccome la soluzione $\hat{\mathbf{x}}^0 = (\frac{60}{13}, \frac{54}{13})$ non è intera, si genera il taglio di Gomory corrispondente alla terza riga, vale a dire quella per cui la parte frazionaria nell'ultima colonna è massima. Tale taglio sarà

$$\frac{7}{26}x_3 + \frac{3}{26}x_4 \geq \frac{10}{13}$$

Tenendo presente che $x_3 = 30 - 2x_1 - 5x_2$ e $x_4 = 6 - 4x_1 + 3x_2$, il taglio può essere espresso nello spazio delle variabili originarie x_1 e x_2 come $x_1 + x_2 \leq 8$. In figura 5 si può vedere la parte di regione ammissibile "tagliata via" da questo vincolo.

Introducendo la variabile ausiliaria x_5 ed espandendo il tableau per tener conto del taglio appena aggiunto, si ottiene il seguente tableau

0	1	$\frac{2}{13}$	$\frac{-1}{13}$	0	$\frac{54}{13}$
1	0	$\frac{3}{26}$	$\frac{5}{26}$	0	$\frac{60}{13}$
0	0	$\frac{-7}{26}$	$\frac{-3}{26}$	1	$\frac{-10}{13}$
0	0	$\frac{7}{26}$	$\frac{3}{26}$	0	$\frac{114}{13}$

Applicando il metodo duale del simplesso, si effettuerà un'operazione di pivot sull'elemento T_{33}, ottenendo la seguente tabella ottima:

0	1	0	$\frac{-1}{7}$	$\frac{4}{7}$	$\frac{26}{7}$
1	0	0	$\frac{1}{7}$	$\frac{3}{7}$	$\frac{30}{7}$
0	0	1	$\frac{3}{7}$	$\frac{-26}{7}$	$\frac{20}{7}$
0	0	0	0	1	8

che fornisce la nuova soluzione $\hat{\mathbf{x}}^2 = (\frac{30}{7}, \frac{26}{7})$, ancora una volta frazionaria. Servendosi della terza riga, si genera il nuovo taglio di Gomory

$$\frac{3}{7}x_4 + \frac{2}{7}x_5 \geq \frac{6}{7}$$

corrispondente, nel piano di x_1 e x_2, a $2x_1 - x_2 \leq 4$. Occorre ora aggiungere una nuova variabile ausiliaria x_6 ed espandere il tableau, ottenendo

0	1	0	$\frac{-1}{7}$	$\frac{4}{7}$	0	$\frac{26}{7}$
1	0	0	$\frac{1}{7}$	$\frac{3}{7}$	0	$\frac{30}{7}$
0	0	1	$\frac{3}{7}$	$\frac{-26}{7}$	0	$\frac{20}{7}$
0	0	0	$\frac{-3}{7}$	$\frac{-2}{7}$	1	$\frac{-6}{7}$
0	0	0	0	1	0	8

Facendo un pivot sull'elemento T_{44} si ottiene il nuovo tableau ottimo

0	1	0	0	$\frac{2}{3}$	$\frac{-1}{3}$	4
1	0	0	0	$\frac{1}{3}$	$\frac{1}{3}$	4
0	0	1	0	-4	1	2
0	0	0	1	$\frac{2}{3}$	$\frac{-7}{3}$	2
0	0	0	0	1	0	8

La soluzione corrente $\hat{\mathbf{x}}^2 = (4, 4)$ è intera e pertanto è la soluzione ottima cercata.