9

Metodi generali per la soluzione di problemi di PLI

Per la soluzione di problemi di PLI non esistono metodi universalmente efficienti. Molto spesso è necessario utilizzare algoritmi "ad hoc" che siano in grado di sfruttare la particolare struttura del problema. Esistono però dei metodi applicabili ad una larga classe di problemi di PLI. Essi sono il metodo del "Branch and Bound" ed il metodo dei piani di taglio. In questo paragrafo verrà descritta in dettaglio la tecnica del "Branch and Bound". Preliminarmente, mettiamo subito in evidenza il fatto che un problema di Programmazione Lineare Intera risulta, in generale, molto piú difficile da risolvere di un problema di Programmazione Lineare; questo è vero anche se il problema intero ha tutte le variabili 0–1.

Consideriamo un problema di Programmazione Lineare Intera,

$$\begin{cases} \min c^T x \\ Ax \geq b \\ x \geq 0, \quad \text{intero.} \end{cases} \tag{PLI}$$

Nel trattare i metodi per la soluzione di problemi di Programmazione Linere Intera faremo la seguente ipotesi sulla regione ammissibile del problema (PLI):

Assunzione A: *la regione ammissibile è costituita da un numero finito di punti*

Questa assunzione non è restrittiva soprattutto in relazione al fatto che nella realtà i problemi che si formulano verificano intrinsecamente questa assunzione. Naturalmente questa assunzione implica che la formulazione del problema è rappresentata da un politopo.

Vediamo ora alcune tecniche che possono essere utilizzate (alcune solo in linea teorica) per risolvere un generico problema di Programmazione Lineare Intera.

9.1 ENUMERAZIONE TOTALE

Abbiamo avuto modo di far notare piú volte che in problema di Programmazione Lineare 0–1 una soluzione ottima può essere determinata teoricamente enumeran-tutte le possibili soluzioni ammissibili; in linea di principio, sarebbe possibile enumerare tutti i vettori binari a n componenti, (verificarne l'ammissibilità) e determinare quelli ammissibili selezionando quelli corrispondenti al valore più basso della funzione obiettivo. Nell'ipotesi che l'Assunzione A sia soddisfatta, questa tecnica di enumerazione totale è teoricamente possibile anche per risolvere un problema di Programmazione Lineare Intera. Purtroppo però, ciò è possibile solo per problemi di dimensioni molto ridotte in quanto c'è una crescita esponenziale dei vettore da esaminare con il numero delle variabili del problema.
Si osservi che un algoritmo di questo tipo (enumerativo) non potrebbe essere assolutamente applicato al caso di un problema di Programmazione Lineare in quanto l'insieme ammissibile in questo caso è non numerabile. Tuttavia, nonostante ciò, come è noto, per la Programmazione Lineare esistono algoritmi (non enumerativi) che permettono di risolvere problemi di Programmazione Lineare a grandi dimensioni, mentre per la Programmazione Lineare Intera non esistono algoritmi in grado di risolvere qualunque problema con tempi di calcolo che non crescono con rapidità esponenziale al crescere dimensione del problema.
In conclusione, possiamo affermare che l'enumerazione totale non rappresenta una tecnica praticabile se non in rari casi di problemi con un numero molto basso di variabili.

9.2 SOLUZIONE APPROSSIMATA PER ARROTONDAMENTO

Come descritto precedentemente, questa strategia si basa sull'idea di risolvere il problema rilassato, cioè il rilassamento lineare, e poi approssimare la soluzione non intera al punto a componenti intere "più vicino". Questa strategia sembrerebbe avere senso soprattutto se ci si aspetta che le variabili assumano valori abbastanza grandi (ad esempio, se si considera la produzione di bene in una azienda costruttrice che si aggira intorno alle 10000 unità mensili, in questo caso se si ha un valore frazionario pari ad esempio a 9837.8 è verosimile "arrotondarlo" a 9838). Anche se si perde l'ottimalità della soluzione, la soluzione "arrotondata" potrebbe avere senso.
Come detto, questa strategia non è in generale una buona strategia. Soprattutto quando le variabili assumono valori relativamente piccoli; in questo caso una siffatta strategia può risultare completamente errata; questo è vero soprattutto in

relazione a problemi di Programmazione Lineare 0–1; in questi problemi le variabili 0–1 indicano scelte alternative e ogni arrotondamento può essere totalmente privo di senso.
Come ulteriore esempio di una situazione che può verificarsi effettuando un arrotondamento della soluzione si può considerare l'Esempio 8.4.1 già visto. L'insieme ammissibile di questo problema intero è costituito dai punti $(0,0),(0,1)$ e $(1,1)$. La soluzione ottima del problema rilassato è nel punto $(5/2,2)$ con valore $9/2$. Arrotondando la componente non intera sia per eccesso, sia per difetto, si ottengono punti che non sono ammissibili e quindi non utilizzabili.

Tuttavia anche nel caso in cui si possa trovare un punto ammissibile "vicino" al punto di ottimo del rilassamento lineare in realtà può accadere che tale punto può essere anche molto "lontano" dalla soluzione ottima intera.

9.3 LA TECNICA DEL "BRANCH AND BOUND"

Si vuole ora esaminare una particolare tecnica che è molto utilizzata per la soluzione di problemi di Programmazione Lineare Intera. Abbiamo già visto che una possibile, semplice tecnica di risoluzione di un problema di Programmazione Lineare Intera, purtroppo quasi mai utilizzabile in pratica, è l'enumerazione totale. Il "Branch and Bound" (BB), invece, è una metodologia di ricerca della soluzione ottima che effettua un'esplorazione *parziale* dell'insieme delle soluzioni ammissibili. In particolare la funzione obiettivo viene calcolata per una sottoinsieme di cardinalità abbastanza piccola delle soluzioni ammissibili con la proprietà di contenere almeno una soluzione ottima.

Per descrivere in dettaglio questa tecnica facciamo riferimento al generico problema di Programmazione Lineare Intera

$$\begin{cases} \min c^T x \\ Ax \geq b \\ x \geq 0, \quad \text{intero} \end{cases} \tag{9.3.1}$$

che può essere riscritto nella forma

$$\begin{cases} \min c^T x \\ x \in S \end{cases}$$

dove $S = \{x \in \mathbb{R}^n \mid Ax \geq b, \quad x \geq 0, \quad \text{intero}\}$ è l'insieme ammissibile che per l'Assunzione A ha cardinalità finita e quindi il problema non può essere illimitato inferiormente. Indicheremo con $x^\star$ l'ottimo del problema (9.3.1) e con $z^* = c^T x^\star$ il suo valore ottimo corrispondente.

La strategia che è alla base della tecnica del "branch and bound" è la decomposizione del problema originario (9.3.1) in sottoproblemi. Questo viene realizzato

effettuando una partizione dell'insieme ammissibile S in una famiglia $\{S_1, \ldots, S_q\}$ di sottoinsiemi di S con $q \geq 2$, cioè tali che

$$S_i \cap S_j = \emptyset \qquad \text{per ogni coppia} \qquad 1 \leq i < j \leq r$$

e

$$\bigcup_{i=1}^{q} S_i = S.$$

A seguito di questa partizione si possono considerare q sottoproblemi (che indichiamo con $Prob^{(i)}$) del tipo

$$\begin{cases} \min c^T x \\ x \in S_i \end{cases} \qquad i = 1, \ldots, q. \qquad (Prob^{(i)})$$

Ora, se $x^{(i)}$ è l'ottimo dell'i-esimo sottoproblema $Prob^{(i)}$ e $z^{(i)} = c^T x^{(i)}$ il valore ottimo corrispondente, si ha che la soluzione ottima del problema originario è data dalla $x^{(i)}$ corrispondente al minimo tra i valori $z^{(i)} = c^T x^{(i)}$, $i = 1, \ldots, q$. Identificando il problema originario (9.3.1) con il problema $Prob^{(0)}$ e il suo insieme ammissibile S con S_0 si può dire che i nuovi problemi generati $Prob^{(1)}, \ldots, Prob^{(q)}$ sono *"figli"* del problema *"padre"* $Prob^{(0)}$.
Se un sottoproblema $Prob^{(i)}$ dovesse risultare, a sua volta, di difficile soluzione si partiziona ulteriormente l'insieme S_i producendo nuovi sottoproblemi (figli) ed iterando la procedura fino a che il problema originario non risulti decomposto in problemi elementari di facile soluzione. Si osservi che questa generazione progressiva di sottoproblemi (figli) produce un *albero genealogico* detto *albero di enumerazione*.

In generale però, risolvere un sottoproblema può essere difficile tanto quanto risolvere il problema originario ed è per questo motivo che invece della soluzione esatta del problema $Prob^{(i)}$ si preferisce calcolare una limitazione inferiore *"lower bound"* L_i di $z^{(i)}$ e cioè un valore $L_i \leq z^{(i)}$. Tale valore viene poi confrontato, ad un certo passo della procedura, con il miglior valore della funzione obiettivo trovato fino a quel momento che chiameremo *valore ottimo corrente* e indicheremo con $\widetilde{z}$. Se il lower bound trovato L_i risulta non inferiore a quello del valore ottimo corrente, ovvero se

$$\widetilde{z} \leq L_i \leq z^{(i)}$$

allora nell'insieme S_i non esiste un punto in cui la funzione obiettivo abbia un valore migliore (cioè minore) di $\widetilde{z}$. Questo permette di sospendere l'esame del sottoproblema $Prob^{(i)}$ senza risolverlo e di non considerarlo ulteriormente nella soluzione del problema originario.

Da quanto detto emerge che la tecnica del "branch and bound" è caratterizzata da due fasi principali:

i) FASE DI "BOUNDING"
Calcolo dei "lower bound" dei sottoproblemi allo scopo di acquisire l'informazione necessaria per capire se scartare o meno un sottoproblema.

ii) FASE DI "BRANCHING"
Generazione di sottoproblemi e quindi dell'albero di enumerazione.

È importante osservare che entrambi le fasi devono essere realizzabili attraverso procedure efficienti. Inoltre il numero dei sottoproblemi generati deve essere estremamente limitato affinché la strategia nel suo complesso risulti efficiente.

Riportiamo, ora, uno schema algoritmico che implementa il metodo del "branch and bound". Indicheremo con $\mathcal{L}$ l'insieme dei sottoproblemi candidati (generati nelle varie fasi di branching) che devono ancora essere analizzati; tali sottoproblemi vengono detti *aperti*. Con $\widetilde{z}$ indicheremo il *valore ottimo corrente* e con $\widetilde{x}$ la *soluzione ottima corrente* che vengono dapprima inizializzati e poi aggiornati nel corso dell'algoritmo non appena vengono individuate soluzioni ammissibili per il problema originario "migliori" cioè con valori inferiori della funzione obiettivo. Per ciascun problema $Prob^{(i)} \in \mathcal{L}$ indicheremo con L_i il valore di un "lower bound" e con $\bar{x}^{(i)}$ un vettore in corrispondenza del quale è raggiunto questo "lower bound", cioè un vettore $\bar{x}^{(i)}$ tale che risulti $L_i = c^T \bar{x}^{(i)}$.

Metodo del "Branch and Bound"

Inizializzazione

- Si inizializzano $\widetilde{z}$ e $\widetilde{x}$ determinando una soluzione ammissibile (facile da calcolare) del problema originario. Naturalmente risulta $\widetilde{z} \geq z^\star$. Se tale soluzione $\widetilde{x}$ non è facile da individuare si lascia per ora non definito $\widetilde{x}$ e si pone $\widetilde{z} = +\infty$.
- Si applica una *strategia di "bounding"* per determinare un "lower bound" L_0 del valore ottimo del problema $Prob^{(0)}$ e un vettore $\bar{x}^{(0)}$ in corrispondenza del quale si ha $L_0 = c^T \bar{x}^{(0)}$.
  - Se $\bar{x}^{(0)} \in S_0$ oppure $L_0 = \widetilde{z}$, l'algoritmo termina; nel primo caso si ha che la soluzione ottima del problema originario è $\bar{x}^{(0)}$, nel secondo tale soluzione ottima è l'ottimo corrente $\widetilde{x}$.
  - Altrimenti si applica una *strategia di "branching"* per generare nuovi sottoproblemi che vengono inseriti nella lista dei sottoproblemi aperti $\mathcal{L}$.

Iterazione generica

Si esamina la lista $\mathcal{L}$ dei sottoproblemi aperti. Se $\mathcal{L} = \emptyset$ si pone $x^\star = \widetilde{x}$ e $z^\star = \widetilde{z}$ e l'algoritmo termina. Se invece $\mathcal{L} \neq \emptyset$ si estrae un sottoproblema aperto $Prob^{(j)}$ dalla lista $\mathcal{L}$ e si procede come segue:

- si applica una *strategia di "bounding"* per calcolare un "lower bound" L_j del valore ottimo del sottoproblema $Prob^{(j)}$ e un vettore $\bar{x}^{(j)}$ in corrispondenza del quale si ha $L_j = c^T\bar{x}^{(j)}$, con la convenzione di porre $L_j = +\infty$ e chiudendo il problema $Prob^{(j)}$ se esso risultasse inammissibile.
  - Se $L_j \geq \widetilde{z}$ allora il sottoproblema $Prob^{(j)}$ viene chiuso in quanto nessuna soluzione "migliore" dell'ottimo corrente può essere contenuta nell'insieme ammissibile del sottoproblema $Prob^{(j)}$.
  - Se invece $L_j < \widetilde{z}$ il sottoproblema $Prob^{(j)}$ potrebbe permettere di migliorare l'ottimo corrente e quindi
    - se $\bar{x}^{(j)} \in S_j$, allora $\bar{x}^{(j)}$ è soluzione ottima del sottoproblema $Prob^{(j)}$ e poiché si sta considerando il caso $L_j < \widetilde{z}$, si ha che $\bar{x}^{(j)}$ è una soluzione ammissibile del problema originario (in quanto $S_i \subseteq S_0$) in cui il valore $L_j = c^T\bar{x}^{(j)}$ è minore (e pertanto migliore) del valore ottimo corrente. Si aggiorna quindi l'ottimo corrente ponendo $\widetilde{x} = \bar{x}^{(j)}$ e $\widetilde{z} = L_j = c^T\bar{x}^{(j)}$ e si chiude il problema $Prob^{(j)}$.
    - se invece $\bar{x}^{(j)} \notin S_i$ allora si applica una *strategia di "branching"* al problema $Prob^{(j)}$ in modo da generare nuovi sottoproblemi che vengono inseriti nella lista dei problemi candidati $\mathcal{L}$.

Evidentemente la risoluzione del problema originario sarà tanto più efficiente quanto migliori saranno i valori dei "lower bound" ed a loro volta tali valori approssimeranno tanto meglio il valore ottimo del sottoproblema quanto più efficace sarà stata la decomposizione del problema originario. Di conseguenza l'efficienza del metodo del "branch and bound" dipende essenzialmente dalla qualità delle strategie che ne caratterizzano la struttura che sono:

a) *la strategia di bounding*, ovvero la strategia per determinare i "lower bound" cioè per calcolare un valore che approssimi per difetto il valore ottimo dei sottoproblemi.

b) *la strategia di branching*, ovvero la strategia per determinare la partizione dell'insieme delle soluzioni ammissibili di un sottoproblema.

c) *la strategia per la scelta del sottoproblema da esaminare*, ovvero come decidere, ad ogni iterazione, quale sottoproblema selezionare dalla lista $\mathcal{L}$ dei problemi aperti.

Ovviamente a seconda della strategia di bounding, di branching e di scelta del sottoproblema da estrarre da $\mathcal{L}$ adottate, lo schema generale appena descritto si concretizzerà in un algoritmo differente.

Qui di seguito descriveremo alcune delle strategie più utilizzate nella pratica.

Strategie di "bounding".
Esistono varie strategie per il calcolo dei "lower bound". Per descriverne due delle piú utilizzate consideriamo

$$P^i = \{x \in \mathbb{R}^n \ \mid \ A^i x \geq b^i, \ \ x \geq 0\}$$

una possibile formulazione del sottoproblema $Prob^{(i)}$ che quindi può essere scritto nella forma

$$\begin{cases} \min c^T x \\ x \in P^i \cap \mathbf{Z}^n. \end{cases} \qquad (Prob^{(i)})$$

1. *Rilassamento lineare.*
   Un possibile modo di calcolare i "lower bound" è quello di considerare il rilassamento lineare del problema $Prob^{(i)}$ cioè il problema ottenuto eliminando il vincolo di interezza, ovvero il problema

   $$\begin{cases} \min c^T x \\ x \in P^i \end{cases} \qquad (9.3.2)$$

   e di porre

   $$L_i = c^T \bar{x}^{(i)}$$

   dove $\bar{x}^{(i)}$ è soluzione ottima del rilassamento lineare, cioè del problema (9.3.2). Infatti, come abbiamo già visto nel paragrafo 8.3, il valore ottimo del problema rilassato è sempre minore o uguale al valore ottimo del problema intero $Prob^{(i)}$ ed inoltre se la soluzione ottima del problema rilassato è intera, allora essa è anche soluzione ottima del problema $Prob^{(i)}$ (si veda la Proposizione 8.3.1). Si osservi che il problema rilassato è risolubile in maniera molto efficiente (ad esempio con il metodo del simplesso).

2. *Rilassamento della formulazione.*
   Per ottenere un "lower bound" è possibile considerare oltre al rilassamento lineare un qualsiasi altro rilassamento del problema intero $Prob^{(i)}$. Infatti, il valore ottimo di qualunque problema del tipo

   $$\begin{cases} \min c^T x \\ x \in P' \end{cases}$$

con P' poliedro tale che $S_i \subseteq P'$, fornisce un "lower bound" per il problema $Prob^{(i)}$. È però molto importante notare che il poliedro P' non è necessariamente una formulazione del problema $Prob^{(i)}$ e quindi può contenere punti a coordinate intere che non appartengono all'insieme ammissibile S_i del problema $Prob^{(i)}$. Quindi non è piú vero che una soluzione ottima intera del problema rilassato è una soluzione ottima del problema intero

Strategie di "branching".
Vediamo ora una semplice strategia per separare un generico problema $Prob^{(i)}$. Limiteremo la trattazione ad una strategia di tipo binario ovvero ad una strategia che genera sempre due sottoproblemi; nonostante la sua semplicità questa strategia si rivela in pratica molto efficiente.
Supponiamo di aver risolto il rilassamento lineare di $Prob^{(i)}$ e sia, come già detto, $\bar{x}^{(i)}$ la sua soluzione ottima e $L_i = c^T \bar{x}^{(i)}$ il corrispondente valore ottimo.
Se $\bar{x}^{(i)}$ ha tutte componenti intere allora $\bar{x}^{(i)} \in S_i$ e quindi è una soluzione ottima del problema $Prob^{(i)}$ e quindi il problema non va separato, ma chiuso.
Se L_i è maggiore o uguale al valore ottimo corrente $\widetilde{z}$, il problema non può dare origine ad un punto in cui il valore della funzione obiettivo sia migliore di quello corrente e non è necessario separarlo per trovare la sua soluzione ottima intera e quindi va chiuso.
Supponiamo quindi che nessuno di questi due casi si sia verificato, cioè $\bar{x}^{(i)}$ abbia almeno una componente frazionaria e $L_i < \widetilde{z}$, e vediamo come separare questo sottoproblema $Prob^{(i)}$.
Sia $\bar{x}_k^{(i)}$ una componente non intera del vettore $\bar{x}^{(i)}$. Separiamo il problema $Prob^{(i)}$ nei seguenti due problemi:

$$Prob^{(i,1)} \begin{cases} \min c^T x \\ x \in S_i \\ x_k \leq \lfloor \bar{x}_k^{(i)} \rfloor \end{cases} \qquad \text{e} \qquad Prob^{(i,2)} \begin{cases} \min c^T x \\ x \in S_i \\ x_k \geq \lceil \bar{x}_k^{(i)} \rceil \end{cases}$$

dove $\lfloor \bar{x}_k^{(i)} \rfloor$ indica la sua parte intera inferiore (ossia il più grande intero minore di $\bar{x}_k^{(i)}$) e $\lceil \bar{x}_k^{(i)} \rceil$ la sua parte intera superiore (ossia il più piccolo intero maggiore di $\bar{x}_k^{(i)}$). $Prob^{(i,1)}$ è ottenuto da $Prob^{(i)}$ semplicemente aggiungendo a $Prob^{(i)}$ il vincolo $x_k \leq \lfloor \bar{x}_k^{(i)} \rfloor$ e $Prob^{(i,2)}$ aggiungendo a $Prob^{(i)}$ il vincolo $x_k \geq \lceil \bar{x}_k^{(i)} \rceil$. È facile verificare che l'unione delle regioni ammissibili di questi due problemi coincide con la regione ammissibile S_i e che la loro intersezione è vuota; abbiamo così realizzato una partizione di S_i.

Strategie per la scelta del sottoproblema da esaminare.
Esistono diverse strategie di scelta, tra queste le più usate sono le seguenti:

1. Scelta del sottoproblema con il minimo "lower bound". Tale scelta ha lo scopo di esaminare per primi quei sottoproblemi in cui è più probabile

trovare una soluzione ottima. Infatti, se la strategia di bounding produce buone approssimazioni dei valori ottimi dei sottoproblemi, a valori bassi del "lower bound" corrisponderanno bassi valori delle soluzioni ottime dei sottoproblemi. Pertanto, esaminando il sottoproblema aperto cui corrisponde il minimo valore del "lower bound" si avrà una maggiore probabilità di individuare la soluzione ottima del problema originario.

2. Scelta con criterio di priorità LIFO (Last In First Out). In questo caso i sottoproblemi da esaminare sono gestiti dalla procedura secondo lo schema a *pila* (*stack*). In particolare, il sottoproblema scelto in $\mathcal{L}$ è quello che da meno tempo si trova in $\mathcal{L}$.

3. Scelta con criterio di priorità FIFO (First In First Out). In questo caso i sottoproblemi da esaminare sono gestiti dalla procedura secondo lo schema a *coda*. In particolare, il sottoproblema scelto in $\mathcal{L}$ è quello che da più tempo si trova in $\mathcal{L}$.

Integrando lo schema algoritmico già esaminato con queste strategie si ottiene un algoritmo effettivamente realizzabile.

Le Figure 9.3.1 e 9.3.2 riassumono con dei diagrammi di flusso i passi fondamentali di un algoritmo *Branch and Bound* per la soluzione del generico problema di PLI

$$\begin{cases} \min c^T x \\ x \in S = P \cap \mathbf{Z}^n. \end{cases}$$

che utilizza il *rilassamento lineare* per calcolare i lower bound.

Osservazione 9.3.1 Se il problema originario è un problema di Programmazione Lineare 0–1, ovvero le variabili del problema possono assumere solo i valori 0 o 1, si può supporre che la formulazione del problema sia contenuta nell'insieme $\{x \in \mathbb{R}^n \mid 0 \leq x_h \leq 1, \quad h = 1, \ldots, n\}$. Quindi ogni componente $\bar{x}_k^{(i)}$ dell'ottimo $\bar{x}^{(i)}$ del rilassato lineare del problema $Prob^{(i)}$ è compresa tra 0 e 1; quindi poiché ovviamente risulta $\lfloor \bar{x}_k^{(i)} \rfloor = 0$ e $\lceil \bar{x}_k^{(i)} \rceil = 1$, i sottoproblemi che eventualmente vengono generati nella fase di "branching" si ottengono ponendo $x_k = 0$ in uno e $x_k = 1$ nell'altro.

Osservazione 9.3.2 Se il problema di Programmazione Lineare Intera in esame è un problema di massimizzazione invece che di minimizzazione (come assunto fino ad ora), la tecnica del "branch and bound" si applica in maniera analoga sostituendo i "lower bound" L_i con "upper bound" U_i, ovvero con delle limitazioni superiori del valore ottimo del problema intero. Ovviamente l'"upper bound" associato ad un sottoproblema con regione ammissibile vuota sarà posto per convenzione uguale a $-\infty$. Inoltre la chiusura dei sottoproblemi dominati dall'ottimo corrente avverrà nel caso in cui $U_i \leq \widetilde{z}$.

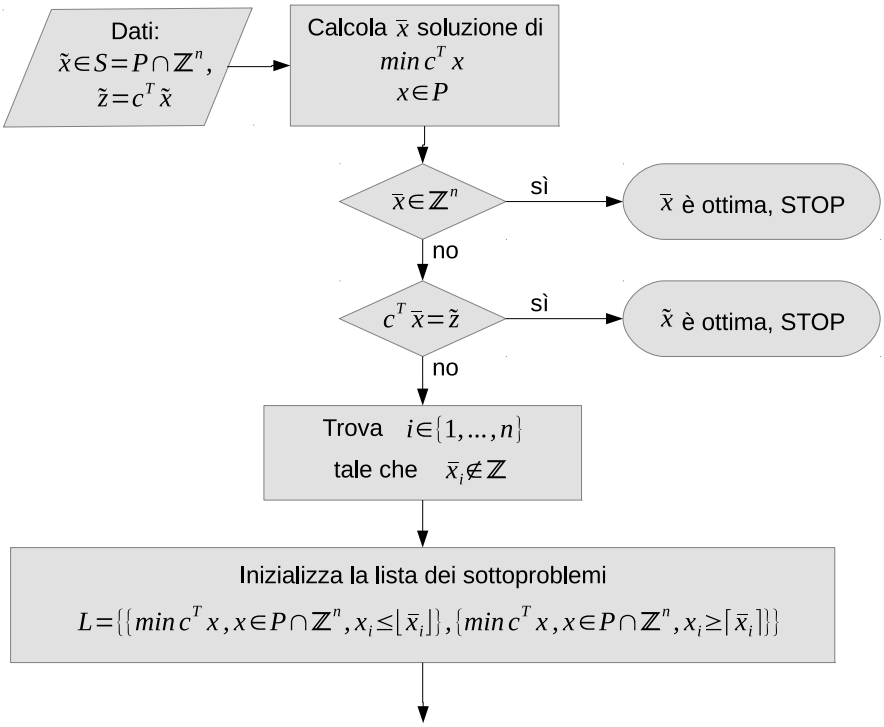

Figura 9.3.1 Inizializzazione del Branch and Bound

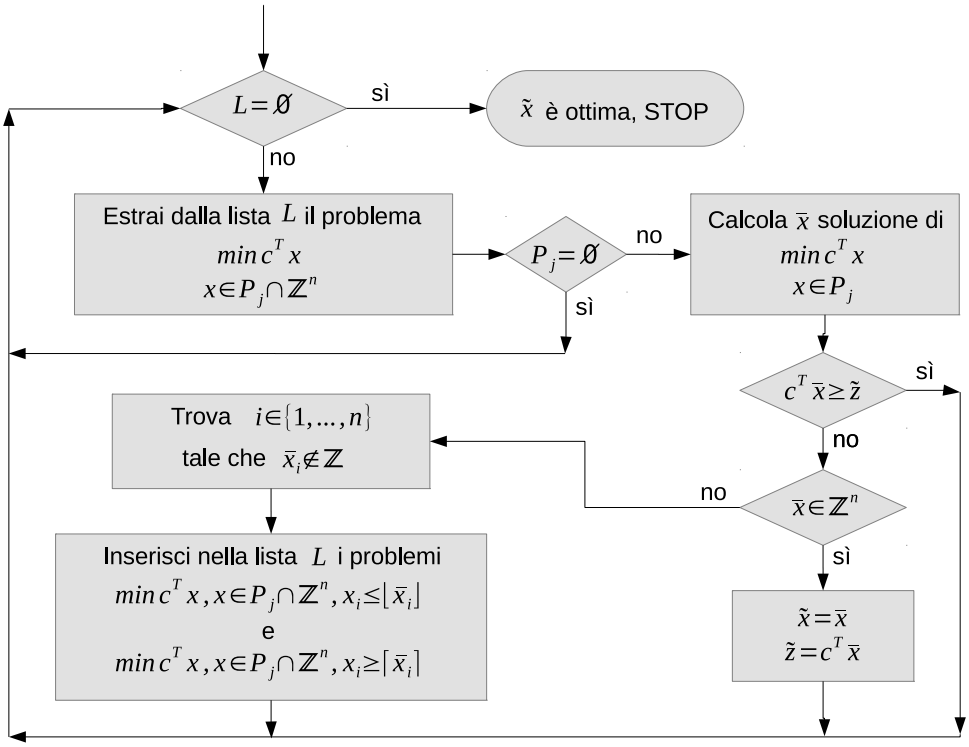

Figura 9.3.2 Generica iterazione del Branch and Bound

Vediamo ora un esempio di applicazione del metodo del "Branch and Bound" appena descritto.

Esempio 9.3.3 *Sia dato il seguente problema lineare intero:*

$$\begin{cases} \min -2x_1 - 8x_2 \\ -2x_1 + 6x_2 \geq -2 \\ x_1 - 4x_2 \geq -5 \\ -2x_1 - 2x_2 \geq -13 \\ x_1 \geq 0, x_2 \geq 0 \\ x_1 \geq 0, x_2 \geq 0, \quad \textit{intere.} \end{cases}$$

Dato un punto ammissibile $(3,2)^T$*, risolvere il problema con la tecnica del Branch and Bound.*

Chiameremo $Prob^{(0)}$ il problema dato. Poiché il problema è in due variabili, possiamo risolvere tutti i rilassamenti lineari per via grafica (si veda la Figura 9.3.3).

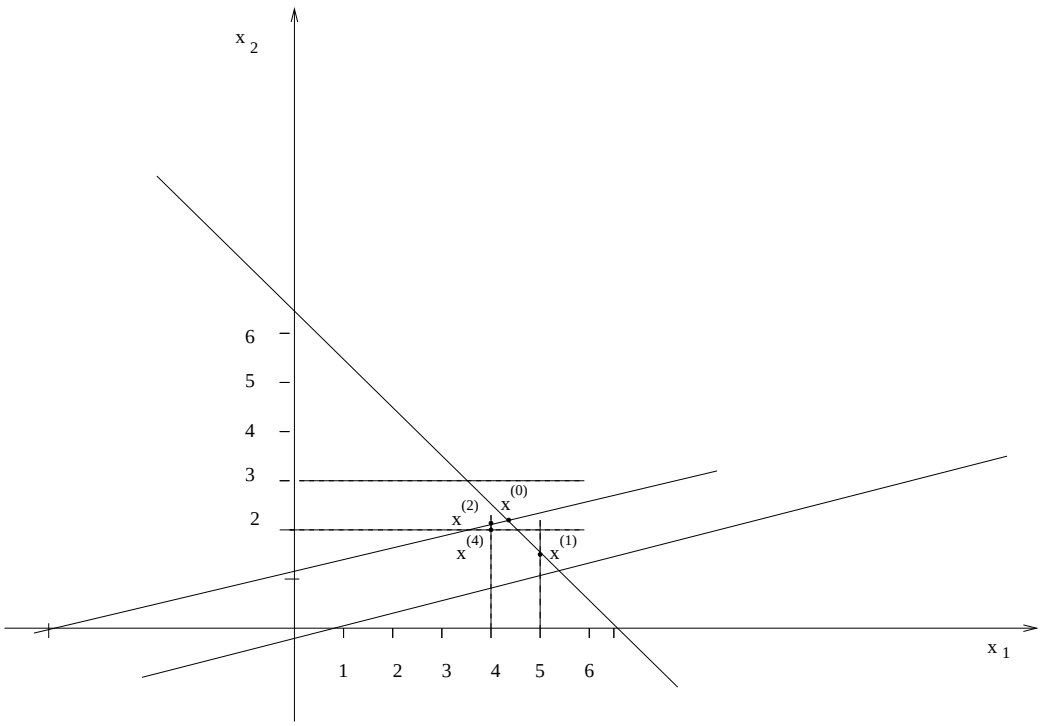

Figura 9.3.3 Rappresentazione grafica del problema dell'Esercizio 9.3.3

Inizializzazione
Innanzitutto inizializziamo l'ottimo corrente utilizzando la soluzione ammissibile fornita, ovvero $\widetilde{x} = (3,2)^T$ e $\widetilde{z} = -22$.

Calcoliamo L_0 risolvendo graficamente il rilassamento lineare di $Prob^{(0)}$ e otteniamo

$$\bar{x}^{(0)} = (21/5, \ \ 23/10)^T, \qquad L_0 = -2 \cdot 21/5 - 8 \cdot 23/10 = -26.8.$$

Poichè $\bar{x}^{(0)}$ non è a componenti intere, non possiamo dichiararlo soluzione del problema. Inoltre poichè $L_0 < \widetilde{z} = -22$, neanche $\widetilde{x}$ può essere dichiarato ottimo del problema. Il vettore $\bar{x}^{(0)}$ non è intero e quindi separiamo rispetto a una sua componente frazionaria, per esempio rispetto a quella con indice piú basso, cioè x_1; vengono allora generati i due sottoproblemi

$$Prob^{(1)} \quad \begin{cases} \min -2x_1 - 8x_2 \\ -2x_1 + 6x_2 \geq -2 \\ x_1 - 4x_2 \geq -5 \\ -2x_1 - 2x_2 \geq -13 \\ x_1 \geq \lceil 21/5 \rceil = 5 \\ x_1 \geq 0, x_2 \geq 0 \\ x_1, x_2 \in Z. \end{cases} \quad \text{e} \quad Prob^{(2)} \begin{cases} \min -2x_1 - 8x_2 \\ -2x_1 + 6x_2 \geq -2 \\ x_1 - 4x_2 \geq -5 \\ -2x_1 - 2x_2 \geq -13 \\ x_1 \leq \lfloor 21/5 \rfloor = 4 \\ x_1 \geq 0, x_2 \geq 0 \\ x_1, x_2 \in Z. \end{cases}$$

e viene inizializzata la lista $\mathcal{L}$:

$$\mathcal{L} = \left\{ Prob^{(1)}, Prob^{(2)} \right\}.$$

Prima iterazione
La lista $\mathcal{L}$ non è vuota, e quindi estraiamo $Prob^{(1)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(2)} \right\}.$$

Siccome $Prob^{(1)}$ risulta ammissibile, risolviamo geometricamente il suo rilassamento lineare. Si ottiene

$$\bar{x}^{(1)} = (5, \ 3/2)^T, \qquad L_1 = -22,$$

Poiché risulta $L_1 = \widetilde{z} = -22$ il problema $Prob^{(1)}$ si può chiudere.

Seconda iterazione
La lista $\mathcal{L}$ non è vuota, e quindi estraiamo l'unico problema in esso contenuto, cioè $Prob^{(2)}$:

$$\mathcal{L} = \emptyset.$$

Siccome $Prob^{(2)}$ risulta ammissibile, risolviamo geometricamente il suo rilassamento lineare. Si ottiene

$$\bar{x}^{(2)} = (4, \ 9/4)^T, \qquad L_2 = -26.$$

Poiché risulta $L_2 = -26 < \widetilde{z} = -22$ ed inoltre $\bar{x}^{(2)}$ non è a componenti intere, il problema $Prob^{(2)}$ non si può chiudere e quindi si effettua un'operazione di separazione rispetto all'unica componente non intera di $\bar{x}^{(2)}$, ovvero la seconda componente; vengono generati i due sottoproblemi

$$Prob^{(3)} \quad \begin{cases} \min -2x_1 - 8x_2 \\ -2x_1 + 6x_2 \geq -2 \\ x_1 - 4x_2 \geq -5 \\ -2x_1 - 2x_2 \geq -13 \\ x_1 \leq 4 \\ x_2 \geq \lceil 9/4 \rceil = 3 \\ x_1 \geq 0, x_2 \geq 0 \\ x_1, x_2 \in Z. \end{cases} \quad \text{e} \quad Prob^{(4)} \quad \begin{cases} \min -2x_1 - 8x_2 \\ -2x_1 + 6x_2 \geq -2 \\ x_1 - 4x_2 \geq -5 \\ -2x_1 - 2x_2 \geq -13 \\ x_1 \leq 4 \\ x_2 \leq \lfloor 9/4 \rfloor = 2 \\ x_1 \geq 0, x_2 \geq 0 \\ x_1, x_2 \in Z. \end{cases}$$

e vengono inseriti nella lista $\mathcal{L}$:

$$\mathcal{L} = \left\{ Prob^{(3)}, Prob^{(4)} \right\}.$$

Terza iterazione
La lista $\mathcal{L}$ non è vuota, e quindi estraiamo $Prob^{(3)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(4)} \right\}.$$

Il $Prob^{(3)}$ risulta inammissibile e quindi si può chiudere.

Quarta iterazione
La lista $\mathcal{L}$ non è vuota, e quindi estraiamo l'unico problema in esso contenuto, cioè $Prob^{(4)}$:

$$\mathcal{L} = \emptyset.$$

Siccome $Prob^{(4)}$ risulta ammissibile, risolviamo geometricamente il suo rilassamento lineare. Si ottiene

$$\bar{x}^{(4)} = (4, 2)^T, \qquad L_4 = -24.$$

Poiché si ha $L_4 = -24 < \widetilde{z} = -22$ e $\bar{x}^{(4)}$ è a componenti intere, si può chiudere il problema $Prob^{(4)}$ e si aggiorna la soluzione ottima corrente e il valore ottimo corrente ponendo

$$\widetilde{x} = (4, 2)^T \qquad \widetilde{z} = -24.$$

Quinta iterazione
Siccome la lista $\mathcal{L}$ è vuota l'algoritmo termina e la soluzione ottima è data da

$$x^\star = \widetilde{x} = (4, 2)^T \qquad \text{con valore ottimo} \qquad z^\star = \widetilde{z} = -24.$$

□

9.3.1 Il problema del knapsack binario.

Dato un problema di knapsack

$$\begin{aligned} &\max c^T x \\ &a^T x \leq b \\ &x \in \{0,1\}^n \end{aligned}$$

il suo rilassamento continuo è

$$\begin{aligned} &\max c^T x \\ &a^T x \leq b \\ &0 \leq x \leq 1. \end{aligned}$$

Possiamo supporre che i coefficienti a_i e c_i, $i = 1, \ldots, n$ siano positivi (vedere osservazione di seguito).
Per determinare la soluzione del problema rilassato si può procedere come segue

(i) Si riordinano le variabili in modo che i rapporti peso ingombro siano ordinati in modo non crescente, cioè

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

(ii) Si determina l'indice critico k per cui

$$\sum_{i=1}^{k} a_i > b \qquad \sum_{i=1}^{k-1} a_i \leq b.$$

(iii) La soluzione ottima è quindi

$$\begin{aligned} &x_i^* = 1, \quad \text{per} \quad i = 1, \ldots, k-1 \\ &x_k^* = \frac{b - \sum_{i=1}^{k-1} a_i}{a_k}, \\ &x_i^* = 0, \quad \text{per} \quad i = k+1, \ldots, n \end{aligned}$$

Osservazione Dato un problema di knapsack continuo limitato, possiamo sempre suppore che il coefficienti a_i e c_i per $i = 1, \ldots, n$ siano positivi. Infatti negli altri casi è possibile ricondursi a questa situazione come descritto nel seguito.

1) La variabile x_i non compare nella funzione obiettivo, ovvero $c_i = 0$. In questo caso si può fissare la variabile x_i in base al segno di a_i:

   – se $a_i > 0$ allora si fissa $x_i = 0$ e il problema può essere riformulato escludendo la variabile i-esima (senza modificare il termine noto b),

- se $a_i < 0$ allora si fissa $x_i = 1$ e il problema può essere riformulato escludendo la variabile i-esima (ovviamente sommando al termine noto b il valore $-a_i$).

2) La variabile x_i non compare nel vincolo, ovvero $a_i = 0$. In questo caso si può fissare la variabile x_i in base al segno di c_i:
   - se $c_i > 0$ allora si fissa $x_i = 1$ e il problema può essere riformulato escludendo la variabile i-esima (senza modificare il termine noto b),
   - se $c_i < 0$ allora si fissa $x_i = 0$ e il problema può essere riformulato escludendo la variabile i-esima (senza modificare il termine noto b).

3) L'i-esimo coefficiente della funzione obiettivo è negativo, ed il corrispondente coefficiente nel vincolo è positivo: $c_i < 0,\ a_i > 0$. In questo caso si può fissare la variabile $x_i = 0$. Il problema può quindi essere riformulato escludendo la variabile i-esima senza modificare il termine noto b.

4) L'i-esimo coefficiente della funzione obiettivo è positivo, ed il corrispondente coefficiente nel vincolo è negativo: $c_i > 0,\ a_i < 0$. In questo caso si può fissare la variabile $x_i = 1$. Il problema può quindi essere riformulato escludendo la variabile i-esima e sommando al termine noto b il valore $-a_i$.

5) Entrambi i coefficienti $c_i < 0,\ a_i < 0$. In questo caso si può sostituire la variabile x_i con un'altra variabile x'_i ponendo $c'_i = -c_i,\ a'_i = -a_i$ (che risultano quindi entrambi positivi) e si somma al termine noto b il valore $-a_i$. Una volta risolto questo problema trasformato è possibile ottenere il volore ottimo della variabile originaria x_i ponendo $x^*_i = 1 - x'^{,*}_i$.

Esempio 9.3.4 *Risolvere con il metodo del Branch and Bound il seguente problema di knapsack binario:*

$$\begin{array}{ll} \max & 1.2x_1 - x_2 + x_3 + x_4 + 1.5x_5 + 0.3x_6 + 0.3x_7 \\ & 2x_1 + 2x_2 + 2x_3 + x_4 + x_5 + x_6 + 2x_7 \leq 3 \\ & x_i \in \{0,1\}, \quad i = 1, \ldots, 7. \end{array}$$

Innanzitutto si fissa $x_2 = 0$. Si deve risolvere il problema nelle variabili rimanenti. Si riordinano le variabili in modo decrescente rispetto al rapporto peso–ingombro c_k/a_k (rinominandole con y); si ottiene il problema

$$Prob^{(0)} \left\{ \begin{array}{l} \max 1.5y_1 + y_2 + 1.2y_3 + y_4 + 0.3y_5 + 0.3y_6 \\ y_1 + y_2 + 2y_3 + 2y_4 + y_5 + 2y_6 \leq 3 \\ y_i \in \{0,1\} \quad i = 1, \ldots, 6. \end{array} \right.$$

Inizializzazione
Una soluzione del rilassamento lineare è data da

$$\bar{y}^{(0)} = \left(1,\ 1, \frac{3-2}{2},\ 0,\ 0,\ 0\right)^T = \left(1,\ 1, \frac{1}{2},\ 0,\ 0,\ 0\right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_0 = 3.1$. Siccome $\bar{y}^{(0)}$ non è intera, essa non può essere la soluzione del problema di knapsack binario. Per inizializzare l'ottimo corrente è sufficiente ottenere una qualunque soluzione ammissibile intera. Ne possiamo individuare una approssimando all'intero inferiore la componente frazionaria di $\bar{y}^{(0)}$, ovvero $\tilde{y} = (1,\ 1,\ 0,\ 0,\ 0,\ 0)^T$. Il valore dell'ottimo corrente è quindi $\tilde{z} = c^T\tilde{y} = 2.5$. Siccome risulta $2.5 = \tilde{z} < U_0 = 3.1$, $\tilde{y}$ non è ottimo.
Si separa rispetto alla variabile y_3 e si ottengono i due sottoproblemi:

$$Prob^{(1)} \begin{cases} \max 1.5y_1 + y_2 + y_4 + 0.3y_5 + 0.3y_6 \\ y_1 + y_2 + 2y_4 + y_5 + 2y_6 \leq 3 \\ y_3 = 0 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

$$Prob^{(2)} \begin{cases} \max 1.5y_1 + y_2 + 1.2 + y_4 + 0.3y_5 + 0.3y_6 \\ y_1 + y_2 + 2y_4 + y_5 + 2y_6 \leq 1 \\ y_3 = 1 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

Si inizializza la lista $\mathcal{L}$:

$$\mathcal{L} = \left\{Prob^{(1)}, Prob^{(2)}\right\}.$$

Prima iterazione
Si estrae $Prob^{(1)}$ dalla lista:

$$\mathcal{L} = \left\{Prob^{(2)}\right\}.$$

Si ottiene:

$$\bar{y}^{(1)} = \left(1,\ 1,\ 0, \frac{1}{2},\ 0,\ 0\right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_1 = 3$. Poiché $U_1 = 3 > 2.5 = \tilde{z}$ e $\bar{y}^{(1)}$ non è intera si separa rispetto alla variabile y_4 e si ottengono i due sottoproblemi:

$$Prob^{(3)} \begin{cases} \max 1.5y_1 + y_2 + 0.3y_5 + 0.3y_6 \\ y_1 + y_2 + y_5 + 2y_6 \leq 3 \\ y_3 = 0 \\ y_4 = 0 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

$$Prob^{(4)} \begin{cases} \max 1.5y_1 + y_2 + 1 + 0.3y_5 + 0.3y_6 \\ y_1 + y_2 + y_5 + 2y_6 \leq 1 \\ y_3 = 0 \\ y_4 = 1 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

e si inseriscono nella lista:

$$\mathcal{L} = \left\{ Prob^{(2)}, Prob^{(3)}, Prob^{(4)} \right\}.$$

Seconda iterazione
Si estrae $Prob^{(2)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(3)}, Prob^{(4)} \right\}.$$

Si ottiene:

$$\bar{y}^{(2)} = (1,\ 0,\ 1,0,\ 0,\ 0)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_2 = 2.7$. Poiché $U_2 = 2.7 > 2.5 = \tilde{z}$ e $\bar{y}^{(2)}$ è intera, si aggiorna l'ottimo corrente $\tilde{y} = \bar{y}^{(2)}$ e $\tilde{z} = 2.7$ e si chiude il $Prob^{(2)}$.

Terza iterazione
Si estrae $Prob^{(3)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(4)} \right\}.$$

Si ottiene:

$$\bar{y}^{(3)} = (1,\ 1,\ 0,0,\ 1,\ 0)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_3 = 2.8$. Poiché $U_3 = 2.8 > 2.7 = \tilde{z}$ e $\bar{y}^{(3)}$ è intera, si aggiorna l'ottimo corrente $\tilde{y} = \bar{y}^{(3)}$ e $\tilde{z} = 2.8$ e si chiude il $Prob^{(3)}$.

Quarta iterazione
Si estrae $Prob^{(4)}$ dalla lista:

$$\mathcal{L} = \emptyset.$$

Si ottiene:

$$\bar{y}^{(4)} = (1,\ 0,\ 0,1,\ 0,\ 0)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_4 = 2.5$. Poiché $U_4 = 2.5 < \tilde{z}$ si chiude il $Prob^{(4)}$.

Quinta iterazione
A questo punto la lista $\mathcal{L}$ è vuota per cui si ha

$$y^* = \tilde{y} = \bar{y}^{(3)} = (1,\ 1,\ 0,\ 0,\ 1,\ 0)^T$$

con valore ottimo pari a 2.8. Nelle variabili originarie la soluzione ottima è $x^* = (0,\ 0,\ 0,1,\ 1,\ 1,\ 0)^T$. □

Esempio 9.3.5 *Risolvere con il metodo del Branch and Bound il seguente problema di knapsack binario:*

$$\begin{array}{ll} \max & 0.8x_1 + 0.6x_2 + 3x_3 + x_4 + 2.2x_5 + 5x_6 + 0.5x_7 \\ & x_1 + 2x_2 + 3x_3 - 2x_4 + 2x_5 + 2x_6 + 2x_7 \leq 4 \\ & x_i \in \{0,1\}, \quad i = 1,\ldots,7. \end{array}$$

Innanzitutto si fissa $x_4 = 1$. Si deve risolvere il problema nelle variabili rimanenti. Si riordinano le variabili in modo decrescente rispetto al rapporto peso–ingombro c_k/a_k (rinominandole con y) e si aumenta di 2 il valore del termine noto (in seguito alla rimozione della variabile x_4); si ottiene il problema

$$Prob^{(0)} \begin{cases} \max 5y_1 + 2.2y_2 + 3y_3 + 0.8y_4 + 0.6y_5 + 0.5y_6 + 1 \\ 2y_1 + 2y_2 + 3y_3 + y_4 + 2y_5 + 2y_6 \leq 6 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

Inizializzazione
Una soluzione del rilassamento lineare è data da

$$\bar{y}^{(0)} = \left(1,\ 1, \frac{6-4}{3},\ 0,\ 0,\ 0\right)^T = \left(1,\ 1, \frac{2}{3},\ 0,\ 0,\ 0\right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_0 = 10.2$. Siccome $\bar{y}^{(0)}$ non è intera, essa non può essere la soluzione del problema di knapsack binario. Per inizializzare l'ottimo corrente è sufficiente ottenere una soluzione ammissibile intera, approssimando all'intero inferiore la componente frazionaria, ovvero $\tilde{y} = (1,\ 1,\ 0,\ 0,\ 0,\ 0)^T$ ed il valore dell'ottimo corrente è $\tilde{z} = 8.2$. Siccome risulta $8.2 = \tilde{z} < U_0 = 10.2$, $\tilde{y}$ non è ottimo.
Si separa rispetto alla variabile y_3 e si ottengono i due sottoproblemi:

$$Prob^{(1)} \begin{cases} \max 5y_1 + 2.2y_2 + 0.8y_4 + 0.6y_5 + 0.5y_6 + 1 \\ 2y_1 + 2y_2 + y_4 + 2y_5 + 2y_6 \leq 6 \\ y_3 = 0 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

$$Prob^{(2)} \begin{cases} \max 5y_1 + 2.2y_2 + 0.8y_4 + 0.6y_5 + 0.5y_6 + 4 \\ 2y_1 + 2y_2 + y_4 + 2y_5 + 2y_6 \leq 3 \\ y_3 = 1 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

Si inizializza la lista $\mathcal{L}$:

$$\mathcal{L} = \left\{ Prob^{(1)}, Prob^{(2)} \right\}.$$

Prima iterazione
Si estrae $Prob^{(1)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(2)} \right\}.$$

Si ottiene:

$$\bar{y}^{(1)} = \left(1,\ 1,\ 0,1,\ \frac{1}{2},\ 0 \right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_1 = 9.3$. Poiché $U_1 = 9.3 > 8.2 = \tilde{z}$ e $\bar{y}^{(1)}$ non è intera, si separa rispetto alla variabile y_5, si ottengono i seguenti due sottoproblemi:

$$Prob^{(3)} \begin{cases} \max 5y_1 + 2.2y_2 + 0.8y_4 + 0.5y_6 + 1 \\ 2y_1 + 2y_2 + y_4 + 2y_6 \leq 6 \\ y_3 = 0 \\ y_5 = 0 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

$$Prob^{(4)} \begin{cases} \max 5y_1 + 2.2y_2 + 0.8y_4 + 0.5y_6 + 1.6 \\ 2y_1 + 2y_2 + y_4 + 2y_6 \leq 4 \\ y_3 = 0 \\ y_5 = 1 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

e si inseriscono nella lista:

$$\mathcal{L} = \left\{ Prob^{(2)}, Prob^{(3)}, Prob^{(4)} \right\}.$$

Seconda iterazione
Si estrae $Prob^{(2)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(3)}, Prob^{(4)} \right\}.$$

Si ottiene:

$$\bar{y}^{(2)} = \left(1,\ \frac{1}{2},\ 1,0,\ 0,\ 0 \right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_2 = 10.1$. Poiché $U_2 = 10.1 > 8.2 = \tilde{z}$ e $\bar{y}^{(2)}$ non è intera, si separa rispetto alla variabile y_2 e si ottengono i seguenti due sottoproblemi:

$$Prob^{(5)} \begin{cases} \max 5y_1 + 0.8y_4 + 0.6y_5 + 0.5y_6 + 4 \\ 2y_1 + y_4 + 2y_5 + 2y_6 \leq 3 \\ y_3 = 1 \\ y_2 = 0 \\ y_i \in \{0,1\} \quad i = 1,\ldots,6. \end{cases}$$

$$Prob^{(6)} \begin{cases} \max 5y_1 + 0.8y_4 + 0.6y_5 + 0.5y_6 + 6.2 \\ 2y_1 + y_4 + 2y_5 + 2y_6 \leq 1 \\ y_3 = 1 \\ y_2 = 1 \\ y_i \in \{0,1\} \quad i = 1, \ldots, 6. \end{cases}$$

e si inseriscono nella lista:

$$\mathcal{L} = \left\{ Prob^{(3)}, Prob^{(4)}, Prob^{(5)}, Prob^{(6)} \right\}.$$

Terza iterazione
Si estrae $Prob^{(5)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(3)}, Prob^{(4)}, Prob^{(6)} \right\}.$$

Si ottiene:

$$\bar{y}^{(5)} = (1,\ 0,\ 1,\ 1,\ 0,\ 0)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_5 = 9.8$. Poiché $U_5 = 9.8 > 8.2 = \tilde{z}$ e $\bar{y}^{(5)}$ è intera, si aggiorna l'ottimo corrente $\tilde{y} = \bar{y}^{(5)}$ e $\tilde{z} = 9.8$ e si chiude il $Prob^{(5)}$.

Quarta iterazione
Si estrae $Prob^{(3)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(4)}, Prob^{(6)} \right\}.$$

Si ottiene:

$$\bar{y}^{(3)} = \left(1,\ 1,\ 0,\ 1,\ 0,\ \frac{1}{2}\right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_3 = 9.25$. Poiché $U_3 = 9.25 < \tilde{z}$ si chiude il $Prob^{(3)}$.

Quinta iterazione
Si estrae $Prob^{(4)}$ dalla lista:

$$\mathcal{L} = \left\{ Prob^{(6)} \right\}.$$

Si ottiene:

$$\bar{y}^{(4)} = (1,\ 1,\ 0,0,\ 1,\ 0)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_4 = 8.8$. Poiché $U_4 = 8.8 < \tilde{z}$ si chiude il $Prob^{(4)}$.

Sesta iterazione
Si estrae $Prob^{(6)}$ dalla lista:

$$\mathcal{L} = \emptyset.$$

Si ottiene:

$$\bar{y}^{(6)} = \left(\frac{1}{2},\ 1,\ 1, 0,\ 0,\ 0\right)^T$$

in corrispondenza della quale abbiamo l'upper bound $U_6 = 8.7$. Poiché $U_6 = 8.7 < \tilde{z}$ si chiude il $Prob^{(6)}$.

Settima iterazione
La lista $\mathcal{L}$ è vuota, quindi si ha

$$y^* = \tilde{y} = \bar{y}^{(5)} = (1,\ 0,\ 1,\ 1,\ 0,\ 0)^T$$

con valore ottimo pari a 9.8. Nelle variabili originarie la soluzione ottima è $x^* = (1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 0)^T$. □

Indice