Programmazione lineare intera

Ipotesi di divisibilità: mentre nella PL le variabili decisionali possono assumere qualsiasi valore reale (numeri interi o decimali), nella PLI **solo valori interi**.

- **PLI pura**: tutte le variabili decisionali devono essere intera
- **PLM mista (programmazione lineare mista)**: alcune variabili sono intere, altre possono essere reali
- **PB (programmazione binaria)**: possono assumere solo i valori 0 o 1

Variabili binarie

Utili per modellare decisioni logiche di tipo "Si/no". Ed anche altri tipi:

- **Vincoli di tipo "almeno uno di N vincoli"**: variabile binaria y_i per ogni vincolo, e un numero molto grande M
  - Se y_i = 1, il vincolo viene allentato da M e non deve essere soddisfatto; se y_i = 0 il vincolo è attivo.
- **La funzione assume solo N possibili valori**: deve assumere uno tra N valori discreti puoi usare N variabili y_i
  - La somma di queste variabili deve essere 1, *solo una di esse sia attiva*
  - Funzione: somma ponderata dei valori possibili
- **Problema del costo fisso**: se un'attività ha un costo iniziale che si applica solo se viene intrapresa (x > 0), si modella
  - Vincolo **x ≤ My** (y: variabile binaria, M: numero molto grande)
    - Se y = 0 l'attività non viene intrapresa, **x deve essere 0**
    - Se y = 1, x può **assumere qualsiasi valore**
- **Rappresentazione binaria di variabili intere generiche**
  - Convertire variabile intera che può assumere qualsiasi valore in **intervallo [0,u]** in una **rappresentazione binaria** usando un **insieme di variabili binarie y_i**

Modellazione di condizioni logiche

Le variabili binarie, combinate con una costante M molto grande, possono essere usate per rappresentare condizioni logiche complesse.

- **Vincolo o-o**: per imporre che almeno uno tra due vincoli sia soddisfatto, si usa una variabile binaria. **Se y=0, il premio è attivo, il secondo è sempre soddisfatto**. **Se y=1, il secondo vincolo è attivo e il primo è sempre soddisfatto**
- **Costo fisso**: per modellare un'attività con un costo fisso che si attiva solo se il livello di attività x è maggiore di 0, si usa una *variabile binaria y*.
  - **Se y=0 il costo fisso non si applica, x deve essere zero**
  - **Se y=1, il costo fisso viene aggiunto alla funzione obiettivo**
- **Rappresentazione di variabili intere**: convertire una variabile intera generica in un insieme di variabili binarie, usando la rappresentazione binaria dei numeri.

Problema del rilassamento lineare

Il numero di soluzioni intere cresce esponenzialmente con il numero di variabili.

Il metodo del simplesso non può essere applicato direttamente perchè le variabili non possono assumere valori continui.
L'arrotondamento di una soluzione ottimale di un problema PL non garantisce nè l'ammissibilità nè l'ottimalità della soluzione intera risultante.

Il rilassamento lineare di un problema PLI è il problema PL identico, ma senza i vincoli di interezza. Utile per due motivi:

- se la soluzione ottima del rilassamento lineare è già un punto intero, allora quella è la soluzione ottima anche per il problema PLI
- Se la soluzione non è intera, il valore della funzione obiettivo del rilassamento lineare fornisce un **limite superiore** (per un problema di massimizzazione) o un **limite inferiore** (per un problema di minimizzazione) per il valore ottimo del problema PLI.

Esempio

Un'azienda deve decidere se costruire una fabbrica e/o magazzino a Saronno o Gallarate, con un budget di **10 milioni di euro**, per **massimizzare** il valore attuale netto.

Variabili decisionali
X1,x2,x3,x4 -> *valgono 1 se si decide di costruire e 0 altrimenti*

Funzione obiettivo
Massimizzare il valore attuale netto totale - max $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

Vincoli

- **Budget**: costo totale non deve superare i 10 milioni
  - $6x_1+3x_2+5x_3+2x_4 \leq 10$
- **Mutua esclusione**: il magazzino può essere costruito o a Saronno o a Gallarate, ma non in entrambe le località
  - $X_3 + x_4 \leq 1$
- **Contingenza**: magazzino può essere costruito solo nella città in cui viene costruita la fabbrica
  - $X_3 \leq x_1$ e $x_4 \leq x_2$

Metodo Branch and Bound

Esplorare sistematicamente l'insieme di tutte le possibili soluzioni, suddividendolo in sottoinsiemi sempre più piccoli, ed eliminando quelli che non possono contenere la soluzione ottima. Tre concetti chiave:

- **Rilassamento**: si risolve il problema di **rilassamento lineare**, ovvero il problema PLI originale senza il vincolo di interezza. La soluzione ottima di questo rilassamento fornisce un limite superiore per il problema PLI *nel caso di massimizzazione*
- **Ramificazione (branch)**: se la soluzione del rilassamento lineare non è intera, si sceglie una variabile con un valore frazionario. Si creano **due nuovi sottoprodotti (rami)** aggiungendo nuovi vincoli che forzano la variabile ad assumere un valore intero

        - Ad esempio, se x = 2.7, si crea il vincolo *$x \leq 2$* e *$x \geq 3$*
- **Potatura (bound)**: ramo dell'albero di ricerca può essere eliminato se si verifica una delle seguenti condizioni:
    - **Inammissibilità**: sottoproblema del rilassamento lineare non ha soluzioni ammissibili
    - **Ottimalità**: soluzione sottoproblema è intera. In questo caso, il valore della funzione obiettivo diventa il **nuovo migliore intero** finora trovato
    - **Superamento del limite**: valore funzione obiettivo del sottoproblema supera il **limite superiore** (massimizzazione) o **inferiore** (minimizzazione) imposto dal miglior intero trovato fino a quel momento.

Il metodo si compone di tre tecniche principali:
- **Branching** (partizione): suddividere il problema in più sotto-problemi più semplici
    - Nel caso dei problemi binari, si fa fissando il valore di una variabile a **0** per un sottoproblema, e a **1** per l'altro
    - Due strategie per scegliere il prossimo nodo da esplorare:
        - **Depth first - più recente**: esplora ramo dell'albero fino a quando non trova una soluzione o un limite. Rischio: esplorare a fondo un sotto-albero con soluzioni di scarsa qualità
        - **Best bound first - miglior limite**: sceglie il nodo più *promettente* (valore Z più alto per massimizzazione).
    - Variabile da usare per il branching: indice più basso, o variabile la cui soluzione non intera è più lontana da un valore intero
- **Bounding** (limite): calcolare un limite superiore (o inferiore se di minimizzazione) per ogni sottoproblema. Si ottiene risolvendo la *versione rilassata* del problema (senza i vincoli di interezza)
    - **Rilassamento lagrangiano**: i vincoli vengono spostati nella funzione obiettivo. La soluzione del problema lagrangiano fornisce un limite almeno *altrettanto buono quanto quello del rilassamento lineare*
- **Fathoming** (eliminazione): scartare i sottoproblemi che non è più necessario analizzare
    - La soluzione ottenuta è **intera** ed **ammissibile**
    - Il limite del sotto-problema è **peggiore** del valore della migliore soluzione intera trovata finora (**soluzione incombente**)
    - Il sottoproblema non ha soluzioni ammissibili

Criteri di arresto alternativi

L'algoritmo standard si ferma quando tutti i nodi sono chiusi (fathomed). Altrimenti, la procedura può terminare quando una soluzione ammissibile Z* si trova entro una certa percentuale alpha del limite superiore Bound.

Metodo del Cutting-Plane

Risolvere il problema di rilassamento lineare e, se la soluzione non è intera, si aggiunge un nuovo **vincolo** (taglio) che:
- non elimina nessuna soluzione intera ammissibile del problema PLI originale
- Elimina la soluzione non intera trovata dal rilassamento lineare.

PROBLEMA RILASSATO $z = 16.5$. I coeff. della fz sono interi, quindi lim. sup. sol. intera $z \leq 16$

Branching:

valore variabile	soluzione	
$x_1 = 0$	$z = 9$	$\rightarrow 9 < 16$ SOLUZIONE INCOMBENTE
$x_1 = 1$	$z \leq 16$	✓ MA NON È INTERO QUINDI È DA ANALIZZARE

Nuovo branching: $x_1 = 1$

$x_2 = 0$	$z \leq 13$	ENTRAMBI NON INTERI BRANCHING SUL VALORE PIÙ ALTO
$x_2 = 1$	$z \leq 16$	

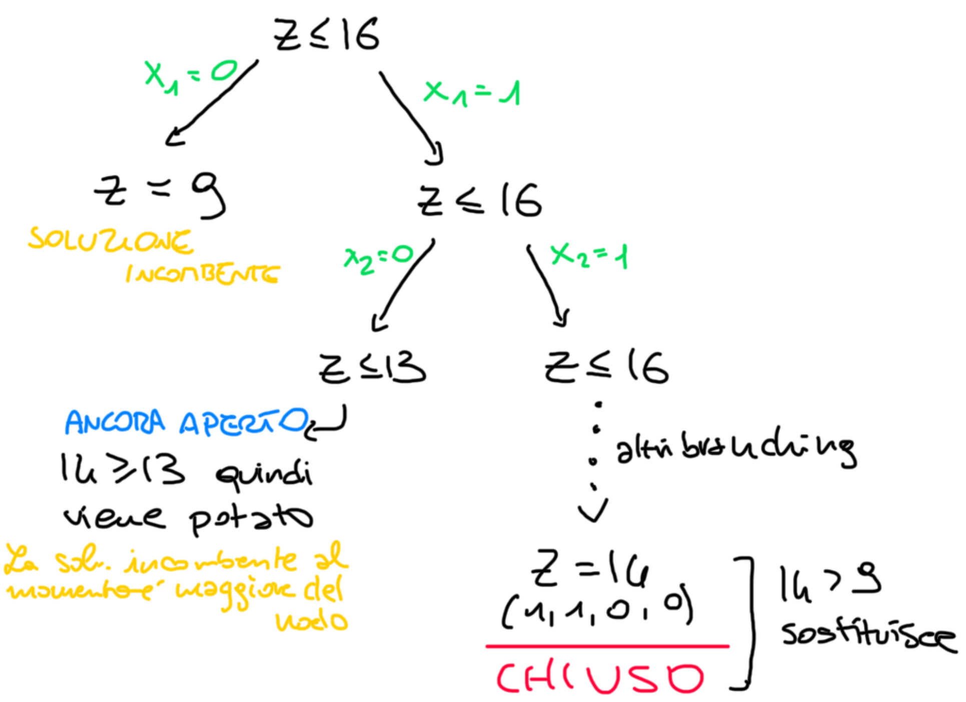
ALTRI BRANCHING...

SOL. TROVATA: $(x_1, x_2, x_3, x_4) = (1,1,0,0)$, $z = 14$

$\Downarrow$

$14 > 9$ sostituisce la soluzione incombente

Rimane aperto 13, quindi si mette a posto quello



B&B per Programmazione Lineare Mista

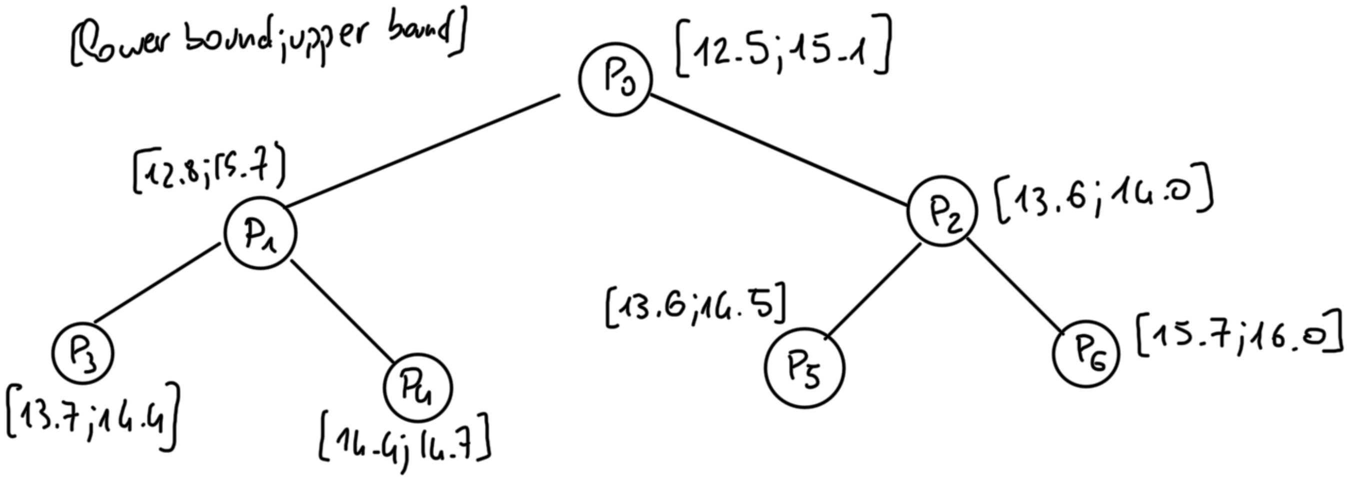
Problemi dove solo alcune variabili devono essere intere. Approccio simile ma con alcune modifiche:

- **Scelta della variabile di branching**: no branching su variabili continue. La scelta si concentra su variabili che non hanno ancora un valore intero

- **Divisione in sotto problemi**: invece di assegnare valori 0 e 1, il problema viene diviso in due sotto-problemi usando vincoli ≤ e ≥ basati sulla parte intera della soluzione

ESERCIZIO 1

Sviluppo B&B con fz. obiettivo di minimo



Come si può capire che è un problema di minimo?

Perché i valori LB di ciascun nodo (quindi il primo numero nelle parentesi) sono **crescenti** ⇒ associati a valutazioni ottimistiche di problemi di minimo

I valori di UB sono **decrescenti**, non possono essere associati a valutazioni ottimistiche di problemi di massimo.

I valori UB sono quindi le valutazioni della fz obiettivo di minimo in soluzioni ammissibili

se un altro nodo ha Lower Bound > Upper Bound attuale, non c'è bisogno di esplorare quel ramo, perché non darà mai una soluzione migliore

Problemi di massimizzazione

Concetto	Significato	Comportamento in albero	Perché è "ottimistico"
Upper bound	Il valore della funzione obiettivo del problema rilassato, senza	**Decresce** o **rimane uguale** di padre in figlio. Ogni volta che ramifichi, il	Il problema rilassato ha **più soluzioni ammissibili** e

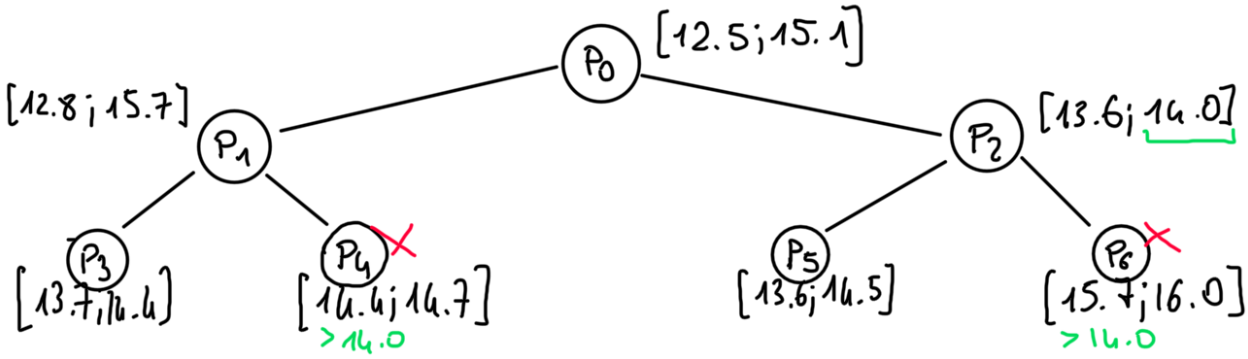
	vincoli di interezza	valore non può aumentare	quindi fornisce la stima **migliore/più alta** che puoi sperare di ottenere per la soluzione intera
Lower bound	Il valore di una soluzione intera ammissibile trovata	Non ha una relazione fissa di crescita o decrescita, ma è il valore che serve come **punto di riferimento**	Questo valore non è *ottimistico* ma è una stima realistica (soluzione valida) che usi per confrontare e potare gli altri rami

Problemi di minimizzazione

Concetto	**Significato**	**Comportamento in albero**	**Perché è "ottimistico"**
Lower bound	Il valore della funzione obiettivo del problema rilassato, senza vincoli di interezza	**Cresce** o **rimane** uguale di padre in figlio. Ogni volta che ramifichi, il valore non può diminuire	Il problema rilassato, essendo meno vincolato, fornisce la stima **più ottimistica/più bassa** che puoi sperare di ottenere
Upper bound	Il valore di una soluzione intera ammissibile trovata	Non ha una relazione fissa di crescita o decrescita, ma è il valore che usi per **confrontare e potare gli altri rami**	Questo valore non è **ottimistico** ma una stima realistica che usi come punto di riferimento per confrontare le soluzioni dei nodi aperti.

Esercizio 2

B&B fz obiettivo di minimo



E' possibile chiudere dei nodi?

visto che è un problema di minimo, dobbiamo vedere

Questa è la prima soluzione ammissibile ed intera. Viene contrassegnata come UB → soluzione incombente

Poi elimino tutti i nodi che hanno LB > soluzione incombente

In quale intervallo è sicuramente compresa la funzione obiettivo?

Abbiamo preso UB = 14.0 come soluzione incombente. Per trovare LB, troviamo il lower bound migliore (valore più basso) negli ultimi nodi figli, quindi 13.6

Quale nodo sarà sviluppato per primo in best first?

Dobbiamo vedere quale è il nodo con il LB più basso tra i figli. Il più basso è P5 [13.6; 14.5]