Si consideri il problema dell'Esempio "PLI generico". Quale sarebbe lo sviluppo dell'albero di B&B con le seguenti varianti:

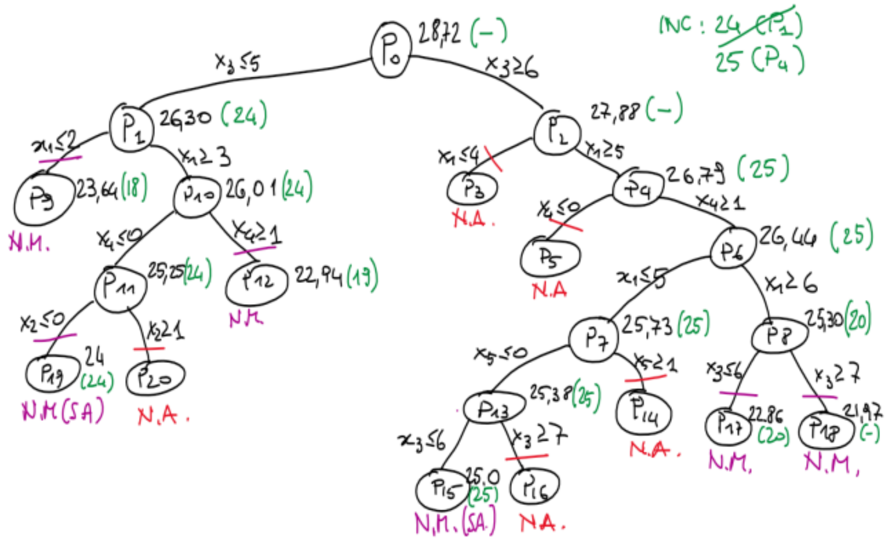
- variante A: provare a generare, ad ogni nodo, una soluzione ammissibile approssimando per difetto la soluzione frazionaria ottenuta con il rilassamento continuo
- variante B: migliorare il bound osservando che tutti i coefficienti e tutte le variabili della funzione obiettivo, **nello specifico problema in esame**, sono interi, quindi il valore della fuzione obiettivo è intero.

Andando a provare fisicamente la versione A delle varianti (soluzione ammissibile approssimando per difetto la soluzione frazionaria) di cui si riporta l'albero completo sotto, alcune osservazioni:

- Tutto ciò che viene segnato non ammissibile è tale perché usando AMPL risulta "infeasible"
- La prima soluzione ammissibile trovata è 24, poi eseguendo il B&B a destra si trova 25
- Rispetto all'albero precedente, l'approssimazione permette di chiudere prima l'albero, ma non di risparmiare il numero di nodi valutati

Esempio "PLI generico - A": soluzione

"n.d.": non disponibile (la soluzione arrotondata non è ammissibile)

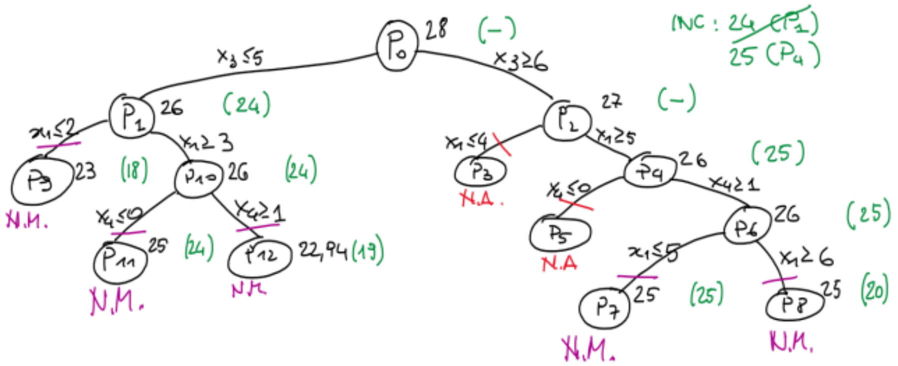


Andando a provare fisicamente la versione B delle varianti (migliorare il bound mettendo ad intero approssimando per difetto [quindi utilizza anche la variante A, ecco perché A+B nell'immagine])la valutazione della f.o. in quanto *nello specifico problema* variabili ed f.o. sono interi), di cui si riporta l'albero completo sotto, alcune osservazioni:

- Approssimare all'intero inferiore individua meno nodi e permette di chiudere subito altri nodi ancora, altrimenti reiterati fino a successivi bound (quindi, si affina meglio la valutazione). Questo funziona, essendo problema di massimo.

Esempio “PLI generico - A e B”: soluzione

"n.d.": non disponibile (la soluzione arrotondata non è ammissibile)



Esercizio 1

Risolvere con il metodo del Branch-and-bound:

$$
\begin{array}{ll}
\min & 1.97x_1 + 3x_2 + 5x_3 + 2.14x_4 + 2x_5 \\
s.t. & -x_1 + 3x_2 + 1x_3 + 2x_4 + x_5 \geq 4 \\
 & 2x_1 + 1.5x_2 + 2x_3 + 3x_4 + x_5 \geq 7 \\
 & x_1, \dots, x_5 \in Z^+
\end{array}
$$

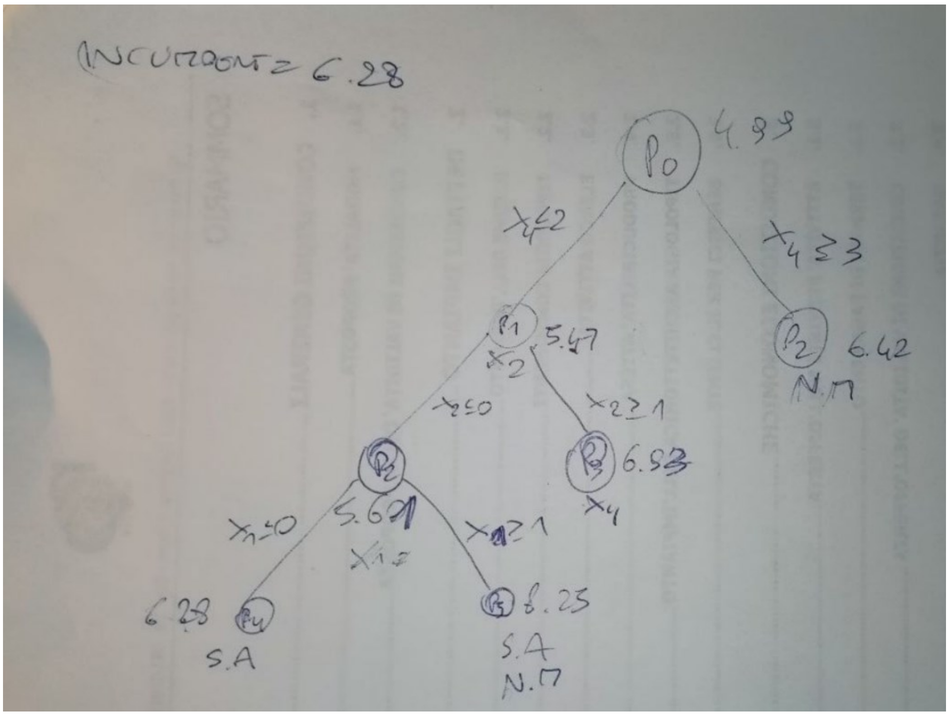
- Branching: binario
- Bound: rilassamento continuo (usare AMPL!)
- Fathoming: standard
- Esplorazione: a piacere (Best Bound First)
- Valutazione soluzioni ammissibili: nessuno (da rilassamento intero)
- Stop: lista nodi aperti vuota

File *.mod*:

```
var x1 >=0;
var x2 >=0;
var x3 >=0;
var x4 >=0;
var x5 >=0;
minimize f: 1.97*x1 + 3*x2 + 5*x3 + 2.14*x4 + 2*x5;
s.t. v1: -x1 + 3*x2 + x3 + 2*x4 + x5 >= 4;
s.t. v2: 2*x1 + 1.5*x2 + 2*x3 + 3*x4 + x5 >= 7;
```

File *.run*:

```
reset;
model es1.mod;
solve;
display f;
display x1,x2,x3,x4,x5;
```



Esercizio 2

Si consideri il problema "Assunzione multiperiodale di personale" e il modello formulato nelle note "Modelli di Programmazione Lineare". Si implementi il modello in AMPL e lo si risolva, per il caso descritto nel testo, con il metodo del Branch-and-Bound, assumento di avere a disposizione soltanto un solver per programmazione lineare a variabili continue.

`assunzionemulti.mod, assunzionemulti.dat`

Per non fare riferimenti a vuoto, inseriamo il relativo modello (pagg. 30/31 dispense "Note di programmazione lineare"):

Variabili decisionali

x_i: neoassunti nel mese i, $\forall\, i \in \{1,2,3,4,5\}$;

y_i: esperti disponibili nel mese i, $\forall\, i \in \{1,2,3,4,5\}$;

w_i: variabile logica legata alla scelta di assumere nel mese i, $\forall\, i \in \{1,2,3,4,5\}$:

$$w_i = \begin{cases} 1 & \text{se si decide di assumere nel mese } i \\ 0 & \text{altrimenti;} \end{cases}$$

z: variabile logica legata alla scelta di ottenere o meno il contributo statale:

$$z = \begin{cases} 1 & \text{se si decide di ottenere il contributo statale} \\ 0 & \text{altrimenti.} \end{cases}$$

Parametri

M: costante sufficientemente elevata (maggiore del numero massimo di apprendist assumibili nei mesi 3, 4 o 5).

Modello PLI

$$\min \quad 500(x_1+x_2+x_3+x_4+x_5)+1000(y_1+y_2+y_3+y_4+y_5)-10000z$$

s.t.

(Mese 1) $y_1 = 20$
$x_1 \leq y_1$
$150(y_1 - x_1) + 70x_1 \geq 2000$

(Mese 2) $y_2 = y_1 + x_1$
$x_2 \leq y_2$
$150(y_2 - x_2) + 70x_2 \geq 4000$

(Mese 3) $y_3 = y_2 + x_2$
$x_3 \leq y_3$
$150(y_3 - x_3) + 70x_3 \geq 7000$

(Mese 4) $y_4 = y_3 + x_3$
$x_4 \leq y_4$
$150(y_4 - x_4) + 70x_4 \geq 3000$

(Mese 5) $y_5 = y_4 + x_4$
$x_5 \leq y_5$
$150(y_5 - x_5) + 70x_5 \geq 3500$

(attiva z) $x_1 + x_2 \geq 10z$
(attiva w) $x_i \leq Mw_i, \forall\, i = 3,4,5$
(limiti) $w_3 + w_4 + w_5 \leq 1$

$x_i, y_i \in \mathbb{Z}_+, \forall\, i \in \{1,2,3,4,5\}$

$z \in \{0,1\}$
$w_i \in \{0,1\}, \forall\, i \in \{3,4,5\}$

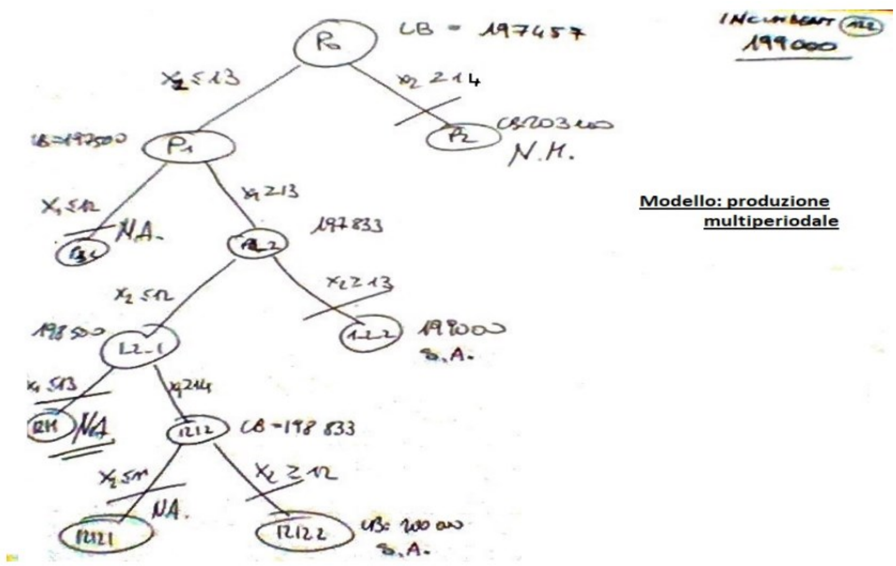
Per il file .mod:

```
### ASSUNZIONE MULTIPERIODALE ###
#################################


#INSIEMI
set mesi;
set mesi_limitati in mesi;
set mesi_iniziali = mesi diff mesi_limitati;

set mSet;

#PARAMETRI
param num_operai_init;
param num;
param den;
param costo_neoassunto := num/den;
param costo_esperto;
param incentivo;
param base_incentivo;
param capacita_operaio;
param capacita_istruttore;
param richiesta{mesi};
param bigM := (sum{i in mesi} richiesta[i]) / capacita_istruttore;

#VARIABILI
var X{mesi} integer >= 0;     #numero neoassunti
var Y{mesi} integer >= 0;     #numero eperti
var Z binary;                 #raccolgo incentivo
var W{mesi_limitati} binary;   #assumo nel mese

#FUNZIONE OBIETTIVO
minimize     costo_totale:
             costo_neoassunto * sum {i in mesi} X[i]
             + costo_esperto  * sum {i in mesi} Y[i]
```

```
            - incentivo      * Z;

#VINCOLI

s.t. operai_iniziali: Y[1] = num_operai_init;
s.t. bilancio_mensile {i in mesi: i > 1}: Y[i] = X[i-1] + Y[i-1];
s.t. sostieni_mensile {i in mesi}: X[i] <= Y[i];
s.t. domanda_mensile  {i in mesi}:
            capacita_operaio * (Y[i]-X[i]) + capacita_istruttore * X[i] >=
richiesta[i];

s.t. attiva_incentivo: sum {i in mesi_iniziali } X[i] >= base_incentivo * Z;

s.t. limite_assunzioni: sum {i in mesi_limitati} W[i] <= 1;
s.t. attiva_W {i in mesi_limitati}: X[i] <= bigM * W[i];
```

Per il file .dat:

```
### ASSUNZIONE MULTIPERIODALE ###
#################################

#INSIEMI
set mesi := 1 2 3 4 5;
set mesi_limitati := 3 4 5;

#PARAMETRI
param num_operai_init := 21;
param num := 1;
param den := 3;
#param costo_neoassunto := 500;
param costo_esperto := 1000;
param incentivo := 10000;
param base_incentivo := 10;
param capacita_operaio := 150;
param capacita_istruttore := 70;
param richiesta :=
1     2000
2     4000
3     7000
4     3000
5     3500
;
```



Modello: produzione multiperiodale