BRANCH AND BOUND FOR AN ILP PROBLEM

STEFANO BIONDI

1. Problem 1

Dato il problema di programmazione lineare intera

$$
\begin{aligned}
&\max \quad 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
&\text{s.t.} \\
&6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\
&x_3 + x_4 \leq 1 \\
&-x_1 + x_3 \leq 0 \\
&-x_2 + x_4 \leq 0 \\
&x_1, x_2, x_3, x_4 \in \{0, 1\}
\end{aligned} \tag{1}
$$

e dato il suo albero delle soluzioni di tutti i possibili problemi rilassati dalla consizione di interezza delle variabili.

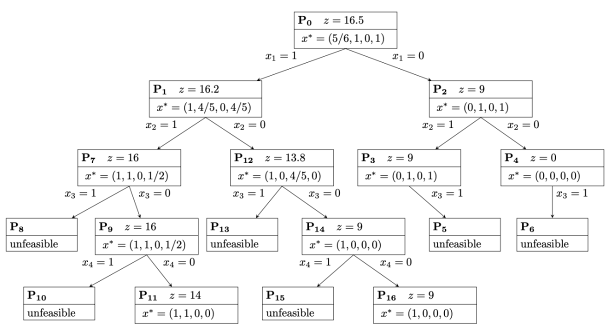

Figura 1. Albero delle soluzioni dei problemi rilassati.

Esercizio 1. Supponendo che il Branch and Bound visiti i sottoproblemi nell'ordine $P_0, P_1, \ldots, P_16$ ecco i passaggi che farebbe:

Inizializzazione: essendo un problema di massimizzazione inizializza $Z_{best} = -\infty$ e risolve il problema P_0, essendo che x^* è a variabili non intere viene eseguita la fase di branching. Notiamo che il problema rilassato genera come risultato della funzione obiettivo 16.5 quindi possiamo dire che la soluzione del ILP è compresa tra $-\infty$ e 16.5

Branching: La prima variabile non intera della soluzione, x_1 viene divisa nella sua parte intera e nella sua parte intera +1 generando i 2 sotto problemi P_1 e P_2. Questo genera 2 problemi nella lista dei problemi candidati alla soluzione.

Bounding P_1: Risolto P_1 con le variabili rilassate abbiamo una soluzione a variabili non intere con valore della funzione obiettivo $Z_{cp} = 16.2 > -\infty$, lower bound. Quindi passiamo al branching di P_1.

Branching P_1: La prima variabile non intera della soluzione, x_2, viene divisa nella sua parte intera e nella sua parte intera $+1$ generando i 2 sotto problemi P_7 e P_12. Quindi la lista dei possibili problemi candidati ora ha P_2, P_7, P_{12}.

Bounding P_2: Risolto P_2 con le variabili rilassate abbiamo una soluzione $x^* = (0, 1, 0, 1)$ a variabili intere. Inoltre essendo che il valore della funzione obiettivo è maggiore di Z_{best}, $9 > -\infty$, impostiamo $Z_{best} = 9$. Quindi anche il lower bound del valore della funzione obiettivo cambierà e avremo che la soluzione del ILP sarà compreso tra 9 e 16.5. Essendo che la lista dei possibili problemi candidati non è vuota si procede al successivo problema.

Bounding P_7: Risolto P_7 con le variabili rilassate abbiamo una soluzione a variabili non intere con valore della funzione obiettivo $Z_{cp} = 16 >$ 9, lower bound. Quindi passiamo al branching di P_7.

Branching P_7: La variabile x_3 viene divisa nella sua parte intera e nella sua parte intera $+1$ generando i 2 sotto problemi P_8 e P_9. Quindi la lista dei possibili problemi candidati ora ha P_8, P_9, P_{12}.

Bounding P_8: Risolto P_8 con le variabili rilassate abbiamo una soluzione infeasible. Quindi la lista dei possibili problemi candidati ora ha P_9, P_{12}.

Bounding P_9: Risolto P_9 con le variabili rilassate abbiamo una soluzione a variabili non intere con valore della funzione obiettivo $Z_{cp} = 16 >$ 9, lower bound. Quindi passiamo al branching di P_9.

Branching P_9: La variabile x_4 viene divisa nella sua parte intera e nella sua parte intera $+1$ generando i 2 sotto problemi P_{10} e P_{11}. Quindi la lista dei possibili problemi candidati ora ha P_{10}, P_{11}, P_{12}.

Bounding P_{10}: Risolto P_{10} con le variabili rilassate abbiamo una soluzione infeasible. Quindi la lista dei possibili problemi candidati ora ha P_{11}, P_{12}.

Bounding P_{11}: Risolto P_{11} con le variabili rilassate abbiamo una soluzione $x^* = (1, 1, 0, 0)$ a variabili intere. Inoltre essendo che il valore della funzione obiettivo è maggiore di Z_{best}, $14 > 9$, impostiamo $Z_{best} = 149$. Quindi anche il lower bound del valore della funzione obiettivo cambierà e avremo che la soluzione del ILP sarà compreso tra 14 e 16.5.La lista dei possibili problemi candidati quindi sarà composta ora dall'unico problema P_{12}.

Bounding P_{12}: Risolto P_{12} con le variabili rilassate abbiamo una soluzione a variabili non intere con valore della funzione obiettivo $Z_{cp} = 13.8 <$ 14, lower bound. Quindi il problema viene eliminato dalla lista che ora risulta vuota.

STOP: La soluzione ottima è stata trovata, $x^* = (1, 1, 0, 0)$ che ha 14 come valore della funzione obiettivo.

In definitiva quindi i nodi valutati dall'algoritmo sono

Problema P_0: inizializzazione, $Z_{real} \in \{-\infty, 16.5\}$
Problema P_1: soluzione non intera, $Z_{real} \in \{-\infty, 16.5\}$
Problema P_2: soluzione intera, $Z_{real} \in \{9, 16.5\}$
Problema P_7: soluzione non intera, $Z_{real} \in \{9, 16.5\}$
Problema P_8: soluzione infeasible, $Z_{real} \in \{9, 16.5\}$
Problema P_9: soluzione non intera, $Z_{real} \in \{9, 16.5\}$
Problema P_{10}: soluzione infeasible, $Z_{real} \in \{9, 16.5\}$
Problema P_{11}: soluzione intera, $Z_{real} \in \{14, 16.5\}$
Problema P_{12}: soluzione non intera con $Z_{cp} < Z_{best}$, $Z_{real} \in \{14, 16.5\}$

Esercizio 2. Risolvendo il problema ILP con la libreria *lpSolveAPI*, impostando le variabili a binary abbiamo che i risultati coincidono.

```
> get.objective(model)
[1] 14
> get.variables(model)
[1] 1 1 0 0
```

FIGURA 2. Risultato ILP utilizzando lpSolveAPI

2. Problem 2

In questo problema si vuole minimizzare il costo totale di assegnamento di clienti dalle regioni $1, 2, 3, 4, 5$ agli uffici Pine Hills, Eustis e Sanford. Nella tabella seguente sono definite le capacità dei 3 uffici, il numero di clienti per ogni regione e il costo medio per cliente assegnato da una regione ad un ufficio.

Region	Pine Hills	Eustis	Sanford	Customers
1	\$6.50	\$7.50	n.a.	30, 000
2	\$7.00	\$8.00	n.a.	40, 000
3	\$8.25	\$7.25	\$6.75	25, 000
4	n.a.	\$7.75	\$7.00	35, 000
5	n.a.	\$7.50	\$6.75	33, 000
Capacity	60, 000	70, 000	40, 000	

Figura 3. Tabella definizione network problem

Esercizio 1. Per disegnare il network flow definiamo le 5 regioni come nodi supply e i i3 uffici come nodi demand. Essendo che in tabella il costo di assegnazione n.a. rappresenta l'impossibilità per quella determinata coppia reigone ufficio vediamo che verranno create 11 variabili

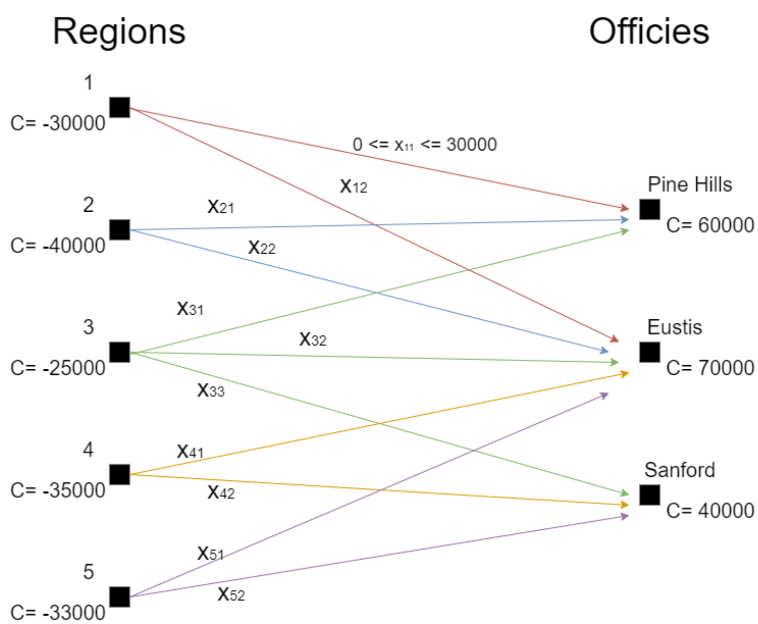

Figura 4. Network Flow

Esercizio 2. La funzione obiettivo da minimizzare è quindi

$$\min 6.5x_{11} + 7.5x_{12} + 7x_{21} + 8x_{22} + 8.25x_{31} + 7.25x_{32} + 6.75x_{33} + 7.75x_{41} + 7x_{42} + 7.5x_{51} + 6.75x_{52}$$

Le constraint sui nodi supply

$$\begin{aligned} x_{11} + x_{12} &\leq 30000 \\ x_{21} + x_{22} &\leq 40000 \\ x_{31} + x_{32} + x_{33} &\leq 25000 \\ x_{41} + x_{42} &\leq 35000 \\ x_{51} + x_{52} &\leq 33000 \end{aligned} \tag{2}$$

Le constraint sui nodi demand

$$\begin{aligned} x_{11} + x_{21} + x_{31} &\leq 60000 \\ x_{12} + x_{22} + x_{32} + x_{41} + x_{51} &\leq 70000 \\ x_{33} + x_{42} + x_{52} &\leq 40000 \end{aligned} \tag{3}$$

La constraint per cui tutti i clienti devono essere assegnati

$$x_{11} + x_{12} + x_{21} + x_{22} + x_{31} + x_{32} + x_{33} + x_{41} + x_{42} + x_{51} + x_{52} = 163000$$

Infine è stato aggiunnto il lower bound per tutte le variabili che devono essere maggiori di 0.

Il modello è stato lanciato con le variabili rilassate, e la soluzione ottima è a variabili intere, quindi non è stato necessario l'utilizzo di algoritmi di ottimimzzazione come il branch and bound.

Esercizio 3. La soluzione ottima è 1155000

```
> get.objective(model)
[1] 1155000
> get.variables(model)
 [1] 20000 10000 40000     0     0 25000     0     0 35000 28000  5000
```

Figura 5. Soluzione Network Flow

Le variabili valgono

$$\begin{aligned} x_{11} &= 20000 \\ x_{12} &= 10000 \\ x_{21} &= 40000 \\ x_{22} &= 0 \\ x_{31} &= 0 \\ x_{32} &= 25000 \\ x_{33} &= 0 \\ x_{41} &= 0 \\ x_{42} &= 35000 \\ x_{51} &= 28000 \\ x_{52} &= 5000 \end{aligned} \tag{4}$$

Quindi dalla regione 1 verranno assegnati 20000 clienti a Pine Hills e 10000 a Eustis, Dalla regione 2 verranno assegnati 40000 clienti a Pine Hills completando la capacità. Dalla regione 3 verranno assegnati tutti i clienti, 25000, a Eustis. I 35000 clienti della regione 4 verranno assegnati tutti a Sanford e i clienti della regione 5 verranno infine assegnati rispettivamente 5000 a Senford, completandone la capacità e 28000 a Eustis.

Eustis è l'unico ufficio che non completerà la capacità avanzando 7000 posti.