

Zaino: branch-and-bound sull'elemento critico

Indice

Perché un algoritmo di branch-and-bound per il KP?

L'algoritmo per il KP

Risoluzione del problema LP_{KP}^k di un nodo

Strategia di scelta della variabile di branching

Che cosa cambia rispetto all'algoritmo generale

Un esempio svolto

Errori tipici

In sintesi

Obiettivi

Alla fine di questo capitolo saremo in grado di:

- ▶ scrivere il problema LP_{KP}^k di un nodo e risolverlo con le formule di Dantzig, senza l'algoritmo del simplesso;
- ▶ fare branching sulla variabile dell'oggetto critico, escludendo o selezionando quell'oggetto;
- ▶ eseguire a mano l'algoritmo di branch-and-bound per il KP, con la regola FIFO o LIFO;
- ▶ spiegare perché l'algoritmo è più semplice di quello generale, e che cosa resta uguale.

Prerequisiti

- ▶ Dal modulo *Programmazione lineare intera*, il capitolo «[Branch-and-bound](#)»: operazioni di branching e di bounding, lista dei problemi, criteri di potatura, lower bound LB e upper bound UB. Qui l'algoritmo generale si applica al problema dello zaino 0/1.
- ▶ Da questo modulo, il capitolo «[Formulazioni dei problemi di zaino](#)»: i modelli ILP_{KP} e LP_{KP} , l'efficienza di un oggetto, l'oggetto critico s , la capacità residua rc e il bound di Dantzig.

Il problema di questo capitolo

Il problema di questo capitolo è il
problema dello zaino 0/1 (KP):

la sua definizione e le sue ipotesi sono nella sua pagina.

Perché un algoritmo di branch-and-bound per il KP?

Introduzione: Perché un algoritmo di branch-and-bound per il KP?

- ▶ L'algoritmo generale di branch-and-bound risolve un problema LP in ogni nodo dell'albero
- ▶ Per il KP questi problemi LP hanno una struttura molto particolare
- ▶ Cominciamo con una piccola sfida, e poi vediamo che cosa diventa l'algoritmo generale per il KP

Una sfida

Uno zaino di capacità 9 e sei oggetti con i seguenti profitti e pesi:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
p_j	6	5	8	9	6	7
w_j	2	3	6	7	5	9

- Trovare il valore ottimo del rilassamento lineare *senza* l'algoritmo del simplesso
- Suggerimento: pensare all'efficienza $\frac{p_j}{w_j}$ di ogni oggetto

Dall'algoritmo generale al KP

- ▶ L'algoritmo generale risolve il problema LP di ogni nodo con l'algoritmo del simplesso duale
- ▶ Per il KP non serve: il problema LP di ogni nodo è ancora un problema LP di zaino, che si risolve ordinando gli oggetti per efficienza
- ▶ Questo capitolo mostra che cosa diventa l'algoritmo generale in questo caso

La sfida:

- ▶ il rilassamento lineare di un'istanza del KP si risolve guardando le efficienze $\frac{p_j}{w_j}$, senza l'algoritmo del simplesso

Branch-and-bound per il KP:

- ▶ il problema LP di ogni nodo è ancora un problema LP di zaino
- ▶ si risolve ordinando gli oggetti per efficienza, non con l'algoritmo del simplesso duale

L'algoritmo per il KP

Introduzione: L'algoritmo per il KP

- ▶ In questa sezione descriviamo un nodo dell'albero di branch-and-bound per il modello ILP_{KP} , e il suo problema LP LP_{KP}^k
- ▶ Vediamo che LP_{KP}^k è ancora un problema LP di zaino, che si risolve con le formule di Dantzig
- ▶ Poi scegliamo la variabile di branching, ed elenchiamo che cosa cambia rispetto all'algoritmo generale

Definizione 1: Nodo di branching e problema LP_{KP}^k (1/2)

Nell'algoritmo di branch-and-bound per il modello ILP_{KP} , i vincoli di branching su una variabile x_j , $j \in \mathcal{J}$, sono $x_j \leq 0$ e $x_j \geq 1$. In un nodo di branching k dell'albero di branch-and-bound abbiamo:

- 1. un sottoinsieme di **oggetti esclusi** $\rightsquigarrow$ indicato con $\mathcal{J}_{out}^k$*
- 2. un sottoinsieme di **oggetti selezionati** $\rightsquigarrow$ indicato con $\mathcal{J}_{in}^k$*
- 3. un sottoinsieme di **oggetti rimanenti** $\rightsquigarrow$ indicato con $\mathcal{J}^k := \mathcal{J} \setminus \{\mathcal{J}_{out}^k \cup \mathcal{J}_{in}^k\}$*

Definizione: Nodo di branching e problema LP_{KP}^k (2/2)

Il problema LP_{KP} associato al nodo k , detto problema LP_{KP}^k , è il seguente:

$$\max \sum_{j \in \mathcal{J}} p_j x_j$$

$$\text{soggetto a } \sum_{j \in \mathcal{J}} w_j x_j \leq c,$$

$$x_j \geq 1, \quad \forall j \in \mathcal{J}_{\text{in}}^k,$$

$$x_j \leq 0, \quad \forall j \in \mathcal{J}_{\text{out}}^k,$$

$$x_j \leq 1, \quad \forall j \in \mathcal{J},$$

$$x_j \geq 0, \quad \forall j \in \mathcal{J}.$$

Escludere e selezionare oggetti

- ▶ I vincoli di branching $x_j \leq 0$ e $x_j \geq 1$ hanno l'effetto di **escludere** o **selezionare** un oggetto.
- ▶ Gli oggetti selezionati danno un profitto costante e usano una parte della capacità: possiamo portarli fuori dal problema

Il problema LP_{KP}^k riscritto

Il problema LP_{KP}^k di un nodo di branching k si può riscrivere in modo equivalente come:

$$\underbrace{\sum_{j \in \mathcal{J}_{in}^k} p_j}_{=p(k)} + \max \sum_{j \in \mathcal{J}^k} p_j x_j$$

$$\text{soggetto a } \sum_{j \in \mathcal{J}^k} w_j x_j \leq c - \underbrace{\sum_{j \in \mathcal{J}_{in}^k} w_j}_{=c(k)},$$

$$x_j \leq 1, \quad \forall j \in \mathcal{J}^k,$$

$$x_j \geq 0, \quad \forall j \in \mathcal{J}^k.$$

Un problema LP di zaino più piccolo

Il problema LP_{KP}^k di un nodo di branching k è **strutturalmente equivalente** al problema LP_{KP} , ma:

1. l'insieme degli oggetti è $\mathcal{J}^k$ invece di $\mathcal{J}$
2. la capacità si riduce a $c(k) := c - \sum_{j \in \mathcal{J}_{in}^k} w_j$
3. c'è una costante nella funzione obiettivo $p(k) := \sum_{j \in \mathcal{J}_{in}^k} p_j$

Risoluzione del problema LP_{KP}^k di un nodo (1/3)

- ▶ Il problema LP_{KP}^k è un problema LP di zaino sugli oggetti di $\mathcal{J}^k$, con capacità $c(k)$: si risolve con le formule di Dantzig, senza alcuna iterazione del simplesso
- ▶ Gli oggetti di $\mathcal{J}^k$ restano ordinati per efficienza non crescente
- ▶ Se $c(k) < 0$ il problema è **inammissibile**
- ▶ Se $\sum_{j \in \mathcal{J}^k} w_j \leq c(k)$ tutti gli oggetti di $\mathcal{J}^k$ entrano, e la soluzione ottima è $\tilde{x}_j = 1$ per ogni $j \in \mathcal{J}^k$: è intera

Risoluzione del problema LP_{KP}^k di un nodo (2/3)

- ▶ Altrimenti l'**oggetto critico** s del nodo (in inglese **split item** o **critical item**) è il primo oggetto di $\mathcal{J}^k$ che non entra per intero
- ▶ La **capacità residua** è la capacità lasciata dagli oggetti che lo precedono:

$$\sum_{j \in \mathcal{J}^k, j < s} w_j \leq c(k) < \sum_{j \in \mathcal{J}^k, j \leq s} w_j, \quad rc = c(k) - \sum_{j \in \mathcal{J}^k, j < s} w_j.$$

Risoluzione del problema LP_{KP}^k di un nodo (3/3)

- ▶ La soluzione ottima è:
 - ▶ $\tilde{x}_j = 1$ per gli oggetti di $\mathcal{J}^k$ prima di s
 - ▶ $\tilde{x}_s = \frac{rc}{w_s}$
 - ▶ $\tilde{x}_j = 0$ per gli oggetti dopo s

- ▶ Il valore ottimo è

$$z(LP_{KP}^k) = p(k) + \sum_{j \in \mathcal{J}^k, j < s} p_j + p_s \frac{rc}{w_s}.$$

Una scansione degli oggetti per nodo

In ogni nodo al più una variabile è frazionaria, quella dell'oggetto critico, e il valore del nodo è il bound di Dantzig di uno zaino più piccolo:

il nodo si risolve con una scansione degli oggetti.

Strategia di scelta della variabile di branching

- ▶ La soluzione ottima di un problema LP_{KP}^k data dalle formule precedenti ha al più una variabile frazionaria, quella associata all'oggetto critico
- ▶ **Il branching si può fare sulla variabile associata all'oggetto critico:** questa strategia di scelta della variabile di branching si chiama strategia dell'oggetto critico
- ▶ La soluzione ottima di un problema LP_{KP} non è unica quando alcuni oggetti hanno la stessa efficienza: le altre soluzioni ottime possono avere più variabili frazionarie, per esempio $x = (\frac{1}{2}, \frac{1}{2})$ per $p = (2, 2)$, $w = (2, 2)$, $c = 2$
- ▶ Qui, come nell'algoritmo generale, usiamo sempre la soluzione data dalle formule

Che cosa cambia rispetto all'algoritmo generale (1/3)

- ▶ L'algoritmo è il branch-and-bound basato sulla PL del modulo *Programmazione lineare intera* ([capitolo «Branch-and-bound»](#)), con la sua lista dei problemi, i suoi criteri di potatura e il suo gap di ottimalità
- ▶ Cambiano le tre operazioni: **bounding**, **branching** e **scelta del nodo**

Che cosa cambia rispetto all'algoritmo generale (2/3)

Bounding

- ▶ Il problema di un nodo si risolve con le formule precedenti
- ▶ Tutti i profitti sono interi, quindi il bound di un nodo è $\lfloor z(LP_{KP}^k) \rfloor$
- ▶ Il nodo è potato:
 - ▶ per bound se $\lfloor z(LP_{KP}^k) \rfloor \leq LB$
 - ▶ per inammissibilità se $c(k) < 0$
 - ▶ per interezza se la sua soluzione è intera (e allora LB si aggiorna quando il valore è maggiore)
- ▶ Un nodo intero è potato per interezza anche quando il suo valore non supera LB

Che cosa cambia rispetto all'algoritmo generale (3/3)

Branching

- ▶ La variabile di branching è quella dell'oggetto critico s : i figli sono $x_s \leq 0$ (down, oggetto escluso) e $x_s \geq 1$ (up, oggetto selezionato)
- ▶ Il nodo non si risolve con un tableau del simplesso ma si descrive con gli insiemi $\mathcal{J}_{\text{in}}^k$, $\mathcal{J}_{\text{out}}^k$ e $\mathcal{J}^k$

Scelta del nodo

- ▶ Il nodo della lista dei problemi si estrae con la regola FIFO o LIFO
- ▶ Il nodo di down-branching si risolve prima del nodo di up-branching; i nodi sono numerati nell'ordine di risoluzione
- ▶ Il lower bound scritto in un nodo è il valore al momento della sua risoluzione

Riepilogo (1/2)

Nodi:

- ▶ un nodo k è descritto dagli insiemi $\mathcal{J}_{\text{in}}^k$, $\mathcal{J}_{\text{out}}^k$ e $\mathcal{J}^k$
- ▶ il suo problema LP_{KP}^k è un problema LP di zaino su $\mathcal{J}^k$, con capacità $c(k)$ e costante $p(k)$

Risoluzione di un nodo:

- ▶ inammissibile se $c(k) < 0$, intera se tutti gli oggetti di $\mathcal{J}^k$ entrano
- ▶ altrimenti le formule di Dantzig con l'oggetto critico s e la capacità residua rc

Riepilogo (2/2)

Branching:

- ▶ sulla variabile dell'oggetto critico: $x_s \leq 0$ (escluso) e $x_s \geq 1$ (selezionato)

Il resto dell'algoritmo generale:

- ▶ il bound di un nodo è $\lfloor z(LP_{KP}^k) \rfloor$
- ▶ lista dei problemi, criteri di potatura e gap di ottimalità restano gli stessi
- ▶ regola FIFO o LIFO, prima il nodo di down-branching

Un esempio svolto

Introduzione: Un esempio svolto

- ▶ In questa sezione eseguiamo a mano l'algoritmo di branch-and-bound per il KP, sull'istanza della sfida
- ▶ In ogni nodo scriviamo gli insiemi di oggetti, la capacità $c(k)$, la costante $p(k)$, l'oggetto critico e la capacità residua
- ▶ Disegniamo l'albero di branch-and-bound alla fine di ogni iterazione

Esempio: il modello ILP_{KP} (1/3)

Consideriamo uno zaino di capacità $c = 9$ e un insieme $\mathcal{I} = \{1, 2, 3, 4, 5, 6\}$ di 6 oggetti ($n = 6$), con i seguenti profitti e pesi:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
p_j	6	5	8	9	6	7
w_j	2	3	6	7	5	9

Esempio: il modello ILP_{KP} (2/3)

Il modello ILP_{KP} per questa istanza del KP è:

$$\begin{aligned} \max \quad & 6x_1 + 5x_2 + 8x_3 + 9x_4 + 6x_5 + 7x_6 \\ \text{soggetto a} \quad & 2x_1 + 3x_2 + 6x_3 + 7x_4 + 5x_5 + 9x_6 \leq 9 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \in \{0, 1\} \end{aligned}$$

Esempio: il modello ILP_{KP} (3/3)

Una soluzione ottima $\tilde{\mathbf{x}}(\text{ILP}_{\text{KP}})$ è:

$$\tilde{x}_1 = 1, \quad \tilde{x}_2 = 0, \quad \tilde{x}_3 = 0, \quad \tilde{x}_4 = 1, \quad \tilde{x}_5 = 0, \quad \tilde{x}_6 = 0$$

Il valore ottimo $z(\text{ILP}_{\text{KP}})$ è 15. La soluzione ottima associata del KP è $\tilde{S} = \{1, 4\}$.

Gli oggetti sono ordinati per efficienza non crescente, poiché:

$$\frac{6}{2} \geq \frac{5}{3} \geq \frac{8}{6} \geq \frac{9}{7} \geq \frac{6}{5} \geq \frac{7}{9}$$

Esempio: branch-and-bound per il modello ILP_{KP}

Un algoritmo di branch-and-bound per il modello ILP_{KP} e l'istanza del KP precedente che usa:

1. la strategia dell'oggetto critico come strategia di scelta della variabile di branching;
2. la regola FIFO come strategia di scelta del nodo di branching;
3. risolvere prima il nodo di down-branching e poi il nodo di up-branching;
4. porre la soluzione incumbent $\bar{\mathbf{x}}(ILP_{KP})$ uguale alla seguente soluzione euristica:

$$\bar{x}_1 = 1, \quad \bar{x}_2 = 0, \quad \bar{x}_3 = 0, \quad \bar{x}_4 = 1, \quad \bar{x}_5 = 0, \quad \bar{x}_6 = 0$$

e porre di conseguenza il lower bound LB.

Esempio: inizializzazione (1/2)

- ▶ Per il problema LP_{KP}^1 , l'oggetto critico è $s = 3$ e la capacità residua è $rc = 4$
- ▶ Una soluzione ottima $\tilde{\mathbf{x}}(LP_{KP}^1)$ è:

$$\tilde{x}_1 = 1, \quad \tilde{x}_2 = 1, \quad \tilde{x}_3 = \frac{2}{3}, \quad \tilde{x}_4 = 0, \quad \tilde{x}_5 = 0, \quad \tilde{x}_6 = 0$$

- ▶ Il valore ottimo è:

$$z(LP_{KP}^1) = 6 \cdot 1 + 5 \cdot 1 + 8 \cdot \frac{2}{3} = \frac{49}{3}$$

Esempio: inizializzazione (2/2)

- ▶ La soluzione ottima $\tilde{\mathbf{x}}(\text{LP}_{\text{KP}}^1)$ è **frazionaria** e il problema LP_{KP}^1 è **ammissibile**
- ▶ Il problema LP_{KP}^1 viene inserito nella lista dei problemi L e il lower bound LB viene posto a 15 ($\text{LB}(\bar{\mathbf{x}}) = 15$)
- ▶ Dati il lower bound LB e l'upper bound UB correnti, abbiamo:

$$15 \leq z(\text{ILP}_{\text{KP}}) \leq \underbrace{\left\lfloor \frac{49}{3} \right\rfloor}_{=16}$$

Esempio: iterazione 1 (1/5)

- ▶ Il problema LP_{KP}^1 viene estratto e cancellato dalla lista dei problemi L
- ▶ La variabile di branching è x_3
- ▶ I problemi LP_{KP}^2 e LP_{KP}^3 vengono creati escludendo ($x_3 \leq 0$) e selezionando ($x_3 \geq 1$) l'oggetto 3

Esempio: iterazione 1 (2/5)

Soluzione del problema LP_{KP}^2

- ▶ Abbiamo

$$\mathcal{J}^2 = \{1, 2, 4, 5, 6\}, \quad \mathcal{J}_{\text{out}}^2 = \{3\}, \quad \mathcal{J}_{\text{in}}^2 = \emptyset$$

- ▶ La capacità è $c(2) = 9$ e la costante nella funzione obiettivo è $p(2) = 0$
- ▶ L'oggetto critico è $s = 4$ e la capacità residua è $rc = 4$
- ▶ Una soluzione ottima $\tilde{\mathbf{x}}(LP_{KP}^2)$ è:

$$\tilde{x}_1 = 1, \quad \tilde{x}_2 = 1, \quad \tilde{x}_4 = \frac{4}{7}, \quad \tilde{x}_5 = 0, \quad \tilde{x}_6 = 0$$

e

$$z(LP_{KP}^2) = 6 \cdot 1 + 5 \cdot 1 + 9 \cdot \frac{4}{7} = \frac{113}{7}$$

Esempio: iterazione 1 (3/5)

Soluzione del problema LP_{KP}^2

- ▶ Il problema LP_{KP}^2 è **ammissibile** e abbiamo:

$$\lfloor z(LP_{KP}^2) \rfloor > LB \quad \text{poiché} \quad \underbrace{\left\lfloor \frac{113}{7} \right\rfloor}_{=16} > 15$$

- ▶ La soluzione ottima $\tilde{x}(LP_{KP}^2)$ è **frazionaria**, quindi il problema LP_{KP}^2 viene **inserito nella lista dei problemi L**

Esempio: iterazione 1 (4/5)

Soluzione del problema LP_{KP}^3

- ▶ Abbiamo

$$\mathcal{J}^3 = \{1, 2, 4, 5, 6\}, \quad \mathcal{J}_{\text{out}}^3 = \emptyset, \quad \mathcal{J}_{\text{in}}^3 = \{3\}$$

- ▶ La capacità è $c(3) = 3$ e la costante nella funzione obiettivo è $p(3) = 8$
- ▶ L'oggetto critico è $s = 2$ e la capacità residua è $rc = 1$
- ▶ Una soluzione ottima $\tilde{\mathbf{x}}(LP_{KP}^3)$ è:

$$\tilde{x}_1 = 1, \quad \tilde{x}_2 = \frac{1}{3}, \quad \tilde{x}_4 = 0, \quad \tilde{x}_5 = 0, \quad \tilde{x}_6 = 0$$

e

$$z(LP_{KP}^3) = 8 + 6 \cdot 1 + 5 \cdot \frac{1}{3} = \frac{47}{3}$$

Esempio: iterazione 1 (5/5)

Soluzione del problema LP_{KP}^3

- ▶ Il problema LP_{KP}^3 è **ammissibile** e abbiamo:

$$\lfloor z(LP_{KP}^3) \rfloor \leq LB \quad \text{poiché} \quad \underbrace{\left\lfloor \frac{47}{3} \right\rfloor}_{=15} \leq 15$$

- ▶ Quindi il nodo di branching è potato per upper bound

Esempio: l'albero dopo l'iterazione 1

- ▶ Alla fine della prima iterazione, il Nodo 2 è inserito nella lista dei problemi L , mentre il Nodo 3 è potato per upper bound
- ▶ L'albero di branch-and-bound corrispondente ha tre nodi: il riquadro di ogni nodo contiene il problema LP del nodo, la sua soluzione ottima e i bound
- ▶ I riquadri dei nodi sono grandi: li guardiamo uno alla volta, nelle prossime tre slide

Esempio: l'albero dopo l'iterazione 1 (1/3)

Nodo 1

$$\max 6x_1 + 5x_2 + 8x_3 + 9x_4 + 6x_5 + 7x_6$$

$$2x_1 + 3x_2 + 6x_3 + 7x_4 + 5x_5 + 9x_6 \leq 9$$

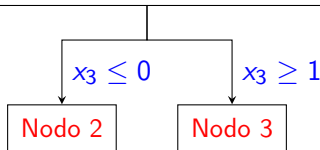
$$x_1, x_2, x_3, x_4, x_5, x_6 \leq 1$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

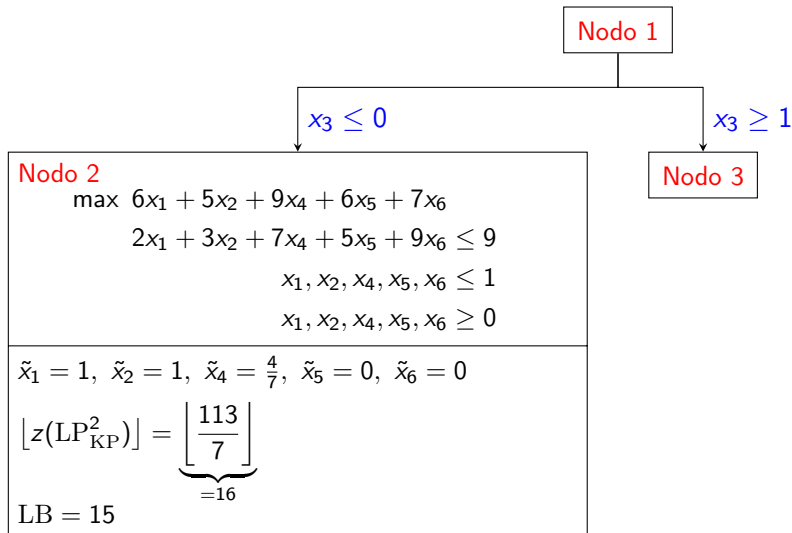
$$\tilde{x}_1 = 1, \tilde{x}_2 = 1, \tilde{x}_3 = \frac{2}{3}, \tilde{x}_4 = 0, \tilde{x}_5 = 0, \tilde{x}_6 = 0$$

$$\lfloor z(\text{LP}_{\text{KP}}^1) \rfloor = \underbrace{\left\lfloor \frac{49}{3} \right\rfloor}_{=16}$$

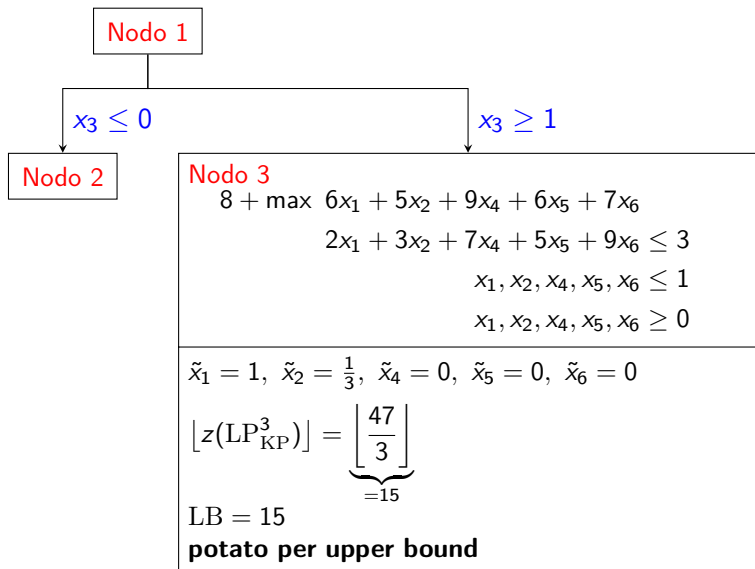
$$\text{LB} = 15$$



Esempio: l'albero dopo l'iterazione 1 (2/3)



Esempio: l'albero dopo l'iterazione 1 (3/3)



Esempio: i bound dopo l'iterazione 1

- ▶ Il Nodo 2 è l'unico nodo della lista dei problemi L
- ▶ Dati il lower bound LB e l'upper bound UB correnti, abbiamo:

$$15 \leq z(\text{ILP}_{\text{KP}}) \leq \underbrace{\left\lfloor \frac{113}{7} \right\rfloor}_{=16}$$

Esempio: iterazione 2 (1/5)

- ▶ Il problema LP_{KP}^2 viene estratto e cancellato dalla lista dei problemi L
- ▶ La variabile di branching è x_4
- ▶ I problemi LP_{KP}^4 e LP_{KP}^5 vengono creati escludendo ($x_4 \leq 0$) e selezionando ($x_4 \geq 1$) l'oggetto 4

Esempio: iterazione 2 (2/5)

Soluzione del problema LP_{KP}^4

- ▶ Abbiamo

$$\mathcal{I}^4 = \{1, 2, 5, 6\}, \quad \mathcal{I}_{out}^4 = \{3, 4\}, \quad \mathcal{I}_{in}^4 = \emptyset$$

- ▶ La capacità è $c(4) = 9$ e la costante nella funzione obiettivo è $p(4) = 0$
- ▶ L'oggetto critico è $s = 5$ e la capacità residua è $rc = 4$
- ▶ Una soluzione ottima $\tilde{\mathbf{x}}(LP_{KP}^4)$ è:

$$\tilde{x}_1 = 1, \quad \tilde{x}_2 = 1, \quad \tilde{x}_5 = \frac{4}{5}, \quad \tilde{x}_6 = 0$$

e

$$z(LP_{KP}^4) = 6 \cdot 1 + 5 \cdot 1 + 6 \cdot \frac{4}{5} = \frac{79}{5}$$

Esempio: iterazione 2 (3/5)

Soluzione del problema LP_{KP}^4

- Il problema LP_{KP}^4 è **ammissibile** e abbiamo:

$$\lfloor z(LP_{KP}^4) \rfloor \leq LB \quad \text{poiché} \quad \underbrace{\left\lfloor \frac{79}{5} \right\rfloor}_{=15} \leq 15$$

- Quindi il nodo di branching è potato per upper bound

Esempio: iterazione 2 (4/5)

Soluzione del problema LP_{KP}^5

- ▶ Abbiamo

$$\mathcal{J}^5 = \{1, 2, 5, 6\}, \quad \mathcal{J}_{\text{out}}^5 = \{3\}, \quad \mathcal{J}_{\text{in}}^5 = \{4\}$$

- ▶ La capacità è $c(5) = 2$ e la costante nella funzione obiettivo è $p(5) = 9$
- ▶ L'oggetto critico è $s = 2$ e la capacità residua è $rc = 0$
- ▶ Una soluzione ottima $\tilde{\mathbf{x}}(LP_{KP}^5)$ è:

$$\tilde{x}_1 = 1, \quad \tilde{x}_2 = 0, \quad \tilde{x}_5 = 0, \quad \tilde{x}_6 = 0$$

e

$$z(LP_{KP}^5) = 9 + 6 \cdot 1 = 15$$

Esempio: iterazione 2 (5/5)

Soluzione del problema LP_{KP}^5

- ▶ Il problema LP_{KP}^5 è **ammissibile** e la sua soluzione ottima $\tilde{x}(LP_{KP}^5)$ è **intera**
- ▶ Il suo valore non migliora la soluzione incumbent, perché $z(LP_{KP}^5) = 15 \leq 15 = LB$, quindi il lower bound LB resta 15
- ▶ Di conseguenza, il nodo di branching è potato per interezza: quando un nodo si potrebbe potare sia per interezza sia per bound, lo si pota per interezza

Esempio: l'albero dopo l'iterazione 2

- ▶ Alla fine della seconda iterazione, il Nodo 4 è potato per upper bound e il Nodo 5 è potato per interezza
- ▶ L'albero di branch-and-bound corrispondente è il sottoalbero del Nodo 2: come prima, guardiamo i suoi tre riquadri uno alla volta

Esempio: l'albero dopo l'iterazione 2 (1/3)

Nodo 2

$$\max 6x_1 + 5x_2 + 9x_4 + 6x_5 + 7x_6$$

$$2x_1 + 3x_2 + 7x_4 + 5x_5 + 9x_6 \leq 9$$

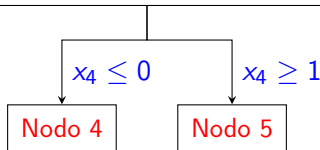
$$x_1, x_2, x_4, x_5, x_6 \leq 1$$

$$x_1, x_2, x_4, x_5, x_6 \geq 0$$

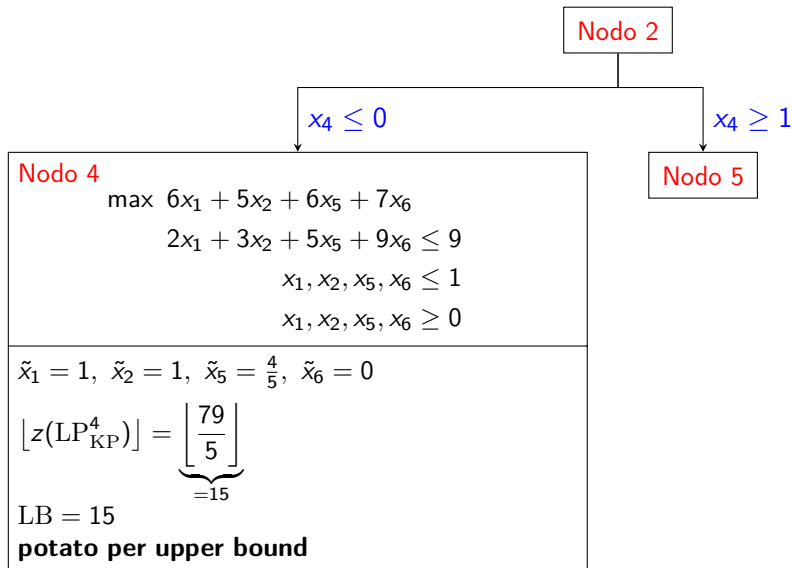
$$\tilde{x}_1 = 1, \tilde{x}_2 = 1, \tilde{x}_4 = \frac{4}{7}, \tilde{x}_5 = 0, \tilde{x}_6 = 0$$

$$\lfloor z(\text{LP}_{\text{KP}}^2) \rfloor = \underbrace{\left\lfloor \frac{113}{7} \right\rfloor}_{=16}$$

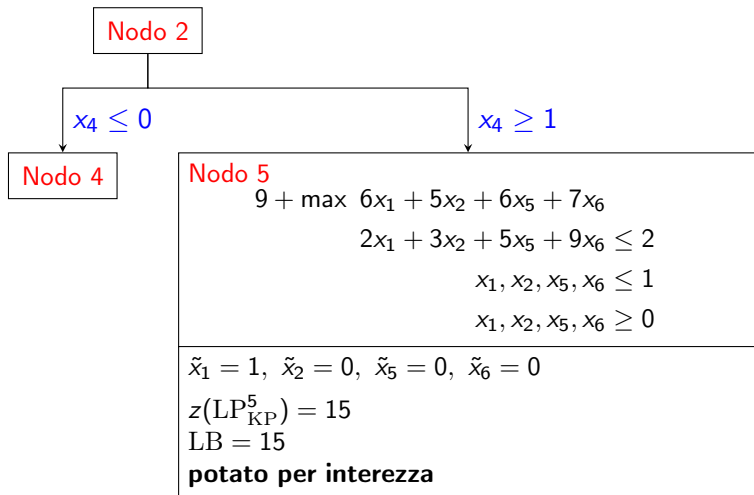
$$\text{LB} = 15$$



Esempio: l'albero dopo l'iterazione 2 (2/3)



Esempio: l'albero dopo l'iterazione 2 (3/3)



Esempio: terminazione

- ▶ Poiché la lista dei problemi L è vuota, il problema ILP_{KP} è risolto all'ottimo
- ▶ Il valore ottimo è $z(ILP_{KP}) = 15$ (il valore del lower bound LB)
- ▶ Una soluzione ottima $\tilde{\mathbf{x}}(ILP_{KP})$ è

$$\tilde{x}_1 = 1, \tilde{x}_2 = 0, \tilde{x}_3 = 0, \tilde{x}_4 = 1, \tilde{x}_5 = 0, \tilde{x}_6 = 0$$

(la soluzione incumbent $\tilde{\mathbf{x}}(ILP_{KP})$)

Esempio: l'albero di branch-and-bound

Questo è l'albero di branch-and-bound della precedente esecuzione dell'algoritmo di branch-and-bound, che riporta per ogni nodo di branching:

il valore ottimo del problema LP_{KP} associato,
il lower bound LB
e i criteri di potatura.

Nodo 1

$$\lfloor z(\text{LP}_{\text{KP}}^1) \rfloor = \underbrace{\left\lfloor \frac{49}{3} \right\rfloor}_{=16}$$

$$\text{LB} = 15$$

Nodo 2

$$x_3 \leq 0$$

$$\lfloor z(\text{LP}_{\text{KP}}^2) \rfloor = \underbrace{\left\lfloor \frac{113}{7} \right\rfloor}_{=16}$$

$$\text{LB} = 15$$

$$x_3 \geq 1$$

Nodo 3

$$\lfloor z(\text{LP}_{\text{KP}}^3) \rfloor = \underbrace{\left\lfloor \frac{47}{3} \right\rfloor}_{=15}$$

$$\text{LB} = 15$$

potato per upper bound

Nodo 4

$$x_4 \leq 0$$

$$\lfloor z(\text{LP}_{\text{KP}}^4) \rfloor = \underbrace{\left\lfloor \frac{79}{5} \right\rfloor}_{=15}$$

$$\text{LB} = 15$$

potato per upper bound

$$x_4 \geq 1$$

Nodo 5

$$z(\text{LP}_{\text{KP}}^5) = 15$$

$$\text{LB} = 15$$

potato per interezza

Riepilogo (1/2)

L'istanza:

- ▶ $c = 9$, sei oggetti già ordinati per efficienza non crescente
- ▶ la soluzione euristica $\{1, 4\}$ dà $LB = 15$; la radice dà $\lfloor \frac{49}{3} \rfloor = 16$

Iterazione 1:

- ▶ branching sull'oggetto critico 3
- ▶ il Nodo 2 ($x_3 \leq 0$) va nella lista, il Nodo 3 ($x_3 \geq 1$) è potato per upper bound

Riepilogo (2/2)

Iterazione 2:

- ▶ branching sull'oggetto critico 4 del Nodo 2
- ▶ il Nodo 4 ($x_4 \leq 0$) è potato per upper bound, il Nodo 5 ($x_4 \geq 1$) per interezza

Terminazione:

- ▶ la lista è vuota dopo 5 nodi
- ▶ la soluzione incumbent $\{1, 4\}$ è ottima, $z(\text{ILP}_{\text{KP}}) = 15$

Errori tipici

Introduzione: Errori tipici

- ▶ Raccogliamo gli errori più frequenti quando si esegue a mano l'algoritmo di branch-and-bound per il KP
- ▶ Riguardano il problema LP di un nodo, il suo valore, il lower bound e la numerazione dei nodi
- ▶ Controllateli prima di svolgere gli esercizi

Osservazione 1: Errori tipici (1/2)

- ▶ ***Dimenticare i vincoli $x_j \leq 1$.*** Senza di essi il problema LP di un nodo metterebbe tutta la capacità sull'oggetto di efficienza migliore: i box dei nodi devono contenere $x_j \leq 1$.
- ▶ ***Calcolare l'oggetto critico sull'istanza originale.*** In un nodo, l'oggetto critico e la capacità residua si calcolano sugli oggetti di $\mathcal{J}^k$ con la capacità $c(k)$, dopo aver fissato gli oggetti di $\mathcal{J}_{\text{in}}^k$ e $\mathcal{J}_{\text{out}}^k$.
- ▶ ***Dimenticare la costante $p(k)$.*** Il valore del nodo è $p(k)$ più il valore degli oggetti di $\mathcal{J}^k$: il profitto degli oggetti selezionati conta.

Osservazione: Errori tipici (2/2)

- ▶ *Usare il lower bound vecchio.* Il lower bound si aggiorna appena un nodo ha una soluzione intera di valore maggiore; i nodi della lista che diventano dominati si scartano.
- ▶ *Numerare i figli saltati.* Un figlio che non viene risolto non riceve un indice di nodo, esattamente come nell'algoritmo generale.

Il problema LP di un nodo:

- ▶ contiene i vincoli $x_j \leq 1$
- ▶ oggetto critico e capacità residua si calcolano su $\mathcal{J}^k$ con capacità $c(k)$
- ▶ il suo valore comprende la costante $p(k)$

L'algoritmo:

- ▶ aggiornare LB appena si trova una soluzione intera migliore
- ▶ un figlio che non viene risolto non riceve un indice di nodo

In sintesi

Introduzione: In sintesi

- ▶ Raccogliamo le idee chiave del capitolo sull'algoritmo di branch-and-bound per il KP
- ▶ Il quadro riassuntivo raccoglie i bound trovati sull'esempio del capitolo

Riepilogo (1/2)

- ▶ L'algoritmo di branch-and-bound per il KP è l'algoritmo basato sulla PL con due semplificazioni: il problema di ogni nodo si risolve con le formule di Dantzig, e la variabile di branching è quella dell'oggetto critico.
- ▶ Un nodo è descritto dagli insiemi $\mathcal{J}_{\text{in}}^k$, $\mathcal{J}_{\text{out}}^k$ e $\mathcal{J}^k$, dalla capacità $c(k)$ e dalla costante $p(k)$.
- ▶ Il bound di un nodo è $\lfloor z(\text{LP}_{\text{KP}}^k) \rfloor$; quando la lista dei problemi è vuota il lower bound e l'upper bound coincidono e la soluzione incumbent è ottima.

Riepilogo (2/2)

Quadro riassuntivo dell'istanza del KP dell'esempio ($c = 9$, $p = (6, 5, 8, 9, 6, 7)$, $w = (2, 3, 6, 7, 5, 9)$). I bound su $z(\text{ILP}_{\text{KP}})$ trovati finora:

Fonte	Lower bound	Upper bound	Da
Bound di Dantzig	–	16	$\lfloor \frac{49}{3} \rfloor$
Questo capitolo, inizializzazione	15	16	soluzione euristica $\{1, 4\}$
Questo capitolo, iterazione 1	15	16	nodo 2 nella lista, $\lfloor \frac{113}{7} \rfloor$
Questo capitolo, terminazione	15	15	lista vuota, 5 nodi