



# Algoritmo di branch-and-bound

# Divide et Impera

Sia

$$z^* = \max \{c^T x : x \in S\} \quad (1)$$

un problema di OC difficile da risolvere

**Domanda**

Voglio decomporre il problema (1) in una collezione di problemi tali che

1. Ogni nuovo problema sia facile da risolvere
2. Le informazioni che ottengo risolvendo la collezione di problemi mi indicano la sol. ottima di (1)

## Un semplice teoremino

Sia  $S = S_1 \cup S_2 \cup \dots \cup S_k$  una decomposizione della regione ammissibile  $S$  in  $k$  insiemi

### Teorema

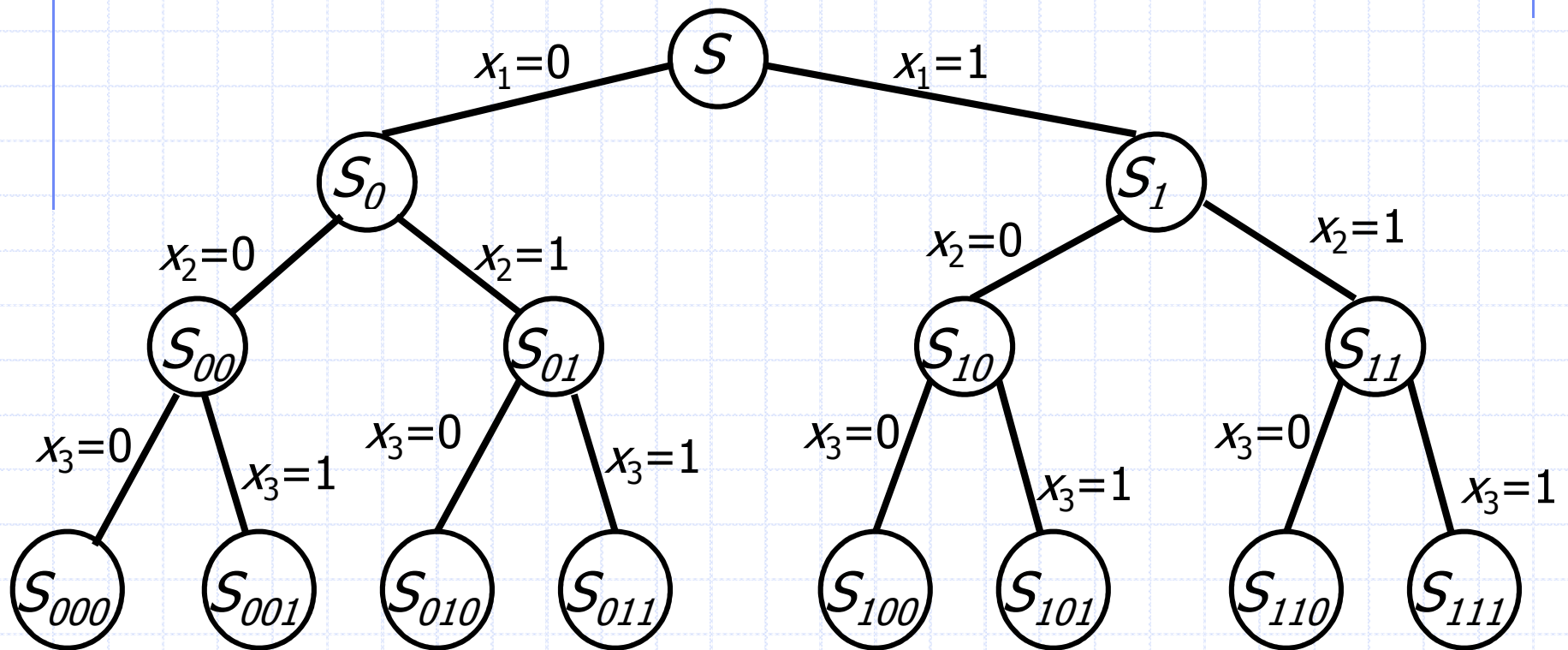
Se  $z^h = \max \{c^T x : x \in S_h\}$  allora  $z^* = \max_h z^h$

### Attenzione

Non ho **richiesto** che  $S_i \cap S_j = \emptyset$ , ovvero che la decomposizione sia una partizione, ma non ho escluso che possa essere una partizione!

## Un esempio

Consideriamo  $S = \{0,1\}^3$



Attenzione...

Stiamo semplicemente enumerando TUTTI gli insiemi ammissibili!!!!

Per un problema di OC conosciamo bound "primali" e "duali"

Domanda

Possiamo utilizzare questi bound per enumerare in modo più efficace?

## Altro teoremino

Sia  $S = S_1 \cup S_2 \cup \dots \cup S_k$  una decomposizione della regione ammissibile  $S$  in  $k$  insiemi.

Sia  $z^* = \max \{c^T x : x \in S_k\}$

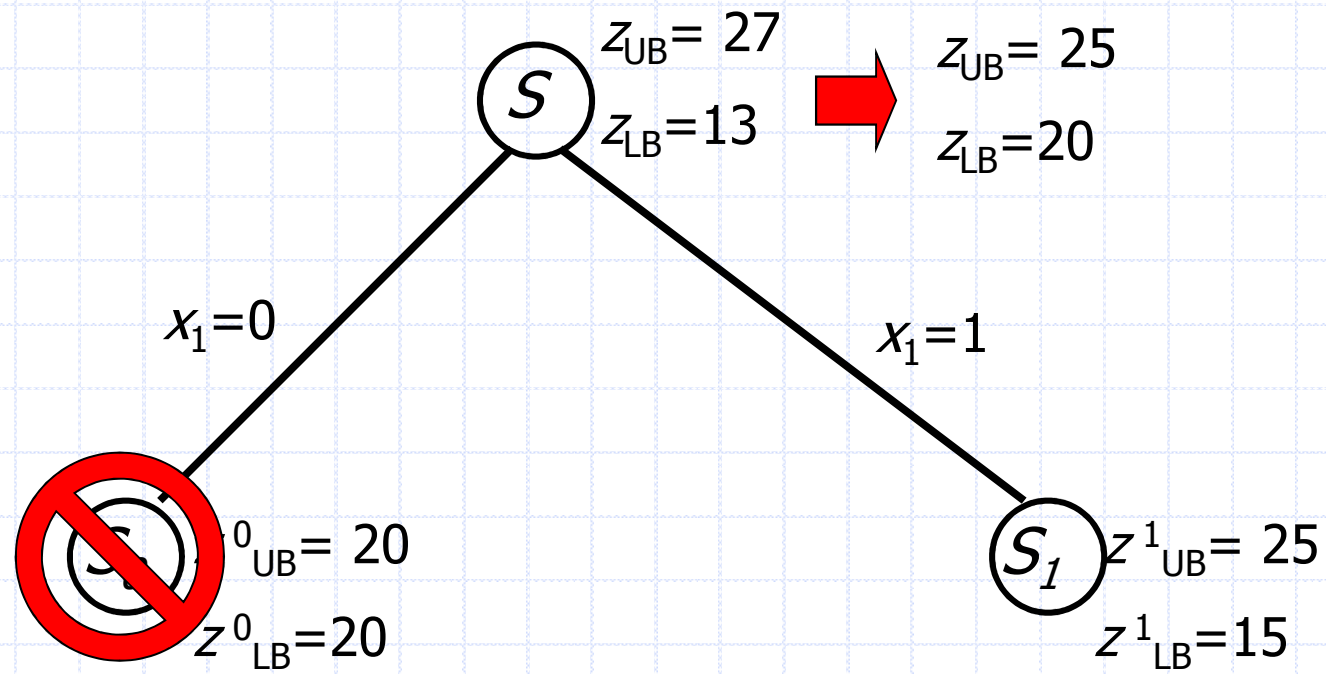
Se  $z_{LB}^*$  e  $z_{UB}^*$  sono rispettivamente un lower e un upper bound per  $z^*$

Allora

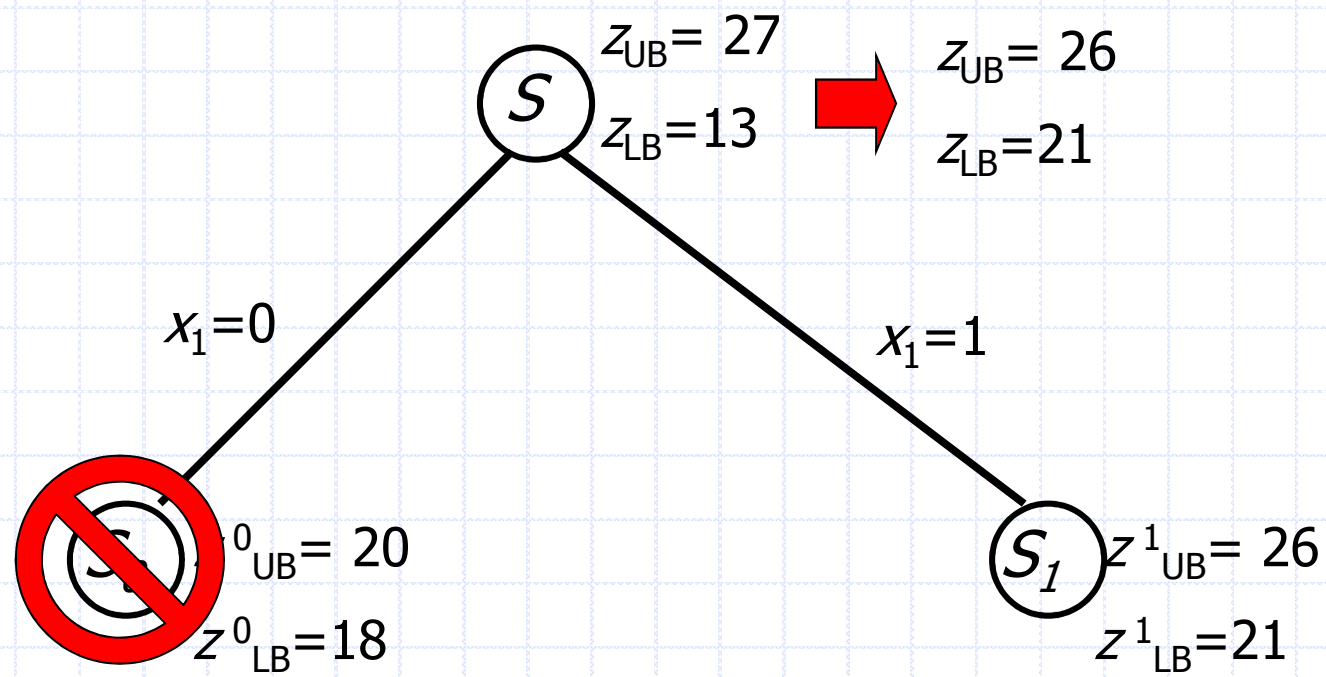
$z_{UB} = \max_k z_{UB}^*$  è un upper bound per  $z^*$

$z_{LB} = \max_k z_{LB}^*$  è un lower bound per  $z^*$

## Potatura per ottimalità

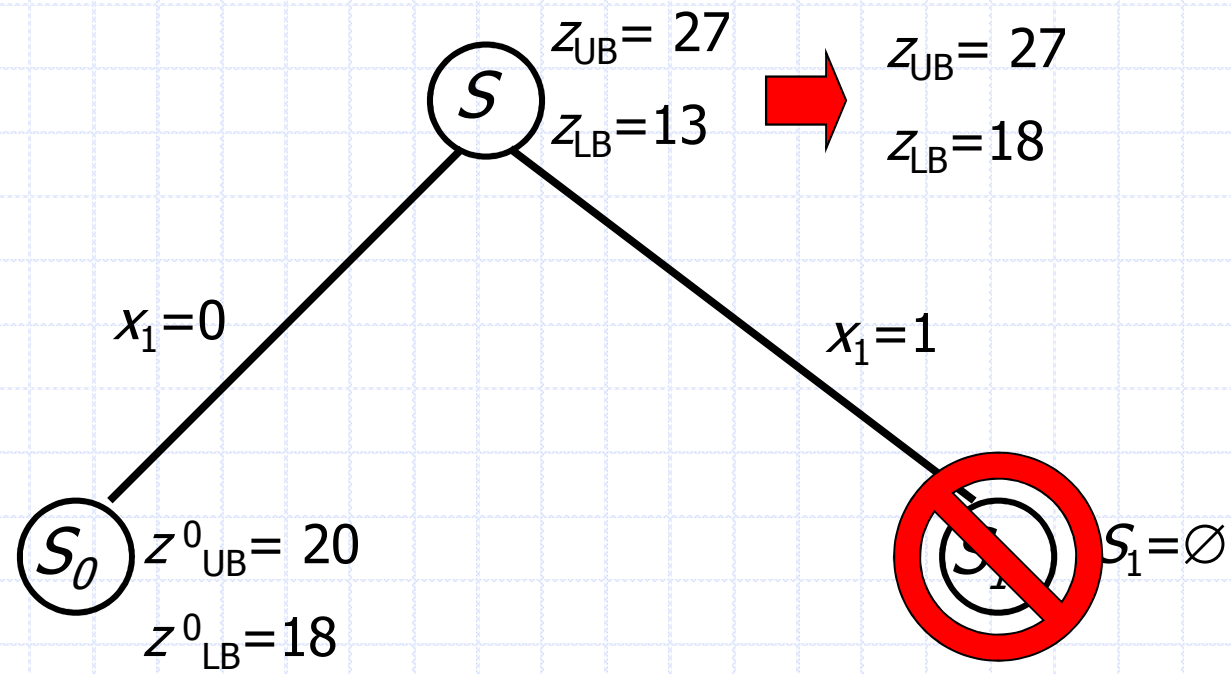


## Potatura per bound

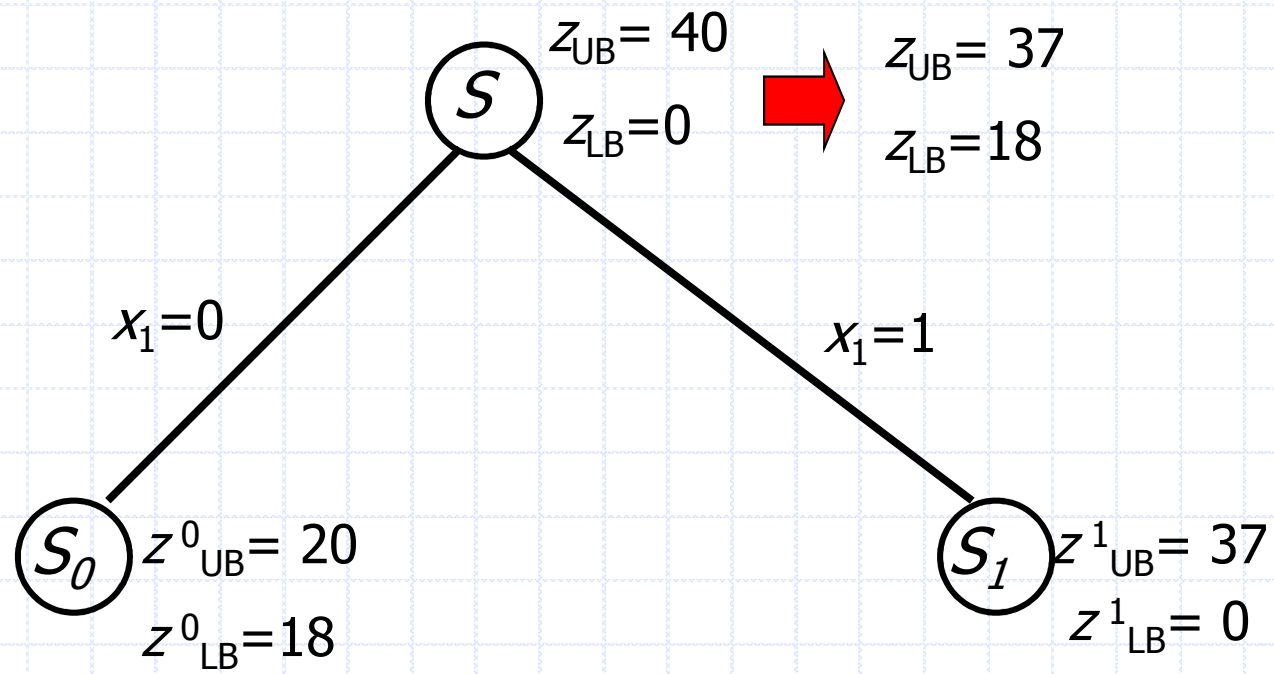




## Potatura per inammissibilità



## Nessuna potatura



## Ricapitolando...

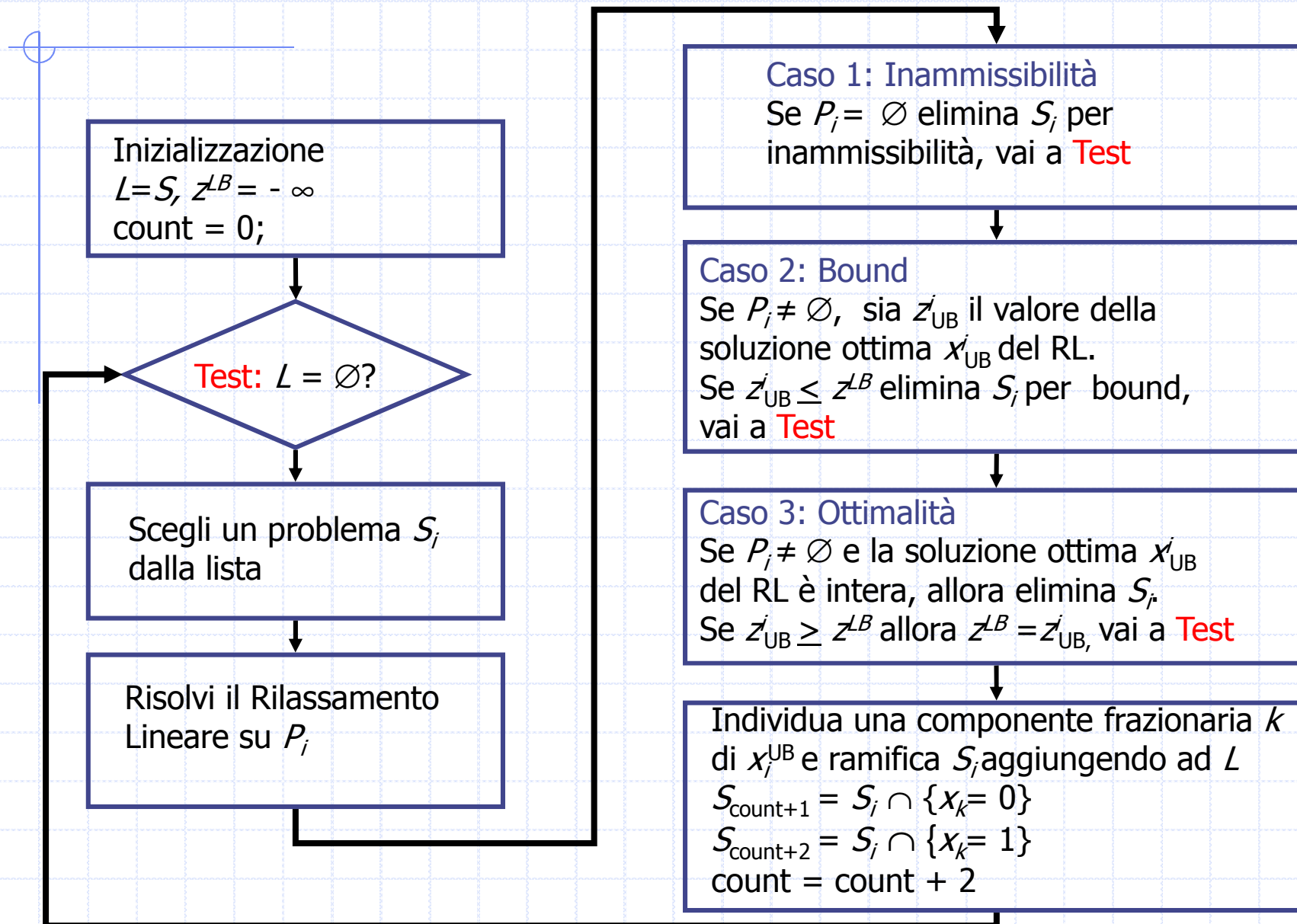
Enumero un insieme di soluzioni “implicitamente” perché “poto” rami dell’albero per:

1. Inammissibilità
2. Bound
3. Ottimalità

Indichiamo con  $L$  la lista dei sottoproblemi e con  $P_i$  la formulazione del sottoproblema  $S_i$ .

L’algoritmo ha il seguente diagramma di flusso:

# Branch-and-Bound



## Esempio

Consideriamo il problema di knapsack

$$\begin{aligned} \max & 9x_1 + 15x_2 + 8x_3 + 6x_4 + 5x_5 + 4x_6 + x_7 \\ & 6x_1 + 11x_2 + 6x_3 + 5x_4 + 5x_5 + 4x_6 + x_7 \leq 19 \quad (1) \\ & x \in \{0,1\}^7 \end{aligned}$$

La soluzione ottima del rilassamento lineare è

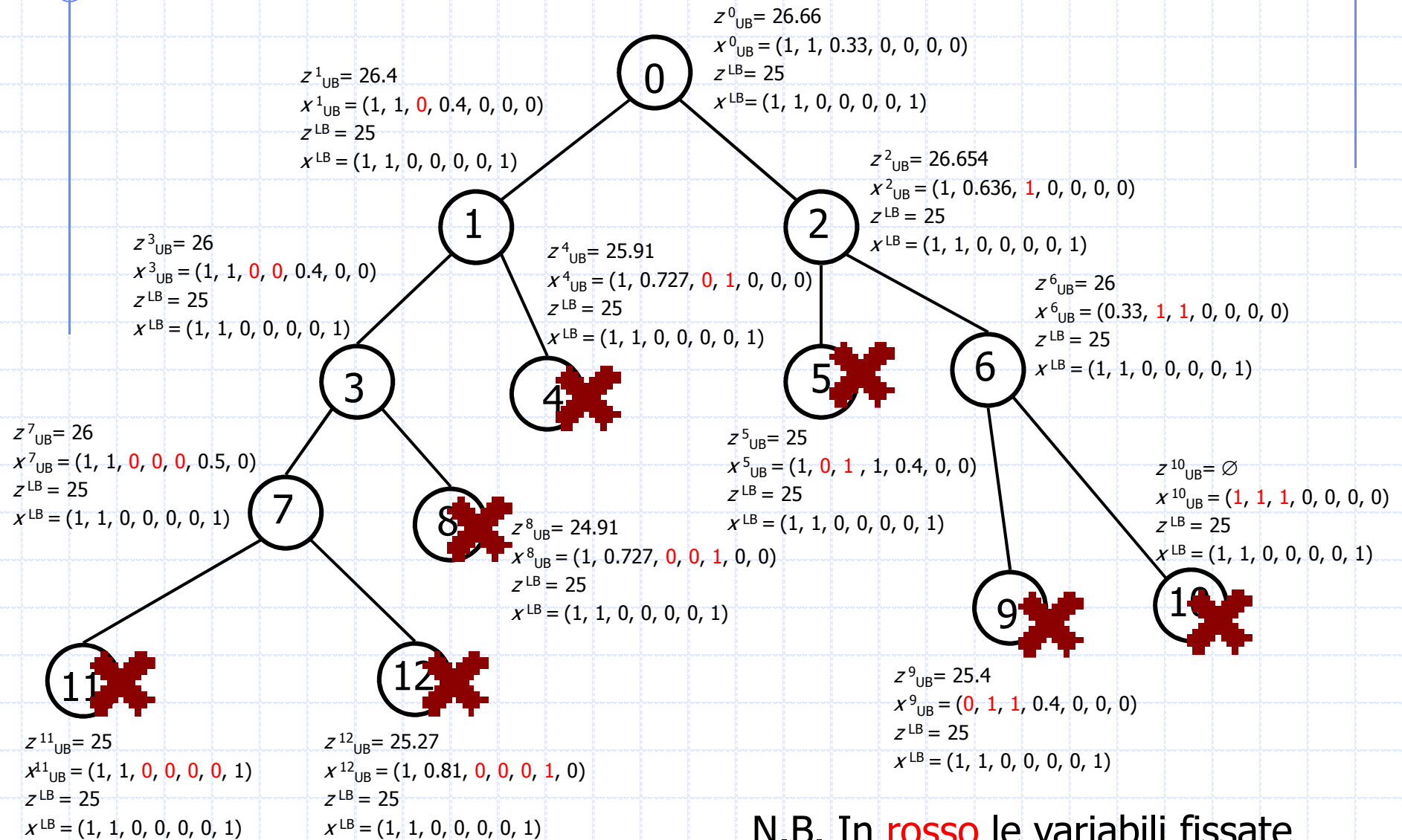
$x^0_{UB} = (1, 1, 1/3, 0, 0, 0, 0)$  di valore 26.666 (UB)

Possiamo anche inizializzare il LB con la soluzione ammissibile  $x^{LB} = (1, 1, 0, 0, 0, 0, 1)$  di valore 25

### Osservazione

Poiché la f.o. ha coefficienti interi, possiamo scrivere la condizione di potatura per bound (caso 2) come  $\lfloor z^i_{UB} \rfloor \leq z^{LB}$

# Esempio





# Rafforzamento di formulazioni e algoritmo del piano di taglio

## Rafforzamento di formulazioni

Il problema di knapsack

$$\max 5x_1 + 2x_2$$

$$3x_1 + 4x_2 \leq 6$$

$$x \in \{0,1\}^2$$

ha come rilassamento lineare

$$\max 5x_1 + 2x_2$$

$$3x_1 + 4x_2 \leq 6$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_1 \leq 1$$

$$x_2 \leq 1$$

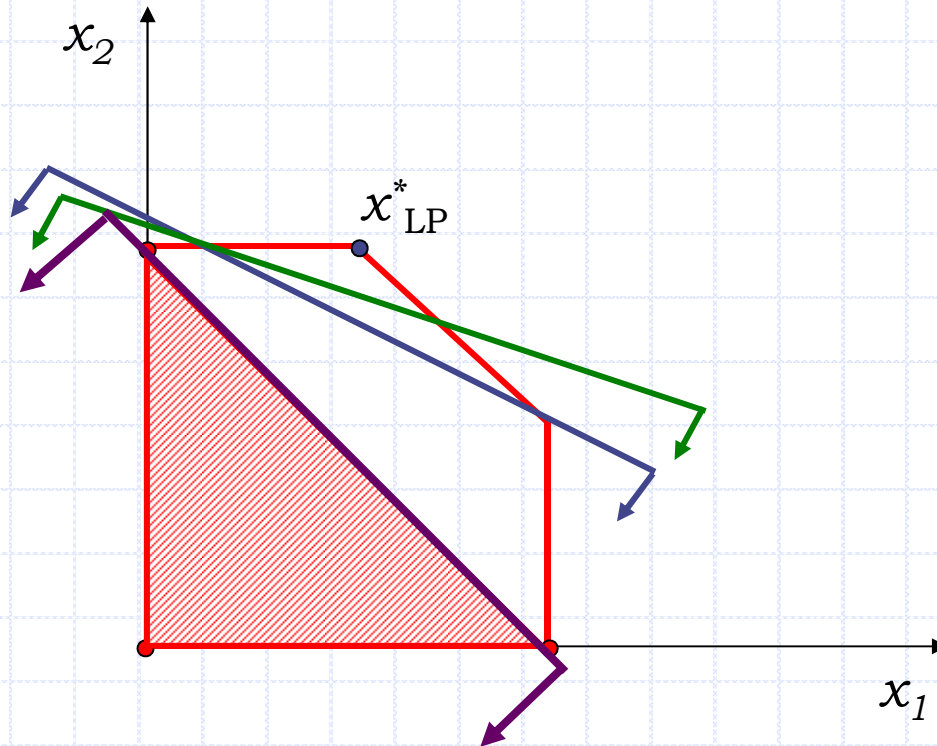
Indichiamo con  $x_{LP}^*$  la soluzione ottima del rilassamento.



# Separazione

## Domanda

Possiamo individuare una disuguaglianza **valida** per  $\text{conv}(F)$  che "separa"  $x_{LP}^*$  da  $\text{conv}(F)$  ?



## Minimal cover

Consideriamo l'insieme

$$X = \{\sum_{j=1, \dots, n} a_j x_j \leq b, x \in \{0, 1\}^n\}$$

### Definizione

Un insieme  $C \subseteq N$  è un *cover* se e solo se

$$\sum_{j \in C} a_j > b.$$

Un cover è *minimale* se, comunque preso  $j \in C$ , l'insieme  $C \setminus \{j\}$  NON è un cover

### Osservazione

Un cover ha associato un vettore caratteristico non ammissibile per  $X$

# Disuguaglianze cover

## Teorema

Se  $C \subseteq N$  è un cover, la disuguaglianza

$$\sum_{j \in C} x_j \leq |C| - 1 \quad (*)$$

è valida per  $\text{conv}(X)$ .

## Dimostrazione

Dimostriamo che se  $x^R \in \{0,1\}^n$  non soddisfa il vincolo  $(*)$  allora  $x^R \notin X$ .

Se  $x^R$  è tale che  $\sum_{j \in C} x_j > |C| - 1$ , allora  $|R \cap C| = |C|$ ,  
quindi  $C \subseteq R$ .

Pertanto:

$$\sum_{j \in N} a_j x_j^R = \sum_{j \in R} a_j \geq \sum_{j \in C} a_j > b$$

ovvero  $x^R \notin X$ . ■

## Esempio

Consideriamo il problema di knapsack

$$\max 15 x_1 + 9 x_2 + 8 x_3 + 6 x_4 + 5 x_5 + 4 x_6 + x_7$$

$$11 x_1 + 6 x_2 + 6 x_3 + 5 x_4 + 5 x_5 + 4 x_6 + x_7 \leq 19$$

$$x \in [0,1]^7$$

$$x_{LP}^* = (1, 1, 1/3, 0, 0, 0, 0)$$

Una disuguaglianza cover **violata da**  $x_{LP}^*$  è la disuguaglianza

$$x_1 + x_2 + x_3 \leq 2$$

Come si individua una disuguaglianza di tipo cover violata da  $x_{LP}^*$ ?

# Problema di separazione

## Domanda

Dato un poliedro  $P$ , formulazione di un problema di PL- $\{0,1\}$ , esiste una disuguaglianza di tipo cover che separa  $x_{LP}^*$  da  $\text{conv}(X)$ ?

## Risposta

- a) **SI** la disuguaglianza esiste e fornisce anche il cover associato  $C$
- b) **NO** non esiste

Un algoritmo  $A$  che risponde (correttamente) alla domanda con una delle due risposte possibili a) o b), si chiama **ORACOLO DI SEPARAZIONE**

## Oracolo di separazione

Riscriviamo la disuguaglianza

$$\sum_{j \in C} x_j \leq |C| - 1 \quad \text{come} \quad \sum_{j \in C} (1 - x_j) \geq 1$$

La disuguaglianza è violata se e solo se

$$\sum_{j \in C} (1 - x_{LPj}^*) < 1$$

Definiamo il seguente problema di OC:

$$SEP = \min \left\{ \sum_{j \in C} 1 - x_{LPj}^*, C \text{ è un cover del vincolo di knapsack} \right\}$$

Un algoritmo per questo problema è un **oracolo di separazione**

## Difatti...

Sia  $z_{\text{SEP}}^*$  il valore della soluzione ottima  $y_{\text{SEP}}^*$  di

$$\min \left\{ \sum_{j \in C} (1 - x_{\text{LP}j}^*), C \text{ è un cover del vincolo di knapsack} \right\}$$

Se  $z_{\text{SEP}}^* < 1$  esiste una dicover violata associata a  $y_{\text{SEP}}^*$

Se  $z_{\text{SEP}}^* \geq 1$  **NON** esiste una disuguaglianza cover violata

# Formulazione del problema di separazione

## Variabili decisionali

$y_j = 1$  se l'indice  $j$  è nel cover

$y_j = 0$  altrimenti

## Funzione obiettivo

$$\min \sum_{j=1}^n (1 - x_{LPj}^*) y_j$$

## Vincoli

$$\sum_{j=1}^n a_j y_j \geq b + 1$$

$$y \in \{0,1\}^n$$



## Trasformazione ...

Ponendo  $w_j = 1 - y_j$  il problema di separazione diventa

$$\min \sum_{j=1}^n (1 - x_{LPj}^*) - \sum_{j=1}^n (1 - x_{LPj}^*) w_j$$

st

$$-\sum_{j=1}^n a_j w_j \geq -\sum_{j=1}^n a_j + b + 1$$

$$w \in \{0,1\}^n$$

ovvero ...

...un problema di knapsack

$$\max \sum_{j=1}^n (1 - x_{\text{LP } j}^*) w_j$$

st

$$\sum_{j=1}^n a_j w_j \leq \sum_{j=1}^n a_j - b - 1$$

$$w \in \{0,1\}^n$$

## Esempio

$$\max 15 x_1 + 9 x_2 + 8 x_3 + 6 x_4 + 5 x_5 + 4 x_6 + x_7$$

$$11 x_1 + 6 x_2 + 6 x_3 + 5 x_4 + 5 x_5 + 4 x_6 + x_7 \leq 19$$

$$x \in \{0,1\}^7$$

$$x_{LP}^* = (1, 1, 1/3, 0, 0, 0, 0)$$

### Problema di separazione

$$\max (1 - 1) w_1 + (1 - 1) w_2 + (1 - 1/3) w_3 + w_4 + w_5 + w_6 + w_7$$

$$11 w_1 + 6 w_2 + 6 w_3 + 5 w_4 + 5 w_5 + 4 w_6 + w_7 \leq 38 - 19 - 1 = 18$$

$$w \in \{0,1\}^7$$

# Separazione

## Problema di separazione

$$\begin{aligned} \max & \quad 2/3 w_3 + w_4 + w_5 + w_6 + w_7 \\ 11 w_1 + 6 w_2 + 6 w_3 + 5 w_4 + 5 w_5 + 4 w_6 + w_7 & \leq 18 \\ w & \in \{0,1\}^7 \end{aligned}$$

## Soluzione ottima

$w^* = (0, 0, 0, 1, 1, 1, 1)$  di valore 4

Riportiamola nello spazio delle  $y$ :

$y^* = (1, 1, 1, 0, 0, 0, 0)$  di valore  $2/3$ , ovvero  $< 1$

## Pertanto...

1. Il valore di  $z^*_{\text{SEP}}$  è  $< 1$ , ovvero esiste una disuguaglianza cover violata
2. La disuguaglianza è scritta in corrispondenza agli indici che hanno valore 1 in  $y^*$ , ovvero

$y^* = (1, 1, 1, 0, 0, 0, 0)$  corrisponde a  $x_1 + x_2 + x_3 \leq 2$

## Ricapitolando

### Algoritmo del piano di taglio (CUT)

Dato il rilassamento lineare di un problema di knapsack

1. Calcola la soluzione ottima  $x_{LP}^*$
2. Se  $x_{LP}^* \in \{0, 1\}^n$  **STOP** altrimenti vai a 3.
3. Definisci il problema di separazione
4. Risolvo il problema di separazione
5. Se esiste una disuguaglianza cover violata la aggiungo alla formulazione corrente e torno al punto 1, altrimenti **STOP**

## Osservazioni

L'algoritmo del piano di taglio può arrestarsi senza aver trovato la soluzione ottima intera. Che cosa si fa in questo caso?

### Algoritmo di Cut-and-branch

Il problema di separazione è un problema di knapsack avente la stessa dimensione del problema originario.

Perché non risolvere all'ottimo direttamente il problema originario?

Cosa succede se si risolve euristicamente il problema di separazione?

# Cut-and-branch

