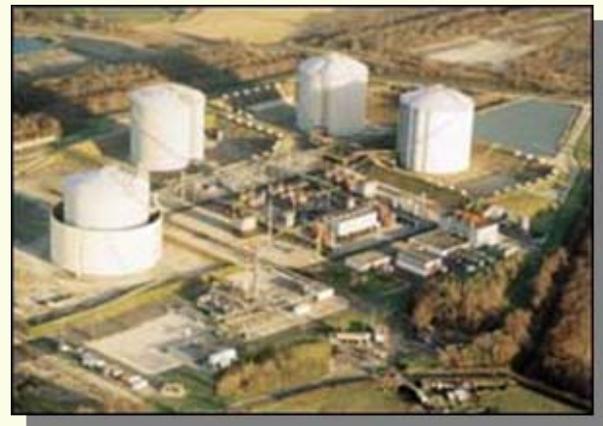
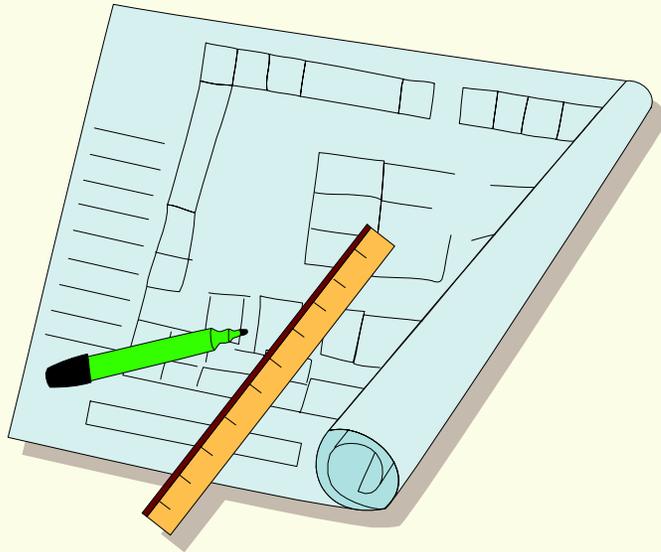


# Ottimizzazione nella sintesi di processo



# Introduzione ai “metodi algoritmici”

- L'ottimizzazione di specifici sottoproblemi nella sintesi di processo riduce significativamente il problema combinatorio legato alla identificazione del migliore flowsheet. Al contempo, proprio per la natura specifica dei sottoproblemi analizzati, (reti di scambio termico, sistemi di calore ed energia, sequenze di distillazione, reti di reattori, ...) è possibile evidenziare le seguenti limitazioni:
  1. Vengono spesso ignorate le possibili interazioni tra flussi di materia e flussi di energia. Come determinare un *compromesso* tra utilizzo delle **materie prime** e **consumi energetici** nella definizione delle correnti di processo?
  2. I problemi evidenziati in precedenza si basano, il più delle volte, su principi fisici e non considerano i costi di investimento e di conduzione. Occorre quindi sviluppare metodologie per la sintesi di processo che sappiano trovare un *compromesso* tra: **materie prime—costi energetici** e **costi di capitale**.
  3. Non viene indicato in genere come determinare sistematicamente una struttura processistica ottimale: **presenza/assenza** di **apparecchiature** e di **correnti** nel processo considerato.



# Introduzione ai “metodi algoritmici”

- Le metodologie calcolistiche volte ad affrontare e risolvere le problematiche sollevate dai tre punti indicati in precedenza sono definite: “**metodi algoritmici**”.
- Tali metodi si basano su tecniche di ottimizzazione e principalmente su metodi di ottimizzazione: Mixed-Integer dove coesistono sia variabili continue che variabili discrete (necessarie alla sintesi di processo).
- Tre sono i punti base per lo sviluppo di un metodo algoritmico:
  1. Rappresentazione del problema
  2. Modellazione del problema
  3. Strategia risolutiva



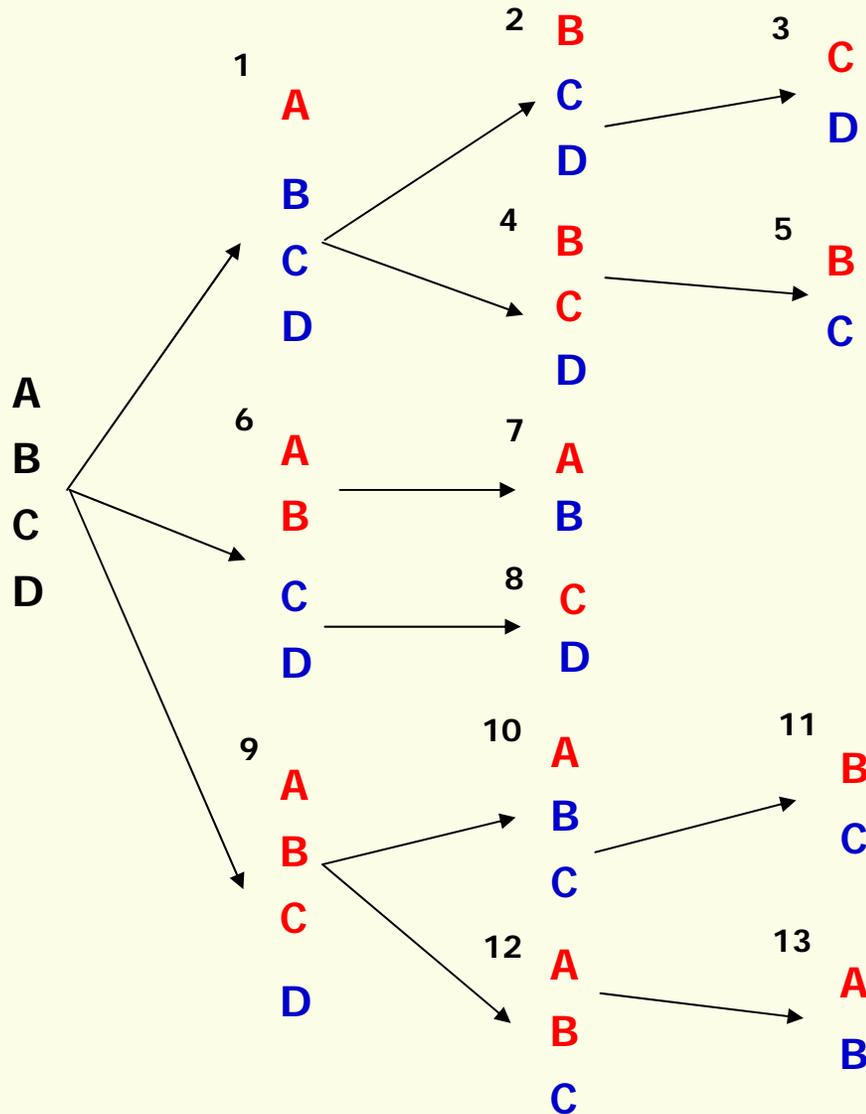
# Rappresentazione del problema

- La rappresentazione del problema secondo approcci differenti gioca un ruolo fondamentale nella successiva risoluzione della sintesi di processo tramite metodi algoritmici.
- Obiettivo principale della rappresentazione è includere esplicitamente o implicitamente una famiglia di flowsheet possibili candidati per la soluzione ottimale.
- In base alla rappresentazione scelta occorre adottare tecniche risolutive distinte.



# Rappresentazione del problema

Esempio 1: separazione netta di una corrente di quattro componenti: A, B, C, D.

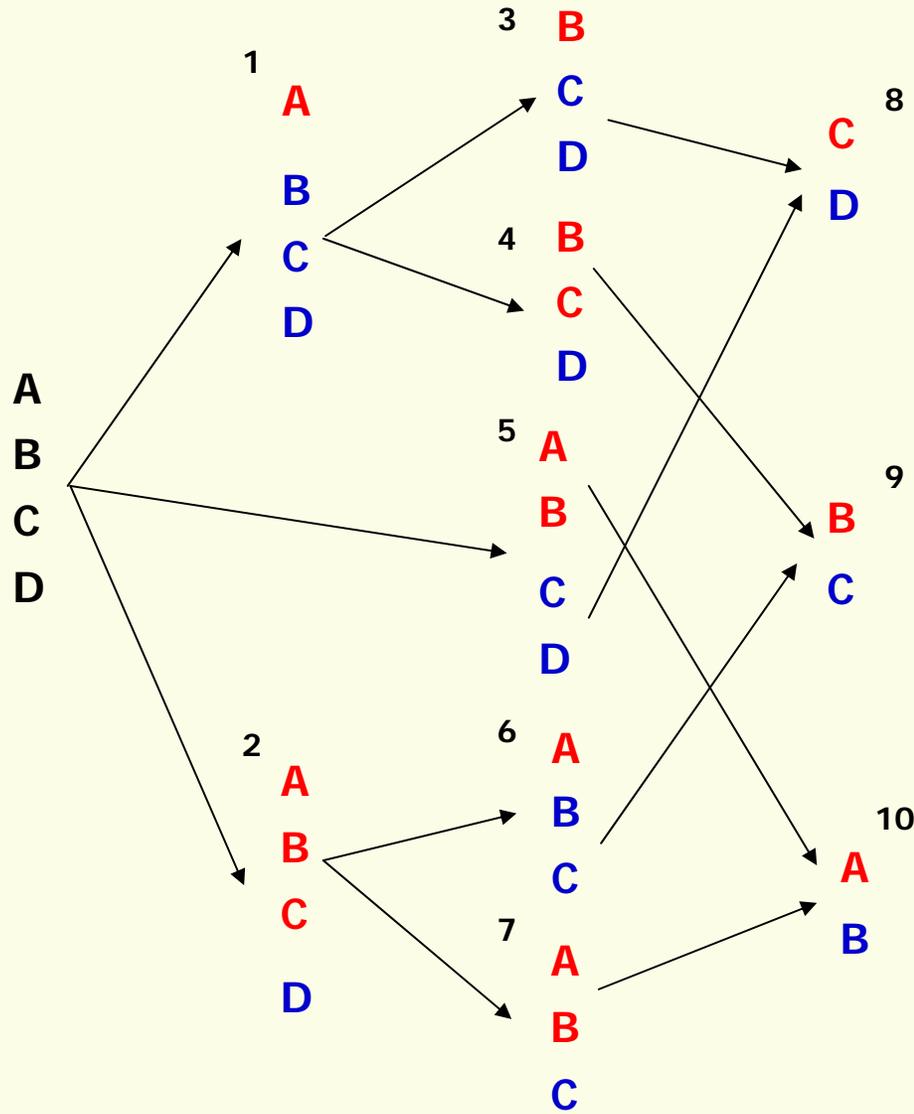


Rappresentazione  
ad ALBERO

Hendry e Hughes, 1972

# Rappresentazione del problema

Esempio 1: separazione netta di una corrente di quattro componenti: A, B, C, D.



Rappresentazione  
a RETE

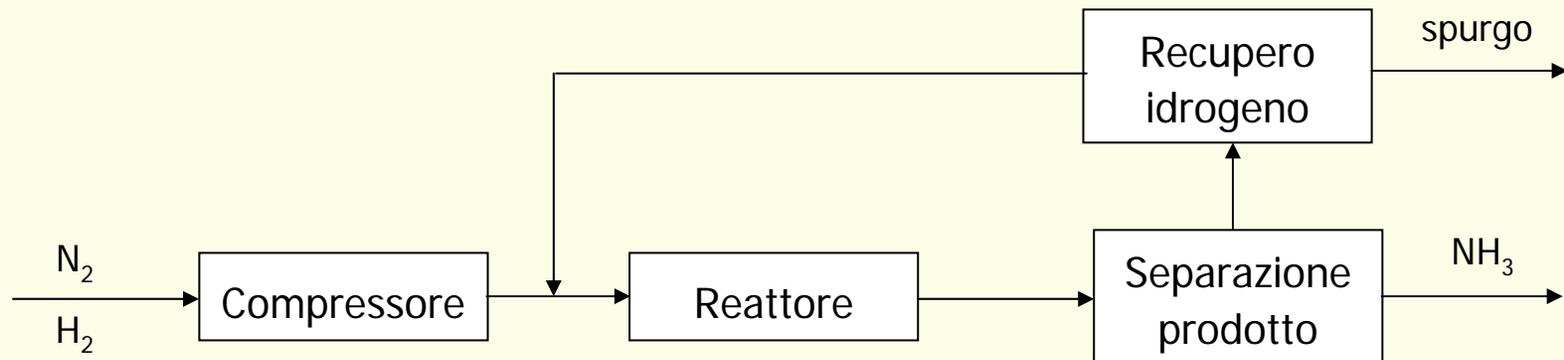
Andrecovich e Westerberg, 1985



# Rappresentazione del problema

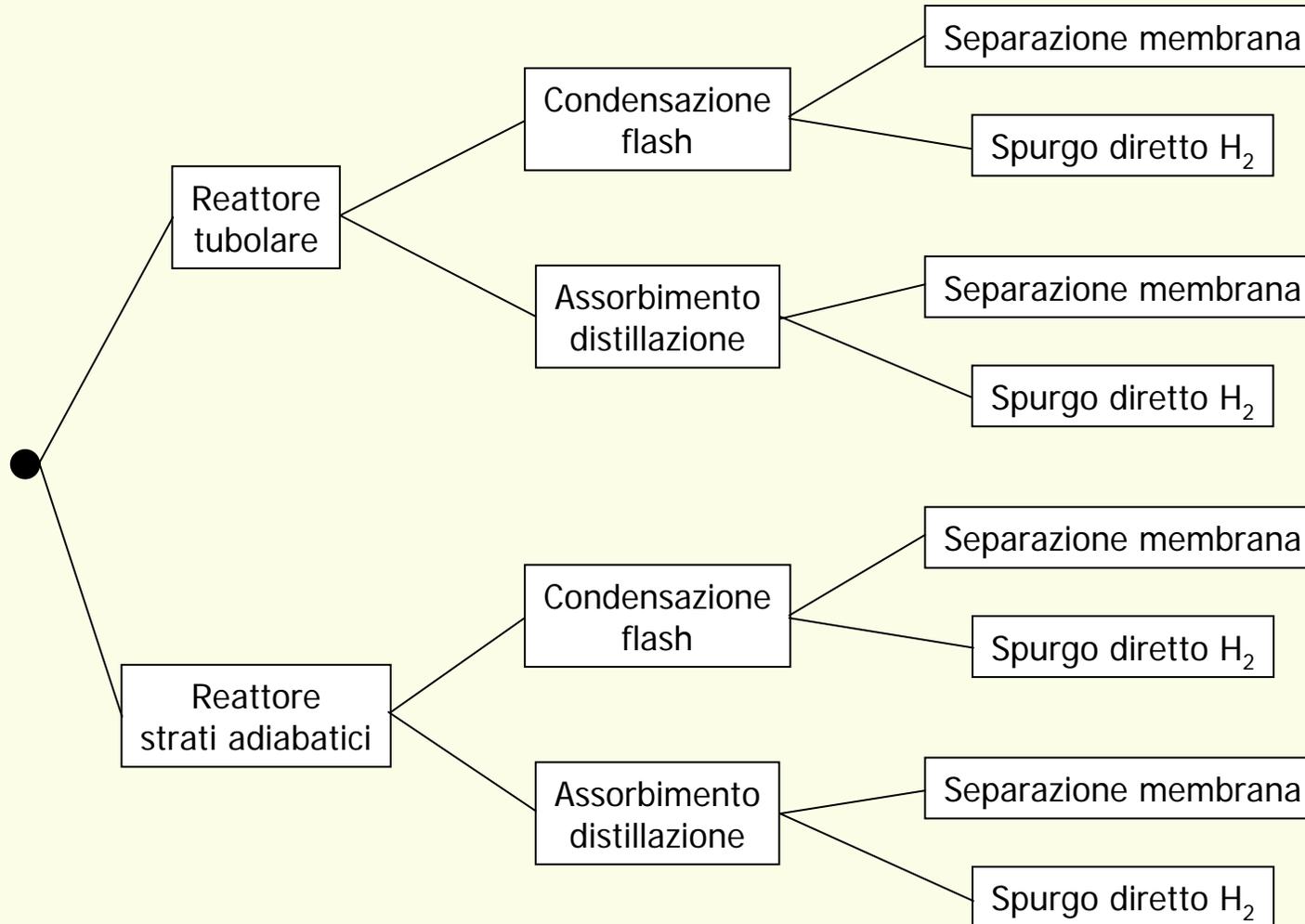
Esempio 2: flowsheet del processo di sintesi dell'ammoniaca

- Occorre distinguere tra le seguenti alternative processistiche:
  1. Reattore tubolare o a strati adiabatici con raffreddamento intermedio
  2. Separazione del prodotto tra condensazione a flash o assorbimento/distillazione
  3. Recupero idrogeno tramite separazione a membrana o eliminazione tramite spurgo



# Rappresentazione del problema

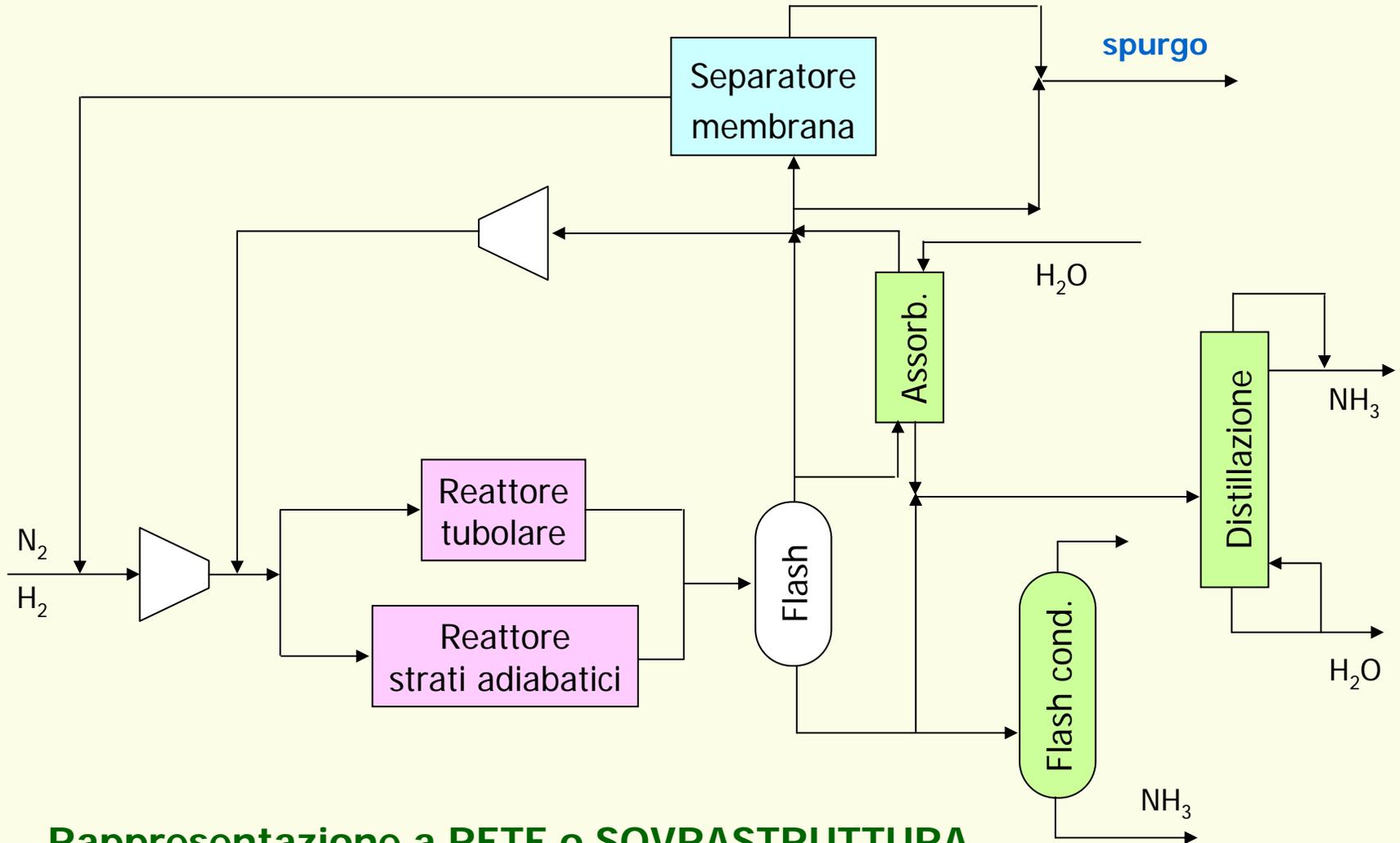
Esempio 2: flowsheet del processo di sintesi dell'ammoniaca



**Rappresentazione ad ALBERO**

# Rappresentazione del problema

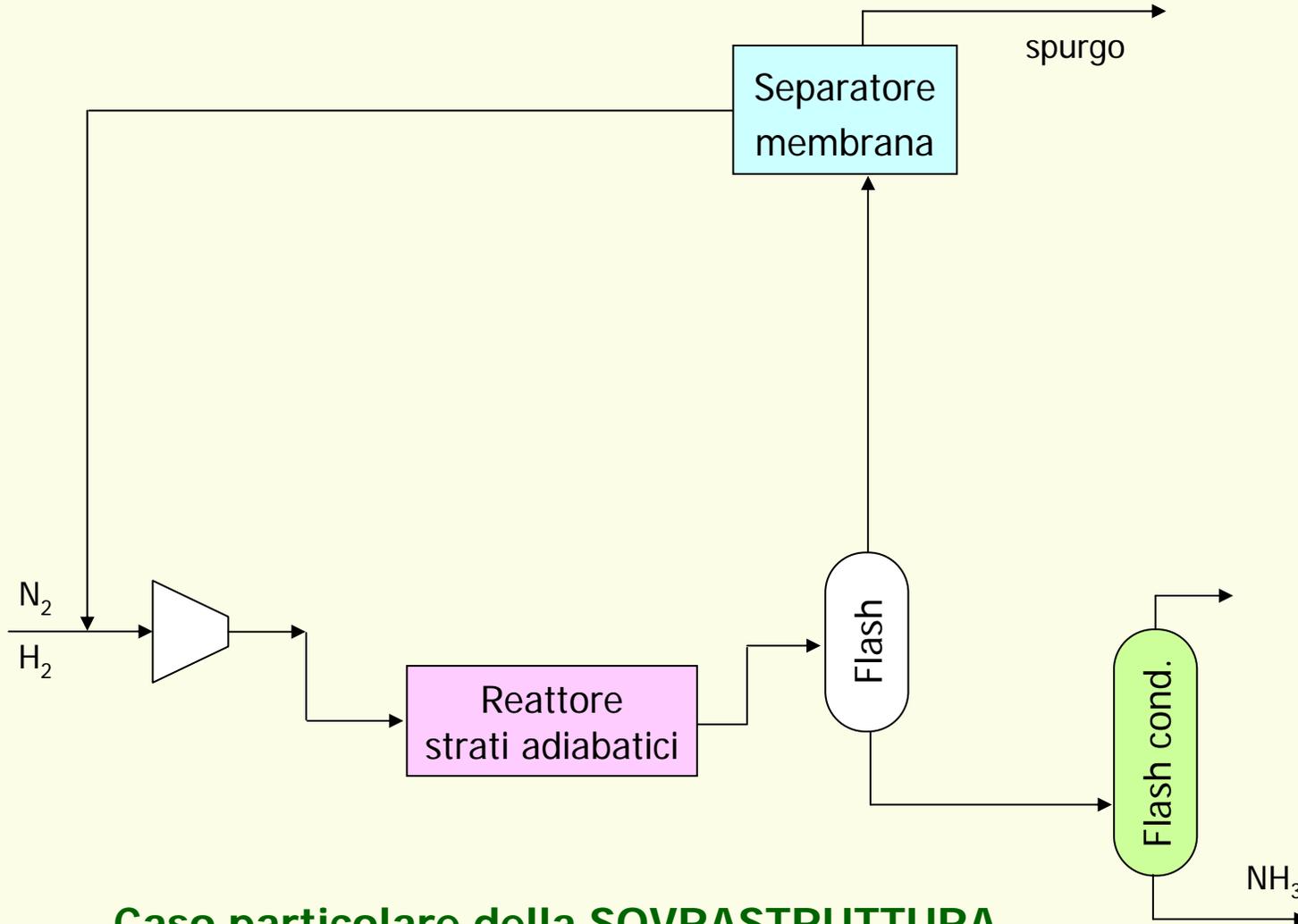
Esempio 2: flowsheet del processo di sintesi dell'ammoniaca



**Rappresentazione a RETE o SOVRASTRUTTURA**

# Rappresentazione del problema

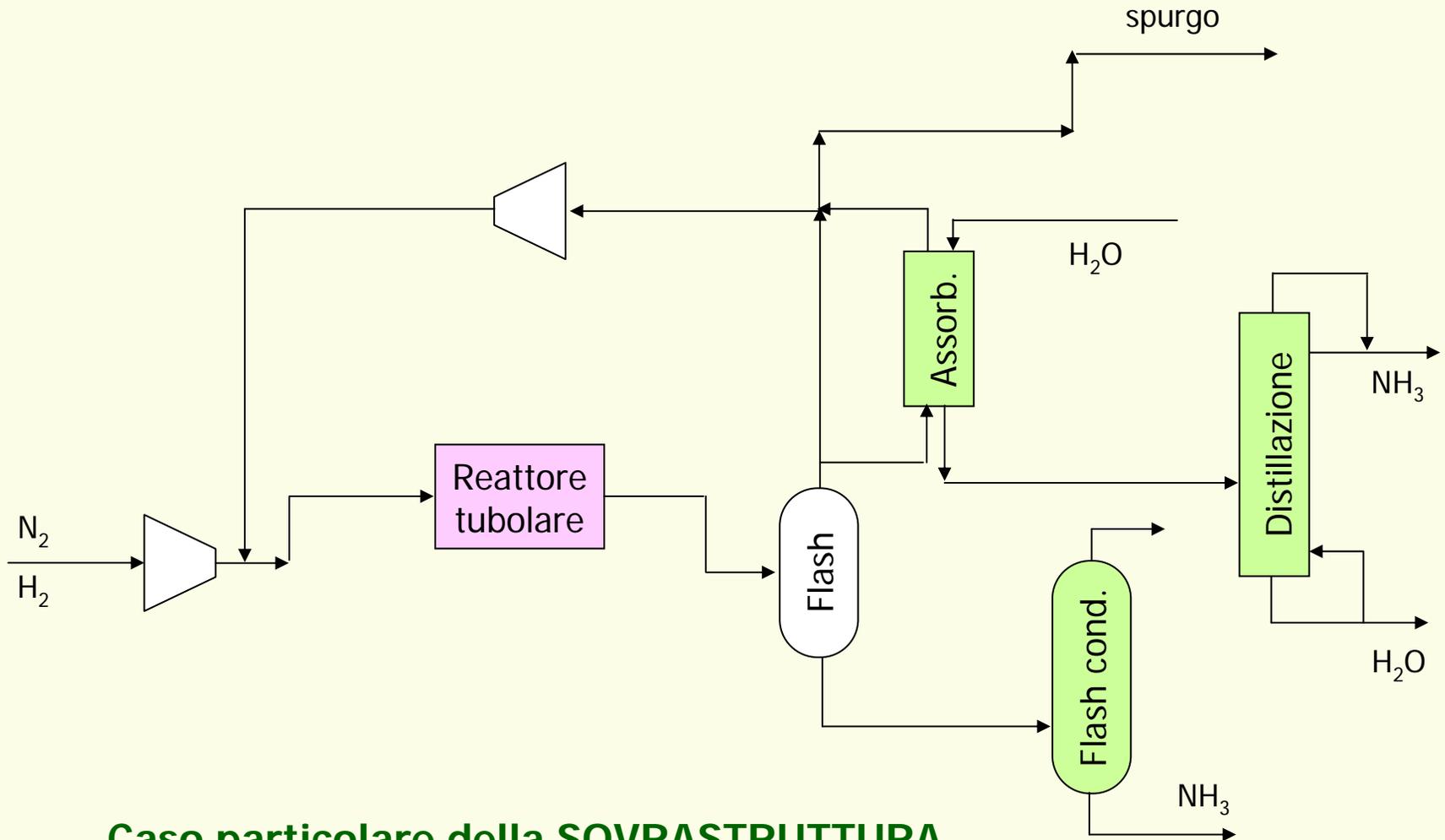
Esempio 2: flowsheet del processo di sintesi dell'ammoniaca



**Caso particolare della SOVRASTRUTTURA**

# Rappresentazione del problema

Esempio 2: flowsheet del processo di sintesi dell'ammoniaca



**Caso particolare della SOVRASTRUTTURA**

# Strategie risolutive per rappresentazioni ad albero

Dopo aver realizzato una specifica rappresentazione del problema occorre definire la strategia di ricerca della struttura ottimale del flowsheet.

Nel caso di **rappresentazione ad albero** è possibile scomporre il problema, analizzando una sequenza di nodi all'interno dell'albero stesso.

Ogni nodo richiede il dimensionamento e la valutazione dei costi di investimento ed esercizio dell'apparecchiatura ad esso connessa.

Esistono due alternative per l'analisi dei nodi:

- **enumerazione esaustiva** (adatta per alberi di piccole dimensioni)
- **enumerazione implicita** (adatta per alberi di medie-grandi dimensioni)

La strategia basata sulla **enumerazione implicita** focalizza l'attenzione su specifici sottoinsiemi dei nodi.

Tale metodo di ricerca è più conosciuto con il termine: "**Branch and Bound**".

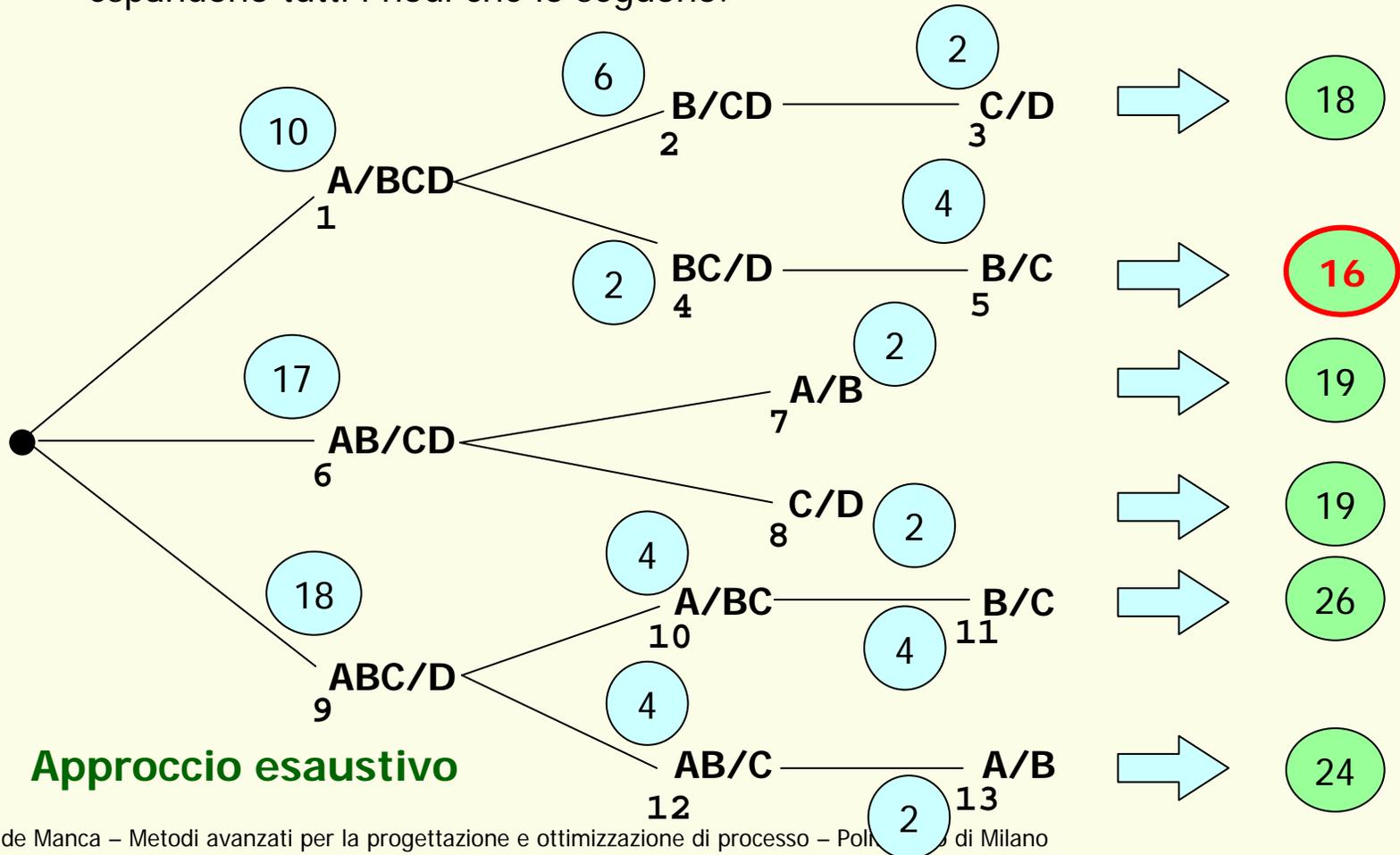


# Metodo "Branch and Bound"

- Il problema che si vuole risolvere è quello della minimizzazione dei costi di investimento e/o operativi di un flowsheet di impianto.
- Considerando un albero abbiamo: il nodo radice, i nodi intermedi ed i nodi terminali.
- Per ogni nodo appartenente all'albero è possibile calcolare il **costo parziale** che è uguale alla somma dei costi dei nodi che lo precedono nel percorso che parte dal nodo radice e conduce al nodo considerato.
- Dato che il costo parziale aumenta monotonicamente lungo un qualsiasi percorso all'interno dell'albero, è possibile evidenziare le seguenti proprietà:
  1. Il costo parziale di ogni nodo intermedio è un limite **INFERIORE** per il costo di ogni nodo seguente.
  2. Il costo parziale di un nodo terminale è un limite **SUPERIORE** per il costo del problema originale. Ogni nodo terminale infatti identifica una possibile soluzione della rappresentazione. Tale soluzione può essere eventualmente quella ottimale.
- Grazie alle due proprietà appena elencate è possibile sfrondare i rami a valle di un nodo che abbia un costo parziale superiore al limite SUPERIORE corrente: **BOUNDING**.
- Occorre definire una metodologia per la numerazione dei nodi ovvero per la loro selezione in sequenza: **BRANCHING**.

# Metodo "Branch and Bound"

- Le due opzioni più comunemente usate per la selezione dei nodi sono:
  - A priorità di **profondità**: viene effettuato un branching (ramificazione) sul nodo selezionato più recentemente.
  - A priorità di **ampiezza**: si seleziona il nodo con il costo parziale più basso e si espandono tutti i nodi che lo seguono.



# "Branch and Bound" a priorità di profondità

- Si procede nel seguente modo:
- Dal nodo radice al nodo 1: costo parziale  $CP = 10$
- Dal nodo 1 al nodo 2:  $CP = 10 + 6 = 16$
- Dal nodo 2 al nodo 3:  $CP = 16 + 2 = 18$ 
  - Dato che il nodo 3 è terminale  $\rightarrow$  il limite superiore attuale è  $LS = 18$
  - Sequenza parziale migliore: (1,2,3)
- Ritorna al nodo 2
- Ritorna al nodo 1
- Dal nodo 1 al nodo 4:  $CP = 10 + 2 = 12 < 18$
- Dal nodo 4 al nodo 5:  $CP = 12 + 4 = 16$ 
  - Dato che il nodo 5 è terminale  $\rightarrow$  il limite superiore attuale è  $LS = 16$
  - Sequenza parziale migliore: (1,4,5)
- Ritorna al nodo 4
- Ritorna al nodo 1
- Ritorna al nodo radice



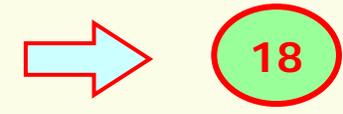
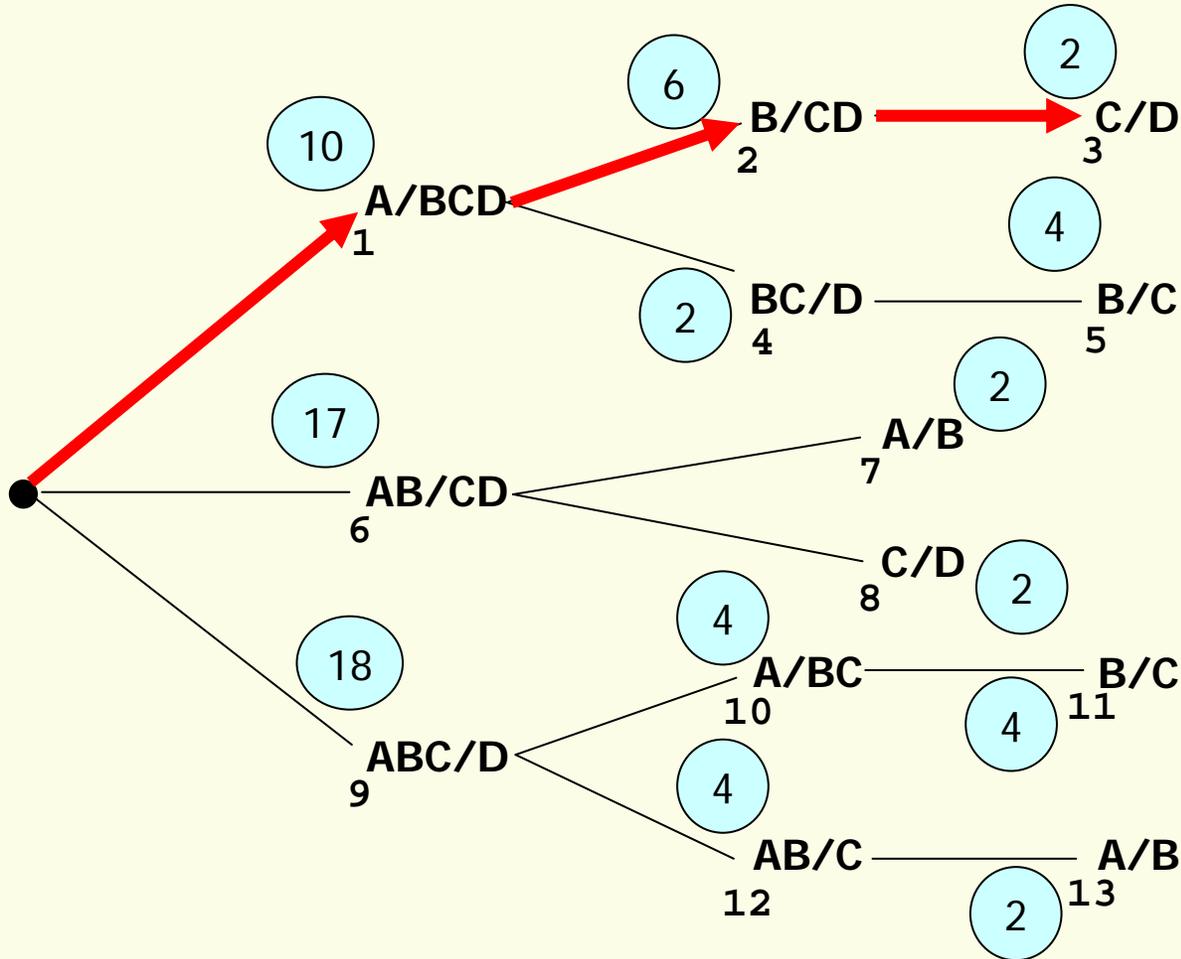
# "Branch and Bound" a priorità di profondità

- Dal nodo radice al nodo 6: costo parziale  $CP = 17$ 
  - Dato che  $CP (= 17) > LS (= 16)$  allora si sfronda il nodo 6 (e tutti i nodi seguenti)
- Ritorna al nodo radice
- Dal nodo radice al nodo 9: costo parziale  $CP = 18$ 
  - Dato che  $CP (= 18) > LS (= 16)$  allora si sfronda il nodo 9 (e tutti i nodi seguenti)
- Ritorna al nodo radice
  - Dato che tutte le diramazioni dal nodo radice sono state esaminate → STOP
  - Sequenza ottimale: (1,4,5) con  $COSTO_{OTT} = 16$
  - Totale apparecchiature analizzate: 7 su 13.

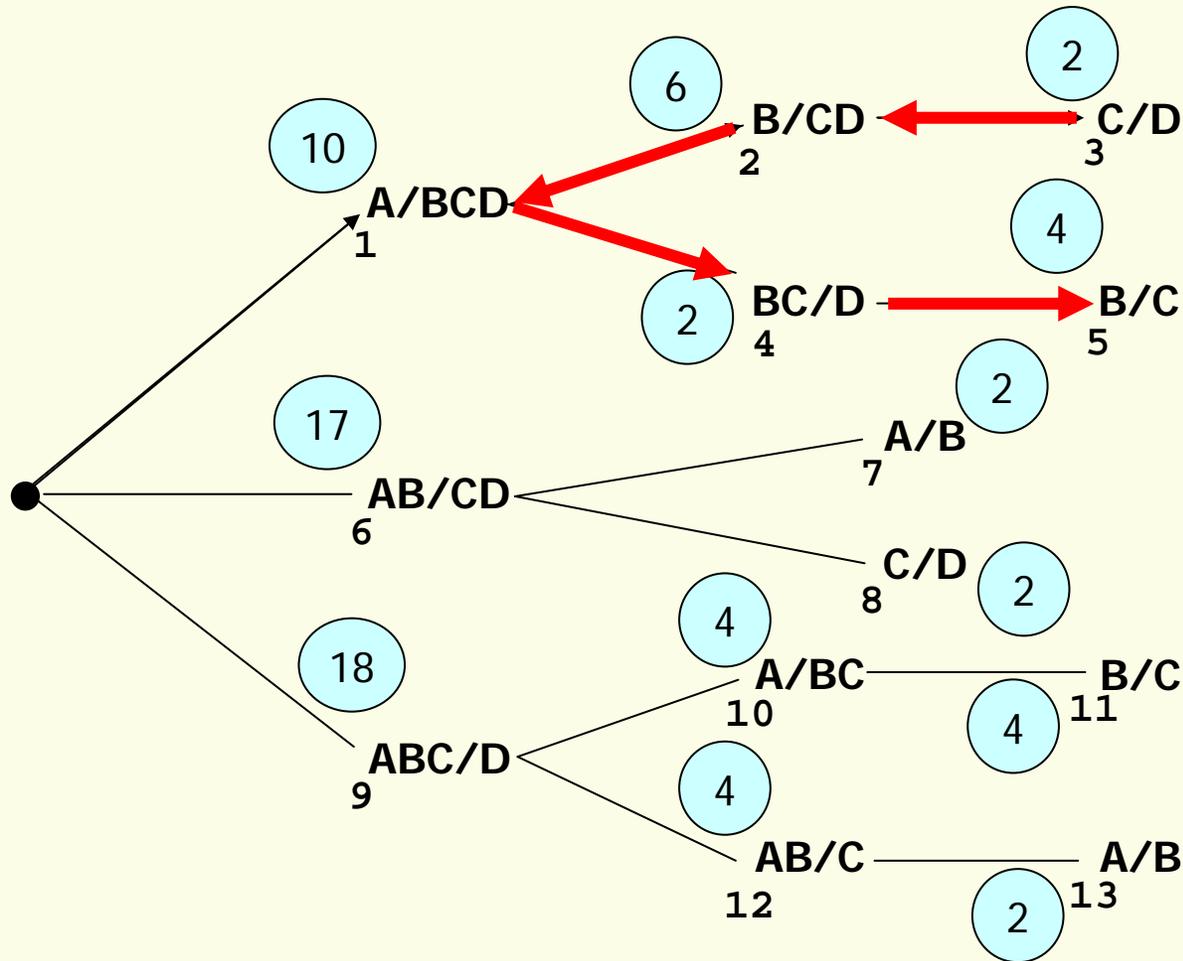


# "Branch and Bound" a priorità di profondità

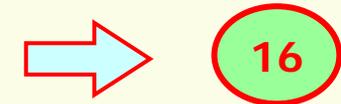
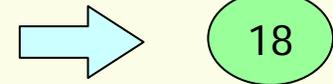
Sequenza ottimale:  
(1,2,3)



# "Branch and Bound" a priorità di profondità

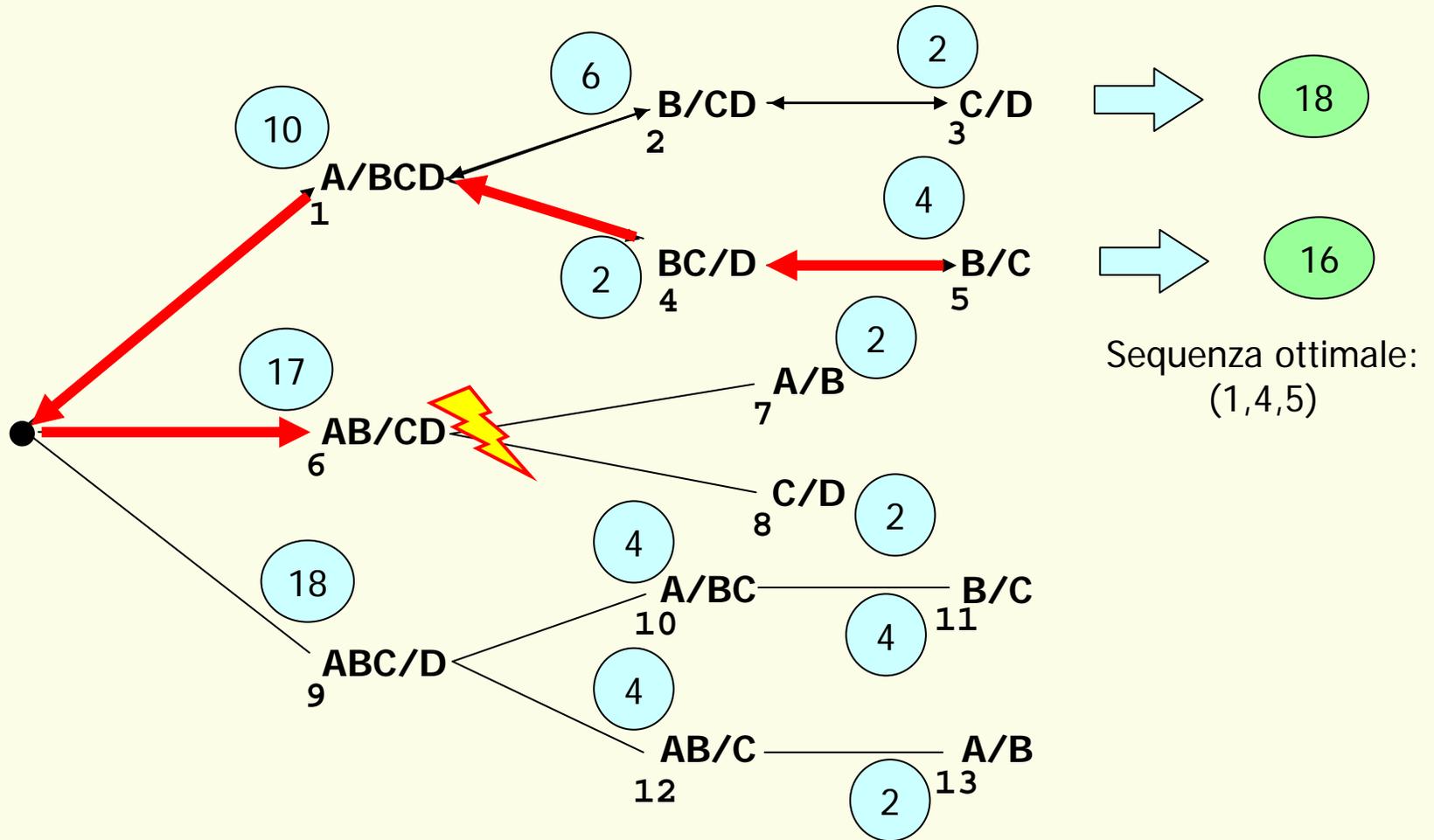


Sequenza ottimale:  
(1,2,3)

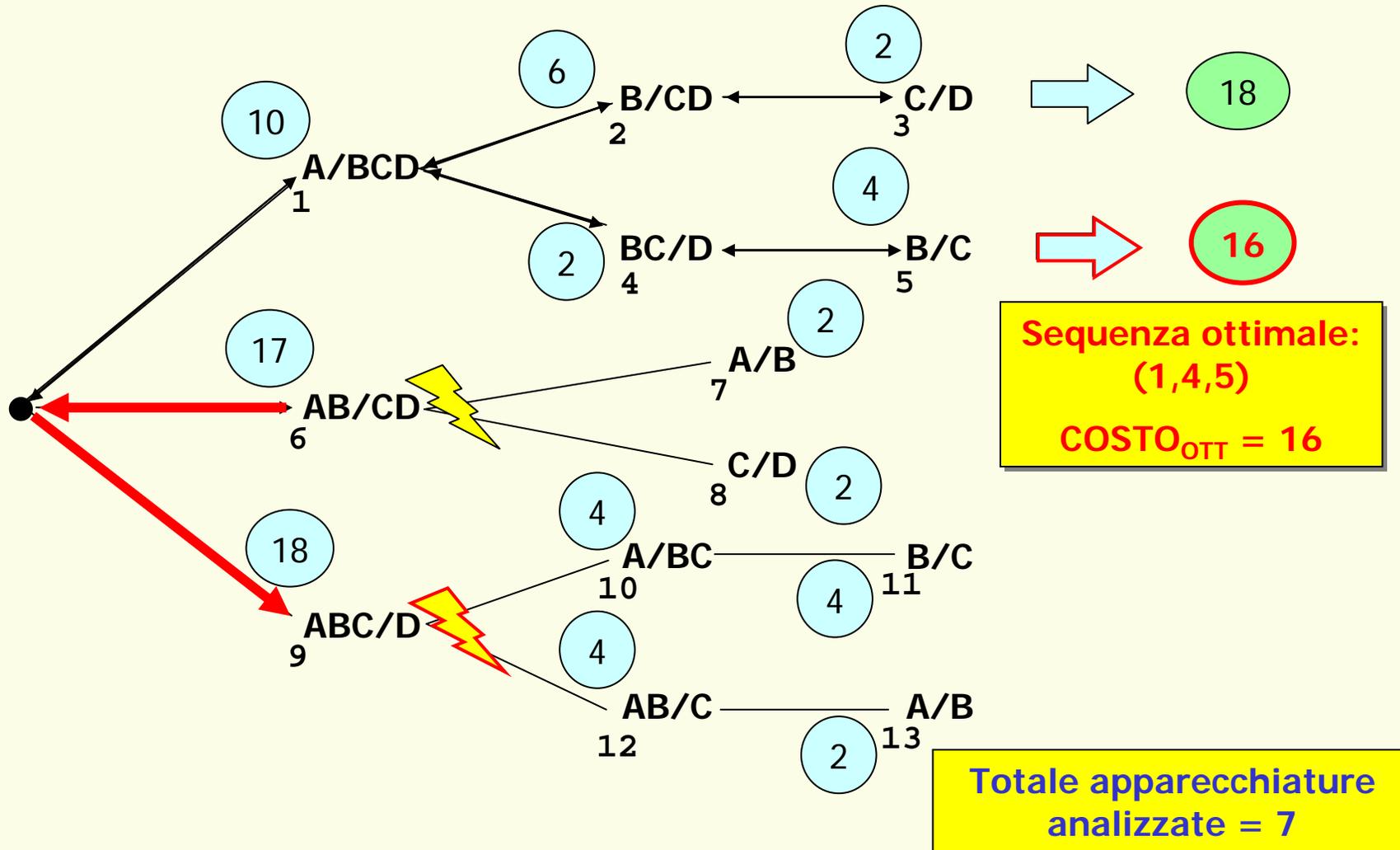


Sequenza ottimale:  
(1,4,5)

# "Branch and Bound" a priorità di profondità



# "Branch and Bound" a priorità di profondità



# "Branch and Bound" a priorità di ampiezza

- Si procede nel seguente modo:
  - Dal nodo radice diramare la ricerca verso i nodi:
    - Nodo 1, CP = 10
    - Nodo 6, CP = 17
    - Nodo 9, CP = 18
  - Selezionare il nodo 1 dato che ha il CP più basso tra i nodi 1,6,9
  - Dal nodo 1 diramare la ricerca verso i nodi:
    - Nodo 2, CP = 10 + 6 = 16
    - Nodo 4, CP = 10 + 2 = 12
  - Selezionare il nodo 4 dato che ha il CP più basso tra i nodi 6,9,2,4
  - Dal nodo 4 diramare la ricerca verso il nodo:
    - Nodo 5, CP = 12 + 4 = 16
  - Dato che il nodo 5 è terminale → il limite superiore attuale LS = 16
    - Sequenza parziale migliore: (1,4,5)
- A partire dai nodi restanti 6,9,2 il nodo 2 ha il CP più basso: CP = 16

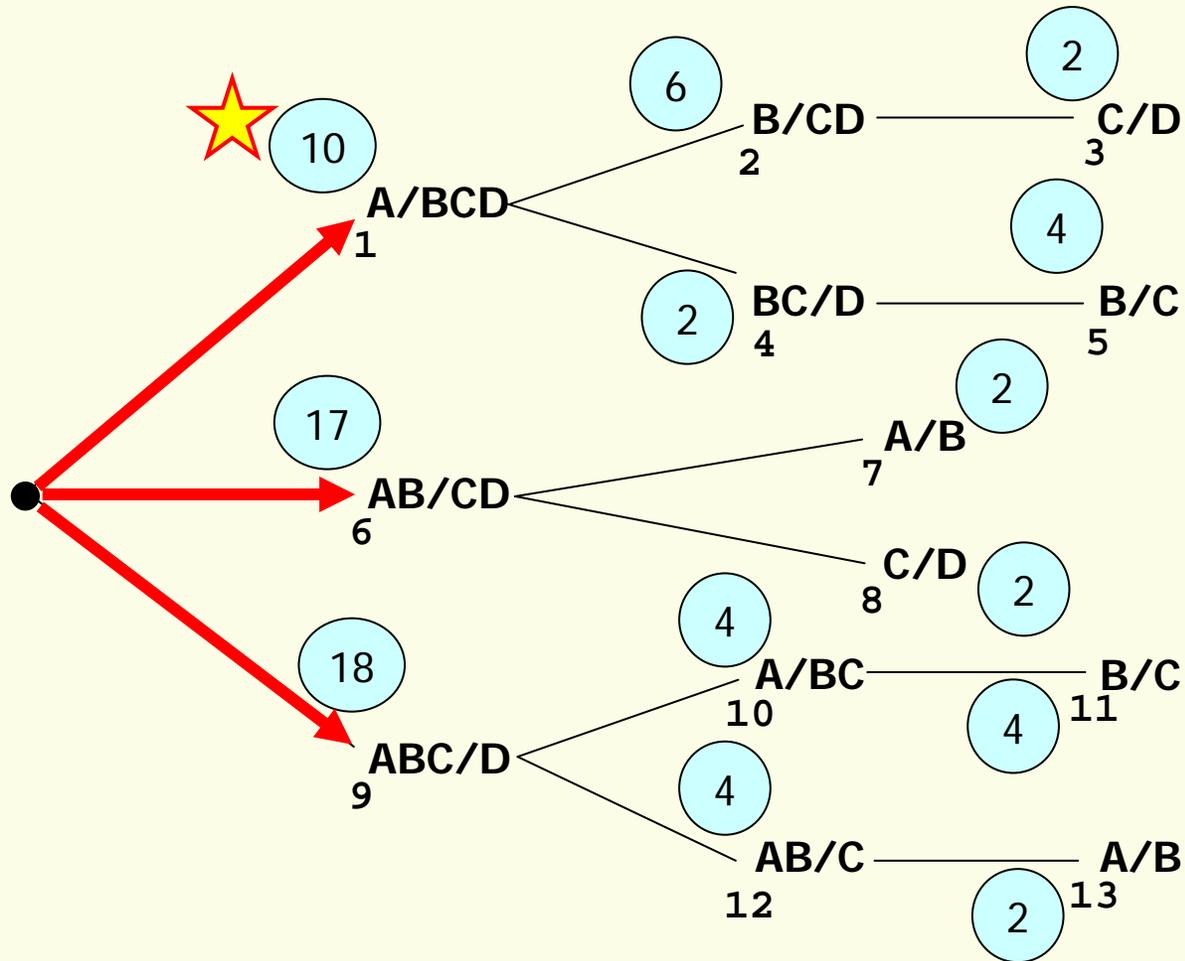


# "Branch and Bound" a priorità di ampiezza

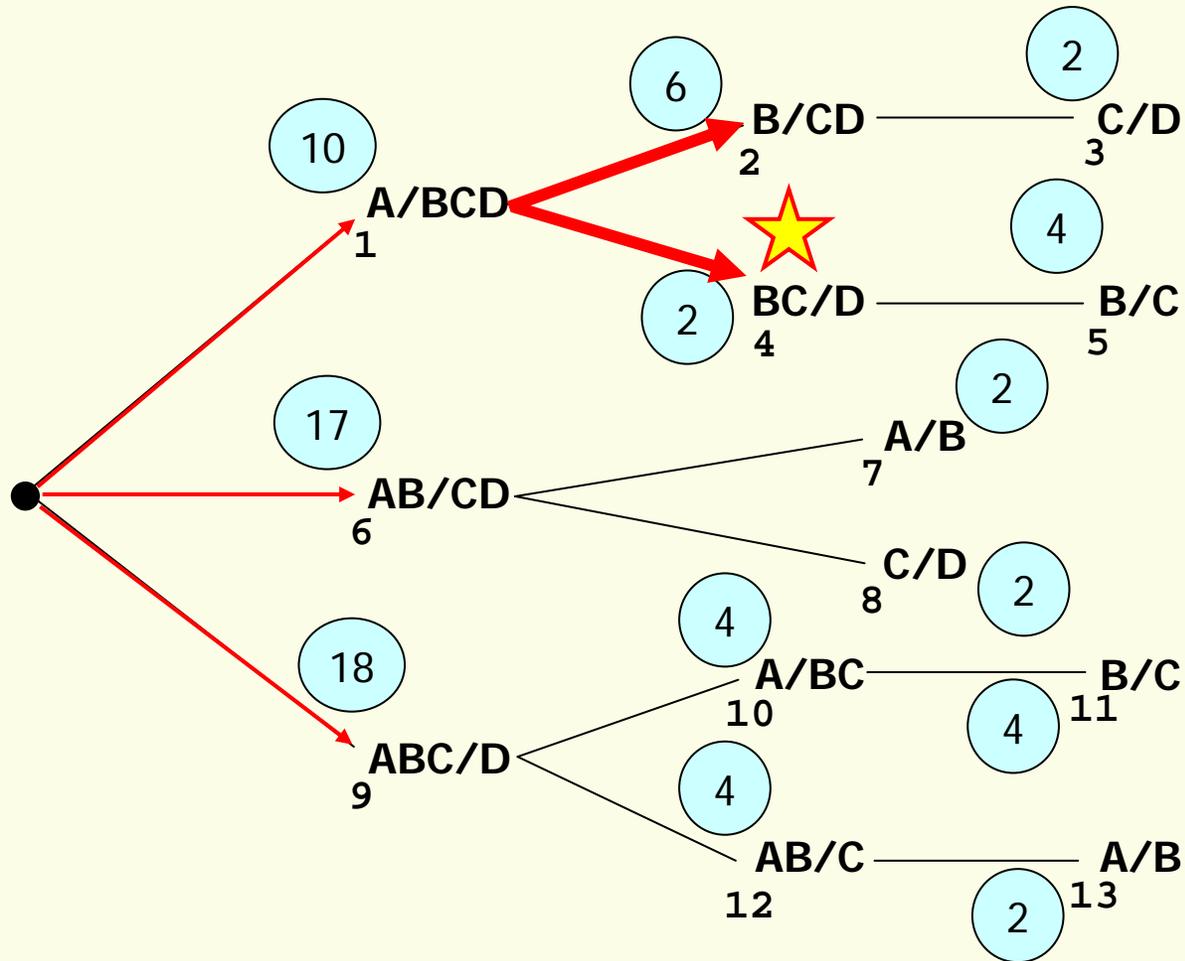
- Dato che il CP del nodo 2 è già uguale a quello del LS, anche se il nodo 2 NON è terminale, allora a maggior ragione anche gli altri nodi 6,9 producono sequenze con costi maggiori del limite superiore migliore LS → sfrondare i rami originati dai nodi: 6,9,2.
- Dato che non esistono altri nodi da cui riprendere la ricerca → STOP
  - Sequenza ottimale: (1,4,5) con  $COSTO_{OTT} = 16$
  - Totale apparecchiature analizzate: 6 su 13.



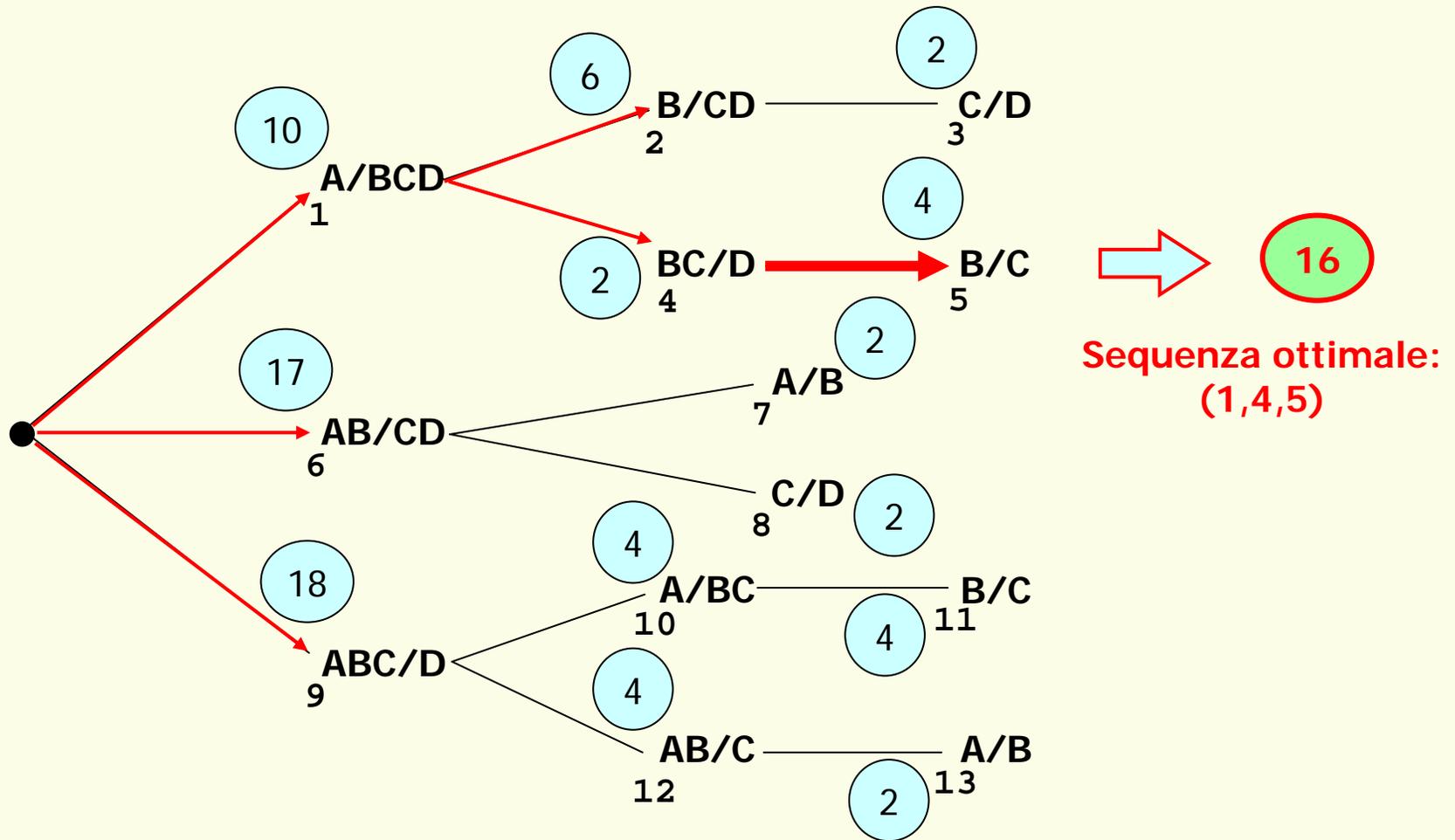
# "Branch and Bound" a priorità di ampiezza



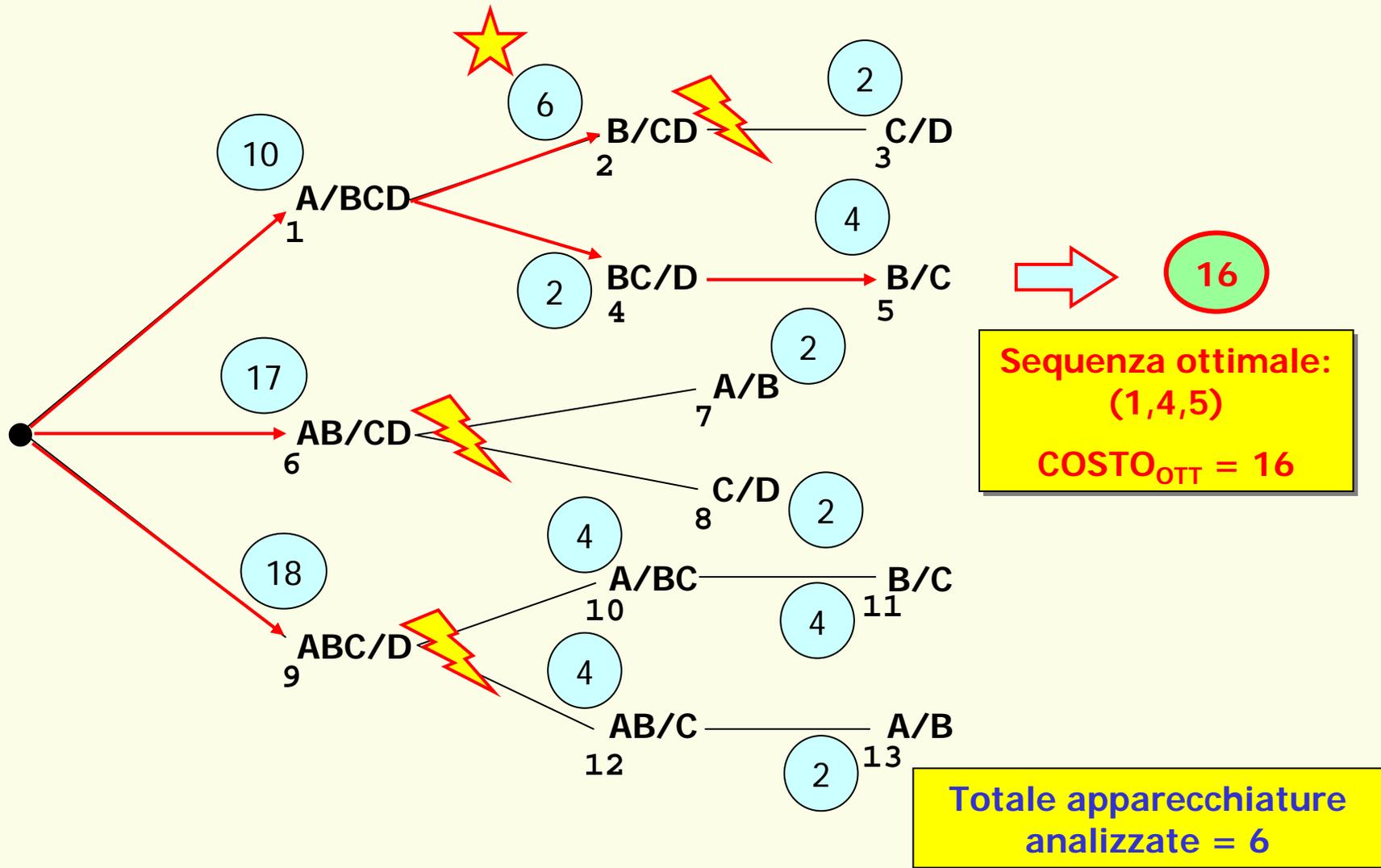
# "Branch and Bound" a priorità di ampiezza



# "Branch and Bound" a priorità di ampiezza



# "Branch and Bound" a priorità di ampiezza



# ***"Branch and Bound"* osservazioni generali**

- In genere la ricerca a priorità di ampiezza richiede un numero di apparecchiature da valutare inferiore a quello prodotto dalla ricerca a priorità di profondità.
- Il metodo a priorità di profondità richiede un'allocazione di memoria inferiore in quanto lavora in modo più sequenziale e meno parallelo rispetto al metodo a priorità di ampiezza.
- Queste metodologie operative vengono anche utilizzate nella risoluzione di problemi MILP (Mixed Integer Linear Programming).



# Strategie risolutive per rappresentazioni a rete

Nel caso della rappresentazione del flowsheet di impianto tramite una rete o analogamente una sovrastruttura l'approccio preferito non è più quello sequenziale basato su successione di nodi. Al contrario si opta per un'**ottimizzazione simultanea** di tutta la rete.

La motivazione principale per tale approccio è che la rete stessa il più delle volte non è sequenziale a causa delle correnti di riciclo ivi presenti.

Il problema di ottimizzazione vede coinvolte due tipologie distinte di variabili:

- Variabili continue: portate, temperature, pressioni, conversioni, ...
- Variabili discontinue: presenza di nodi, presenza di apparecchiature

Le variabili continue sono in genere limitate tra due estremi (quello inferiore la maggior parte delle volte corrisponde con lo zero: portate, frazioni molari, concentrazioni, pressioni relative, ...). Le variabili discrete in genere sono binarie ad indicare la presenza (1) o l'assenza (0) della apparecchiatura o del nodo processistico.

La funzione obiettivo da ottimizzare è funzione di entrambe le tipologie di variabili. Da ciò scaturiscono i problemi: MILP o MINLP.

**MILP:** Mixed Integer Linear Programming: funzione obiettivo e vincoli sono lineari.

**MINLP:** Mixed Integer Non-Linear Programming: funzione obiettivo e/o vincoli sono non lineari.



# Formulazioni matematiche alternative

Il problema di ottimizzazione viene riassunto nella formula:

$$\begin{array}{ll} \textit{Min} & f(x, y) \\ \textit{s.t.} & h(x) = 0 \\ & g(x, y) \leq 0 \\ & x \geq 0, y \in \{0, 1\}^m \end{array}$$

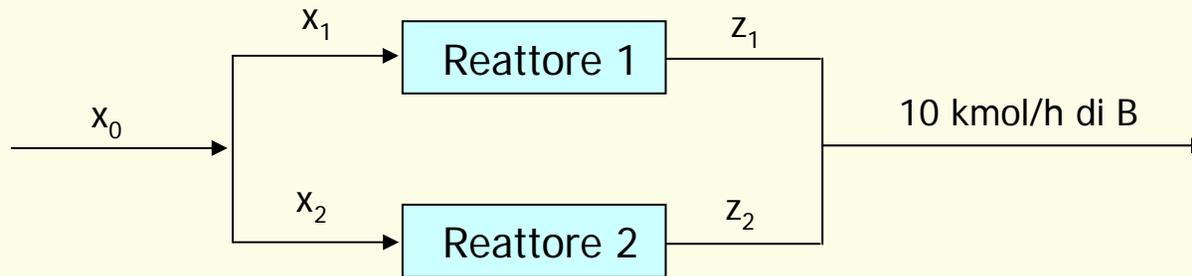
Al fine di introdurre le implicazioni logiche legate all'uso di **variabili discrete binarie** per l'ottimizzazione di un layout di processo (sintesi del flowsheet ottimale), si presenta un esempio molto semplice.



# Formulazioni matematiche alternative

## Esempio: Selezione ottimale di reattori

Si desidera selezionare la tipologia di reattori più adatta per condurre la reazione:  $A \rightarrow B$ . Utilizzando il reattore 1 la conversione è del 80% ma il costo è superiore. Utilizzando il reattore 2 la conversione è del 66.7% ed il costo è inferiore. Occorre produrre 10 kmol/h di B ed il costo del reagente A è di 5 €/kmol.



## Svolgimento

Assegnate le variabili binarie  $y_1$  e  $y_2$   $\{0,1\}$  che indicano la presenza/assenza dei reattori 1 e 2, il costo di investimento di ogni reattore (quota fissa + quota funzione della capacità) è:

$$C_1 = 7.5y_1 + 1.4x_1, \quad C_2 = 5.5y_2 + 1.0x_2,$$

Con  $x_1$  e  $x_2$  le rispettive correnti in alimentazione ai due reattori posti in parallelo.

**N.B.:** i due reattori possono coesistere. Si desidera altresì evidenziare che se un reattore è assente allora anche la corrispondente corrente di alimentazione deve essere nulla.

# Formulazioni matematiche alternative

La condizione di alimentazione nulla  $x_i$  per assenza del reattore  $i$ -esimo si traduce matematicamente in:

$$x_1 - 20 y_1 \leq 0 \quad \text{con } x_1 \geq 0 \qquad x_2 - 20 y_2 \leq 0 \quad \text{con } x_2 \geq 0$$

Dove 20 è un valore sufficientemente alto (limite superiore) per le portate alimentate ai due reattori. Si noti che se  $y_1=0 \rightarrow x_1=0$ .

Il problema di ottimo economico si traduce in:

$$\text{Min } C = 7.5y_1 + 1.4x_1 + 5.5y_2 + 1.0x_2 + 5.0 (x_1 + x_2) =$$

$$7.5y_1 + 6.4x_1 + 5.5y_2 + 6.0x_2$$

$$\text{s.t.} \quad 0.8 x_1 + 0.67 x_2 = 10$$

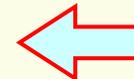
$$x_1 - 20 y_1 \leq 0 \qquad x_2 - 20 y_2 \leq 0$$

$$x_1, x_2 \geq 0 \qquad y_1, y_2 = \{0, 1\}$$

Il problema risultante a causa della linearizzazione dei costi di investimento dei reattori è lineare con variabili continue e variabili discrete binarie: **MILP**.

Risolvendo il problema tramite una ricerca esaustiva sulle variabili binarie si ottiene:

y1	y2	C [€/h]
0	0	non fattibile
<b>1</b>	<b>0</b>	<b>87.5</b>
0	1	95.5
1	1	93.0



# Inferenza logica

La presenza di variabili discrete di tipo binario si fa risentire anche a livello di costruzione dei vincoli logici in grado di descrivere certi assunti processistici di tipo qualitativo.

**Esempio 1:** si supponga di assegnare durante la ricerca dell'ottimo nella sintesi di processo il vincolo: *"un solo reattore può essere selezionato da un insieme di possibili candidati"*.

Matematicamente avremo:  $\sum_{i \in R} y_i = 1$

**Esempio 2:** si rappresenti con un formalismo matematico la seguente condizione di vincolo: *"se l'assorbitore per il recupero del prodotto è presente o il separatore a membrana è selezionato allora non bisogna adottare la separazione criogenica"*.

$$y_{ASS} + y_{MEMB} + 2y_{CRIO} \leq 2$$

Si nota come la trasformazione della condizione semantica 2 in un costrutto matematico equivalente non è banale. Si sottolinea altresì come la condizione "o" in realtà racchiuda a livello logico un'affermazione "e/o".

È anche possibile suddividere il vincolo logico 2 in due espressioni distinte:

$$\begin{cases} y_{ASS} + y_{CRIO} \leq 1 \\ y_{MEMB} + y_{CRIO} \leq 1 \end{cases} \quad \Rightarrow \quad y_{ASS} + y_{MEMB} + 2y_{CRIO} \leq 2$$

# Inferenza logica

Al fine di derivare in modo sistematico i vincoli che contengono variabili logiche (binarie: 0,1) è opportuno basarsi su di un approccio rigoroso ed una metodologia sistematica.

Vengono introdotte delle trasformazioni che convertono le espressioni logiche in una rappresentazione matematica equivalente.

- Ad ogni proposizione  $P_i$  che rappresenta una selezione od un'azione si assegna una variabile binaria  $y_i$ .
- La negazione, o il complemento, di  $P_i$  è  $\neg P_i$  e vale:  $1 - y_i$ .
- Il valore logico VERO vale 1 mentre il valore logico FALSO vale 0.

Valgono le seguenti relazioni:

Relazione logica	Commenti	Espressione booleana	Rappresentazione come disuguaglianza lineare
OR logico		$P_1 \vee P_2 \vee \dots \vee P_r$	$y_1 + y_2 + \dots + y_r \geq 1$
AND logico		$P_1 \wedge P_2 \wedge \dots \wedge P_r$	$y_1 \geq 1 \quad y_2 \geq 1 \quad \dots \quad y_r \geq 1$
Implicazione	$P_1 \Rightarrow P_2$	$\neg P_1 \vee P_2$	$1 - y_1 + y_2 \geq 1$
Equivalenza	$(P_1 \Rightarrow P_2) \wedge (P_2 \Rightarrow P_1)$	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$y_1 = y_2$
XOR logico	Esattamente una delle variabili è vera	$P_1 \underline{\vee} P_2 \underline{\vee} \dots \underline{\vee} P_r$	$y_1 + y_2 + \dots + y_r = 1$

# Inferenza logica

Tramite i costrutti logico—matematici, indicati nella tabella precedente, è possibile trasformare le espressioni logiche in vincoli lineari di uguaglianza o disuguaglianza.

Un approccio volto alla trasformazione delle condizioni logiche in costrutti matematici è quello della “**congiunzione in forma normale**” che raccoglie un insieme di condizioni tramite operatori AND:  $Q_1 \wedge Q_2 \wedge \dots \wedge Q_s$ .

Affinché la forma normale sia vera occorre che ogni condizione  $Q_i$  sia vera indipendentemente dalle altre.

Dato che ogni condizione  $Q_i$  è una disgiunzione di proposizioni  $P_i$  connesse da operatori OR: allora può essere espressa tramite la disuguaglianza:  $P_1 \vee P_2 \vee \dots \vee P_r$ .

Clocksink e Mellish (1981) hanno proposto una metodologia per tradurre le espressioni logiche in congiunzioni in forma normale.

1. Sostituzione dell'implicazione con la sua disgiunzione equivalente:

$$P_1 \Rightarrow P_2 \quad \Leftrightarrow \quad \neg P_1 \vee P_2$$

2. Trasformazione della negazione tramite il teorema di De Morgan:

$$\neg (P_1 \wedge P_2) \quad \Leftrightarrow \quad \neg P_1 \vee \neg P_2$$

$$\neg (P_1 \vee P_2) \quad \Leftrightarrow \quad \neg P_1 \wedge \neg P_2$$

3. Distribuzione ricorsiva dell'operatore OR sull'operatore AND

$$(P_1 \wedge P_2) \vee P_3 \quad \Leftrightarrow \quad (P_1 \vee P_3) \wedge (P_2 \vee P_3)$$



# Inferenza logica

**Esempio 3:** si traduca in termini matematici la seguente condizione: *“se l'assorbitore per il recupero del prodotto è presente o il separatore a membrana è selezionato allora non bisogna adottare la separazione criogenica”*.

- Assegnando le seguenti proposizioni:  $P_A$  = seleziona l'assorbitore  
 $P_M$  = seleziona il separatore a membrana  
 $P_C$  = seleziona il separatore criogenico
- La condizione semantica viene tradotta nell'espressione logica:  $P_A \vee P_M \Rightarrow \neg P_C$
- L'implicazione logica “ $\Rightarrow$ ” viene rimossa e porta a:  $\neg (P_A \vee P_M) \vee \neg P_C$
- Applicando il teorema di De Morgan si ha:  $(\neg P_A \wedge \neg P_M) \vee \neg P_C$
- Distribuendo infine l'OR sull'AND:  $(\neg P_A \vee \neg P_C) \wedge (\neg P_M \vee \neg P_C)$
- È ora possibile introdurre le variabili binarie e trasformare le istruzioni AND logiche in costrutti matematici:  
$$1 - y_A + 1 - y_C \geq 1$$
$$1 - y_M + 1 - y_C \geq 1$$
- Che è infine equivalente alle condizioni già precedentemente dedotte intuitivamente:

$$y_A + y_C \leq 1 \text{ e } y_M + y_C \leq 1 \text{ che sommate danno: } y_A + y_M + 2y_C \leq 2$$

Il vantaggio di questo approccio è che un'espressione logica complessa è stata trasformata “automaticamente” in un costrutto matematico basandosi su regole precise e definite.



# Inferenza logica

**Esempio 4:** si trasformi l'espressione logica:  $(P_1 \wedge P_2) \vee P_3 \Rightarrow (P_4 \vee P_5)$  in un'equivalente espressione matematica.

- Rimuoviamo l'implicazione:  $\neg [(P_1 \wedge P_2) \vee P_3] \vee (P_4 \vee P_5)$
- Appliciamo il teorema di De Morgan:  $[\neg (P_1 \wedge P_2) \wedge \neg P_3] \vee (P_4 \vee P_5)$
- Appliciamo nuovamente il teorema di De Morgan:  $[(\neg P_1 \vee \neg P_2) \wedge \neg P_3] \vee (P_4 \vee P_5)$
- Distribuiamo infine l'OR sull'AND:  $[(\neg P_1 \vee \neg P_2) \vee (P_4 \vee P_5)] \wedge [\neg P_3 \vee (P_4 \vee P_5)]$
- Che è equivalente a:  $(\neg P_1 \vee \neg P_2 \vee P_4 \vee P_5) \wedge (\neg P_3 \vee P_4 \vee P_5)$
- È ora possibile introdurre le variabili binarie e trasformare le istruzioni AND logiche in costrutti matematici:  
$$1 - y_1 + 1 - y_2 + y_4 + y_5 \geq 1$$
$$1 - y_3 + y_4 + y_5 \geq 1$$
- ovvero:  
$$y_1 + y_2 - y_4 - y_5 \leq 1$$
$$y_3 - y_4 - y_5 \leq 0$$

## Considerazioni:

Ancora una volta abbiamo visto come una condizione logica sia stata trasformata in un insieme di disequazioni cioè di vincoli su variabili binarie facilmente trattabili da algoritmi di ottimizzazione di tipo: MILP o MINLP.

È questo un problema di inferenza logica per il quale, dato un insieme di  $n$  proposizioni logiche si desidera provare se la condizione ad esse connessa è vera.



# Inferenza logica

Il problema di inferenza logica si riduce quindi a provare la verità di una conclusione a partire da un insieme di proposizioni valide:

$$\begin{array}{ll} \text{Prove} & Q_u \\ \text{s.t.} & B(Q_1, Q_2, \dots, Q_s) \end{array}$$

Dove  $Q_u$  è la condizione che esprime la conclusione da provare mentre  $B$  è l'insieme di condizioni:  $Q_i, i=1, \dots, s$ .

Trasformando le condizioni logiche in un insieme di vincoli lineari di uguaglianza e disuguaglianza, Cavalier e Soyster (1987) hanno dimostrato che **il problema di inferenza logica è equivalente al seguente problema MILP**:

$$\begin{array}{ll} \text{Min} & Z = \sum_{i \in I(u)} c_i y_i \\ \text{s.t.} & Ay \geq a \\ & y \in \{0, 1\}^n \end{array}$$

Dove  $Ay \geq a$  è l'insieme delle disequazioni ottenute dalla traduzione delle condizioni  $B(Q_1, Q_2, \dots, Q_s)$  in forma matematica e  $I(u)$  corrisponde all'insieme di variabili binarie (*gdI*) che compaiono nella condizione  $Q_u$ .



# Inferenza logica

La condizione  $Q_u$  è sempre vera se l'algoritmo di minimizzazione determina  $Z = 1$ .

Se viceversa  $Z = 0$  allora esiste una combinazione delle variabili  $y_i$  che produce una situazione in cui  $Q_u$  è falsa. Possiamo cioè affermare che  $Q_u$  non è sempre vera.

Risulta spesso comodo risolvere il problema nella “**forma rilassata**” ovvero utilizzando variabili  $y_i$  continue anziché discrete (come proposto da Hooker, 1988). In tale caso è possibile affermare che:

1. Se  $Z_{ril} > 0$  allora la condizione  $Q_u$  è sempre vera anche se  $Z_{ril} < 1$ . Infatti dato che  $Z_{ril}$  è un estremo inferiore per la soluzione effettiva  $Z$  del problema MILP ciò implica che non esiste una soluzione intera con  $Z = 0$ . Quindi la soluzione intera deve essere:  $Z = 1$ .
2. Se  $Z_{ril} = 0$  e la soluzione (vettore delle  $y_{ril}$ ) è frazionaria ed unica allora la condizione  $Q_u$  è sempre vera dato che non esiste una soluzione intera con  $Z = 0$ .

Infine se  $Z_{ril} = 0$  e la soluzione è frazionaria ma **non** unica allora non è possibile trarre alcuna conclusione a partire dalla soluzione del problema LP rilassato. Ciò è dovuto al fatto che potrebbe esistere anche una soluzione intera tale che  $Z_{ril} = Z = 0$ .

# Bibliografia

- Andrecovich, M. J., Westerberg, A. W., *MILP formulation for heat-integrated distillation sequence synthesis*, AIChE J., **31**, 1461, (1985)
- **Biegler, L. T., Grossmann, I. E., Westerberg, A. W., "Systematic Methods of Chemical Process Design", Prentice Hall PTR, Upper Saddle River, (1997)**
- Cavalier, T. M., Soyster, A. L., *Logical Deduction via Linear Programming*, IMSE, **87**, 147, (1987)
- Clocksin, W. F., Mellish, C. S., "Programming in Prolog", Springer Verlag, New York, (1981)
- Hendry, J. E., Hughes, R. R., *Generating separation process flowsheets*, Chem. Eng. Progress, **68**, 69, (1972)
- Hooker, J. N., *Resolution vs cutting plane solution of inference problems: Some computational experience*, Operations Research Letters, **7**, 1, (1988)
- Raman, R., Grossmann, I. E., *Relation between MILP modeling and logical inference for chemical process synthesis*, Comp. Chem. Eng., **15**, 73, (1991)
- Raman, R., Grossmann, I. E., *Modeling and computational techniques for logic based integer programming*, Comp. Chem. Eng., **18**, 563, (1994)
- Williams, H. P., "Model building in mathematical programming", Wiley, New York, (1985)

