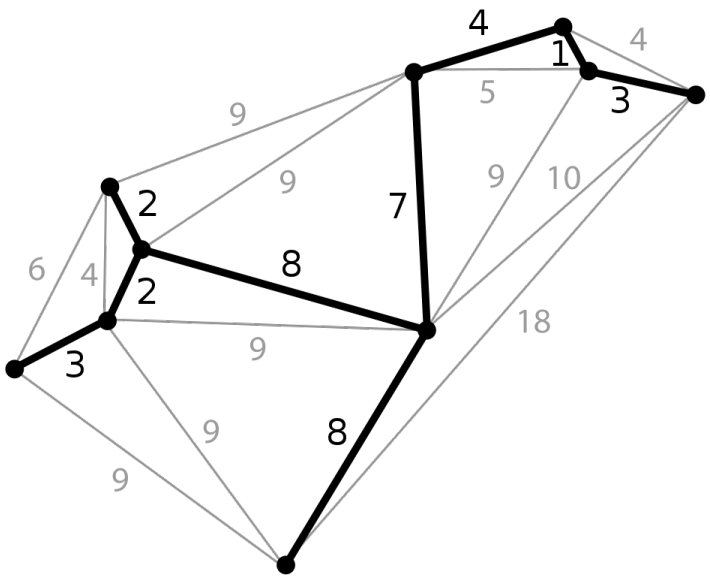



# METODI ED ALGORITMI DI OTTIMIZZAZIONE PER IL PROBLEM SOLVING

Docente:
Aristide Mingozzi

Adattamento:
Edoardo Rosa

---

Anno Accademico 20014-2015

# INDICE


**1 Modelli e formulazioni matematiche** **1**
- 1.1 The Traveling Salesman Problem . . . 1
  - 1.1.1 Formulazioni Matematiche del TSP . . . 2
  - 1.1.2 Eliminazione subtours di Miller, Tucker, Zemlin (1960) . . . 3
  - 1.1.3 Il Traveling salesman problem con time windows (TSPTW) . . . 4
- 1.2 Project scheduling with resource constraints (PSR) . . . 6
  - 1.2.1 Esempio di PSR . . . 6
  - 1.2.2 Formulazione del PSR . . . 7
- 1.3 Fixed Charge Transportation Problem (FCTP) . . . 8
  - 1.3.1 Descrizione del FCTP . . . 8
  - 1.3.2 Formulazione del FCTP . . . 8
- 1.4 Assegnamento dei veicoli alle baie di carico . . . 10
  - 1.4.1 Formulazione matematica F . . . 10
- 1.5 Lot Sizing Problem . . . 12
  - 1.5.1 Lot sizing senza vincoli di capacità . . . 12

**2 Introduzione alla programmazione lineare a numeri interi** **16**
- 2.1 Arrotondamento ad una soluzione non-intera . . . 17
- 2.2 Unimodularità . . . 20
- 2.3 Metodo dei piani di taglio . . . 22
  - 2.3.1 Piani di taglio . . . 22
  - 2.3.2 Gomory cuts . . . 23
- 2.4 Metodi Branch and Bound . . . 28
  - 2.4.1 Tipi di Branching . . . 31
  - 2.4.2 Bounds . . . 32
  - 2.4.3 Eliminazione di alcuni vincoli . . . 37
  - 2.4.4 Rilassamento Surrogato . . . 37

2.5 Assegnamento Generalizzato . . . . . 38
2.5.1 Formulazione matematica . . . . . 39
2.5.2 Rilassamento lagrangiano . . . . . 39
2.5.3 Algoritmo Branch & Bound . . . . . 45

**A Prova** **46**
A.1 Pippo . . . . . 46

# ELENCO DELLE FIGURE

1.1 Grafo orientato . . . . . 3
1.2 Grafo H delle precedenze . . . . . 6
1.3 Esempio della rete di flusso (modello di Wagner-Whitin) . . . . . 14
1.4 . . . . . 14
1.5 . . . . . 15

2.3 * Problemi risolti . . . . . 32

**Copertina**: `http://commons.wikimedia.org/wiki/File:Minimum_spanning_tree.svg`

# ELENCO DELLE TABELLE

2.1 Tableau ottimo. Soluzione continua! . . . . . 25
2.2 Tableau ottimo. . . . . 28

# CAPITOLO 1

# MODELLI E FORMULAZIONI MATEMATICHE

## 1.1 The Traveling Salesman Problem

Il Traveling Salesman Problem (TSP) è il problema più noto dell'ottimizzazione combinatoria. Siano date $n$ città e i costi $c_{ij}$ per andare dalla città $i$ alla città $j$. Si vuole determinare un cammino che parte da una città (diciamo $i_1$), visitare una ed una sola volta tutte le rimanenti città e terminare nella città di partenza $i_1$. Inoltre si vuole che il costo di tale cammino sia minimo.
Ha molteplici applicazioni pratiche e teoriche perche è la struttura di molti problemi pratici.
Si è soliti modella il TSP come segue:

- è dato un grafo orientato (o non orientato) $G = (N, A)$ dove $N$ è un insieme di $n$ vertici e $A$ è un insieme di $m$ archi.

  Ad ogni arco *(i,j)* $\in A$ è associato un costo $c_{ij}$.

  Un circuito hamiltoniano di $G$ è un circuito che passa per ogni vertice una ed una sola volta.
  Il costo di un circuito hamiltoniano di $G$ è pari alla somma dei costi degli archi che compongono il circuito;

- il problema del TSP è di trovare un grafo $G$, con una data matrice dei costi $[c_{ij}]$, un circuito hamiltoniano di costo minimo.

### 1.1.1 Formulazioni Matematiche del TSP

In letteratura esistono molteplici (e a volte fantasiose) formulazioni del TSP.
Presentiamo le due formulazioni più note e su cui si basano i metodi esatti più efficienti.

#### 1.1.1.1 TSP asimmetrico

I costi $c_{ij}$ non verificano $c_{ij} = c_{ji} \; \forall \; i, j$ con $i < j$.
Sia $x_{ij}$ una variabile $(0-1)$ associata ad ogni arco $(i,j) \in A$ dove $x_{ij} = 1$ se l'arco $(i,j)$ è nella soluzione ottima e $x_{ij} = 0$ altrimenti.

$$Min \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \tag{1.1}$$

$$s.t. \sum_{i \in N} x_{ij} = 1, \;\; \forall j \in N \tag{1.2}$$

$$\sum_{j \in N} x_{ij} = 1, \;\; \forall i \in N \tag{1.3}$$

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} \geq 1, \;\; \forall S \subset N \tag{1.4}$$

$$x_{ij} \in \{0,1\} \; , \; \forall (i,j) \in A \tag{1.5}$$

Il vincolo 1.4 impone che ogni soluzione ammissibile debba contenere almeno un arco $(i,j)$ con $i \in S$ e $j \in N \setminus S$ per ogni sottoinsieme $S$ di $N$. Un'alternativa al vincolo 1.4 è:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \;\; \forall S \subset N \tag{1.4'}$$

#### 1.1.1.2 TSP simmetrico

Sia dato un grafo non-orientato $G = (N, A)$ con $c_{ij} = c_{ji} \; , \forall i, j \in N$.
Gli archi di $A$ sono numerati da 1 a $m$. L'arco di indice $l$ corrisponde a $(\alpha_l, \beta_l)$ con $\alpha_l < \beta_l$.
$A_i$ è il sottoinsieme degli indici degli archi che incidono sul vertice $i$:

$$A_i = \{l : \; l = 1, m \;\; s.t. \; \alpha_l = i \; or \; \beta_l = i\}$$

Per una dato $S \in N$ e $\bar{S} = N \setminus S$ indichiamo con $(S, \bar{S})$ il sottoinsieme degli indici degli archi per cui $\alpha_l \in S$ e $\beta_l \in \bar{S}$ oppure $\alpha_l \in \bar{S}$ e $\beta_l \in S$.

Ad ogni arco di incide $l$ è associato un costo $d_l = c_{\alpha_l \beta_l}$ e $x_l \in \{0,1\}$ è una variabile che vale 1 se e solo se l'arco di indice $l$ è nella soluzione ottima.

$$Min \sum_{l=1} d_l \, x_l \tag{1.6}$$

$$s.t. \sum_{l \in A_i} x_l = 2, \ \forall i \in N \tag{1.7}$$

$$\sum_{l \in (S,\bar{S})} x_l \geq 1, \ \forall S \subset N \tag{1.8}$$

$$x_l \in \{0,1\}, \ \ l = 1, ..., m \tag{1.9}$$

### 1.1.2 Eliminazione subtours di Miller, Tucker, Zemlin (1960)

Sia $u_i$ una variabile intera il cui valore sappresenta la posizione che il vertice $i$ occupa nel tour.

Es. tour (1,4,5,3,2,1) per TSP con n=5 vertici, si ha $u_1 = 1, \ u_2 = 5, \ u_3 = 4, \ u_4 = 2, \ u_5 = 3$

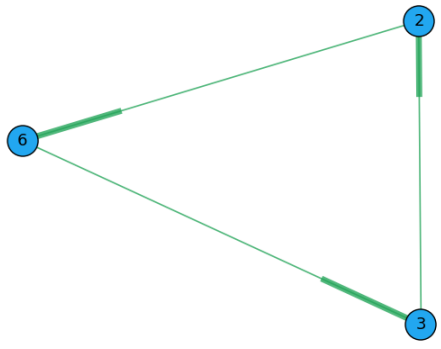
Miller, Tucker e Zemlin propongono in alternativa a:

$$\sum_{i \in S} \sum_{j \in N \setminus S} x_{ij} \geq 1, \ \ \forall S \subset N \tag{*}$$

hanno imposto i seguenti vincoli:

$$u_i - u_j + n x_{ij} \leq n - 1, \ \ i = 1, ..., n \ , \ j = 2, ..., n \tag{1.10}$$

Ogni tour hamiltoniano soddisfa questi vincoli e ogni subtour li viola.




$$u_2 - u_6 + n \ \cdot x_{2,6} \leq n - 1$$
$$u_6 - u_3 + n \ \cdot x_{6,3} \leq n - 1$$
$$u_3 - u_2 + n \ \cdot x_{3,2} \leq n - 1$$
$$\downarrow$$
$$3n \leq 3(n-1)$$

Figura 1.1: Grafo orientato

### 1.1.3 Il Traveling salesman problem con time windows (TSPTW)

È una variante del TSP che ha molte applicazioni.

Sia dato un grafo orientato $G = (V, A)$ di $n + 1$ vertici ($V = \{0, 1, ..., n\}$).

Ad ogni arco $(i, j) \in A$ sono associati

- un costo $c_{ij} \geq 0$
- un tempo di percorrenza $\theta_{ij} \geq 0$

Ad ogni vertice è associato un intervallo $[r_i, d_i]$ chiamato "time window" che rappresenta l'orario in cui il vertice $i$ può essere vistato dal "salesman".

Ovvero il salesman può visitare $i$ ad ogni tempo $t \in \mathbb{Z}^+$ con $r_i \leq t \leq d_i$.
Il problema consiste nel trovare una sequenza dei vertici di $G$ che parte dal vertice 0 al tempo 0 e finisce al nodo 0 tale che sia il minimo il costo del circuito e il tempo di arrivo al nodo $i$ sia nell'intervallo $[r_i, d_j]$, $\forall i \in V$.
Si consideri la sequenza $(0, i, .., i_{k-1}, i_k, ..., i_n, 0)$ e sia $t_{i_k}$ il tempo di arrivo al vertice $i_k$, $k = 0, 1, ..., n + 1$.

I tempi di arrivo sono calcolati come:

$$t_0 = 0 \tag{1.11}$$

$$t_{i_k} = \max\{t_{i_{k-1}} + \theta_{i_{k-1}} \cdot i_k,\ r_{i_k}\} \tag{1.12}$$

#### 1.1.3.1 Formulazione del TSPTW

Sia $x_{ij}$ una variabile binaria intera che assume il valore 1 se il vertice $i$ è visitato immediatamente prima di $i$ e $x_{ij} = 0$ altrimenti.

$$Min \quad \sum_{(i,j)\in A} c_{ik} x_{ij} \tag{1.13}$$

$$s.t. \quad \sum_{i\in A_j^-} x_{ij} = 1, \ \ \forall j \in V \tag{1.14}$$

$$\sum_{j\in A_i^+} x_{ij} = 1, \ \ \forall i \in V \tag{1.15}$$

$$t_i + \theta_{ij} - t_j \leq M(1 - x_{ij}, \ \ \forall (i, j) \in A, \ j \neq 0) \tag{1.16}$$

$$t_i \leq d_i, \ \ \forall i \in V \tag{1.17}$$

$$t_i \geq r_i, \ \ \forall i \in V \tag{1.18}$$

$$x_{ij} \in \{0, 1\}, \ \ \forall \in A \tag{1.19}$$

$$t_i \in \mathbb{N}^+, \ \ \forall i \in V \tag{1.20}$$

dove

$$A_i^+ = \{j \in V : (i,j) \in A\}$$
$$A_i^- = \{j \in V : (i,j) \in A\}$$
$$M \ \textit{un intero grande a piacere}$$
$$r_0 = d_0 = 0$$

## 1.2 Project scheduling with resource constraints (PSR)

È dato un insieme $\mathbb{X} = \{1, ..., n\}$ di $n$ jobs.

Sono disponibili $m$ risorse dove ogni risorsa $k$ ha una disponibilità $b_k$ ad ogni istante del periodo di scheduling.

Ogni job $i$ ha un tempo di processo $d_i$ e la sua esecuzione, una volta iniziata, non può essere interrotta.

Il job $i$ per essere eseguito richiede $b_{ik}$ unità della risorsa $k$ per ciascun intervallo di tempo in cui rimane in esecuzione.
È dato un grafo $G = (X, H)$ di precedenze, dove ogni arco $(i, j) \in H$ impone che il job $j$ può iniziare solo dopo che il job $i$ è stato completato.

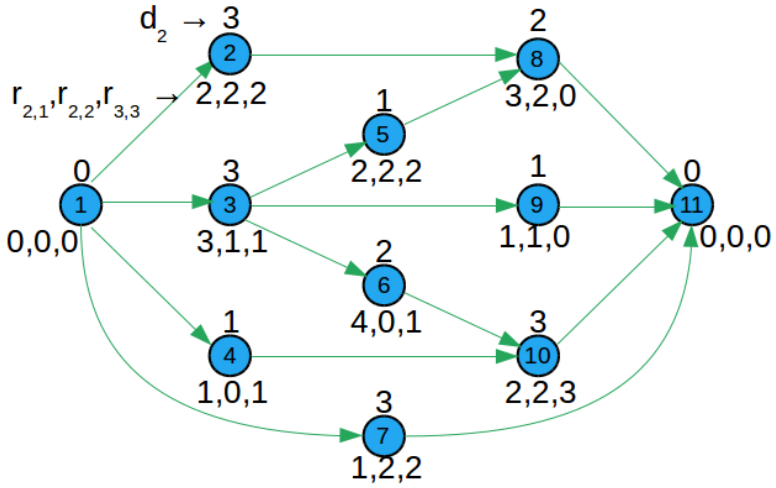
- Si vuole determinare il tempo di inizio di processo di ogni job in modo che siano soddisfatti i vincoli di precedenza, i vincoli sulle risorse e sia minima la durata complessiva del progetto

### 1.2.1 Esempio di PSR

Siano dati $n = 11$ jobs e $m = 3$ risorse con $b_1 = b_2 = b_3 = 4$ e un grafo $H$ delle precedenze corrispondenti agli archi della figura 1.2.

Si osservi che i jobs 2 e 3 non possono essere eseguiti in parallelo poiché $r_{2,1} + r_{3,1} = 5 > b_1$!

Figura 1.2: Grafo H delle precedenze



### 1.2.2 Formulazione del PSR

Sia $\xi_{it}$ una variabile binaria 0-1 che vale 1 se e solo se il job $i$ viene messo in esecuzione al tempo $t$.

Sia $T_{max}$ un upper bound sulla durata del progetto.

$$Min \sum_{t=1}^{T_{max}} t\, \xi_{nt} \tag{1.21}$$

$$s.t. \sum_{t=1}^{T_{max}} t\, \xi_{it} = 1, \quad i = 1, ..., n \tag{1.22}$$

$$\sum_{t=1}^{T_{max}} t\, \xi_{jt} - \sum_{t=1}^{T_{max}} t\, \xi_{it} \geq d_i, \quad \forall (i,j) \in H \tag{1.23}$$

$$\sum_{i=1}^{n} r_{ik} \sum_{\tau=t-d_i+1}^{t} \xi_{i\tau} \leq b_k, \quad t = 1, ..., T_{max} \text{ e } k = 1, ..., m \tag{1.24}$$

$$\xi_{it} \in \{0,1\}, \quad i = 1, ..., n \text{ e } t = 1, ..., T_{max} \tag{1.25}$$

Si osservi che:

$$\sum_{\tau=t-d_i+1}^{t} \xi_{i\tau} = 1 \quad \textit{se il job i in esecuzione al tempo t}$$

#### 1.2.2.1 Esempio

Sia $d_i = 4$.

Se $\xi_{i3} = 1$, allora i è in esecuzione nei tempi 3,4,5 e 6. Infatti avremo:
$\sum_{\tau=t-d_i+1}^{t} \xi_{i\tau} = 1$ per $t = 3, 4, 5, 6$ e $\sum_{\tau=t-d_i+1}^{t} \xi_{i\tau} = 0$ per $t < 3$ e $t > 6$

## 1.3 Fixed Charge Transportation Problem (FCTP)

Il Problema del Trasporto di Carico Fisso è una generalizzazione del classico Problema del Trasporto.

Si differenzia nel definire che il costo per la spedizione di una quantità non-zero di beni, da ogni origine alla sua destinazione, è composto da un costo proporzionale all'ammontare dei beni inviati più un costo fisso.

### 1.3.1 Descrizione del FCTP

Il FCTP è definito su un grafo completo e bipartito $G = (S, T, A)$ dove $S = 1, 2, ..., m$ è un insieme di $m$ sorgenti e $T = 1, 2, ..., n$ è un insieme di $n$ destinazioni.

Per ogni sorgente $i \in S$ è disponibile è una quantità intera $a_i > 0$ di merce e per ogni destinazione $j \in T$ è necessaria una quantità intera $b_j > 0$ di merce dalle sorgenti.
L'insieme $A$ degli archi è definito come: $A = \{(i, j) : \ i \in S, \ j \in T\}$; ogni arco $(i, j) \in A$ è associato ad un costo unitario $c_{ij}$ per il trasporto di una unità della merce dalla sorgente $i$ alla destinazione $j$ più un costo fisso $f_{ij}$ for usare l'arco $(i, j)$.

Senza perdere di generalità si assume che:

$$\sum_{i \in S} a_i = \sum_{j \in T} b_j$$

### 1.3.2 Formulazione del FCTP

Sia $x_{ij}$ una variabile rappresentante la quantità di merce trasportata dalla sorgente $i$ alla destinazione $j$ e $y_{ij}$ una variabile (0-1) che vale 1 se e solo se $x_{ij} > 0$.

Sia $m_{ij} = mina_i, b_j, \ (i, j) \in A$.

Una semplice formulazione matematiche del FCTP è:

$$z(F0) = min \sum_{i \in S} \sum_{j \in T} (c_{ij} x_{ij} + f_{ij} y_{ij}) \tag{1.26}$$

$$s.t. \ \sum_{j \in T} x_{ij} = a_i, \ \ i \in S \tag{1.27}$$

$$\sum_{i \in S} x_{ij} = b_j, \ \ j \in T \tag{1.28}$$

$$x_{ij} \leq m_{ij} y_{ij}, \quad (i,j) \in A \tag{1.29}$$

$$x_{ij} \geq 0, \qquad (i,j) \in A \tag{1.30}$$

$$y_{ij} \in \{0,1\} \tag{1.31}$$

Si denota con $LF0$ il rilassamento lineare del problema $F0$ e con $z(LF0)$ il costo della soluzione ottima. Notare che, per ogni soluzione ottima di $LF0$, le variabili $x_{ij} > 0$ corrispondono ad una soluzione base fattibile dei vincoli 1.27 e 1.28, e $y_{ij} = x_{ij}/m_{ij}$ *con* $(i,j) \in A$.

## 1.4 Assegnamento dei veicoli alle baie di carico

Sia dato un insieme $N$ di veicoli che devono scaricare presso un deposito che ha un insieme $L$ di linee di scarico.

Per ogni linea di scarico $j \in L$ è definito l'insieme degli istanti di tempo $T_j$ in cui è operativa. Per ogni veicolo $i \in N$ sono noti:

- il sottoinsieme di linee $L_i \subseteq L$ compatibili con le operazioni di scarico richieste dal veicolo;
- iltempo di arrivo $a_i$ del veicolo al deposito;
- la durata dello scarico $d_{ij}$ sulla linea $j \in L_i$.

Si assume che lo scarico di un veicolo non possa essere interrotto, ovvero, se lo scarico del veicolo $i$ sulla linea $j \in L_i$ inizia al tempo $t$, allora la linea $j$ deve essere disponibile per tutti gli istanti di tempo $\tau = t, ..., t + d_{ij} - 1$ (ovvero $\tau \in T_j$ per ogni $\tau = t, ..., t + d_{ij} - 1$). Indichiamo con $I_{ij}$ l'insieme degli istanti di tempo in cui può iniziare lo scarico del veicolo $i$ sulla linea $j \in L_i$, ovvero per ogni $t \in I_{ij}$ si assume che la linea $j$ disponibile per ogni istante $\tau = i, ..., d_{ij} - 1$.

Sia $c_{ijt}$ è il costo per iniziare lo scarico del veicolo $i \in N$ sulla linea $j \in Li$ al tempo $t \in I_{ij}$.
Il problema richiede che ogni veicolo sia assegnato ad una linea di scarico compatibile in modo che ogni scarico sia fatto senza interruzioni e sia minimo il costo dell'assegnamento.

### 1.4.1 Formulazione matematica F

Per ogni $i \in N$, $j \in L_i$ e $t \in I_{ij}$ poniamo $\delta_{ijt\tau} = 1$ per $\tau = t, ..., t + d_{ij} - 1$ e $\delta_{ijt\tau} = 0$ per ogni $\tau \in T_j$ tale che $\tau < t$ oppure $\tau > t + d_{ij} - 1$.

Indichiamo con $N_j \subseteq N$ il sottoinsieme di veicoli che possono essere scaricati sulla linea $j$, ovvero $N_j = \{i \in N : \ j \in L_i\}$.

#### 1.4.1.1 Variabili

$x_{ijt}$ è una variabile (0-1) che vale 1 se e solo se il veicolo $i \in N$ inizia lo scarico sulla linea $j \in L_i$ al tempo $t \in I_{ij}$.

$s_{j\tau}$ è una variabile (0-1) che vale 1 se e solo se la linea $j$ non viene utilizzata nell'istante di tempo $\tau$.

La formulazione matematica $F$ del problema è la seguente.

$$z(F) = min \sum_{j \in L} \sum_{i \in N_j} \sum_{t \in I_{ij}} c_{ijt} + x_{ijt} + \sum_{j \in L} \sum_{\tau \in T_j} g_{j\tau} s_{j\tau} \tag{1.32}$$

$$s.t. \sum_{j \in L_i} \sum_{t \in I_{ij}} x_{ijt} = 1, \quad i \in N \tag{1.33}$$

$$\sum_{i \in N_j} \sum_{t \in I_{ij}} \delta_{ijt\tau} x_{ijt} + s_{j\tau} = 1, \quad j \in L,\ \tau \in T_j \tag{1.34}$$

$$x_{ijt} \in 0, 1, \qquad i \in N,\ j \in L_i,\ t \in I_{ij} \tag{1.35}$$

$$s_{j\tau} \in 0, 1, \qquad j \in L,\ \tau \in T_j \tag{1.36}$$

Il vincolo 1.33 impone che ad ogni veicolo venga assegnato una linea compatibile ed un tempo di scarico a sua volta compatibile sia con il veicolo stesso che con la linea a lui assegnata.

Il vincolo 1.34 impone che per ogni linea ed ogni istante di tempo compatibile con la linea vi sia in scarico al più un solo veicolo.
La formulazione $F$ richiede $\hat{n} = |N| \times |L| \times \hat{I}$ variabili, dove $\hat{I} = max|I_{ij}| : i \in N,\ j \in L_i$ e al più $\hat{m} = |N| + |L| \times \hat{T}$ vincoli, dove $\hat{T} = max|T_j| :\ j \in L$.

Supponiamo di discretizzare il tempo a 5 minuti, che ogni linea sia disponibile al più 10 ore (i.e. $\hat{T} = 120$) e che un veicolo quando arriva non possa aspettare più di 5 ore (i.e. $\hat{I} = 60$). Avremo $\hat{n} = 200 \cdot 20 \cdot 60 = 240.000$ e $\hat{m} = 200 + 20 \cdot 120 = 2600$.

## 1.5 Lot Sizing Problem

Il termine ***Lot Sizing*** indica il processo decisionale mediante il quale un'azienda definisce la politica ottima di investimenti, produzione e stoccaggio dei prodotti per soddisfare le richieste dei clienti nel rispetto dei vincoli di produzione e di magazzino.

Non esiste un unico modello di lot sizing che rappresenti in modo generale le varie realtà operative. Sistemi di produzione anche marginalmente diversi possono richiedere modelli aventi complessità computazionale molto diverse.

Non esiste in letteratura un modello generale che contenga come sottocasi tutti i problemi reali noti di lot sizing.

Per questi motivi non esistono software commerciali general pourpose.
Diverse aziende di consulenze nel settore della supply chain vendono software basati su modelli semplificati che non necessariamente producono soluzioni operative ma lasciano all'utente il compito di modificare manualmente la soluzione prodotta per tener conto delle specifiche complessità del problema reale.
I problemi reali sono varianti complesse delle seguenti tre classi di lot sizing problem di un singolo prodotto che sono risolvibili in tempo polinomiale:

- lot sizing senza vincoli di capacità produttiva;
- lot sizing con back logging senza vincoli di capacità;
- lot sizing con vincoli di capacità.

Molti problemi reali possono essere risolti rilassando in modo lagrangiano i vincoli reali per cui il problema lagrangiano risultante corrisponde ad uno dei tre problemi suddetti.

### 1.5.1 Lot sizing senza vincoli di capacità

Si consideri un'azienda che deve pianificare la propria produzione per un orizzonte temporale di $T$ periodi (ad esempio, $T$ mesi).

Per ciascun periodo $t = 1, ..., T$ sono noti:

$d_t$ domanda complessiva dei clienti;

$A_t$ costo fisso di set up per attivare la produzione;

$p_t$ costo per produrre un'unità di prodotto;

$h_t$ costo per unità di prodotto presente nel magazzino alla fine del periodo $t$.

Per ciascun periodo $t$, deve essere deciso il numero di unità che devono essere prodotto al fine di soddisfare la domanda in ciascun periodo.
Si suppone che la quantità prodotto nel periodo $t$ sia subito disponibile e che la quantità non

venduta alla fine di ogni mese viene depositata in magazzino. L'obiettivo è di minimizzare i costi complessivi di set up, produzione e stoccaggio.

#### 1.5.1.1 Formulazione Matematica (modello di Wagner-Whitin)

Variabili decisonali associate a ciascun periodo t=1,...,T

$x_t$ quantità prodotta all'inizio del periodo $t$;

$I_t$ livello del magazzino alla fine del periodo $t$;

$y_i \in (0,1) : y_t = 1$ se nel periodo $t$ vi è produzione, $y_t = 0$ altrimenti.

$$Min\ z = \sum_{t=1}^{T}(p_t x_t + h_t I_t + A_t y_t) \tag{1.37}$$

$$x_t + I_{t-1} = I_t + d_t,\ t = 1, ..., T \tag{1.38}$$

$$x_t \leq M y_t,\ t = 1, ..., \tag{1.39}$$

$$x_t,\ I_t \geq 0,\ t = 1, ..., T \tag{1.40}$$

$$y_t \in \{0, 1\},\ t = 1, ..., T \tag{1.41}$$

$$dove\ M = \sum_{t=1}^{T} d_t \text{ e, per semplicità, si suppone che } I_0 = 0. \tag{1.42}$$

#### 1.5.1.2 Metodo di soluzione

Al modello si associa il grafo $R = (N, A)$ senza vincoli di capacità sugli archi tale che ogni soluzione del problema corrisponde ad un flusso in $R$.
Il grafo $R$ si compone di $2T + 1$ nodi:

- nodo sorgente S da cui parte un flusso pari a $\sum_{t=1}^{T} d_t$;
- per ciascun periodo $t$ una coppia di nodi $U_t$, $V_t$ dove:

  $U_t$ rappresenta il magazzino,

  $V_t$ corrisponde alla domanda.

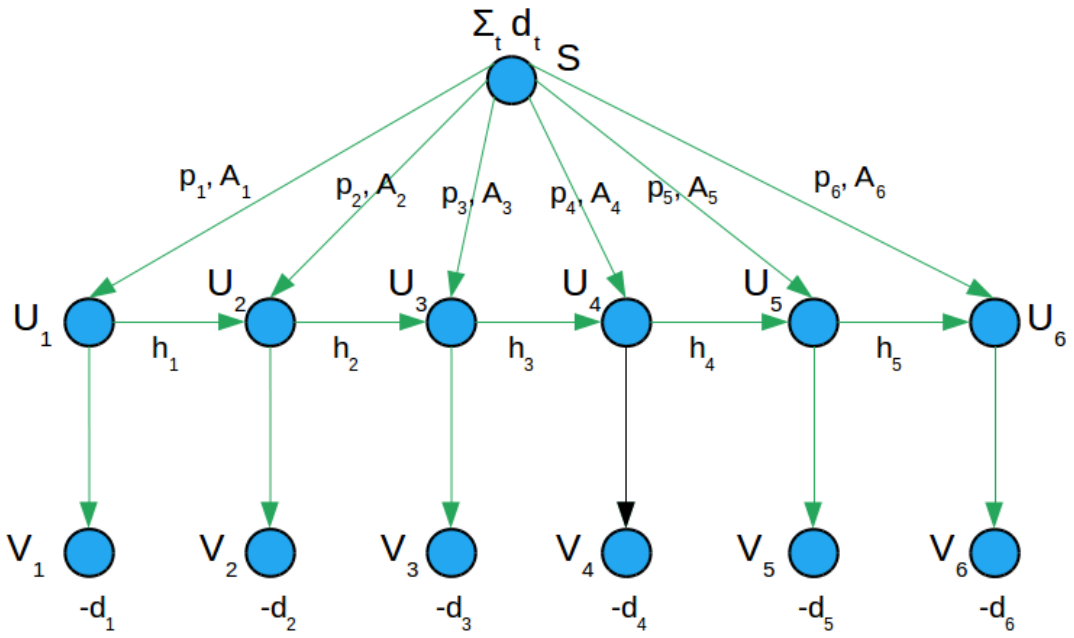
Per ciascun periodo $t = 1, .., T$ vi sono gli archi:

$(S, U_t)$ il cui flusso corrisponde alla produzione $x_t$;

$(U_t, U_{t+1})$ il cui flusso è pari al livello $I_t$ del magazzino alla fine del periodo $t$;

$(U_t, V_t)$ il cui flusso deve essere pari alla domanda $d_t$.

Figura 1.3: Esempio della rete di flusso (modello di Wagner-Whitin)

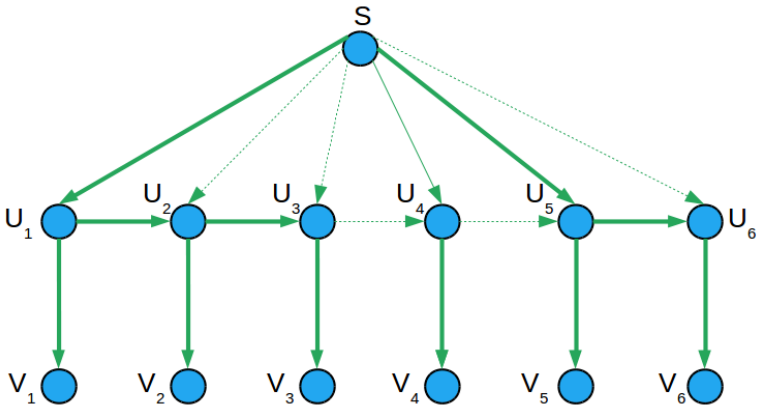



#### 1.5.1.3 Proprietà della soluzione ottima

**Teorema.** In una soluzione ottima non può mai avvenire che la domanda del periodo $t$ venga soddisfatta sia dalla produzione che dal magazzino, ovvero:

$$I_{t-1} \cdot x_t = 0;\ t = 1, ..., T$$

Figura 1.4



#### 1.5.1.4 Algoritmo di soluzione (di complessità $O(T^2)$)

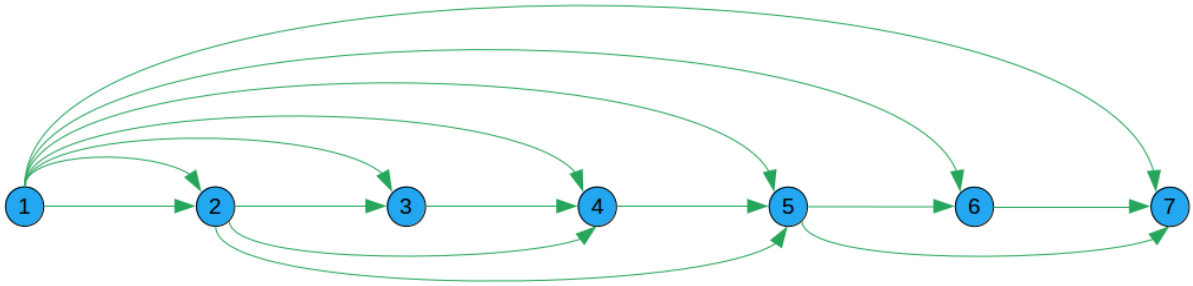
Si costruisca un grafo aciclico di $T+1$ vertici.
Si definiscano gli archi $j,k)$ per $j=0,...,T-1$ e $k=j+1,...,T$.

L'arco $(j,k)$ rappresenta la decisione di produrre all'inizio del periodo $j+1$ quanto serve per soddisfare le domanda complessiva dei periodo $j+1,\ j+2,...,k$.

Il costo $M_{jk}$ dell'arco (j,k) è pari al costo per produrre nel periodo j+1 la quantità $\sum_{r=j+1}^{k} d_r$ più i costi di stoccaggio:

$$M_{jk} = A_{j+1} + p_{j+1} \sum_{r=j+1}^{k} d_r + \sum_{t=j+1}^{k-1} h_t( \sum_{r=t+1}^{k} d_r)$$

Figura 1.5




Ogni soluzione del modello di Wagner-Whitin corrisponde ad un cammino da 0 a $t$ in questo grafo aciclico.
Il cammino di costo minimo fornisce la soluzione ottima.

# CAPITOLO 2

# INTRODUZIONE ALLA PROGRAMMAZIONE LINEARE A NUMERI INTERI

Si consideri il seguente problema.

$$Min\ cx \tag{2.1}$$
$$Ax = b \tag{2.2}$$
$$x \geq 0 \tag{2.3}$$
$$x\ intero \tag{2.4}$$

Le variabili devono assumere valori interi:

$$Es: \quad x_i = Numero\ di\ uomini\ che\ devono\ essere\ assegnati\ al\ lavoro\ i. \tag{2.5}$$
$$= Numero\ di\ automezzi\ che\ devono\ operare\ il\ trasporto\ lungo\ la\ "tratta\ i". \tag{2.6}$$
$$\tag{2.7}$$

## 2.1 Arrotondamento ad una soluzione non-intera

Si risolva il problema ignorando i vincoli $[x : intero]$. Le variabili che risultano non intere, nella soluzione ottima del problema continuo, vengano arrotondate al valore intero più vicino.

$$Es: \quad Min\ z = -2x_1 + 3x_2 \tag{2.8}$$
$$x_1 + x_2 \geq 3 \tag{2.9}$$
$$3x_1 + x_2 \leq 6 \tag{2.10}$$
$$x_2 \leq 5 \tag{2.11}$$
$$x_1,\ x_2 \geq 0\ ed\ intere \tag{2.12}$$

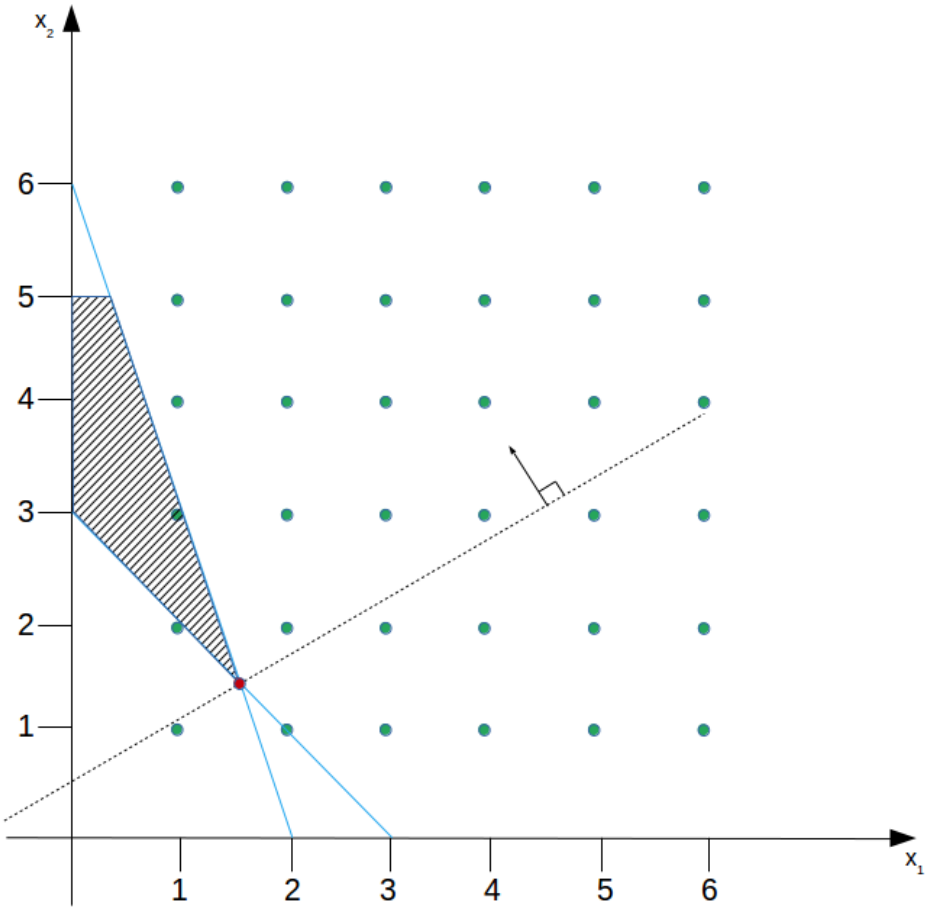



Figura 2.1:
Soluzione continua: $z = \frac{3}{4}$; $x_1 = \frac{3}{2}$, $x_2 = \frac{3}{2}$
Soluzione intera: $z = 4$; $x_1 = 1$, $x_2 = 2$

In questo esempio la soluzione arrotondata coincide con la soluzione ottima.

$$Es: \quad Min\ z = 8x_1 + 6x_2 \tag{2.13}$$
$$4x_1 + 3x_2 \geq 6 \tag{2.14}$$
$$x_1,\ x_2 \geq 0\ ed\ intere \tag{2.15}$$

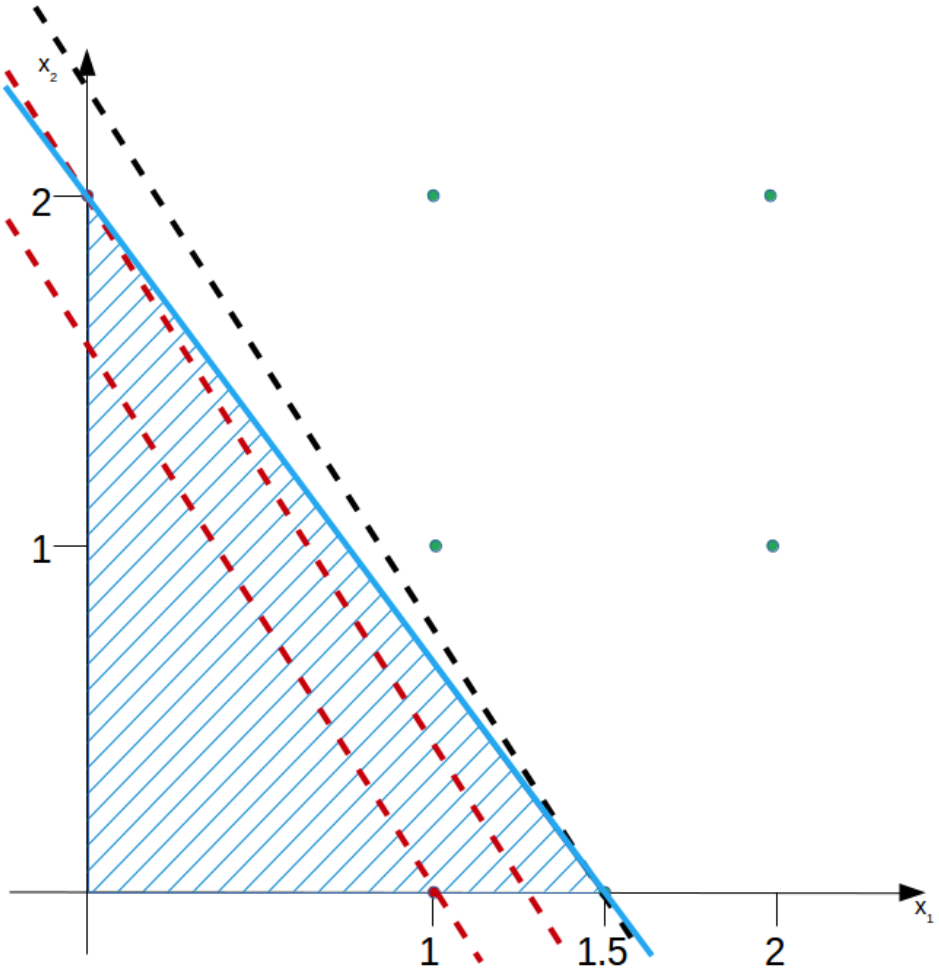
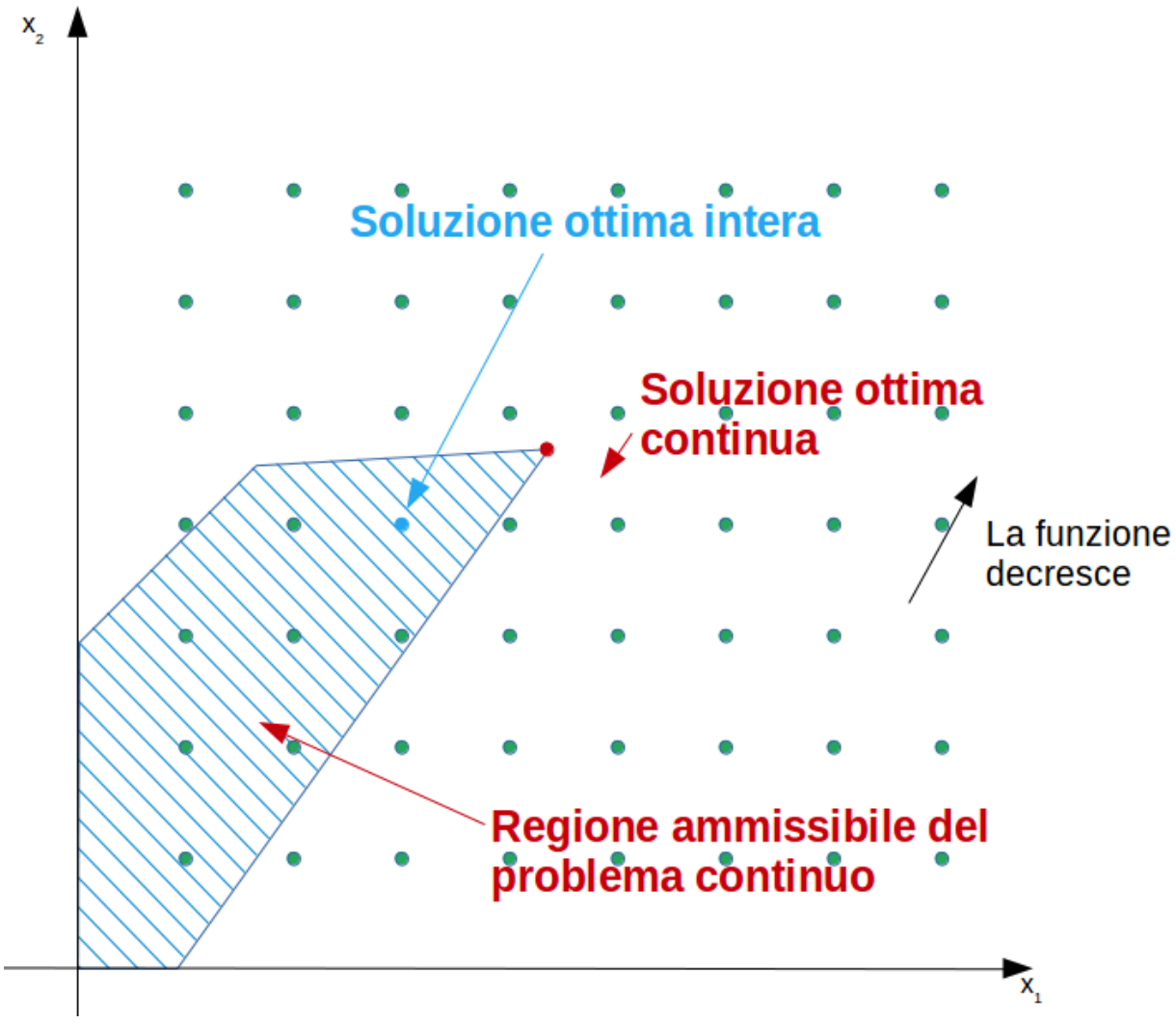



Figura 2.2:
Soluzione continua: $z = 12$; $x_1 = 1,5$, $x_2 = 0$
Soluzione arrotondata $z = 8$; $x_1 = 1$, $x_2 = 0$
Soluzione intera: $z = 10$; $x_1 = 0$, $x_2 = 2$

La soluzione arrotondata si discosta notevolmente dalla soluzione ottima.

$$Es: \quad Min\ z = 8x_1 + 6x_2 \tag{2.16}$$
$$4x_1 + 3x_2 \geq 6 \tag{2.17}$$
$$x_1,\ x_2 \geq 0\ ed\ intere \tag{2.18}$$




I quattro punti interi più vicini alla soluzione continua non sono ammissibili.

## 2.2 Unimodularità

La matrice intera A di $m$ righe ed $n$ colonne è totalmente unimodulare se ogni sua sottomatrice quadrata B non singolare è unimodulare, ovvero $det(B) = \pm 1$.

**Teorema.** Se la matrice intera $A$ è totalmente unimodulare allora tutti i punti estremi dell'insieme pd. convesso $X = x : \ Ax = b, \ x \geq 0$ sono interi per ogni vettore intero $b$.

**Dimostrazione.** Sia B una base ammissibile e $x_b$ le variabili base: $Bx_B = b$.
Per la regola di Cramer:

$$x_{b_i} = \frac{\det(B_i)}{\det(B)} \tag{2.19}$$

Dove $B_i$ si ottiene da $B$ sostituendo la i-esima colonna di $B$ con $b$. È ovvio che $\det(B_i)$ è un numero intero e quindi anche ciascun $x_{B_i}$ è intero.

**Teorema.** Una matrice intera $A$ i cui elemento sono $0, +1, -1$ è totalmente unimodulare se:

1. In ogni colonna $A$ compaiono al più due elementi non-nulli (cioè $1, -1$);
2. L'insieme delle righe R può essere suddiviso in due insieme disgiunti $R_1$ e $R_2$ ($R_1 \cup R_2 = R$) per cui:
   (a) Se una colonna contiene due elementi non-nulli dello stesso segno allora la riga corrispondente ad uno dei due elementi appartiene a $R_1$ mentre la riga relativa all'altro elemento è in $R_2$;
   (b) Se una colonna contiene due elementi di segno opposto entrambe le righe appartengono allo stesso insieme.

**Esempi.**

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 \end{bmatrix}, \quad A = \begin{bmatrix} -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 \end{bmatrix}$$

$$R = 1, 2, 3, 4 \qquad R = 1, 2, 3, 4, 5$$
$$R_1 = 1, 2, 3, 4 \qquad R_1 = 1, 2, 3$$
$$R_2 = \emptyset \qquad R_2 = 4, 5$$

La totale unimodularità della matrice $A$ è **condizione sufficiente** affinchè la soluzione ottima $x^*$ sia intera per

$$\begin{aligned} &Min\ cx \\ &\quad Ax = b\ (bintero) \\ &\quad x \geq 0 \end{aligned}$$

La condizione non è **necessaria**.

**Esempio:**

dato il sistema di vincoli

$$\begin{aligned} 6x_1 + x_2 &= 7 \\ 2x_1 + x_2 &= 3 \end{aligned}$$

L'unica soluzione è $(x_1 = 1,\ x_2 = 1)$ mentre la matrice

$$A = \begin{bmatrix} 6 & 1 \\ 2 & 1 \end{bmatrix}$$

non risulta essere totalmente unimodulare.

## 2.3 Metodo dei piani di taglio

Sia dato il problema

$$ILP \begin{cases} Min\ z_{ILP} = cx \\ Ax = b \\ x \geq 0\ e\ intero \end{cases}$$

Supponiamo A,c,b interi.

Si consideri il problema rilassato che si ottiene da 'ILP' ignorando i vincoli di interezza. Indichiamo tale problema con ***LP***.

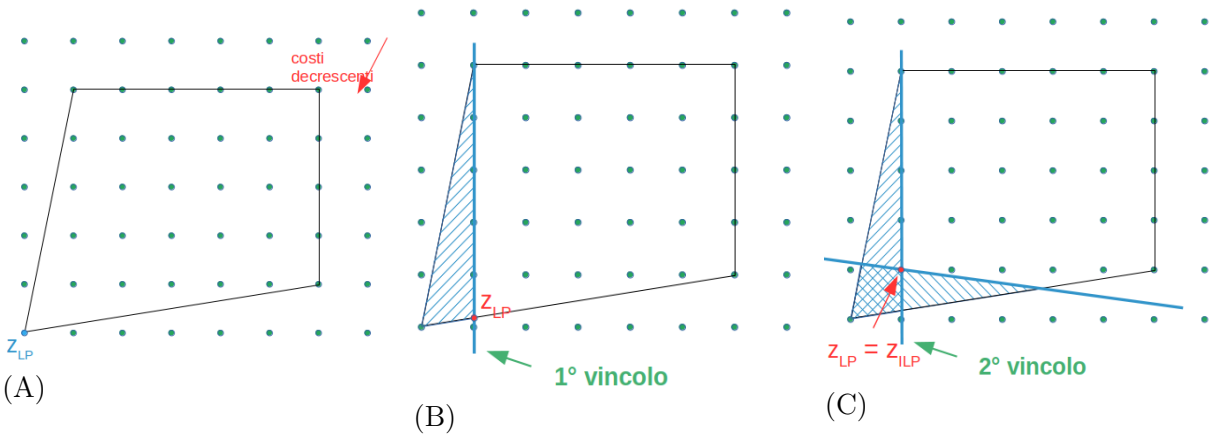
$$LP \begin{cases} z_{LP} = Min\ cx \\ ax = b \\ x \geq 0 \end{cases}$$

È noto che $z_{LP} \leq z_{ILP}$.

### 2.3.1 Piani di taglio

Si risolva $LP$; se la soluzione è intera tale soluzione è anche l'ottimo di $ILP$.

Altrimenti vengono aggiunti a $LP$ vincoli, *che non escludono soluzioni intere*, fino a che la soluzione del problema $LP$ risultante non risulti intera.




In $(A)$ viene mostrata la regione ammissibile di $LP$ ed il punto di ottimo.

In $(B)$ viene mostrata la regione ammissibile di $LP$ più un vincolo che rende non-ammissibile l'ottimo ottenuto in $(A)$ ma che non esclude nessuno dei punti interi.

In $(C)$ viene mostrato come l'aggiunta di un secondo vincolo rende la soluzione intera.

**Nell'esempio sono sufficienti 2 vincoli per rendere la soluzione intera.**

In generale bisogna aggiungere vincoli fino a che la soluzione non risulti intera o si scopra che il problema non ha soluzioni intere.

### 2.3.2 Gomory cuts

Si consideri il tableau ottimo relativo a LP:

| | z | $x_1$ | ... | $x_m$ | $x_{m+1}$ | ... | $x_j$ | ... | $x_n$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | ... | 0 | | | | | | |
| $x_1$ | | 1 | | | | | | | | $\bar{b}_1$ |
| | | | 1 | | | | | | | |
| | | | ... | | | | | | | |
| $x_r$ | | | | 1 | $y_r^{m+1}$ | ... | $y_r^j$ | ... | $y_r^n$ | $\bar{b}_r$ |
| | | | | ... | | | | | | |
| $x_m$ | | | | | 1 | | | | | $\bar{b}_m$ |

Supponiamo la soluzione ottima non intera.

*Sia $\bar{b}_r$ non intero.*

L'equazione associata a $x_r$ è:

$$x_r + \sum_{j=m+1}^{n} y_r^j x_j = \bar{b}_r \tag{2.20}$$

Poniamo:

$$y_r^j = I_r^j + F_r^j, \text{ dove } I_r^j = \lfloor y_r^j \rfloor, \ (0 \leq F_r^j < 1) \tag{2.21}$$

Inoltre:

$$\bar{b}_r = I_r + F_r \text{ essendo } 0 \leq F_r < 1 \tag{2.22}$$

Sostituendo, la 2.20 diviene:

$$x_r + \sum_{j=m+1}^{n} (I_r^j + F_r^j) x_j = (I_r + F_r) \tag{2.23}$$

o anche:

$$\underbrace{x_r + \sum_{j=m+1}^{n} I_r^j x_j - I_r}_{\text{intero per ogni x intero}} = \underbrace{F_r - \sum_{j=m+1}^{n} F_r^j x_j}_{< 1 \text{ per } x \geq 0} \tag{2.24}$$

Ne segue:

$$F_r - \sum_{j=m+1}^{n} F_r^j x_j \leq 0 \tag{2.25}$$

La soluzione corrente non soddisfa il vincolo 2.20 in quanto $x_j = 0\ j = m+1, ..., n$ mentre $F_r > 0$ poiché $\bar{b}_r$ si è supposto non intero.

Se il vincolo 2.20 viene aggiunto al problema $LP$ allora la soluzione corrente risulterà non ammissibile.
Per determinare una nuova soluzione che soddisfi il vincolo 2.20 può essere impiegato il *Simplesso Duale* partendo dal tableau ottimo relativo alla soluzione corrente.

Al tableau va aggiunto il vincolo:

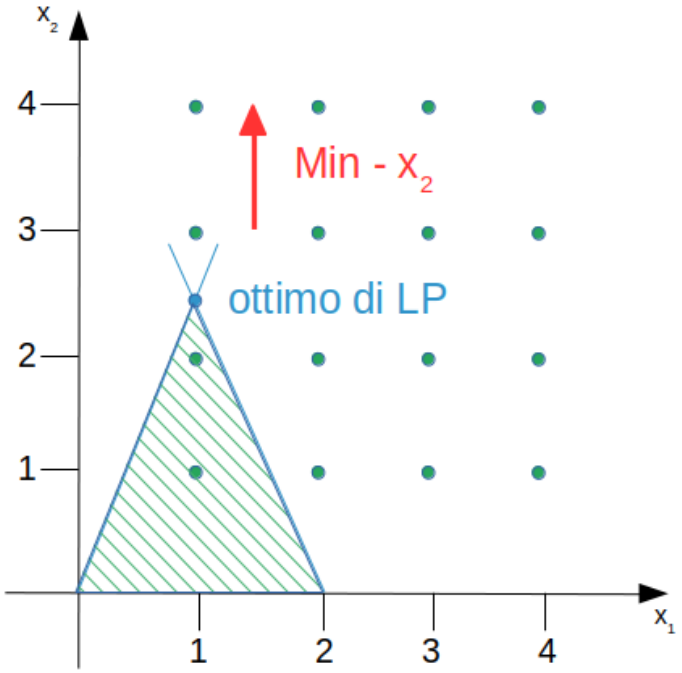
$$-\sum_{j=m+1}^{j} F_r^j x_j + s = -F_r \tag{2.26}$$

La nuova variabile slack $s$ è una nuova variabile base.
Il nuovo tableau è non ammissibile per il Primale ma duale ammissibile.

**Esempio.**

$$\begin{aligned} Min\ & -x_2 \\ & 3x_1 + 2x_2 \leq 6 \\ & -3x_1 + 2x_2 \leq 0 \\ & x_1,\ x_2 \geq 0 \\ & x_1,\ x_2\ intere\ \geq 0 \end{aligned}$$

Si noti che l'ottimo cade nel punto $x = (1; 1)$.




| | z | $x_1$ | $x_2$ | $x_3$ | $x_4$ | RHS |
|---|---|---|---|---|---|---|
| z | 1 | 0 | 1 | 0 | 0 | 0 |
| $x_3$ | 0 | 3 | 2 | 1 | 0 | 6 |
| $x_4$ | 0 | -3 | ② | 0 | 1 | 0 |

| | z | $x_1$ | $x_2$ | $x_3$ | $x_4$ | RHS |
|---|---|---|---|---|---|---|
| z | 1 | 3/2 | 0 | 0 | -1/2 | 0 |
| $x_3$ | 0 | ⑥ | 0 | 1 | -1 | 6 |
| $x_2$ | 0 | -3/2 | 1 | 0 | 1/2 | 0 |

| | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | RHS |
|---|---|---|---|---|---|---|
| z | 1 | 0 | 0 | -1/4 | -1/4 | -3/2 |
| $x_1$ | 0 | 1 | 0 | 1/6 | -1/6 | 1 |
| $x_2$ | 0 | 0 | 1 | 1/4 | 1/4 | 3/2 |

Tabella 2.1: Tableau ottimo. Soluzione continua!

Ricordando che il cut da aggiungere è:

$$-\sum_{j=m+1}^{n} F_r^j x_j + s = -F_r \tag{2.27}$$

Dalla riga di $x_2$ si ha la sequente equzione:

$$x_2 + \frac{1}{4}x_3 + \frac{1}{4}x_4 = \frac{3}{2} \tag{2.28}$$

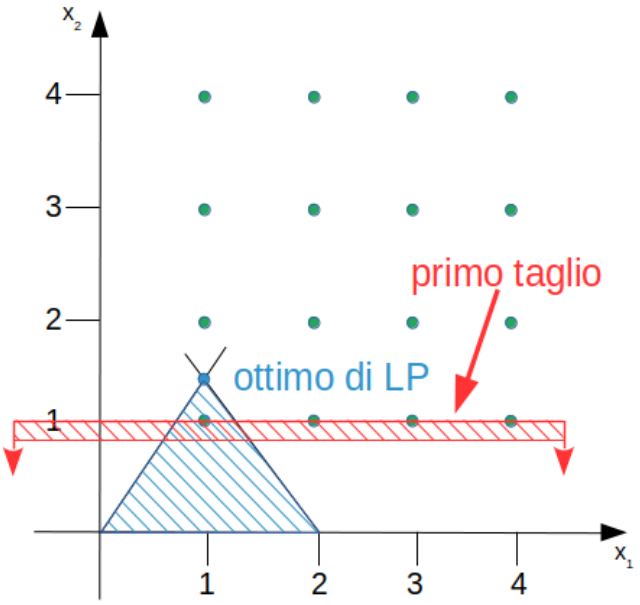
da cui:

$$-\frac{1}{4}x_3 - \frac{1}{4}x_4 + s_1 = -\frac{1}{2} \tag{2.29}$$

Si osservi che per definizione di $x_3$ e $x_4$ si ha

$$x_3 = 6 - 3x_1 - 2x_2 \text{ ed } x_4 = 3x_1 - 2x_2 \tag{2.30}$$

Sostituendo in 2.29 si ottiene $x_2 \leq 1$:




Aggiungendo il cut al tableau ottimo precedente:

| | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | RHS |
|---|---|---|---|---|---|---|---|
| z | 1 | 0 | 0 | -1/4 | -1/4 | 0 | -3/2 |
| $x_1$ | 0 | 1 | 0 | 1/6 | -1/6 | 0 | 1 |
| $x_2$ | 0 | 0 | 1 | 1/4 | 1/4 | 0 | 3/2 |
| $s_1$ | 0 | 0 | 0 | **-1/4** | -1/4 | 1 | 1 |

Continuando con il simplesso duale:

| | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | RHS |
|---|---|---|---|---|---|---|---|
| z | 1 | 0 | 0 | 0 | 0 | -1 | -1 |
| $x_1$ | 0 | 1 | 0 | 0 | -1/3 | 2/3 | 2/3 |
| $x_2$ | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| $s_1$ | 0 | 0 | 0 | 1 | 1 | -4 | 2 |

Dalla riga di $x_1$ si ha i cut:

$$-\frac{2}{3}x_4 - \frac{2}{3}s_1 + s_2 = -\frac{2}{3} \tag{2.31}$$

Il nuovo tableau, quindi, diviene:

e ottimizzando con li simplesso duale: L'algoritmo converge in un numero finito di passi purché venga impiegata una appropiata regola lessicografica per la scelta del pivot.

| | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | RHS |
|---|---|---|---|---|---|---|---|---|
| z | 1 | 0 | 0 | 0 | 0 | -1 | 0 | -1 |
| $x_1$ | 0 | 1 | 0 | 0 | -1/3 | 2/3 | 0 | 2/3 |
| $x_2$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $s_1$ | 0 | 0 | 0 | 1 | 1 | -4 | 0 | 2 |
| $s_2$ | 0 | 0 | 0 | 0 | -2/3 | -2/3 | 1 | 2/3 |

| | | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $s_1$ | $s_2$ | RHS |
|---|---|---|---|---|---|---|---|---|
| z | 1 | 0 | 0 | 0 | 0 | -1 | -1/2 | -1 |
| $x_1$ | 0 | 1 | 0 | 0 | 0 | -1 | 0 | 1 |
| $x_2$ | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| $s_1$ | 0 | 0 | 0 | 1 | 0 | -5 | 3/2 | 2 |
| $s_2$ | 0 | 0 | 0 | 0 | 1 | 1 | -3/2 | 1 |

Tabella 2.2: Tableau ottimo.

#### 2.3.2.1 Come evitare un numero indefinito di righe e colonne

Qualora una variabile di slack $s_i$, associata all'i-esimo cut, entra in base, si elimina sia il cut sia la variabile $s_i$.

In questo modo il numero delle righe aggiunte (relative ai cuts) non supera n-m.

## 2.4 Metodi Branch and Bound

Sia $P_0$ un problema a cui corrisponde l'insieme $S_0$ di soluzioni ammissibili.

Ad esempio

$$P_0 \begin{cases} Min \;\; cx \\ \qquad Ax = b \\ \qquad x \geq 0 \; e \; intero \end{cases}$$

$$S_0 = \{x : Ax = b, \; x \geq 0 \; intero\} \tag{2.32}$$

***Principio base dei metodi Branch and Bound***
Suddividre il problema $P_0$ nei sottoproblemi $P_1, P_2, ..., P_k$ a cui corrispondono gli insiemi di soluzioni ammissibili $S_1, S_2, ..., S_k$. La suddivisione è tale per cui

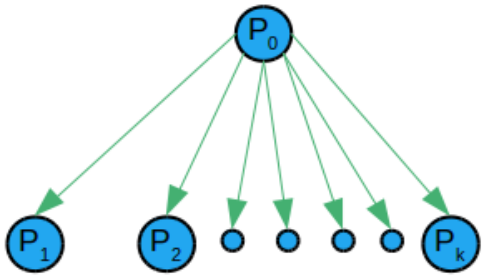
$$S_1 \cup S_2 \cup ... \cup S_k = S_0 \tag{2.33}$$

È evidente che:

$$\min_{x \in S_0} cx = MIN\{\min_{x \in S_1} cx, \min_{x \in S_1} cx, ..., \min_{x \in S_k} cx\} \tag{2.34}$$

La risoluzione di ogni sottoproblema $P_1, P_2, ..., P_k$ può risultare molto più semplice della risoluzione di $P_0$.

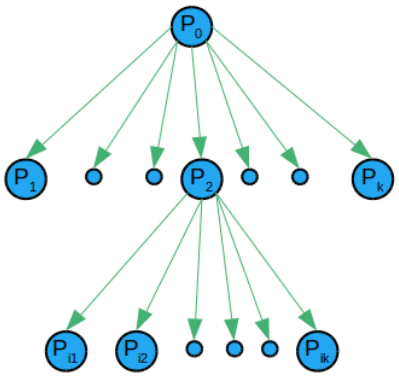
Possiamo rappresentare la suddivisione di $P_0$ in $P_1, P_2, ..., P_k$ mediante un albero




Nel caso in cui la riduzione di uno o più sottoproblemi risulti *difficile* questi possono essere ulteriormente suddivisi.

Supponiamo che $P_i$ risulti difficile, allora può essere suddiviso nei sottoproblemi $P_{i1}, P_{i2}, ..., P_{ik}$ a cui corrispondono gli insiemi di soluzioni ammissibili $S_{i1}, S_{i2}, ..., S_{ik}$ tali che $S_{i1} \cup S_{i2} \cup ... \cup S_{ik} = S_i$

Si ha il seguente albero




Risolvere $P_0$ equivale a risolvere $P_1, ..., P_{i-1}, (P_{i1}, P_{i2}, ..., P_{ir}), P_{i+1}..., P_k$

Il processo di suddivisione di un problema in un numero finito di sottoproblemi viene chiamato **BRANCHING**.

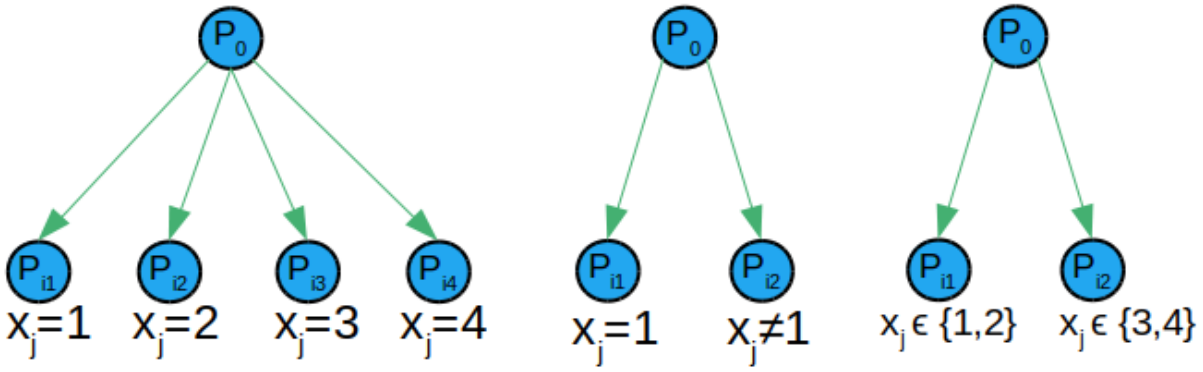
Una buona strategia di branching consiste nel suddividere $P_i$ nei sottoproblemi $P_{i1}, P_{i2}, ..., P_{ir}$ in modo che, per ogni coppia $P_{i\alpha}, P_{i\beta}$ con $(\alpha \neq \beta)$, gli insiemi $S_{i\alpha}$ e $S_{i\beta}$ siano disgiunti.

$$S_{i\alpha} \cap S_{i\beta} = \emptyset$$

Se la condizione sopra è soddisfatta allora $\{S_i, ..., S_{ir}\}$ è una **PARTIZIONE** di $S_i$.
Si noti che la condizione non è *necessaria* ma rende computazionalmente efficiente il processo di branching.

#### 2.4.0.1 Esempi di Branching

Si consideri il problema $P_i$ in $n$ variabili dove la variabile $x_j$ può assumere i valori $1, 2, 3, 4$.

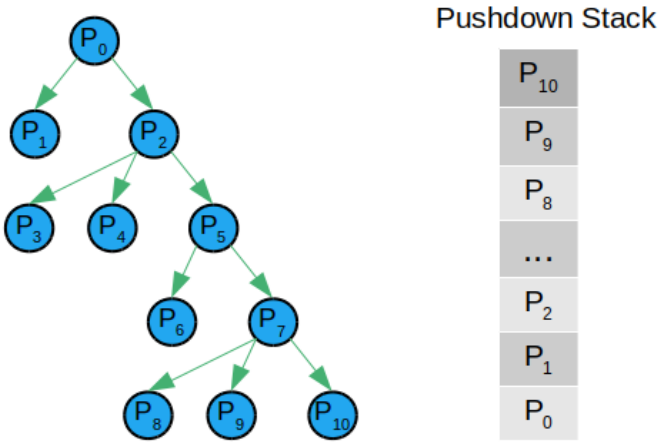



### 2.4.1 Tipi di Branching

Ogni sottoproblema che non può essere risolto può essere suddiviso in sottoprblemi più piccoli. Dato un insieme di sottoproblemi da suddividere quale sottoproblema suddividere per primo?

#### 2.4.1.1 Dept-first search

In questo tipo di branching il sottoproblema che viene suddiviso per primo è l'ultimo generato. Ciò si ripete fino ad ottenere un sottoproblema che può essere risolto.




**BACKTRACKING**: quando un sottoproblema è risolto viene scelto il penultmo sottoproblema generato e su questo viene effettuato il branching.

#### 2.4.1.2 Breadth-first search

Il branching procede da livello a livello, ovvero il problema $P_0$ è suddiviso in $P_1, P_2, ..., P_k$ che sono i sottoproblemi a livello 1.

Ogni sottoproblema a livello 1 viene suddiviso in un numero di sottoproblemi che costituiscono il livello 2.

In generale quando viene esaminato un sottoproblema a livello $K$ sono stati già esaminati tutti i sottoproblemi a livello $K-1$.

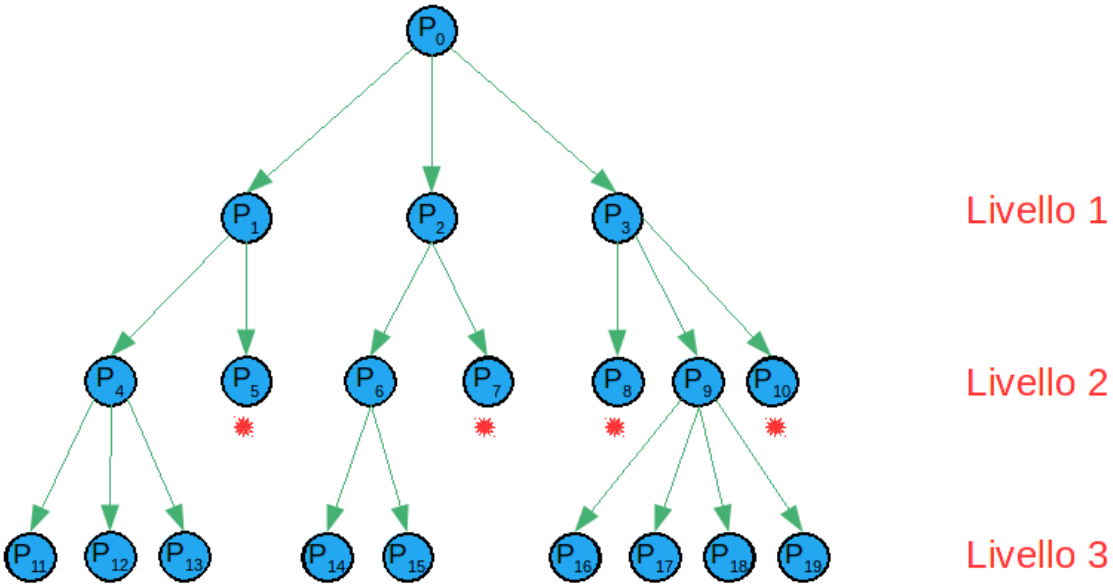



Figura 2.3: * Problemi risolti

### 2.4.2 Bounds

La ricerca della solutione ottima di $P_0$ è completa quando sono stati risolti tutti i sottoproblemi generati.
Questo processo può essere migliorato calcolando, per ogni sottoproblemi $P_j$ un *bound* (*Lower Bound* se il problema è di minimizzazione).

**Lower Bound**: diremo che $LB_i$ è un lower bound al sottoproblema $P_i$ se

$$LB_i \leq \underset{x \in S_i}{Min}\{cx\} \tag{2.35}$$

**Upper Bound**: diremo che $UB_i$ è un upper bound al sottoproblema $P_i$ se

$$UB_i \leq \underset{x \in S_0}{Min}\{cx\} \tag{2.36}$$

È possible trascurare il sottoproblema $P_i$ se $LB_i \geq UB$, infatti, poiché $LB_i \leq \underset{x \in S_i}{Min}\{cx\}$ si avrebbe $\underset{x \in S_i}{Min}\{cx\} \geq UB$ e quindi il sottproblema $P_i$ non contiene la soluzione ottima.

#### 2.4.2.1 Calcolo del Lower Bound

Sia dato il problema

$$P_0 \begin{cases} Min \;\; z = cx \\ \qquad\;\; Ax = b \\ \qquad\;\; x \geq 0 \; e \; intero \end{cases}$$

Sia $z^*$ il costo della soluzione ottima di $P_0$.

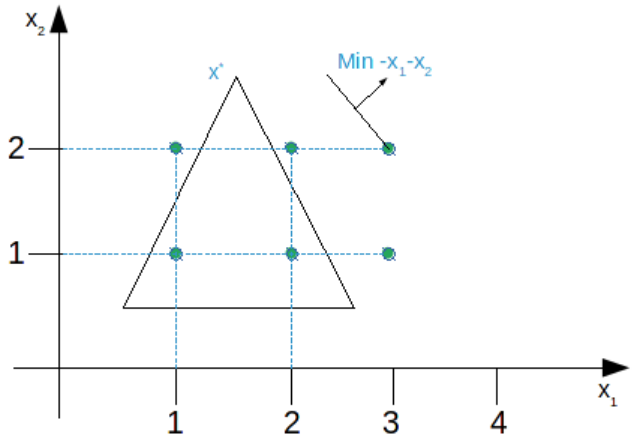
I seguenti medoti producono validi Lower Bounds a $P_0$.

**Rilassamento continuo** Si ignori il vincolo $x$ intero; il problema risultante è risolvibile con la programmazione lineare.
Sia $z^*_{LP}$ il costo di tale soluzione; si ha

$$z^*_{LP} \leq z^* \tag{2.37}$$

- Se la soluzione del problema continuo è intera allora è anche la soluzione ottima intera e quindi $z^*_{LP} = z^*$;
- Se la soluzione è frazionaria allora può essere usato un metodo Branch & Bound per trovare la soluzione ottima intera.

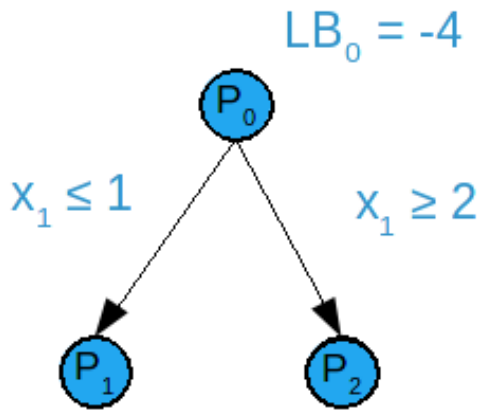
**Esempio.**




Rilassamento continuo $x^* = (\frac{3}{2}, \frac{5}{2})$, $z^*_{LP} = cx^* = -4$

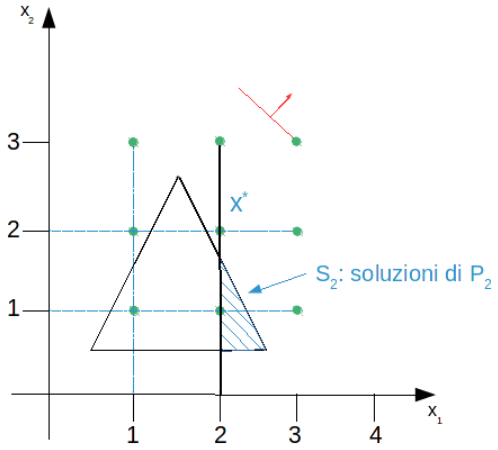
$P_0$ è suddiviso in due sottoproblemi $P_1$ *e* $P_2$

- $P_1$ imponiamo che $x_1 \leq 1$
- $P_2$ imponiamo che $x_1 \geq 2$




Si usi una strategia Depth-First e quindi si esamini il problema $P_2$.

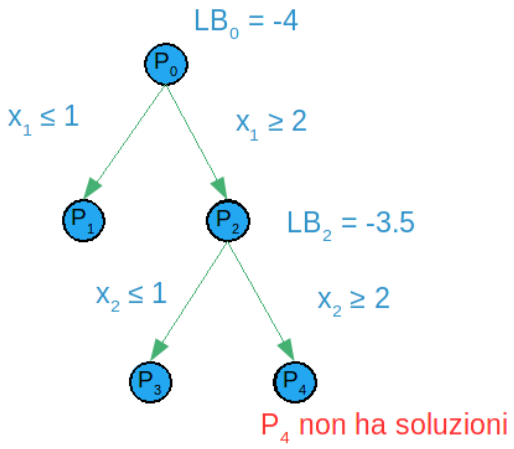
**Esame del sottoproblema $P_2$**
Il lower bound si ottiene imponendo il vincolo $x_1 \geq 2$.



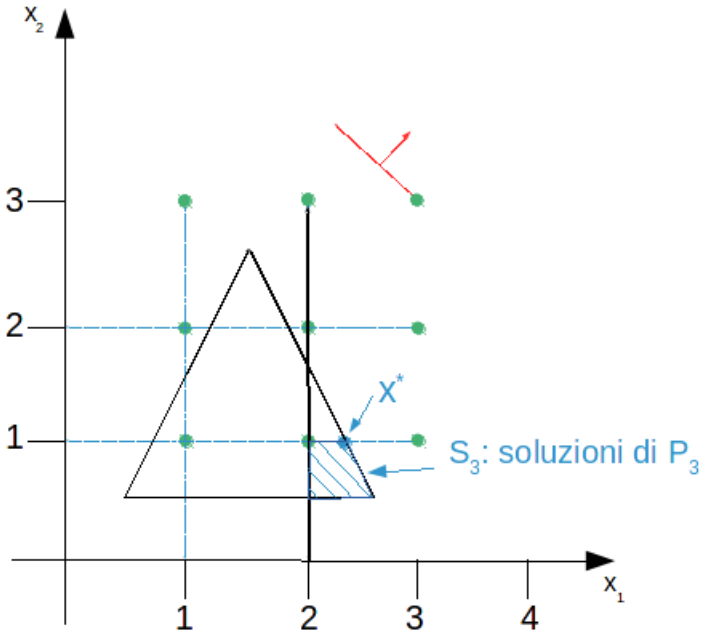

L'ottimo per $P_2$ è $z^*_{LP} = -3.5$ con componenti $x^*_1 = 2$ e $1 < x^*_2 < 2$

Il problema $P_2$ vien suddiviso in $P_3$ e $P_4$ dove

- $P_3$ imponiamo $x_1 \geq 2$ e $\boldsymbol{x_2 \leq 1}$
- $P_4$ imponiamo $x_1 \geq 2$ e $\boldsymbol{x_2 \geq 2}$



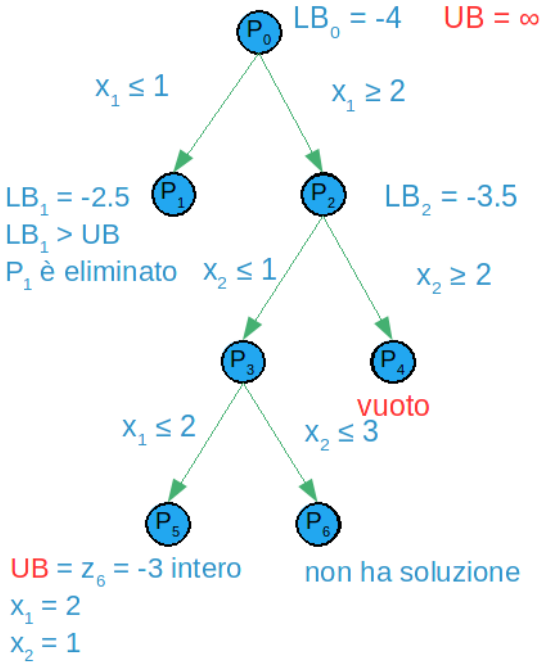

**Esame del sottoproblema $P_3$**



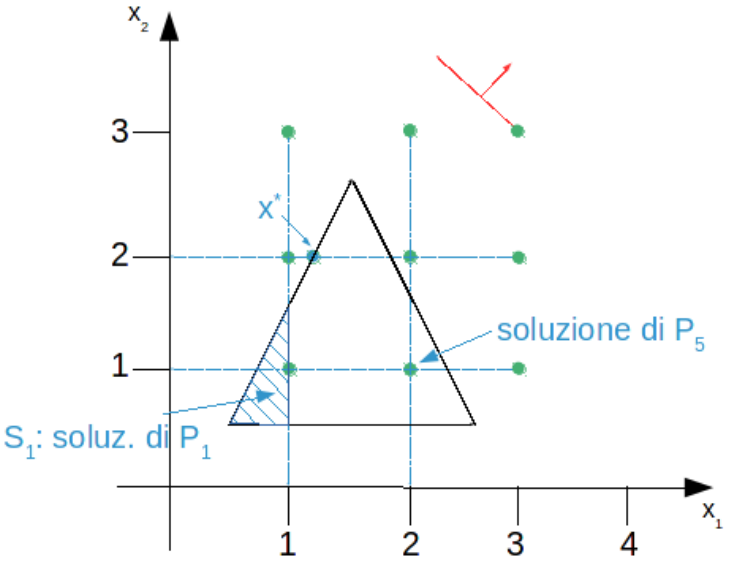

L'ottimo di $P_3$ è $z^*_{LP} = -3.25$ con componenti $x^*_2 = 1$ e $2 < x^*_1 < 3$.

Il problema $P_3$ viene suddiviso in $P_5$ e $P_6$ dove

- $P_5$ imponiamo $x_1 \geq 2$, $x_2 \leq 1$ e $\underline{\boldsymbol{x_1 \leq 2}}$
- $P_6$ imponiamo $x_1 \geq 2$, $x_2 \leq 2$ e $\underline{\boldsymbol{x_1 \geq 3}}$




Al nodo dell'albero, corrispondente al problema $P_5$ si è ottenuta la prima soluzione ammissibile di $P_0$ di costo $-3$. Quindi poniamo $UB = 3$.
Il backtracking conduce ad esaminare il problema $P_1$




La soluzione ottima di $P_1$ è $z^*_{LP} = -2.5$, quindi, $LB_1 = -2.5$ e poiché $LB_1 > UB$ il sottoproblema $P_1$ non può condurre ad alcuna soluzione migliore di quella trovata per $P_5$.

Essendo stati esaminati tutti i nodi dell'albero l'algoritmo termina e $z^* = -3$ è la soluzione ottima.

### 2.4.3 Eliminazione di alcuni vincoli

Si consideri il problema

$$P_0 \begin{cases} Min \;\; z = cx \\ \qquad Ax = b \;\; m_1 \; vincoli \\ \qquad Dx = h \;\; m_2 \; vincoli \\ \qquad x \geq 0 \;\; intero \end{cases}$$

Si consideri il problema RP che deriva da $P_0$ eliminando i vincoli $Ax = b$

$$RP \begin{cases} Min \;\; z_{RP} = cx \\ \qquad Dx = b \\ \qquad x \geq 0 \; e \; intero \end{cases}$$

Sia $z^*_{RP}$ il valore ottimo di RP; si ha

$$z^*_{RP} \leq z^* \quad (\; z^* \text{ ottimo di } P_0 \;) \tag{2.38}$$

L'estensione di questo metodo è il Rilassamento Lagrangiano mediante il quale è possibile tener conto dei vincoli rilassati nella funzione obiettivo.

### 2.4.4 Rilassamento Surrogato

Sia dato il problema

$$P_0 \begin{cases} Min \;\; z = cx \\ \qquad \displaystyle\sum_{j=1}^{n} a_{ij} x_j \geq b_i \;\; i = 1, ..., m_1 \\ \qquad Dx = h \;\; m_2 \; vincoli \\ \qquad x \geq 0 \;\; intero \end{cases}$$

Si consideri il probelma che si ottiene da $P_0$ sostituendo i primi $m_1$ vincoli $a^i x \geq b_i$ con una loro combinazione lineare

$$SP \begin{cases} Min \;\; z_{SP} = cx \\ \qquad \displaystyle\sum_{j=1}^{m_1} \pi_i \sum_{j=1}^{n} a^j_{ij} \geq \sum_{i=1}^{m_1} \pi_i b_i \;\; \pi_i \geq 0 \\ \qquad Dx = h \;\; m_2 \; vincoli \\ \qquad x \geq 0 \;\; intero \end{cases}$$

$z^*_{SP}$, ottimo di SP, è un valido lower bound a $P_0$.

$$P \begin{cases} z^* = \ Min\ cx \\ \qquad s.t.\ Ax \geq b \\ \qquad x \geq 0 \ \ intero \end{cases}$$

Rilassamento lineare

$$LP \begin{cases} z^*_{LP} = \ Min\ cx & = \ Max\ \omega b \\ \qquad s.t.\ Ax \geq b & \omega A \leq c \\ \qquad x \geq 0 \ \ intero & \omega \geq 0 \end{cases}$$

Sia $\omega^*$ la soluzione duale ottima.

$$SP \begin{cases} z^*_{SP} = \ Min\ cx \\ \qquad (\omega^* A)x \geq \omega^* b \\ \qquad x \geq 0 \ \ intero \end{cases}$$

Ottenuto tramite **rilassamento surrogato**.

**Teorema.** $z^*_{SP} \geq z^*_{LP}$ Poiché $w^*$ è la soluzione ottima del duale di LP si ha

$$z^*_{LP} = \omega^* b \quad \text{e} \quad c - \omega^* A x^* \tag{2.39}$$

Sia $x^*$ la soluzione ottima intera di SP; si ha:

$$(c - \omega^* A)x^* \geq 0 \quad \text{ovvero} \quad cx^* \geq \omega^* A x^* \tag{2.40}$$

ma anche (da SP):

$$\omega^* A x^* \geq \omega^* b \tag{2.41}$$

Da 2.40 e 2.41:

$$cx^* \geq \omega^* A x^* \geq \omega^* b = z^*_{LP} \tag{2.42}$$

ovvero

$$z^*_{SP} = cx^* \geq z^*_{LP} \tag{2.43}$$

■

## 2.5 Assegnamento Generalizzato

Allocazione ottimale di $n$ oggetti in $m$ contenitori in modo che ogni oggetto sia assegnato ad un solo contenitore e non sia superata la portata di ogni contenitore.

Indichiamo con:

$b_i$ : portata del contenitore $i, i = 1, ..., m$

$a_{ij}$ : spazio del contenitore $i$ occupato dall'oggetto $j$ (se $j$ viene assegnato a $i$)

$c_{ij}$ : costo per assegnare al contenitore $i$ l'oggetto $j$

$x_{ij}$

$= 1$ se l'oggetto viene assegnato al contenitore $i$

$= 0$ altrimenti

### 2.5.1 Formulazione matematica

$$Min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{2.44}$$

$$\sum_{i=1}^{m} x_{ij},\ j = 1, ..., n \tag{2.45}$$

$$\sum_{j=1}^{n} a_{ij}x_{ij},\ i = 1, ..., m \tag{2.46}$$

$$x_{ij} \in \{0, 1\},\ \forall i, j \tag{2.47}$$

Dove:

2.45 ogni oggetto $j$ deve essere assegnato ad un solo contenitore

2.46 il "peso" complessivo assegnato ad ogni contenitore $i$ non deve superare la portata $b_i$

### 2.5.2 Rilassamento lagrangiano

#### 2.5.2.1 (a) Rispetto ai vincoli 2.45

$$L(u) = Min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} - \overbrace{\sum_{j=1}^{n} u_j(\sum_{i=1}^{m} x_{ij} - 1)}^{-u(Ax-b)} = \tag{2.48}$$

$$L(u) = Min \sum_{i=1}^{m}(\sum_{j=1}^{n}(c_{ij} - u_j)x_{ij}) + \sum_{j=1}^{n} u_j \tag{2.49}$$

$$s.t. \sum_{j=1}^{n} a_{ij}x_{ij} \leq b_i,\ i = 1, ..., m \tag{2.50}$$

$$x_{ij} \in \{0, 1\} \tag{2.51}$$

L'ottimo $L(u)$ si ottiene risolvendo $m$ problemi di *Knapsack* per il contenitore $i$ del tipo:

$$z_i = Min \sum_{j=1}^{n}(c_{ij} - u_j)x_{ij}$$
$$\sum_{j=1}^{n} a_{ij}x_{ij} \leq b_i$$
$$x_{ij} \in \{0,1\}$$

Per cui $L(u) = \sum_{i=1}^{m} z_i \sum_{j=1}^{n} u_j$

**Esempio.**

$m = 2$ contenitori; $n = 4$ oggetti.

$$a_{ij} = \begin{bmatrix} 5 & 7 & 4 & 2 \\ 3 & 1 & 6 & 4 \end{bmatrix} \; c_{ij} = \begin{bmatrix} 3 & 3 & 4 & 9 \\ 2 & 6 & -9 & 3 \end{bmatrix}$$

$b = (30, 12)$

Poniamo $u^0 = 0$ e $z^* = 3$ (soluzione euristica iniziale)

$$L(u^0) = \min_x \sum_{i=1}^{m}(\sum_{j=1}^{n}(c_{ij} - u_j^0)x_{ij} + \sum_{j=1}^{n} u_j^0) \tag{2.52}$$

$$L(u^0) =; \min_x 3x_{11} + 3x_{12} + 4x_{13} + 9x_{14} + 2x_{21} + 6x_{22} - 9x_{23} + 3x_{24} + 0 \tag{2.53}$$

$$5x_{11} + 7x_{12} + 4x_{13} + 2x_{14} \leq 30 \tag{2.54}$$

$$3x_{21} + x_{22} + 6x_{23} + 4x_{24} \leq 12 \tag{2.55}$$

$$x_{11}, ..., x_{24} \in \{0,1\} \tag{2.56}$$

Si decompone in due problemi: $L(u^0) = z_1 + z_2 + 0$

$$1^0 problema \begin{cases} z_1 = \; Min\; 3x_{11} + 3x_{12} + 4x_{13} + 9x14 \\ \qquad 5x_{11} + 7x_{12} + 4x_{13} + 2x_{14} \leq 30 \end{cases}$$

Soluzione ottima: $z_1 = 0;\; x_{11}^0 = x_{12}^0 = x_{13}^0 = x_{14}^0 = 0$

$$2^0 problema \begin{cases} z_1 = \; Min\; 2x_{21} + 6x_{22} - 9x_{23} + 3x24 \\ \qquad 3x_{21} + x_{22} + 6x_{23} + 4x_{24} \leq 12 \end{cases}$$

Soluzione ottima: $z_2 = -9;\; x_{21}^0 = x_{22}^0 = 0,\; x_{23}^0 = 1,\; x_{24}^0 = 0$

Quindi $L(u^0) = z_1 + z_2 + 0 = 0 - 9 + 0 = -9$

La soluzione di $L(u*0)$ non soddisfa i vincoli 2.45; infatti:
$x = (x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24})$
$x^0 = (\ 0,\ \ 0,\ \ 0,\ \ \ 0,\ \ \ 0,\ \ 0,\ \ 1,\ \ 0)$

$$Vincoli\ 2.45 \begin{cases} x_{11} + x_{21} = 1 \ \ (j=1) \\ x_{12} + x_{22} = 1 \ \ (j=2) \\ x_{13} + x_{23} = 1 \ \ (j=3) \\ x_{14} + x_{24} = 1 \ \ (j=4) \end{cases}$$

Si noti che la soluzione di $L(u^0)$ viola i vincoli 2.45 per $j = 1, 2, 4$ mentre soddisfa quello per $j = 3$.

**Aggiornamento delle penalità {u}**

$$u^k = u^{k-1}\alpha_k \frac{(z^* - L(u^{k-1}))}{||Ax^{k-1} - b||^2} \cdot (Ax^{k-1} - b) \tag{2.57}$$

Poniamo $k = 1$ ed $\alpha_1 = 2$; si è già assunto $z^* = 3$

$$u_j^k = u_j^{k-1} - 2 \cdot \frac{(+3 - (-9))}{\sum_{j=1}^{n}(\sum_{i=1}^{m} x_{ij} - 1)^2} \cdot (\sum_{i=1}^{m} x_{ij} - 1) \tag{2.58}$$

$$u_1^1 = u_1^0 - 2 \cdot \frac{12}{3} \cdot (-1) = 0 + 8 = 8 \tag{2.59}$$

$$u_2^1 = u_2^0 - 2 \cdot 4 \cdot (-1) = 0 + 8 = 8 \tag{2.60}$$

$$u_3^1 = u_3^0 - 2 \cdot 4 \cdot (0) = 0 + 0 = 0 \tag{2.61}$$

$$u_4^1 = u_4^0 - 2 \cdot 4 \cdot (-1) = 0 + 8 = 8 \tag{2.62}$$

**Calcolo di $L(u1)$**

$$L(u^1) = \ Min \sum_{i=1}^{m}(\sum_{j=1}^{n}(c_{ij} - u_j^1)x_{ij}) + \sum_{j=1}^{n} u_j^1 \tag{2.63}$$

$c_{ij} = \begin{bmatrix} 3 & 3 & 4 & 9 \\ 2 & 6 & -9 & 3 \end{bmatrix}$
$u^1 = (8, 8, 0, 8)$

Come fatto in precedenza $L(u^1) = z_1 + z_2 + 24$ dove:

$$1^0 problema \begin{cases} z_1 = \ Min\ 5x_{11} - 5x_{12} + 4x_{13} + x14 \\ \qquad 5x_{11} + 7x_{12} + 4x_{13} + 2x_{14} \leq 30 \\ \qquad x_{11}, ..., x_{14} \in \{0, 1\} \end{cases}$$

Soluzione ottima: $z_1 = -10;\ x^1_{11} = x^1_{12} = 1;\ x^1_{13} = x^1_{14} = 0$

$$2^0 problema \begin{cases} z_2 = \ Min\ -6x_{21} - 2x_{22} - 9x_{23} - 5x24 \\ \qquad 3x_{21} + x_{22} + 6x_{23} + 4x_{24} \leq 12 \\ \qquad x_{21}, ..., x_{24} \in \{0, 1\} \end{cases}$$

Soluzione ottima: $z_2 = -17;\ x^1_{21} = x^1_{22} = x^1_{23} = 1;\ x^1_{24} = 0$

Quindi $L(u^1) = z_1 + z_2 + 24 = -10 - 17 + 24 = -3$

I vincoli 2.45 sono violati dalla soluzione $x1$ di $L(u^1)$.

$x = (x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24})$
$x^1 = (\ 1,\ \ 1,\ \ 0,\ \ \ 0,\ \ \ 1,\ \ 1,\ \ 1,\ \ 0)$

$$\begin{cases} x_{11} & & & + x_{21} & & & & = 1 \\ & x_{12} & & & + x_{22} & & & = 1 \\ & & x_{13} & & & + x_{23} & & = 1 \\ & & & x_{14} & & & + x_{24} & = 1 \end{cases}$$

*Tutti i vincoli, eccetto il terzo, sono violati!*

**Aggiornamento delle penalità**

Poniamo $k = 2$ e manteniamo $\alpha_2 = 2$ in quanto $L(u^1) > L(u^0)$

$$u^k_j = u^{k-1}_j - \alpha_2 \cdot \frac{(z^* - L(u^{k-1}))}{\sum_{j=1}^{n}(\sum_{i=1}^{m} x_{ij} - 1)^2} \cdot (\sum_{i=1}^{m} x_{ij} - 1) \tag{2.64}$$

$$u^2_1 = u^1_1 - 2 \cdot \frac{6}{3} \cdot (1) = 8 - 4 = 4 \tag{2.65}$$

$$u^2_2 = u^1_2 - 2 \cdot 2 \cdot (1) = 8 - 4 = 4 \tag{2.66}$$

$$u^2_3 = u^1_3 - 2 \cdot 2 \cdot (0) = 0 - 0 = 0 \tag{2.67}$$

$$u^2_4 = u^1_4 - 2 \cdot 2 \cdot (-1) = 8 + 4 = 12 \tag{2.68}$$

**Calcolo di $L(u^2)$**

$$L(u^2) = \ Min \sum_{i=1}^{m}(\sum_{j=1}^{n}(c_{ij} - u^2_j)x_{ij}) + \sum_{j=1}^{n} u^1_2 \tag{2.69}$$

$$L(u^2) = z_1 + z_2 + 20 \tag{2.70}$$

dove $z_1$ e $z_2$ sono i valori ottimi dei problemi sequenti

$c_{ij} = \begin{bmatrix} 3 & 3 & 4 & 9 \\ 2 & 6 & -9 & 3 \end{bmatrix}$
$u^2 = (4, 4, 0, 12)$

$$1^0 problema \begin{cases} z_1 = \ Min \ -x_{11} - x_{12} + 4x_{13} - 3x14 \\ \qquad 5x_{11} + 7x_{12} + 4x_{13} + 2x_{14} \leq 30 \\ \qquad x_{11}, ..., x_{14} \in \{0, 1\} \end{cases}$$

Soluzione ottima: $z_1 = -5;\ x^2_{11} = x^2_{12} = 1,\ x^2_{13} = 0,\ x^2_{14} = 1$

$$2^0 problema \begin{cases} z_2 = \ Min \ -2x_{21} + 2x_{22} - 9x_{23} - 9x24 \\ \qquad 3x_{21} + x_{22} + 6x_{23} + 4x_{24} \leq 12 \\ \qquad x_{21}, ..., x_{24} \in \{0, 1\} \end{cases}$$

Soluzione ottima: $z_2 = -18;\ x^2_{21} = x^2_{22} = 0,\ x^2_{23} = x^2_{24} = 1$

Quindi $L(u^2) = z_1 + z_2 + 24 = -5 - 18 + 20 = -3$

L'unico vincolo violato è $x_{14} + x_{24} = 1$; per cui

$$u^3_j = u^2_j - \alpha_3 \cdot \frac{(z^* - L(u^2))}{\sum_{j=1}^{n} (\sum_{i=1}^{m} x_{ij} - 1)^2} \cdot (\sum_{i=1}^{m} x_{ij} - 1) \tag{2.71}$$

poniamo $\alpha_3 = \alpha_2/2 = 1$. Le nuove penalità sono:

$$u^3_1 = u^2_1 - 6 \cdot (0) = 4 \tag{2.72}$$
$$u^3_2 = u^2_2 - 6 \cdot (0) = 4 \tag{2.73}$$
$$u^3_3 = u^2_3 - 6 \cdot (0) = 0 \tag{2.74}$$
$$u^3_4 = u^2_4 - 6 \cdot (1) = 6 \tag{2.75}$$

**Calcolo di $L(u^3) = z_1 + z_2 + 14$ dove**

$$1^0 problema \begin{cases} z_1 = \ Min \ -x_{11} - x_{12} + 4x_{13} + 3x14 \\ \qquad 5x_{11} + 7x_{12} + 4x_{13} + 2x_{14} \leq 30 \\ \qquad x_{11}, ..., x_{14} \in \{0, 1\} \end{cases}$$

Soluzione ottima: $z_1 = -2;\ x^3_{11} = x^3_{12} = 1,\ x^3_{13} = x^3_{14} = 0$

$$2^0 problema \begin{cases} z_2 = \ Min \ -2x_{21} + 2x_{22} - 9x_{23} - 3x24 \\ \qquad 3x_{21} + x_{22} + 6x_{23} + 4x_{24} \leq 12 \\ \qquad x_{21}, ..., x_{24} \in \{0, 1\} \end{cases}$$

Soluzione ottima: $z_2 = -12;\ x^3_{21} = x^3_{22} = 0,\ x^3_{23} = x^3_{24} = 1$

Quindi $L(u^2) = -2 - 12 + 14 = 0$.

■ Si noti che $x^3$ è ammissibile e quindi la soluzione è ottima.

#### 2.5.2.2 (b) Rispetto ai vincoli 2.46

$$L(u) = Min \overbrace{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{m}\lambda_i(\sum_{j=1}^{n} a_{ij}x_{ij} - b_i)}^{cx-\lambda(Ax-b)} = \tag{2.76}$$

$$L(u) = Min \sum_{j=1}^{n}(\sum_{i=1}^{m}(c_{ij} - \lambda_i a_{ij})x_{ij}) + \sum_{i=1}^{m}\lambda_i b_i \tag{2.77}$$

$$\sum_{i=1}^{m} x_{ij} = 1 \tag{2.78}$$

$$x_{ij} \in \{0,1\} \tag{2.79}$$

L'ottimo si ottiene ponendo, per ogni $j$

$$x_{i^*j} = 1 \quad \text{per} \quad c_{i^*j} - \lambda_{i^*j}a_{i^*j} = \underset{i}{Min}\{c_{ij} - \lambda_i a_{ij}\} \tag{2.80}$$

Ovvero, ogni oggetto $j$ viene assegnato al contenitore $i^*$ rispetto al quale $j$ ha costo minimo.

**Esempio** *(Problema precedente con $m = 2$, $n = 4$)*

$$a_{ij} = \begin{bmatrix} 5 & 7 & 4 & 2 \\ 3 & 1 & 6 & 4 \end{bmatrix} \; c_{ij} = \begin{bmatrix} 3 & 3 & 4 & 9 \\ 2 & 6 & -9 & 3 \end{bmatrix}$$

$b = (30, 12)$

Poniamo $\lambda = (0, 0)$ e assumiamo $z^* = 3$ (come in precedenza).

$$L(\lambda^0) = Min \sum_{j=1}^{n}(\sum_{i}^{m}(c_{ij} - \lambda_i a_{ij})x_{ij} + \sum_{i=1}^{m}\lambda_i b_i) \tag{2.81}$$

$$L(\lambda^0) = 3x_{11} + 3x_{12} + 4x_{13} + 9x_{14} + 2x_{21} + 6x_{22} - 9x_{23} + 3x_{24} + 0 \tag{2.82}$$

$$\begin{cases} x_{11} + x_{21} = 1 \\ x_{12} + x_{22} = 1 \\ x_{13} + x_{23} = 1 \\ x_{14} + x_{24} = 1 \end{cases} \tag{2.83}$$

Poniamo $x_{i^*j} = 1$ dove $c_{i*j} = \underset{i}{Min}\{c_{ij}\};\ \ \forall j$

Dal $1^o$ vincolo $x_{11} = 0,\ x_{21} = 1$
Dal $2^o$ vincolo $x_{12} = 1,\ x_{22} = 0$
Dal $3^o$ vincolo $x_{13} = 0,\ x_{23} = 1$
Dal $4^o$ vincolo $x_{14} = 0,\ x_{24} = 1$

Quindi $L(\lambda^0) = -1$

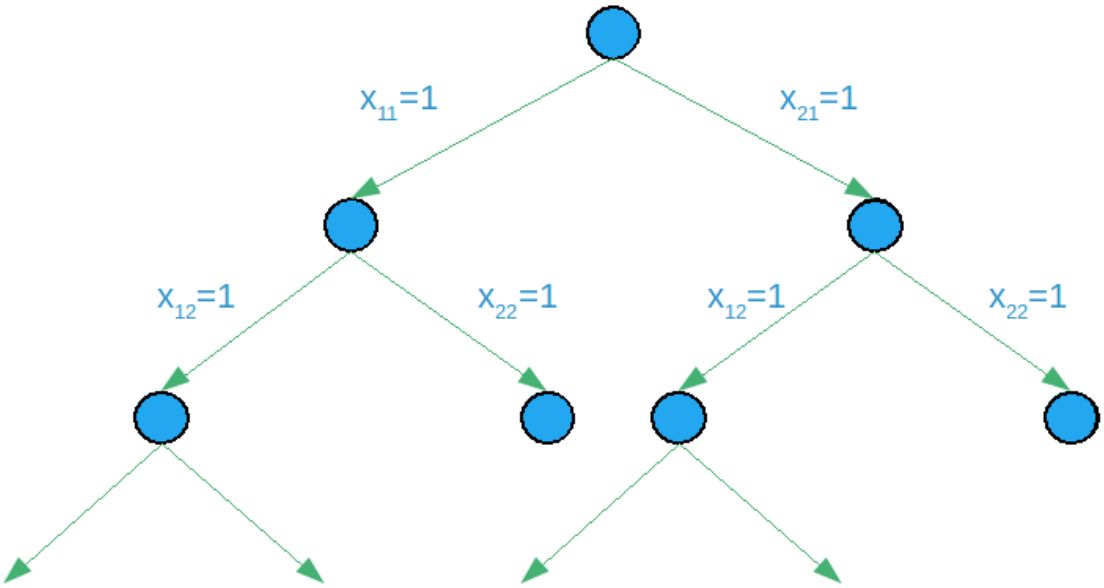
Soluzione ottenuta per $L(u^0)$
$x = (x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24})$
$x^1 = (\ 0,\ \ 1,\ \ \ 0,\ \ \ 0,\ \ \ 1,\ \ 0,\ \ 1,\ \ 1)$

$5x_{11} + 7x_{12} + 4x_{13} + 2x_{14} \leq 30$ soddisfatto
$3x_{21} + x_{22} + 6x_{23} + 4x_{24} \leq 12$ violato!

Ulteriori iterazioni possono migliorare il lower bound...

### 2.5.3 Algoritmo Branch & Bound

**(Ad esempio)**




Ad ogni livello $j$ viene deciso in quale contenitore inserire l'oggetto $j$.

Da ogni nodo vengono generati $m$ nodi.

# APPENDICE A

## PROVA

### A.1 Pippo