NETWORK OPTIMIZATION

II° SEMESTRE
2021

9/03

INTRODUCTION

NETWORK FLOWS → APPROX. ALGOS → Tool: GUROBI 9.1

NETWORK OPTIMIZATION → NP-HARD PROBLEMS → Tool: NetworkX (PY. LIBRARY)

Book: INTEGER PROGRAMMING (L. WALSEY, EDITION 1)

Mixed Integer Linear Program

DATA

A : $m \times n$ RATIONAL MATRIX

G : $m \times p$ ″

c : n-DIMENSIONAL RATIONAL VECTOR

h : p-DIMENSIONAL ″

b : m-DIMENSIONAL ″

VARs.

x : n-DIMENSIONAL VAR. VECTOR

y : p-DIMENSIONAL INTEGER VAR. VECTOR

MILP

$$\boxed{\max \quad c^T x + h^T y}$$

s.t.

$$\boxed{Ax + Gy \le b}$$

$$\boxed{\begin{matrix} x \ge 0 \\ y \ge 0, \end{matrix}}$$

$$\boxed{y \text{ INTEGER}}$$

~~DECISION~~ VARs

OBJECTIVE FUNCT.

LINEAR CONSTs

RESTRICTIONS

INTEGRALITY REQ

ILP vs MIXED ILP

- MIXED ILP HAS INTEGER AND NON INTEGER VARIABLES

Integer Linear Program

DATA

A : $m \times n$ RATIONAL MATRIX

c : n-DIMENSIONAL RATIONAL VECTOR

b : m-DIMENSIONAL //

VARs

x : n-DIMENSIONAL INTEGER VAR. VECTOR

ILP

$$\boxed{\max c'x}$$
$$\boxed{Ax \leq b}$$
$$\boxed{x \geq 0, \text{ integer}}$$

SOL. Assegnamento di valori alle vars.

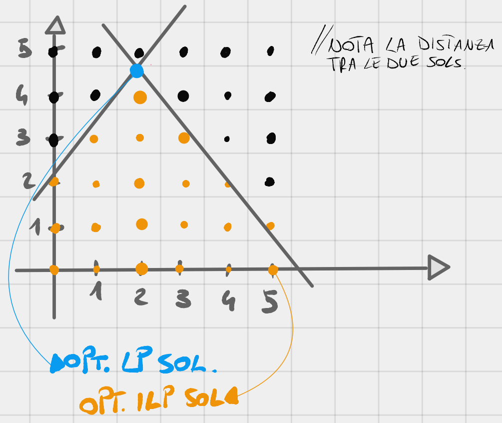
FEASIBLE SOL. Sol. che soddisfi i constraint

OPT. SOL. Feasible sol. che max/min l'OBJECTIVE FUNCTION.

ES.

$$\max \; 1x_1 + 0.4x_2$$
$$50x_1 + 31x_2 \leq 250$$
$$-3x_1 + 2x_2 \leq 4$$
$$x_1, x_2 \geq 0$$
$$\cancel{x_1, x_2 \text{ integer}}$$

LINEAR RELAXATION



Binary Integer Program

DATA

A : m X n RATIONAL MATRIX
c : n-DIMENSIONAL RATIONAL VECTOR
b : m-DIMENSIONAL //

VARs

x : n-DIMENSIONAL INTEGER VAR. VECTOR

BIP

$$\boxed{\max c'x}$$
$$\boxed{Ax \leq b}$$
$$\boxed{x \in \{0,1\}^n}$$

Combinatorial Optimization Problem

given...

- $N = \{1, \_\_\_, n\}$,
- $C = \{c_1, \_\_, c_n\}$ weights vector
- F famiglia di feasible sols. $S \subseteq N$

... find

- $$\min_{S \subseteq N} \{\sum_{j \in S} c_j : S \in \mathcal{F}\} \quad (\max_{S \subseteq N} \{\sum_{j \in S} c_j : S \in \mathcal{F}\})$$

NOTA: ogni comb. opt problem può essere formulato come BIP

- S è rappresentato dall' INCIDENCE VECTOR $x = (x_1, x_2, \ldots, x_n)$
T.C.

$$x_j = \begin{cases} 1 & \text{if } j \in S, \\ 0 & \text{otherwise} \end{cases}$$

↳ Un Problema di ottimizzazione combinatoria si affronta enumerando le feasible sols. e prendendo la migliore

KNAPSACK

given...
- $N = \{1, \_, n\}$ items
- a_j weight, c_j profit per ogni item
- b capacity

... find
- sottoinsieme di item la cui somma dei pesi non ecceda la capacità e che massimizzi la somma dei profitti

ES (KNAPSACK)

o_i	1	2	3	4
a_i	2	3	4	1
c_i	10	14	12	8

$b = 5$

① costruiamo F

$F = \{ \{0\}, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{1,4\}, \dots \}$

② OPT $\in \{1,2\}$

↳ come la formuliamo come BIP?

① scelta vars

VARS DEF.

$$x_j = \begin{cases} 1 & \text{if item } j \text{ is selected,} \\ 0 & \text{otherwise} \end{cases}$$

② trova consts

CONSTRAINTS DEF.

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

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

③ scegli obs func.

OBS. FUNCT.

$$\max \sum_{j=1}^{n} c_j x_j$$

16/03

Set Covering, Set Packing, Set Partitioning

given...
- $M = \{1, \_, m\}$, $N = \{1, \_, n\}$ INSIEMI FINITI
- $\{M_1, \_, M_n\}$ COLLEZIONE DI SOTTOINSIEMI DI M
- $c_j \ \forall M_j$

Nota: F è un insieme di M_j

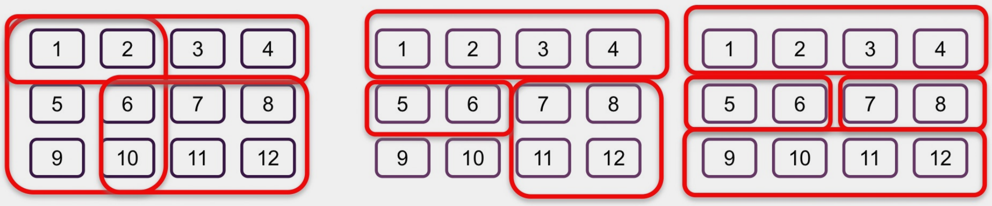
...
- $F \subseteq N$ è
  - COVER $\boxed{\cup_{j \in F} M_j = M}$
  - PACKING $\boxed{M_h \cap M_k = \emptyset \quad \forall h, k, h \neq k}$
  - PARTITION: se è sia cover che packing

Esempio

$M = \{1,2,3,4,5,6,7,8,9,10,11,12\}$

$M_1 = \{1,2,3,4\}$
$M_2 = \{1,2,5,6,9,10\}$
$M_3 = \{7,8,11,12\}$
$M_4 = \{5,6\}$

$M_5 = \{6,7,8,10,11,12\}$
$M_6 = \{9,10,11,12\}$
$M_7 = \{7,8\}$



M_1,M_2,M_5 is a **cover** — M_1,M_3,M_4 is a **packing** — M_1,M_4,M_6,M_7 is a **partition**

- Una COVER è un insieme di sottoinsiemi che prende ogni elemento infatti possono esserci elementi in comune tra i sottoinsiemi.
- Un PACKING invece è un insieme di sottoinsiemi che non hanno elementi in comune (NO INTERSEZIONI)
- Una PARTITION è un PACKING e una COVER

FORMULAZIONE

- VARIABLES $x_j = \begin{cases} 1 \text{ if } j \in F \\ 0 \text{ otherwise} \end{cases}$

- A MATRICE DI INCIDENZA $n \times m$ $\quad a_{ij} = \begin{cases} 1 \text{ if } i \in M_j \\ 0 \text{ otherwise} \end{cases}$

Esempio

$M = \{1,2,3,4,5,6,7,8,9,10,11,12\}$

$M_1 = \{1,2,3,4\}$
$M_2 = \{1,2,5,6,9,10\}$
$M_3 = \{7,8,11,12\}$
$M_4 = \{5,6\}$
$M_5 = \{6,7,8,10,11,12\}$
$M_6 = \{9,10,11,12\}$
$M_7 = \{7,8\}$

$A_{n \times m}$ ↳

n\m	M_1	M_2	M_3	M_4	M_5	M_6	M_7
1	1	1					
2	1	1					
3	1						
4	1						
5		1		1			
6		1		1	1		
7			1		1		1
8			1		1		1
9		1				1	
10		1			1	1	
11			1		1	1	
12			1		1	1	

$$\min c'x \quad \text{s.t. } Ax \geq 1 \quad x \in \{0,1\}^n$$

Set covering

$$\max c'x \quad \text{s.t. } Ax \leq 1 \quad x \in \{0,1\}^n$$

Set packing

$$\min(\text{ or } \max)\ c'x \quad \text{s.t. } Ax = 1 \quad x \in \{0,1\}^n$$

Set partitioning

17/03

MODELLARE CONDIZIONI LOGICHE
(con CONSTRAINT LINEARI e VARS BINARIE)

• Siano x, y, z VARIABILI BINARIE

NOT $z = \neg x \to \boxed{z = 1 - x}$

AND

x	y	$z = x \wedge y$
0	0	0
0	1	0
1	0	0
1	1	1

→ $z = 0$ if $x = 0$ OR $y = 0$ ⇝

→ $z = 1$ if $x = 1$ AND $y = 1$ ⇝

$$\boxed{\begin{cases} z \le x \\ z \le y \\ x + y - z \le 1 \end{cases}}$$

OR

x	y	$z = x \vee y$
0	0	0
0	1	1
1	0	1
1	1	1

$$\boxed{\begin{cases} x + y \ge z \\ z \ge x \\ z \ge y \end{cases}}$$

XOR

x	y	$z = x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

→ COVERED BY ① o ②

→ " " (① o ②) e ③

$$\boxed{\begin{cases} x - y \le z & ① \\ y - x \le z & ② \\ x + y \ge z & ③ \\ x + y + z \le 2 & ④ \end{cases}}$$

↳ ④ TAGLIA VIA LA SOL. $x = y = z = 1$

MODELLING DEPENDENT DECISIONS
(con CONSTRAINT LINEARI e VARS BINARIE)

IDEA

• Possiamo usare BIN. VARS e LINEAR CONSTR. per rappresentare la DIPENDENZA TRA DUE SCELTE

esempio

Suppose x and y are binary variables such that:

$$x = \begin{cases} 1 & \text{if project } x \text{ is selected,} \\ 0 & \text{otherwise} \end{cases} \qquad y = \begin{cases} 1 & \text{if project } y \text{ is selected,} \\ 0 & \text{otherwise} \end{cases}$$

Suppose that project x can be selected **only if** project y has already been selected.

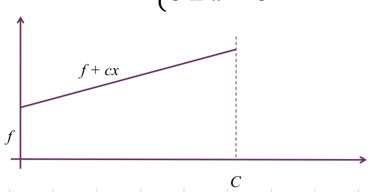
This can be expressed by the (linear) constraint

$$x - y \le 0$$

MODELLING FIXED COST

• Vogliamo minimizzare il fixed cost se un servizio è attivato

$$\min h(x) = \begin{cases} f + cx \text{ if } 0 < x \le C \\ 0 \text{ if } x = 0 \end{cases} \Big\} \text{ funzione non lineare}$$



⇒ $f + cx$ e C è l'upperbound

f: fixed cost; cx: costo variabile

23/03

UNCAPACITATED FACILITY LOCATION PROBLEM

$$\min \sum_{j=1}^{n} f_j y_j + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}$$
$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for } i = 1,\dots,m$$
$$x_{ij} \le y_j \quad \forall i,j$$
$$x_{ij} \in \{0,1\}$$
$$y_j \in \{0,1\}$$

1- Aggiungiamo una VAR. BIN. extra y = "Il servizio è attivato?"

$$y = \begin{cases} 1 & \text{if } 0 < x \le C \\ 0 & \text{if } x = 0 \end{cases}$$

$\Rightarrow \min \ fy + Cx$

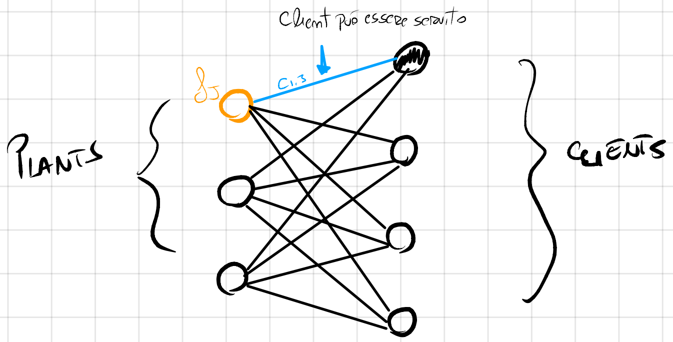
$x - Cy \le 0$ → se $y=0$ non ha senso pagare, quindi $\Rightarrow x \le 0$
→ se $y=1$ $\Rightarrow 0 \le x \le C$

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

$x \ge 0$

↳ si nota che se $y=1$ il minimo x è $x=0$, questa feasible solution va in contrasto con la definizione di y, ma una OPT. SOL. avrà sempre $y=0$ se $x=0$

• Dato $N=\{1,\dots,n\}$ di IMPIANTI PER DELLE STRUTTURE
$M=\{1,\dots,m\}$ di CLIENTS
f_j costo per aprire una STRUTTURA in J
c_{ij} costo per fare servire il client i dalla STRUTTURA J



VARS.

• VARIABLES

$$y_j = \begin{cases} 1 \text{ if facility } j \text{ is open,} \\ 0 \text{ otherwise} \end{cases}$$ ASSOCIATA ALLA STRUTTURA J

$$x_{ij} = \begin{cases} 1 \text{ if client } i \text{ is served from facility } j, \\ 0 \text{ otherwise} \end{cases}$$ ASSOCIATA A UN ARCO (i,j)

OBJ FUNCT

• OBBIETTIVO

$$\min \sum_{j=1}^{n} f_j y_j + \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}$$

CONSTRS.

(1) Ogni client è servito da una sola STRUTTURA

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for } i = 1,\dots,m$$

(2) Un client può essere servito solo se la STRUTTURA c'è

$$x_{ij} \le y_j \quad \forall i,j$$

MODELLING RESTRICTED SET OF VALUES

X può assumere valori solo in $\{a_1, a_2, \ldots, a_m\}$

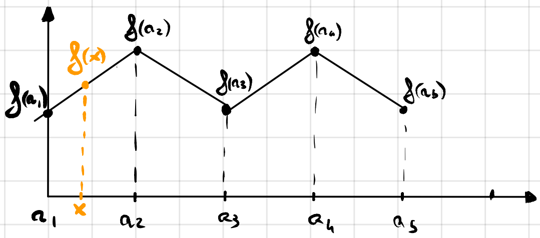
Introduciamo una variabile y_i per ogni a_i

$$\boxed{\begin{cases} x = \sum_{i=1}^{m} a_i y_i \\ \sum_{i=1}^{m} y_i = 1 \\ y_i \in \{0,1\} \quad , \quad i = 1, \ldots, m \end{cases}}$$

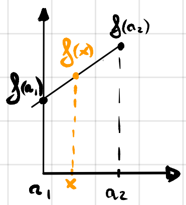
26/03

PIECEWISE LINEAR COST FUNCTION

- Modellare come BP una funzione lineare a tratti



- Una funzione lineare a tratti è composta da m coppie (a_i, f_i)



COMBINAZIONE CONVESSA

$$\begin{cases} f(x) = \lambda_1 f(a_1) + \lambda_2 f(a_2) \\ \lambda_1 + \lambda_2 = 1 \\ \lambda_1, \lambda_2 \geq 0 \end{cases}$$

- generalizzando la combinazione convessa a un intervallo $[a_h, a_{h+1}]$

$$\begin{cases} f(x) = \lambda_h f(a_h) + \lambda_{h+1} f(a_{h+1}) \\ \lambda_h + \lambda_{h+1} = 1 \\ \lambda_h, \lambda_{h+1} \geq 0 \end{cases}$$

$\Rightarrow$ Una LINEAR PIEWISE funct. è esprimibile come LINEAR func. nel settore h

BINARY VARS

- $y_h = \begin{cases} 1 & \text{, se } x \text{ è in } [a_h, a_{h+1}] \ \forall h = 1, \dots, m-1 \\ 0 & \text{, altrimenti} \end{cases}$

OBJ. FUNC

- $\min \sum_{i=1}^{m} \lambda_i f(a_i) = \lambda_1 f(a_1) + \lambda_2 f(a_2) + \dots$

CONSTRS

$$\boxed{\begin{array}{ll} (1) & \sum_{i=1}^{m} \lambda_i = 1 \\ (2) & \lambda_1 \le y_1 \\ (3) & \lambda_i \le y_{i-1} + y_i \quad , \ i = 2, \dots, m-1 \\ (4) & \lambda_m \le y_{m-1} \\ (5) & \sum_{i=1}^{m-1} y_i = 1 \end{array}}$$

→ Tutti questi constrs impongono una sola condizione

$$\boxed{y_h = 1 \Rightarrow \lambda_i = 0 \ \forall i \ne h, h+1}$$

BOUNDS

$\lambda_h \ge 0 \ \forall h = 1, \dots, m$

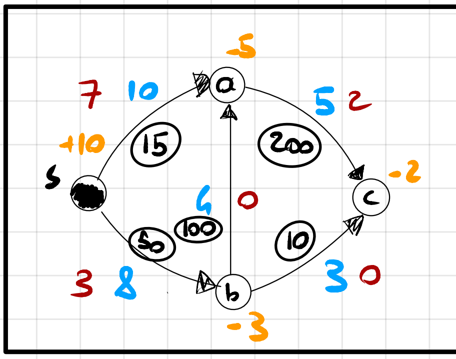
$y \in \{0, 1\} \ \forall h = 1, \dots, m-1$

FIXED CHARGE NETWORK PROBLEM

given

- $G = (N, A)$ diretto
- $s \in N$ detto SOURCE
- DEMAND VECTOR $b \in \mathbb{Z}^{|N|}$ | $b_s > 0$, $b_i \le 0 \ \forall i \ne s \mid i \in N$, $\sum_{i \in N} b_i = 0$

  ASSOCIATO AI NODI



- $u \in \mathbb{Z}_+^{|A|}$ CAPACITY VECTOR

  ASSOCIATO AGLI ARCHI

- $c \in \mathbb{Z}_+^{|A|}$ ACTIVATION COST VECTOR degli archi

find

- $x \in \mathbb{Z}_+^{|A|}$ FEASIBLE FLOW (BALANCE e CAPACITY CONSTs) che minimizzi il costo di attivazione degli archi

VARS

- $y_{ij} \begin{cases} 1 & \text{, se } (i,j) \text{ è ATTIVATO} \\ 0 \end{cases}$

OBJ

- $\min \sum_{(i,j) \in A} c_{ij} y_{ij}$ // ACTIVATION COST

CONSTR.

- $0 \le x_{ij} \le y_{ij} u_{ij} \quad \forall (i,j) \in A$ // ACTIVATION CONSTRAINT

  $\sum_{(i,j) \in \delta^+(i)} x_{ij} - \sum_{(j,i) \in \delta^-(i)} x_{ji} = b_i \quad \forall i \in N$ // FLOW BALANCE CONSTRAINT

DISJUNCTIVE CONSTRAINTS

INDICATOR

- Dati due constraints, supponiamo di volerli modellare come 'OR'

$$a'x \ge b$$
$$c'x \ge \delta$$
$$a, c, b, \delta \ge 0$$
$$x \ge 0$$

$$a'x \ge yb$$
$$c'x \ge (y-1)\delta$$
$$y \in \{0,1\}$$

→ Introduciamo una BINARY VARIABLE y detta INDICATOR e poniamo in DISGIUNZIONE i CONSTRAINTS

- Generalizzando per m constraints $a_i' x \ge b_i, \ i = \{1, \dots, m\}$

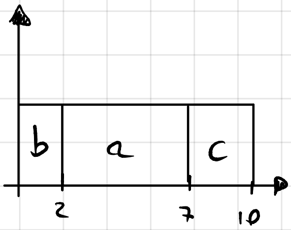
→ se vogliamo almeno k degli m constraints soddisfatti, introduciamo m INDICATORS

$$a_i' x \ge b_i y_i$$
$$\sum_{i=1}^{m} y_i \ge k$$
$$y_i \in \{0,1\}$$

SINGLE MACHINE SCHEDULING PROBLEM

- Dati n jobs con processing time p_j, e una sola macchina M; t_j è lo START TIME del job j

	p_j
a	5
b	2
c	3



- Considerando due job h e k:

$t_k \ge t_h + p_h$ se $h \to k$

$t_h \ge t_k + p_h$ se $k \to h$

- I due constraint non possono essere vera allo stesso tempo, quindi introduciamo l'INDICATOR

$$y_{hk} = \begin{cases} 1, & \text{se } h \to k \\ 0 \end{cases}$$

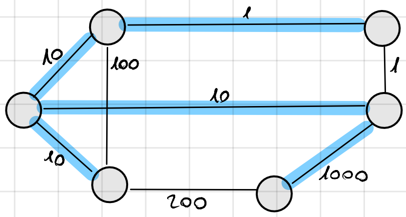
- Possiamo porre ora i constraints in disjunction, nota che M è una BIG COSTANT, poiché non possiamo moltiplicare il RHS per y; dobbiamo quindi sottrarre per rendere l'RHS $= \emptyset$

- $M = T + \max_{j \in J} \{p_j\}$

$$t_k \ge t_h + p_h - M(1 - y_{hk})$$
$$t_h \ge t_k + p_k - M y_{hk}$$

MINIMUM SPANNING TREE

SPANNING TREE



#SPANNING TREE (NON MINIMO)
↳ COST = 1031

MST

- Dati $G=(V,E)$, $c_e \geq 0 \ \forall e \in E$
- Trova uno SPANNING TREE di minimo COSTO

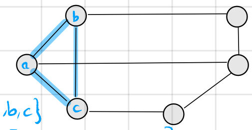
VARs

- VARIABLES

$$\boxed{x_e \begin{cases} 1, & \text{se } e \text{ è selezionato} \\ 0 \end{cases}}$$

NOTA

- Introduciamo $E(S)$: insieme di archi indotti dai nodi in $S \subset V$



$S = \{a,b,c\}$
$E(S) = \{(a,b),(a,c),(b,c)\}$

Se S è un TREE, qual è la massima cardinalità $|E(S)|$?

CONSTRs

- CONSTRAINTS

$$\boxed{\sum_{e \in E} x_e = |V|-1}$$

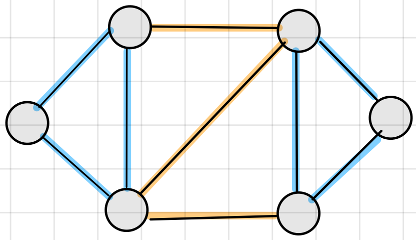
SUBTOUR INEQUALITIES (NUMERO ESPONENZIALE DI CONSTRs)

$$\boxed{\sum_{e \in E(S)} x_e \leq |S|-1 \quad \forall S \subset V, 2 < |S| \leq |V|-1}$$

// SUBTOUR ELIMINATION CONSTRAINT

- Possiamo formularlo meglio?

- SPANNING TREE NON CONNESSO
- Definiamo $\delta(S)$



$$\delta(S) = \{(i,j) \in E \mid i \in S, j \notin S\}$$

CUTSET INEQUALITIES (NUMERO ESPONENZIALE DI CONSTRAINTS ANCHE QUA)

- $$\boxed{\sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subseteq V, S \neq \{\emptyset\}}$$

⇒ ANCORA NON CI SIAMO! Dobbiamo ridurre il numero di constraints

FORMULAZIONE 1

$$\begin{aligned}
\min \ & \sum_{e \in E} c_e x_e \\
\text{s.t.} \ & \sum_{e \in E} x_e = |V|-1, \\
& \sum_{e \in E(S)} x_e \leq |S|-1 \qquad \forall S \subset V, 2 < |S| \leq |V|-1, \\
& x \in \{0,1\}^{|E|}
\end{aligned}$$

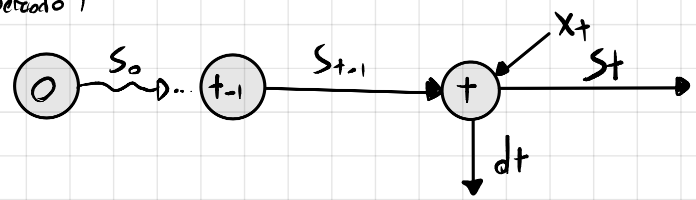
FORMULAZIONE 2

$$\begin{aligned}
\min \ & \sum_{e \in E} c_e x_e \\
\text{s.t.} \ & \sum_{e \in E} x_e = |V|-1, \\
& \sum_{e \in \delta(S)} x_e \geq 1 \qquad \forall S \subset V, S \neq \emptyset, V \\
& x \in \{0,1\}^{|E|}
\end{aligned}$$

UNCAPACITATED LOT SIZING (NELLE SLIDES NON CI STA)

- Fabbriche che producono un solo prodotto
- T : time horizon (composto di periodi t)
- f_t : fixed cost della produzione nel periodo t
- P_t : unita di produzione nel periodo t
- h_t : unit storage cost
- d_t : domanda per il prodotto nel periodo t

- In un generico periodo t



VARS

- VARIABLES

$\boxed{X_t : \text{Quantità di beni prodotti al periodo } t}$ → supponiamo sia una variabile continua

$\boxed{S_t : \text{Stock alla fine del periodo } t}$ → " " "

$$\boxed{y_t = \begin{cases} 1, & \text{se il bene è prodotto nel periodo } t \\ 0 \end{cases}}$$

OBJ

- OBJECTIVE FUNCTION

$$\boxed{\min \underbrace{\sum_{i=1}^{T} P_t x_t}_{\text{PRODUCTION COST}} + \underbrace{\sum_{i=1}^{T} h_t \cdot S_t}_{\text{STORAGE COST}} + \underbrace{\sum_{i=1}^{T} f_t y_t}_{\text{SETUP COST}}}$$

CONSTRS

- CONSTRAINTS

$$\boxed{S_{t-1} + X_t = d_t + S_t \quad , \; t = 1, \dots, T}$$

NOTA

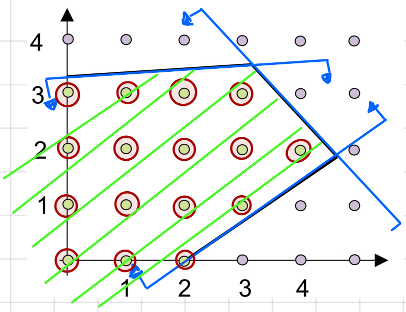
- La produzione può essere avviata al periodo t se il setup della fabbrica è stato fatto

che valore scegliamo per M?

$$\boxed{X_t \leq M y_t \, , \; t = 1, \dots, T}$$

$$\boxed{S_t, X_t \geq 0 \, , \quad y_t \in \{0, 1\}}$$

IP: FEASIBLE REGION



- $Z^* := \max c'x$
  - $x \in P$ → Area definita dai constraints
  - $x \in \cancel{\mathbb{Z}^n}$ → Insieme dei punti interi
  - RELAXATION
- L'intersezione tra $x \in P$ e $x \in \mathbb{Z}^n$ è la FEASIBLE REGION

LINEAR RELAXATION

- $Z^{LP} := \max c'x$
  $Ax \leq b$
  $x \geq 0$

- Se la soluzione x^{LP} del problema rilassato è INTERA ⇒ è ottima anche per il l'IP di partenza perché

$$Z^{LP} \geq Z^*$$