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 : $m \times m$ RATIONAL MATRIX
G : $m \times p$ "
c : M-DIMENSIONAL RATIONAL VECTOR
h : P-DIMENSIONAL "
b : m-DIMENSIONAL "

VARS.

x : m-DIMENSIONAL VAR. VECTOR
y : m-DIMENSIONAL INTEGER VAR. VECTOR

MILP

$$\boxed{\max \quad c^T x + h^T y}$$
s.t.
$$\boxed{Ax + Gy \leq b}$$
$$\boxed{\begin{matrix} x \geq 0 \\ y \geq 0, \end{matrix}}$$
$$\boxed{y \text{ INTEGER}}$$

~~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 × 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.

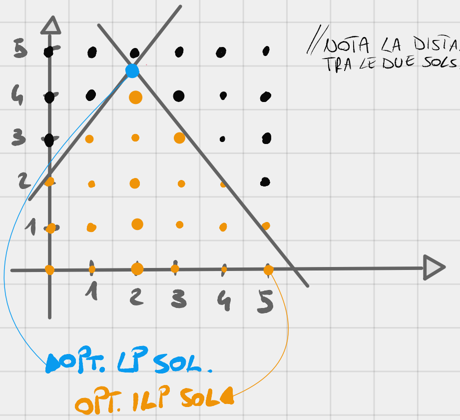
FEASIBLE SOL. Sol. che soddisfi i constraint

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

ES.

$$\max\ 1x_1 + 0.64x_2$$
$$50x_1 + 31x_2 \leq 250$$
$$-3x_1 + 2x_2 \leq 4$$
$$x_1, x_2 \geq 0$$
$$\cancel{x_1, x_2 \text{ INTEGER}}$$

LINEAR RELAXATION



//NOTA LA DISTANZA TRA LE DUE SOLS.

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 ∈ $\{1,2\}$

↳ 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
- $c_j \; \forall M_j$

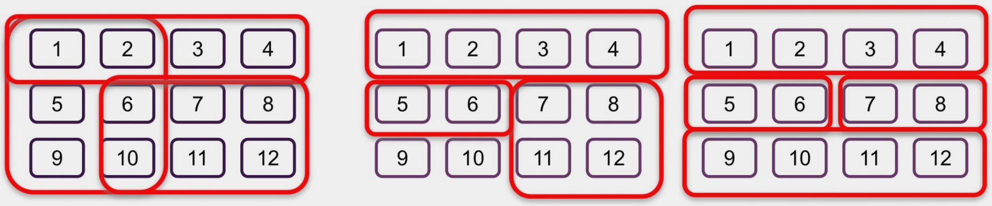
Nota: F è un insieme di M_j

...
- $F \subseteq N$ 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 $\boxed{\text{se è sia cover che packing}}$

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 $n \times m$ $a_{ij} = \begin{cases} 1 \text{ if } i \in M_j \\ 0 \text{ otherwise} \end{cases}$

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	

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

Set covering

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

Set packing

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

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 $y = 1$ ⇝

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

OR

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

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

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

↳ 4 taglia via la sol. $x = y = z = 1$

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

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

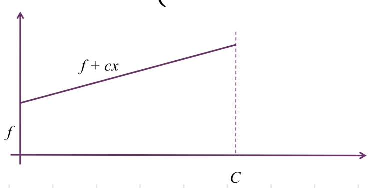
$$x - y \le 0$$

MODELLING FIXED COST

• Vogliamo minimizzare il fixed cost se un servizio è attivato

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

} funzione non lineare



⇒ $f + cx$ e C è l'upperbound

f: fixed cost; cx: costo variabile

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

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

$\Rightarrow \min \; fy + Cx$

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

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

$x \geq 0$

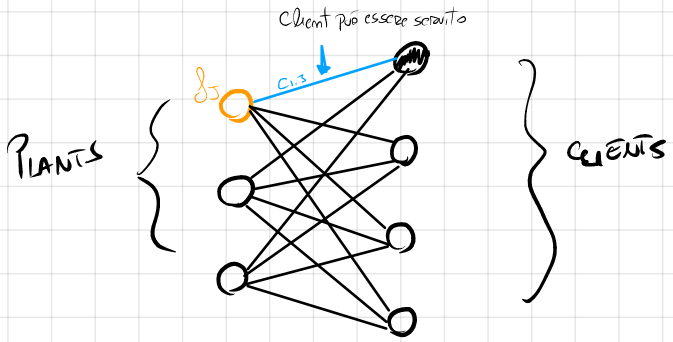
↳ si nota che se $y=1$ il minimo x è $x=0$, questa feasible solution va in contrasto con la definizione di y, ma una OPT. SOL. avrà sempre $y=0$ se $x=0$

23/03

UNCAPACITATED FACILITY LOCATION PROBLEM

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

• Dato $N = \{1, \ldots, n\}$ di IMPIANTI PER DELLE STRUTTURE
$M = \{1, \ldots, m\}$ 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

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

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

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

MODELLING RESTRICTED SET OF VALUES

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

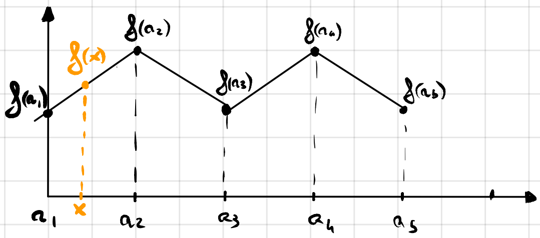
Introduciamo una variabile y_i per ogni a_i

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

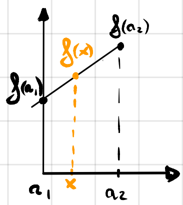
26/03

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 PIECEWISE 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

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

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

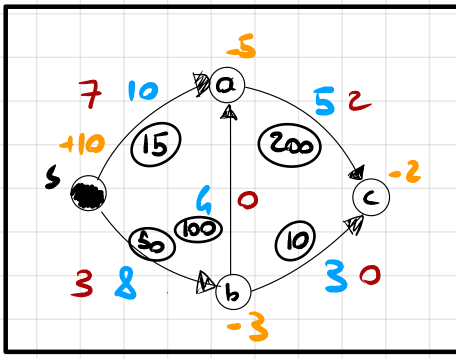
FIXED CHARGE NETWORK PROBLEM

given

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

  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.

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

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

DISJUNCTIVE CONSTRAINTS

INDICATOR

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

$$a'x \geq b$$
$$e'x \geq f$$
$$a, e, b, f \geq 0$$
$$x \geq 0$$

$$a'x \geq yb$$
$$e'x \geq (y-1)f$$
$$y \in \{0,1\}$$

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

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

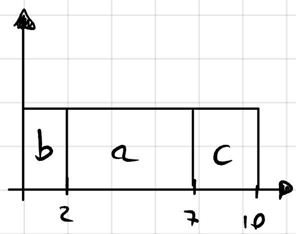
↳ se vogliamo almeno k degli m constraints soddisfatti, introduciamo m INDICATORS

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

SINGLE MACHINE SCHEDULING PROBLEM

- Dati n Jobs con processing time P_J, e una sola macchina M; t_J è lo STARTTIME del Job J

	P_J
a	5
b	2
c	3



- Considerando due Job h e k:

$t_k \geq t_h + P_h$ se $h \to k$

$t_h \geq t_k + P_h$ se $k \to h$

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

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

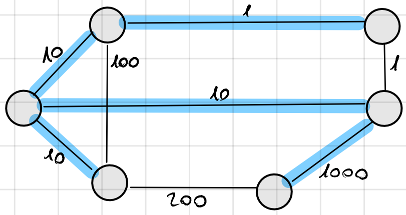
- Possiamo porre ora i constraints in disjunction, nota che M è una BIG COSTANT, poiché non possiamo moltiplicare il RHS per y; dobbiamo quindi sottrarre per rendere l'RHS $= \emptyset$

- $$M = T + \max_{s \in S} \{P_s\}$$

$$t_k \geq t_h + P_h - M(1 - y_{hk})$$
$$t_h \geq t_k + P_k - M y_{hk}$$

MINIMUM SPANNING TREE

SPANNING TREE



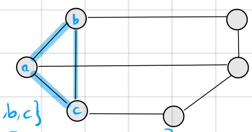
#SPANNING TREE (NON MINIMO)
↳ COST = 1031

MST

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

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, quel è la massima cardinalità $|E(S)|$?

VARs

- VARIABLES

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

CONSTRs

- CONSTRAINTS

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

SUBTOUR INEQUALITIES (NUMERO ESPONENZIALE DI CONSTRs)

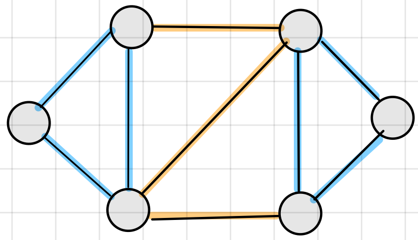
$$\boxed{\sum_{e \in E(S)} x_e \leq |S|-1 \quad \forall S \subset V,\ 2<|S| \leq |V|-1}$$

// 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

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

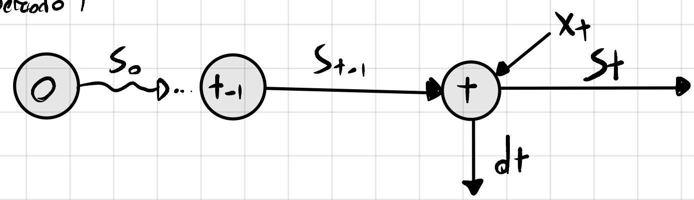
FORMULAZIONE 2

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

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: unità 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

$$\boxed{s_{t-1} + x_t = d_t + s_t \quad , \; t = 1, \ldots, T}$$

NOTA

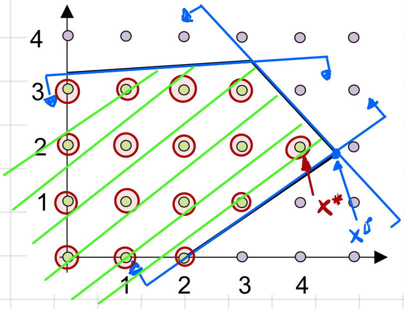
- La produzione può essere avviata al periodo t se il setup della fabbrica è stato fatto

che valore scegliamo per M?

$$\boxed{x_t \leq M y_t \, , \; t = 1, \ldots, T}$$

$$\boxed{s_t, x_t \geq 0 \, , \quad y_t \in \{0, 1\}}$$

IP: FEASIBLE REGION



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

FEASIBLE REGION → $S = P \cap \mathbb{Z}^n$

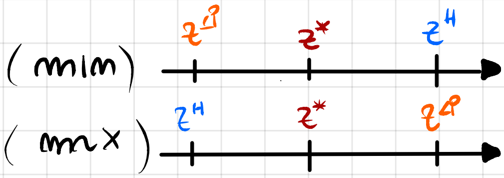
LINEAR RELAXATION

- Se X^{LP} è INTERA, allora coincide con X^*
- Il valore Z^{LP} associato al problema rilassato è un LOW/UP BOUND per Z^*

$$(\min)\quad Z^{LP} = \min_{x \in P} c'x \le \min_{x \in S} c'x = Z^*$$

$$(\max)\quad Z^{LP} = \max_{x \in P} c'x \ge \min_{x \in S} c'x = Z^*$$

- Il valore Z^H associato a una qualsiasi soluzione intera X^H è un UP/LOW BOUND a Z^*

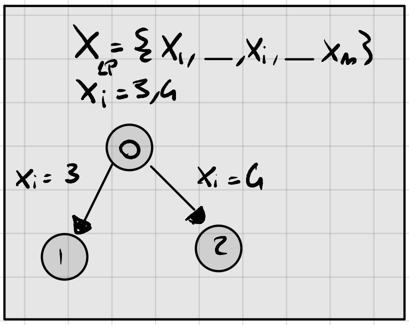


BRANCHING

- Operazione per restringere i bound di Z^*
- Data una soluzione rilassata $X^{LP} := \{x_1, \dots, x_i, \dots, x_n\}$, prendiamo x_i frazionale e suddividiamo il problema in due sottoproblemi

$$z_1^* := \max c'x \quad Ax \le b \quad x \ge 0 \quad \boxed{x_h \le \lfloor x_h^{LP} \rfloor} \quad x \in \mathbb{Z}^n$$

$$z_2^* := \max c'x \quad Ax \le b \quad x \ge 0 \quad \boxed{x_h \ge \lceil x_h^{LP} \rceil} \quad x \in \mathbb{Z}^n$$



- Il BRANCHING definisce

$$P_1 = \{x \in P : x_h \le \lfloor x_h^{LP} \rfloor\} \quad S_1 = P_1 \cap \mathbb{Z}_+^n$$

$$P_2 = \{x \in P : x_h \ge \lceil x_h^{LP} \rceil\} \quad S_2 = P_2 \cap \mathbb{Z}_+^n$$

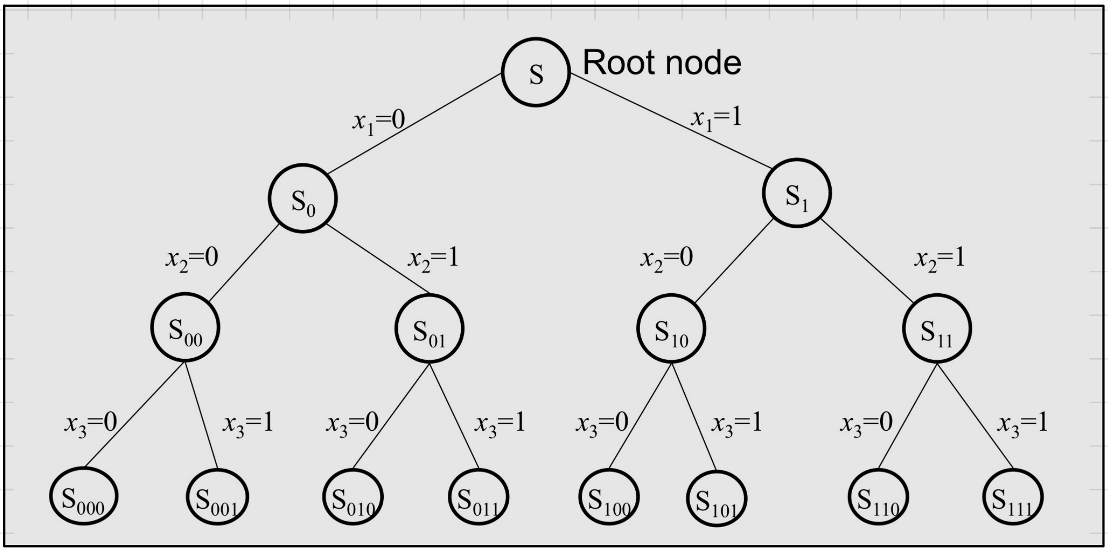
- I due sottoproblemi soddisfano certe proprietà

$$x^{LP} \notin P_1 \text{ and } x^{LP} \notin P_2$$

$$S_1 \cup S_2 = S,\ S_1 \cap S_2 = \emptyset$$

$$z^* = \max\{z_1^*, z_2^*\}$$

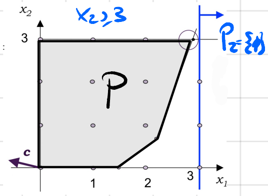
ENUMERATION TREE
(per vars binarie)



PRUNING

- Possiamo evitare di proseguire su alcuni rami se si verificano certe condizioni

  - **By INFEASIBILITY**: l'insieme definito dai constraint P_t è vuoto ⇒ anche quello delle soluzioni S_t lo è



  - **By OPTIMALITY**: se x_t^{LP} è INTERA allora è ottima per il sottoproblema $x_t^{LP} \in S_t$
    - se $x_t^{LP} \in S_t \Rightarrow x_t^{LP}$ feasible anche per S
    - INOLTRE:
      ↳ se $x_t^{LP} \geq z^H / x_t^{LP} \leq z^H \Rightarrow$ Aggiorniamo $z^H = x_t^{LP}$

  - **By BOUND**: se x_t^{LP} FRAZIONALE e $z_t^{LP} \leq z^H$

PREPROCESSING

- Tecniche per ridurre la taglia di un ILP MODEL

BOUND TIGHTENING

- Prendiamo un problema e miglioriamo i bounds delle variabili

$$\max \; 2x_1 + x_2 - x_3$$
$$5x_1 - 2x_2 + 8x_3 \leq 15$$
$$8x_1 + 3x_2 - x_3 \geq 9$$
$$x_1 + x_2 + x_3 \leq 6$$

BOUNDS

$$0 \leq x_1 \leq \cancel{3} \; \tfrac{9}{5}$$
$$0 \leq x_2 \leq 1$$
$$1 \leq x_3$$

(1)

$$\cancel{5}x_1 \leq \frac{15 + 2x_2 - 8x_3}{5} \Rightarrow x_1 \leq \frac{15 + 2 - 8}{5} = \boxed{\tfrac{9}{5}} \checkmark$$

(1)

(1) Prendiamo il primo constraint e la prima variabile;
(2) Controlliamo se rispetta il suo bound
(3) troviamo un altro bound per x_1 e sostituiamo (scegli bene, devi migliorare i bounds)

⇒ Continuiamo per tutti i constraints e le variabili

② ↳ Constr. 1, Var. x_3

$x_2=1$, $x_1=0$

$$8x_3 \le 15 + 2x_2 - 5x_1 = 17 \to x_3 \le \boxed{17/8} \Rightarrow$$

$$\begin{array}{ll} \max & 2x_1 + x_2 - x_3 \\ & 5x_1 - 2x_2 + 8x_3 \le 15 \\ & 8x_1 + 3x_2 - x_3 \ge 9 \\ & x_1 + x_2 + x_3 \le 6 \\ \text{BOUNDS} & 0 \le x_1 \le 9/5 \\ & 0 \le x_2 \le 1 \\ & 1 \le x_3 \le 17/8 \end{array}$$

③ ↳ Constr 1, VAR x_2

$x_1=0$, $x_3=1$

$$2x_2 \ge 5x_1 + 8x_3 - 15 \to \boxed{x_2 \ge -7/2}$$

FAIL

$$\begin{array}{ll} \max & 2x_1 + x_2 - x_3 \\ & 5x_1 - 2x_2 + 8x_3 \le 15 \\ & 8x_1 + 3x_2 - x_3 \ge 9 \\ & x_1 + x_2 + x_3 \le 6 \\ \text{BOUNDS} & 0 \le x_1 \le 9/5 \\ & 0 \le x_2 \le 1 \\ & 1 \le x_3 \le 17/8 \end{array}$$

④ ↳ Constr 2, VAR x_1

$x_2=1$, $x_3=1$

$$8x_1 \ge 9 - 3x_2 + x_3 \to \boxed{x_1 \ge \frac{7}{8}} \Rightarrow$$

$$\begin{array}{ll} \max & 2x_1 + x_2 - x_3 \\ & 5x_1 - 2x_2 + 8x_3 \le 15 \\ & 8x_1 + 3x_2 - x_3 \ge 9 \\ & x_1 + x_2 + x_3 \le 6 \\ \text{BOUNDS} & 7/8\ \cancel{0} \le x_1 \le 9/5 \\ & 0 \le x_2 \le 1 \\ & 1 \le x_3 \le 17/8 \end{array}$$

NOTA · Possiamo anche ripetere gli step

AGAIN ② ↳ Constr 1, VAR x_3

$x_2=1$, $x_1=7/8$

$$8x_3 \le 15 + 2x_2 - 5x_1 \Rightarrow \boxed{x_3 \le \frac{101}{64}} \Rightarrow$$

$$\begin{array}{ll} \max & 2x_1 + x_2 - x_3 \\ & 5x_1 - 2x_2 + 8x_3 \le 15 \\ & 8x_1 + 3x_2 - x_3 \ge 9 \\ & x_1 + x_2 + x_3 \le 6 \\ \text{BOUNDS} & 7/8\ \cancel{0} \le x_1 \le 9/5 \\ & 0 \le x_2 \le 1 \\ & 1 \le x_3 \le \cancel{17/8}\ 101/64 \end{array}$$

⇒ Cosa otteniamo?

RIDUZIONE · Il constraint $x_1+x_2+x_3 \le 6$ diventa inutile perché $\underset{=9/5}{x_1} + \underset{=1}{x_2} + \underset{=101/64}{x_3} < 6$

$$\Rightarrow \begin{array}{ll} \max & 2x_1 + x_2 - x_3 \\ & 5x_1 - 2x_2 + 8x_3 \le 15 \\ & 8x_1 + 3x_2 - x_3 \ge 9 \\ & \cancel{x_1 + x_2 + x_3 \le 6} \\ \text{BOUNDS} & 7/8 \le x_1 \le 9/5 \\ & 0 \le x_2 \le 1 \\ & 1 \le x_3 \le 101/64 \end{array}$$

LINEAR PREPROCESSING TH.

· Considera $S = \{x \mid a_0x_0 + \sum_{s=1}^{m} a_s x_s \le b,\ l_s \le x_s \le u_s,\ s=0,\dots,m\}$

· Gli step sono:

(1) BOUND TIGHTENING: due sottocasi

if $a_0 > 0 \to x_0 \le \left(b - \sum_{s|a_s>0} a_s l_s - \sum_{s|a_s<0} a_s u_s\right)/a_0$ → UPPER BOUND

if $a_0 < 0 \to x_0 \ge \left(b - \sum_{s|a_s>0} a_s l_s - \sum_{s|a_s<0} a_s u_s\right)/a_0$ → LOWER BOUND

(2) REDUNDANCY CHECK

if $\sum_{s|a_s>0} a_s u_s + \sum_{s|a_s<0} a_s l_s \le b$ → THE CONSTRAINT IS REDUNDANT.

(3) INFEASIBILITY CHECK

if $\sum_{s|a_s>0} a_s l_s + \sum_{s|a_s<0} a_s u_s > b \to S = \{\emptyset\}$

(4) VARIABLE FIXING

Per un problema della forma

$$\boxed{\begin{aligned} \max\ & cx \\ & Ax \\ & l \le x \le u \end{aligned}}$$

- if $a_{ij} \geq 0 \ \forall i = 1, \dots, m \land c_j < 0 \Rightarrow x_j = l_j$
- if $a_{ij} \leq 0 \ \forall i = 1, \dots, m \land c_j > 0 \Rightarrow x_j = u_j$

PREPROCESSING PER BIP

$$\begin{aligned}
7x_1 + 3x_2 - 4x_3 - 2x_4 &\leq 1 \quad (1)\\
-2x_1 + 7x_2 + 3x_3 &\leq 6 \quad (2)\\
-2x_2 - 3x_3 - 6x_4 &\leq -5 \quad (3)\\
3x_1 - 2x_3 &\geq -1 \quad (4)
\end{aligned}$$

$$x \in \{0,1\}^4$$

- Un BIP con 4 vars ha al più 16 soluzioni $\Rightarrow$ Possiamo ridurle?
- Trasformiamo ogni constraint in $a\tilde{x} \leq b$ con $a \geq 0$

$$(1) \quad 7x_1 + 3x_2 - 4x_3 - 2x_4 \leq 1$$

$$\Downarrow \leftarrow x_4 = 1 - x_4,\ x_3 = 1 - x_4$$

$$7x_1 + 3x_2 + 4\tilde{x}_3 + 2\tilde{x}_4 \leq 1 + 4 + 2 = 7$$

↳ ci chiediamo se possiamo settare x_1 e x_2 a 1 $\Rightarrow$ No!
formalizziamo questo fatto come CONSTRAINT

$$x_1 + x_2 \leq 1 \qquad (1, 1, *, *)$$

ora lo facciamo con altri assegnamenti

$$x_1 + \tilde{x}_3 \leq 1 \qquad (1, *, 1, *)$$
$$x_1 + \tilde{x}_4 \leq 1 \qquad (1, *, *, 1)$$

↳ I nuovi constraint ci dicono

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ x_1 + \tilde{x}_3 &\leq 1 \\ x_1 + \tilde{x}_4 &\leq 1 \end{aligned} \Rightarrow \boxed{\begin{aligned} x_1 + x_2 &\leq 1 \\ x_1 &\leq x_3 \\ x_1 &\leq x_4 \end{aligned}}$$

- Secondo

$$(2) \quad -2x_1 + 7x_2 + 3x_3 \leq 6 \Rightarrow +2\tilde{x}_1 + 7x_2 + 3x_3 \leq 8$$

$$\Rightarrow \boxed{\begin{aligned} \tilde{x}_1 + x_2 &\leq 1 \\ x_2 + x_3 &\leq 1 \end{aligned}} \Rightarrow \boxed{\begin{aligned} x_2 &\leq x_1 \\ x_2 + x_3 &\leq 1 \end{aligned}}$$

③ $+2\tilde{x}_2 + 3\tilde{x}_3 + 6\tilde{x}_4 \le -5 + 2 + 3 + 6 = 6$

$\Rightarrow \boxed{\begin{array}{l}\tilde{x}_3 + \tilde{x}_4 \le 1 \\ \tilde{x}_2 + \tilde{x}_4 \le 1\end{array}} \Rightarrow \boxed{\begin{array}{l}-x_3 - x_4 \le -1 \\ -x_2 - x_4 \le -1\end{array}} \Rightarrow \boxed{\begin{array}{l}x_3 + x_4 \ge 1 \\ x_2 + x_4 \ge 1\end{array}}$

④ $3x_1 - 2x_3 \ge -1 \Rightarrow -3x_1 + 2x_3 \le 1 \Rightarrow 3\tilde{x}_1 + 2x_3 \le 4$

$\boxed{x_1 + \tilde{x}_3 \le 1} \Rightarrow \boxed{x_3 \le x_1}$

· Possiamo ora riscrivere il problema aggiungendo le implicazioni

$7x_1 + 3x_2 - 4x_3 - 2x_4 \le 1$

$\boxed{x_1 + x_2 \le 1}$

$\boxed{x_1 \le x_3}$

$x_1 \le x_4$

$-2x_1 + 7x_2 + 3x_3 \le 6$

$\boxed{x_2 \le x_1}$

$x_2 + x_3 \le 1$

$-2x_2 - 3x_3 - 6x_4 \le -5$

$x_3 + x_4 \ge 1$

$\boxed{x_2 + x_4 \ge 1}$

$3x_1 - 2x_3 \ge -1$

$\boxed{x_3 \le x_1}$

$x \in \{0,1\}^4$

$\boxed{x_2 = \emptyset}$ $\boxed{x_4 = 1}$ $\boxed{x_1 = x_3}$

· Quindi $\Rightarrow \boxed{\begin{array}{l}x_1 = x_3 \\ x_2 = 0 \\ x_4 = 1\end{array}}$ e le soluzioni sono solo 2 $\boxed{\begin{array}{l}(1,0,1,1) \\ (0,0,0,1)\end{array}}$

ESERCIZIO

$$\boxed{\begin{array}{ll}\max & 5x_1 - 7x_2 - 10x_3 + 3x_4 - 5x_5 \\ & x_1 + 3x_2 - 5x_3 + x_4 + 6x_5 \le 0 \\ & -2x - 6x_2 + 3x_3 - 2x_4 - 2x_5 \le -6 \\ & 2x_2 - 2x_3 - x_4 + x_5 \le -2\end{array}}$$

① $x_1 + 3x_2 + 5\tilde{x}_3 + x_4 + 4x_5 \le 5 \Rightarrow$

$$\left\{\begin{array}{lll} x_1 + \tilde{x}_3 \le 1 & \Rightarrow & x_1 \le x_3 \\ x_2 + \tilde{x}_3 \le 1 & \Rightarrow & \boxed{x_2 \le x_3} \\ x_4 + \tilde{x}_3 \le 1 & \Rightarrow & x_4 \le x_3 \\ x_5 + \tilde{x}_3 \le 1 & \Rightarrow & x_5 \le x_3 \\ x_2 + x_5 \le 1 & \Rightarrow & x_2 + x_5 \le 1 \end{array}\right.$$

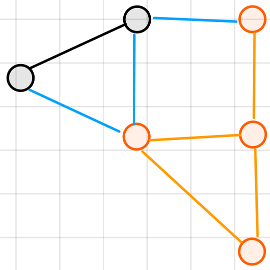
(4) $+2\tilde{x}_1 + 6\tilde{x}_2 + 3x_3 + 2\tilde{x}_4 + 2\tilde{x}_5 \le 8 \Rightarrow \{ \tilde{x}_2 + x_3 \le 1 \Rightarrow \boxed{x_3 \le x_2}$

(5) $2x_2 + 2\tilde{x}_3 + \tilde{x}_4 + x_5 \le 1 \quad \begin{cases} \boxed{x_2 = 0} \\ \tilde{x}_3 = 0 \Rightarrow \boxed{x_3 = 1} \end{cases}$

⇒ INFEASIBLE (perché?) $1 = x_3 \le x_2 = 0$ ⊥

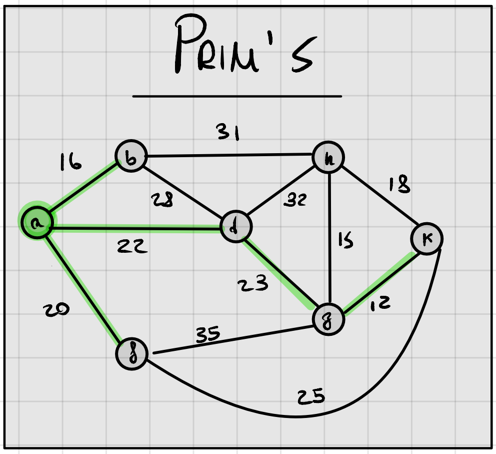
MINIMUM SPANNING TREE

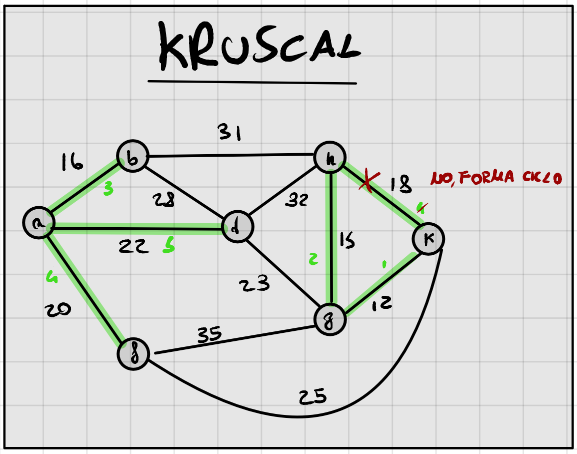
- S, $E(S)$, $\delta(S)$



- Prim's Algorithm
- Kruscal Algorithm

} Poly Algos per MST



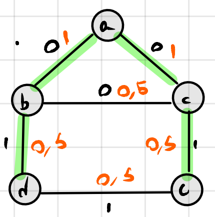


MST ILP

- $x_e = \begin{cases} 1 & \text{se } e \text{ è selezionato} \\ 0 & \text{altrimenti} \end{cases}$

$\min \sum_{e\in E} c_e x_e$	$\min \sum_{e\in E} c_e x_e$
$\sum_{e\in E} x_e = \lvert V\rvert - 1$ // CARDINALITY CONSTRAINT	$\sum_{e\in E} x_e = \lvert V\rvert - 1$ // CARDINALITY CONSTRAINT
$\sum_{e\in E(S)} x_e \le \lvert S\rvert - 1 \quad \forall S\subset V \wedge 2 \le \lvert S\rvert \le \lvert V\rvert - 1$ //impedisce la formazione di cicli	$\sum_{e\in \delta(S)} x_e \ge 1 \quad \forall S\subset V,\ S\neq\emptyset$ //subtour elimination constr.

- Il # CONSTRAINTS sono esponenziale, dobbiamo ridurre il numero per renderlo polinomiale
- Una formulazione è COMPACT se #CONSTRs e #VARs polinomiale ⇒ Quindi entrambe le formulazioni sono NON-COMPACT



- Abbiamo una sol ottima del valore $z^* = 1+1 = 2$
- Rilassando otteniamo $z^{CUT} = 1{,}5$

→ La seconda formulazione è meglio della prima perché: ANCHÉ SE RILASSATA RITORNA SEMPRE UNA SOLUZIONE INTERA

XCASA → · Genera nx graph con 10 nodi
· Genera MST instance
· Calcola MST con la formulazione data a lezione

DELAYED CONSTRAINT GENERATION

ALGORITMO

(1) Inizializza una formulazione P con un subset vuoto di subtour elimination constraints, e cardinality constraint

$$P = \{\max cx : \sum_{e \in E} x_e = |V|-1, \ 0 \le x \le |V|-1\}$$

(2) Risolvi il MASTER LP e sia x^*_{LP} la sol ottima

(3) IF : x^*_{LP} soddisfa tutte le SEC INEQUALITIES ⇒ FINE
subtour elimination constraints

ELSE : Trova un insieme di nodi S t/c $\sum_{e \in E(S)} x^*_e > |S|-1$

(4) $\sum_{e \in E(S)} x_e \le |S|-1$ è aggiunto al master

(5) Torna allo step (2)

SEPARATION PROBLEM

Data una soluzione x^*_{LP} al RILASSAMENTO LINEARE di un problema P, esiste un constraint del tipo

$$\sum_{e \in E(S)} x_e \le |S|-1 \ \Big| \ \sum_{e \in E(S)} x^*_e > |S|-1 \ ?$$

Cerchiamo di formalizzarlo in un prob di combinatoria

IDEA

· Introduciamo $z_j = \begin{cases} 1 & \text{se } j \in S \\ 0 \end{cases}$ per $j = \{1, \ldots, |V|\}$

· Un SEC è violato da x^*_{LP}, se esiste un sottoinsieme di nodi S t/c

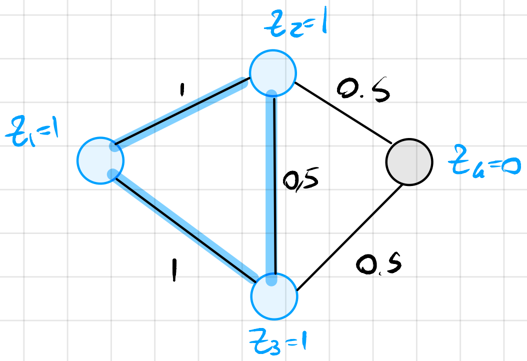
$$\sum_{e \in E(S)} x^*_e > |S|-1$$

· Allora $\boxed{\sum_{e \in E(S)} x^*_e - |S| > -1}$

• Perciò risolviamo $\max_{S \subseteq V} \{ \sum_{e \in E(S)} x_e - |S| \}$ e controlliamo il valore della soluzione:

se $\begin{cases} > -1 \Rightarrow \text{SEC FOUND} \\ \leq -1 \Rightarrow \text{SEC NON ESISTE} \end{cases}$

$$\Rightarrow \boxed{\max_{z_j \in \{0,1\}} \{ \sum_{e=(i,j) \in E} x_e^* \cdot z_i z_j - \sum_{s \in V} z_s \} > -1}$$



$\max_{z_j \in \{0,1\}} \{ \sum_{e=(i,j) \in E} x_e^* \cdot z_i z_j - \sum_{s \in V} z_s \} > -1$

$\Rightarrow 2,5 - 3 > -1$ ✗ NO

• Notiamo che la soluzione che massimizza la formula è $z_s = 0$, però è una soluzione banale perciò:

ALMENO UN z_k VA SETTATO 1

Notiamo anche che il problema è QUADRATICO causa di $z_i z_s$, va LINEARIZZATO trasformando il '$\times$' in un 'AND' creando $\omega_{ij} = z_i \cdot z_j$ e poi vanno aggiunti i constraint per gli AND:

$$\boxed{\begin{aligned} &\max \sum_{e=(i,j) \in E} x_e^* \omega_{ij} - \sum_{j \in V} z_j \\ &\left.\begin{aligned} \omega_{ij} &\leq z_i \\ \omega_{ij} &\leq z_j \end{aligned}\right\} \omega_{ij} \leq \min\{z_i, z_j\} \\ &\boxed{\cancel{\omega_{ij} \geq z_i + z_j - 1}} \rightarrow \text{RIDONDANTE} \\ &z_k = 1 \quad z \in \{0,1\}^{|V|} \\ &\left.\cancel{\omega_{ij} \in \{0,1\}^{|E|}}\right\} \begin{aligned} 0 \leq z \leq 1 \\ 0 \leq \omega \leq 1 \end{aligned} \end{aligned}}$$

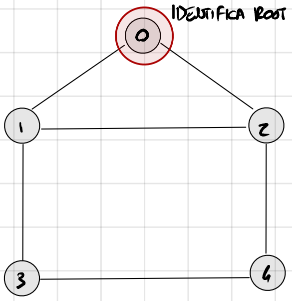
facciamo qualche UPDATE:

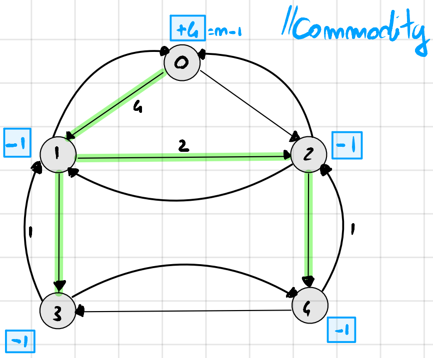
$\Rightarrow$ In definitiva il problema diventa:

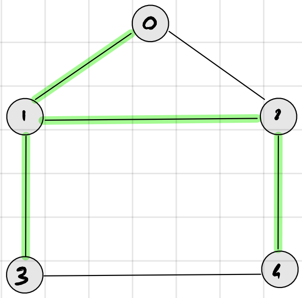
$$\boxed{\begin{aligned} &\max \sum_{e=(i,j) \in E} x_e^* \omega_{ij} - \sum_{j \in V} z_j \\ &\omega_{ij} \leq \min\{z_i, z_j\} \\ &z_k = 1 \quad z \in \{0,1\}^{|V|} \\ &0 \leq z \leq 1 \\ &0 \leq \omega \leq 1 \end{aligned}}$$

MST COMPACT FORMULATIONS

① SINGLE COMMODITY FLOW FORMULATION







- Rendiamo il grafo bidirezionale e impostiamo i LOWER BOUND
- I nodi che tocchiamo mangiano flow
  - Se troviamo, con un flow di $n-1$ uscente dalla root, un max flow, abbiamo trovato un MST sul grafo di partenza

- $x_e = \begin{cases} 1 & \text{se } e \text{ è nel TREE} \\ 0 \end{cases}$

- f_{ij} = limite di flow su (i,j)

- $\min \sum_{e \in E} c_e x_e$

$$\sum_{j \in \delta^+(0)} f_{0j} - \sum_{i \in \delta^-(0)} f_{i0} = n-1$$ // flow balance su root

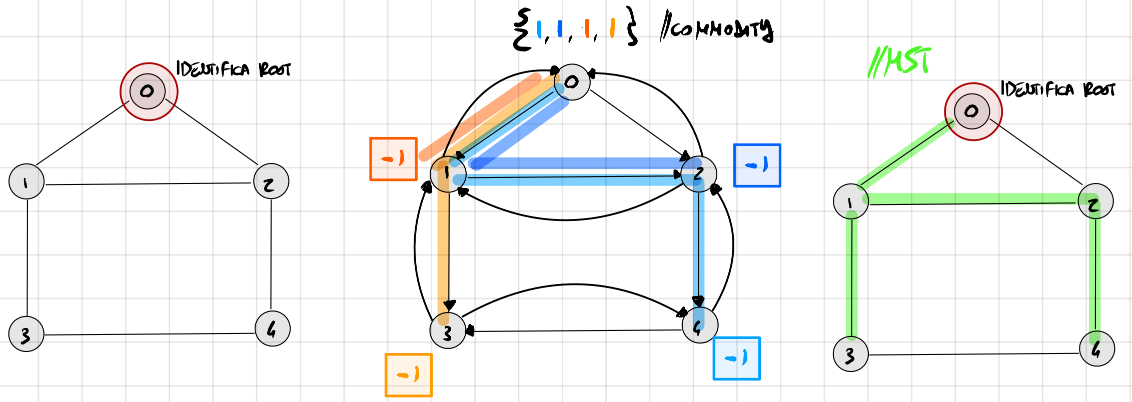
$$\sum_{i \in \delta^-(v)} f_{iv} - \sum_{i \in \delta^+(v)} f_{v} = 1 \quad \forall v \in V \mid v \neq \{0\}$$ // ciò che esce meno ciò che entra è 1; sono $n-1$ constraints

$$\left.\begin{aligned} f_{ij} &\leq (n-1) x_e \quad \forall e \in E \\ f_{ji} &\leq (n-1) x_e \quad \forall e \in E \end{aligned}\right\}$$

// $x_e \geq f_{ij} / M$, $M \to n-1$, $x_e \in \{0,1\}$
// $f_{ij} > 0 \Rightarrow x_e > \varepsilon \Rightarrow x_e = 1$
// $f_{ij} = 0 \Rightarrow x_e = 0$

$$\sum_{e \in E} x_e = n-1$$

② MULTI COMMODITY FLOW FORMULATION



- $X_e \begin{cases} 1, & \text{se } e \text{ è in Tree} \\ 0 \end{cases}$

- f_{ij}^k = Flow di commodity k portato da (i,j), $k = 1, \dots, m$

- y_{ij} = capacity per il flow di ogni commodity k nell'arco (i,j)

- min $\sum_{e \in E} c_e x_e$

$$\sum_{j \in \delta^+(0)} f_{0j}^k - \sum_{i \in \delta^-(0)} f_{i0}^k = 1 \qquad \forall k \neq \{0\}$$

$$\sum_{i \in \delta^-(v)} f_{iv}^k - \sum_{j \in \delta^+(v)} f_{vj}^k = 0 \qquad \forall k \neq \{0\}, \forall v \in V \mid v \neq \{0\}, v \neq k$$

//BALANCE
↳ è per i nodi intermedi, perché k è il terminal node

$$\sum_{i \in \delta^-(v)} f_{ik}^k - \sum_{j \in \delta^+(v)} f_{kj}^k = 1 \qquad \forall k \neq \{0\}$$

//i nodi terminali devono avere 1 flow entrante e non uscente

$$f_{ij}^k \leq y_{ij} \qquad \forall (i,j), k \neq \{0\}$$

$$y_{ij} + y_{ji} = x_e$$

$$\sum_{\{i,j\} \in E} (y_{ij} + y_{ji}) = n - 1$$

$$x_e \in \{0,1\}, \; y_{ij} \in \{0,1\}, \; f_{ij} \geq 0$$

STAINER TREE

- INPUT

  $G = (V, E)$ simmetrico
  costo $c_e \ \forall e \in E$
  insieme di nodi TERMINALI $T \subseteq V$
  insieme di nodi di STAINER $S = V / T$

- OUTPUT
  - Trova un SUBTREE di costo minimo che tocca tutti i nodi terminali

PRIZE COLLECTING STAINER TREE

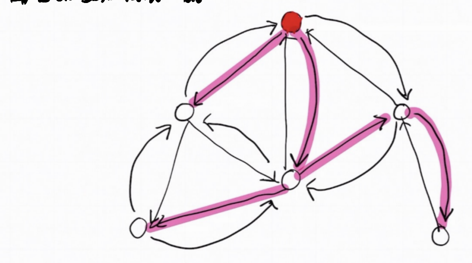
- INPUT

  $G = (V, E)$ simmetrico
  costo $c_e \ \forall e \in E$
  root node $\{o\}$
  REVENUE $p_j > 0 \ \forall j \in \{V \setminus \{o\}\}$

- OUTPUT

  Trova SUBTREE T rooted in $\{o\}$ che massimizzi la somma delle REVENUE dei nodi in T meno la somma dei costi degli archi in T

ARBORESCENCE
↳ ∃ un solo Path tra due nodi



- Consideriamo $B = (V, E)$ grafo BIDIREZIONALE ottenuto da G ($\circ \overset{c}{-} \circ \Rightarrow \circ \overset{B}{\rightleftarrows} \circ$)

$\Rightarrow$ PCST consiste nel trovare un ARBORESCENCE OTTIMA in B rooted in $\{o\}$

- VARS

$$x_{ij} \begin{cases} 1 & \text{se } (i,j) \in \text{ARBORESCENCE} \\ 0 \end{cases}$$

$$y_i \begin{cases} 1 & \text{se } i \in \text{ARBORESCENCE} \\ 0 \end{cases}$$

- OBJ

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} - \sum_{j \in V \setminus \{o\}} p_j y_j$$

- 1^a FORMULAZIONE

$$y_o = 1$$

$$\sum_{i \in \delta^-(j)} x_{ij} = y_j \quad \forall j \in V \setminus \{o\}$$

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \leq y_k \quad \forall S \subset V,\ o \in S,\ k \in V \setminus S$$

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

$$y_i \in \{0,1\} \ \forall i \in V$$

- 2^a FORMULAZIONE
  - u_i Distanza di i da $\{o\}$ nell'ARBORESCENCE

$$\sum_{i \in \delta^-(j)} x_{ij} = y_{j} \quad \forall j \in V \setminus \{o\}$$

$$(n+1) x_{ij} + u_i - u_j \leq n \quad \forall (i,j) \in A$$

$$x_{jk} \leq y_j \quad \forall j \in V \setminus \{o\}, \forall k \in \delta^+(j)$$

$$0 \leq u_j \leq n \quad \forall j \in V$$

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

$$y_i \in \{0,1\} \ \forall i \in V$$

Se l'arco (j,k) è stato preso, anche il nodo j lo è stato

Presi due nodi dell'arborescenza i e j con i più vicino a $\{o\}$ è ovvio che $u_i - u_j < 0$ quindi $(n+1) + u_i - u_j \leq n$

ATSP

ATSP NOTEBOOK

$\boxed{\text{ATSP model con FS e RS}}$ → Opzionale: Qui possiamo aggiungere tutti i SEC di cardinalità 2 (BOOST PERFORMANCE)

↓

~~CUTTING PLANE method~~ Da un LP rilassato lo risolviamo e vediamo se dei SEC sono violati, ripetiamo finché ci sono

→ non trova OPT*

↓

$\boxed{\text{BRANCH AND CUT}}$ Embedding del cutting plane in un ENUMERATION FRAMEWORK

↳ per farlo creiamo una CALLBACK e la passiamo a 'model.optimize(CALLBACK)'

↳ La CALLBACK aggiunge "LAZY CUTS"

(1) se x^* è feasible con il SEPARATION PROBLEM

DFS:

- INIZIALIZZO IL MODELLO CON FS, RS e SEC di cardinality 2
- DEFINISCI Callback [lazy cuts] con il SEP. ALG.
- Optimize (callback)

POLYNOMIAL (#CONSTRS/#VARS) TSP FORMULATION

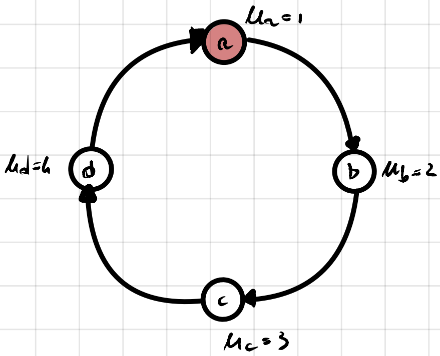
MTZ formulation

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

(FS) $\sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i\in N$

(RS) $\sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall j\in N$

$u_i - u_j + 1 \le (n-1)(1-x_{ij}) \quad \forall (i,j)\in A,\ i\neq 1, j\neq 1$



DESROCHER & LAPORTE formulation

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij}$$

(FS) $\sum_{j\in\delta^+(i)} x_{ij} = 1 \quad \forall i\in N$

(RS) $\sum_{j\in\delta^-(i)} x_{ji} = 1 \quad \forall j\in N$

$u_i - u_j + (n-1)x_{ij} + (n-3)x_{ji} \le n-2 \quad \forall (i,j)\in A,\ i\neq 1, j\neq 1$

- Dato un problema:

$$\boxed{\begin{array}{ll} \min & cx \\ & Ax \geq b \\ & x \geq 0 \end{array}}$$

← A ha CONSECUTIVE 1 PROPERTY

- Lo trasformiamo nella forma Standard $Ax \geq b \Rightarrow Ax - Iy = b$

- Aggiungo una riga di $\emptyset$
- Sottraggo ogni riga con quella sopra e risolvo il problema

$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} x_1 & x_2 & x_3 & x_4 & x_5 & y_1 & y_2 & y_3 & y_4 \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 1 & 1 & -1 & 0 & 0 & 0 \\
1 & 0 & 0 & -1 & 0 & 1 & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & -1 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\
-1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 4 \\ -7 \\ -7 \end{bmatrix}
$$

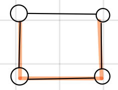
NETWORK SIMPLEX ALGORITHM

$$\min \sum_{(i,j)\in A} c_{ij}\, x_{ij}$$

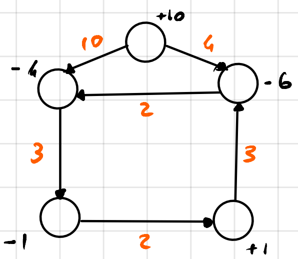
s.t.

$$\sum_{(i,j)\in \delta^+(i)} x_{ij} - \sum_{(j,i)\in \delta^-(i)} x_{ji} = b(i) \qquad i \in N$$

$$x_{ij}$$

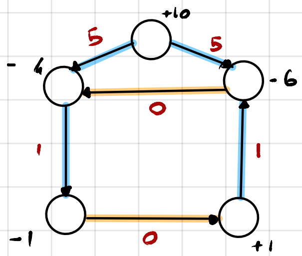
- se G è connesso ⇒ G ammette uno spanning tree T ()
- Una feasible solution x per P è detta TREE SOLUTION, se dato uno spanning tree $T \in T_G$:

$$\begin{cases} \sum_{(i,j)\in \delta^+(i)} x_{ij} - \sum_{(j,i)\in \delta^-(i)} x_{ji} = b(i) & i \in N \\ \\ x_{ij} = 0 \quad \forall (i,j) \notin T & \text{//tutte le vars fuori da } T \text{ sono } \emptyset \end{cases}$$



c_{ij}

$u_{ij} = \infty$



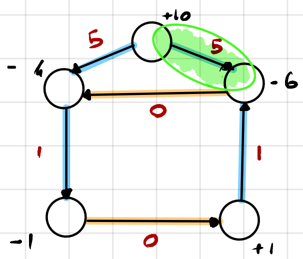
feasib. x_{ij}

Arcs | $x_{ij} = 0$

Arcs | $x_{ij} > 0$ ⇒ T

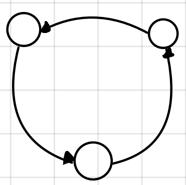
⇒ We have a TREE SOLUTION T

⇒ The value of the solution is $10 \cdot 5 + 4 \cdot 5 + 3 \cdot 1 + 3 \cdot 1 = 76$



NOTA che preso un arco poi partizionare il grafo in modo che il flow che va da PART$_1$ a PART$_2$ è portato da esattamente un ARCO

LEMMA Dato un tree T, esiste sempre un unica TREE SOLUTION associata a T



FTS → x è determinata univocamente

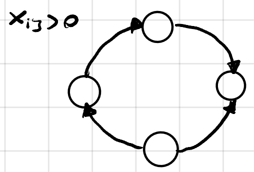
X → porta a più FST

TH se P ammette una feas. solution allora ammette una FTSolution

se P " " optimal solution " " un OTTIMA FTSolution

<u>Nota</u> Possiamo determinare un algoritmo che si sposta solo tra FTSolution fino a trovare l'ottimo

PROOF Sia X feas. per P ma non FTSol.

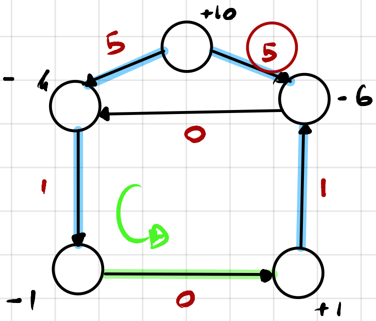


$\varepsilon = \min x_{ij}$ tra i REVERSE ARCS

$x'_{ij} = x_{ij} - \varepsilon$ se (ij) REVERSE

$x'_{ij} = x_{ij} + \varepsilon$ se (ij) FORWARD

MUOVERSI TRA FTS

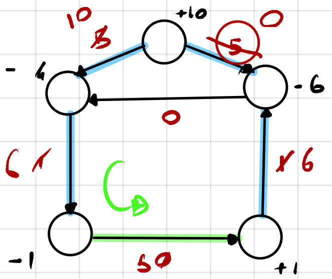


- Partendo da FTSol
- Come ci spostiamo a un altra FTSol
- Aggiungo un arco a FTS

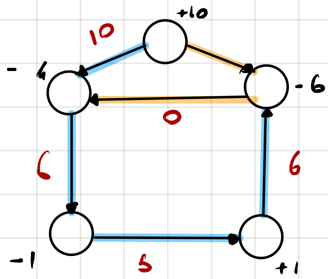
↳ ora ho un ciclo, e l'arco aggiunto è orientato C

↳ tra ogni arco nel ciclo scelgo quello orientato verso ∂ con max flow

↳ incremento il flow di ◯ agli archi verso 'C' e lo sottraggo a quelli verso '∂'



PIVOTING →

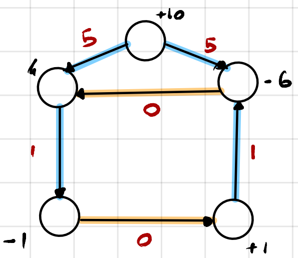


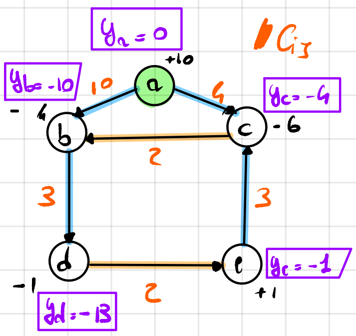
T Novo FTSol

L archi con $x_{ij} \neq 0 \mid (ij) \notin T$

- Come checkiamo l'ottimalità?

- <u>Ric.</u> REDUCED COST: $\bar{C}_{ij} = C_{ij} - y_i + y_j$





- Presa FTSol
- Scegliamo ROOT
- troviamo $y_h = -\sum_{(i,j)\,forward} C_{ij} + \sum_{(i,j)\,Reverse} C_{ij}$ sul path $r \to h$ su T