

NETWORK OPTIMIZATION

II^º 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. WATSON, 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

ILP VS MIXED ILP

- MIXED ILP HAS INTEGER AND NON INTEGER VARIABLES

$$\max C^T x + h^T y$$

s.t.

$$Ax + Gy \leq b$$

$$x \geq 0$$

$$y \geq 0,$$

DECISION VARS

OBJECTIVE FUNCT.

LINEAR CONSTS

RESTRICTIONS

INTEGRITY REQ

y INTEGER

Integer Linear Program

DATA

A : $m \times m$ RATIONAL MATRIX

c : m -DIMENSIONAL RATIONAL VECTOR

b : m -DIMENSIONAL "

VARs

x : m -DIMENSIONAL INTEGER VAR. VECTOR

ILP

$$\max c'x$$

$$Ax \leq b$$

$$x \geq 0, \text{ integer}$$

SOL.

Assegnamento di valori alle vars.

FEASIBLE SOL.

Sol. che soddisfà i constraint

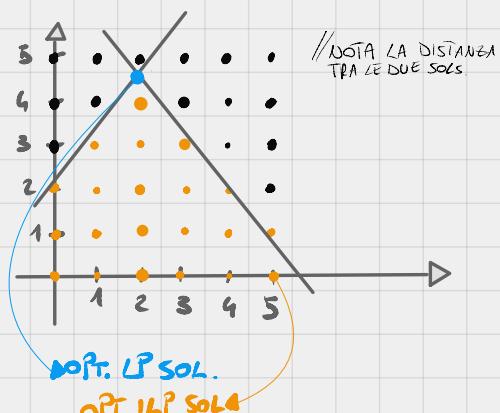
OPT. SOL.

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

ES.

$$\begin{aligned} \max \quad & 1x_1 + 0.4x_2 \\ \text{s.t.} \quad & 50x_1 + 31x_2 \leq 250 \\ & -3x_1 + 2x_2 \leq 4 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \text{ integer} \end{aligned}$$

LINERAR
RELAXATION



Binary Integer Program

DATA

A : $m \times m$ RATIONAL MATRIX

c : M-DIMENSIONAL RATIONAL VECTOR

b : m-DIMENSIONAL //

VARs

x : m-DIMENSIONAL INTEGER VAR-VECTOR

BIP

$$\max c'x$$

$$Ax \leq b$$

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

Combinatorial Optimization Problem

Given...

$$\cdot N = \{1, \dots, m\}$$

$\cdot c = \{c_1, \dots, c_m\}$ weights vector

$\cdot \mathcal{F}$ famiglia di feasible sols. $S \subseteq N$

... find

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

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

$\cdot S$ è rappresentato da $1'$ INCLENCE VECTOR $x = (x_1, x_2, \dots, x_n)$

T.C.

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

\hookrightarrow Un problema di ottimizzazione combinatoria si affronta
enumerando le feasible sols. e prendendo la migliore

KNAPSACK

given... $N = \{1, \dots, n\}$ Items

- a_j weight, c_j profit per item
- b capacity

... find

- sottosinsieme di item la cui somma dei pesi non ecceda b , capacità e che maximizzi la somma dei profitti

ES
(KNAPSACK)

	1	2	3	4	$\} \quad \begin{array}{l} \textcircled{1} \text{ costriamo } F \\ F = \{\sum_{j=1}^2 a_j x_j, \dots, \sum_{j=1}^4 a_j x_j\} \\ \{1, 2\}, \{1, 3\}, \dots \end{array}$
a_i	2	3	4	1	
c_i	10	14	12	8	
$b = 5$					$\textcircled{2} \text{ OPT } \in \{1, 2\}$

↳ Come lo formuliamo come BIP?

① Sceglia vars
vars def.

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

② trou consts.

CONSTRAINTS DEF.

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

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

③ Scegli obb func.

OBS. FUNCT.

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

16/03

Set Covering, Set Packing, Set Partitioning

given...

- $M = \{1, \dots, m\}$, $N = \{1, \dots, n\}$ INSIEMI FINITI
- $\{M_1, \dots, M_m\}$ COLLEZIONE DI SOTTOSIEMI DI M
- $C_S \notin M_S$

Note: F è un insieme di M_S

...

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

Esempio

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

$$M_1 = \{1, 2, 3, 4\}$$

$$M_2 = \{1, 2, 5, 6, 9, 10\}$$

$$M_3 = \{7, 8, 11, 12\}$$

$$M_4 = \{5, 6\}$$

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

$$M_6 = \{9, 10, 11, 12\}$$

$$M_7 = \{7, 8\}$$

1	2	3	4
5	6	7	8
9	10	11	12

1	2	3	4
5	6	7	8
9	10	11	12

1	2	3	4
5	6	7	8
9	10	11	12

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 sottinsiemi che prende ogni elemento. Infatti possono esserci elementi in comune tra i sottinsiemi.
- Un **PACKING** invece è un insieme di sottinsiemi 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 $m \times m$** $a_{ij} = \begin{cases} 1 & \text{if } i \in M_j \\ 0 & \text{otherwise} \end{cases}$

Esempio

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

$$M_1 = \{1, 2, 3, 4\}$$

$$M_2 = \{1, 2, 5, 6, 9, 10\}$$

$$M_3 = \{7, 8, 11, 12\}$$

$$M_4 = \{5, 6\}$$

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

$$M_6 = \{9, 10, 11, 12\}$$

$$M_7 = \{7, 8\}$$

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

MODELLORE CONDIZIONI LOGICHE (con constraint lineari e vars binarie)

- Siamo x, y, z variabili binarie

NOT

$$z = \neg x \rightarrow z = 1 - x$$

AND

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

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

XOR

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

covered by ① or ②

" " "(② or ③) or ④

↳ ④ Take via la sol. $x=y=z=1$

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

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

$$\begin{cases} x - y \leq z \\ y - x \leq z \\ x + y \geq z \\ x + y + z \leq 2 \end{cases}$$

MODELLORE DEPENDENT DECISIONS (con constraint lineari e vars binarie)

IDEA

- Possiamo usare BN.VARS e LINEAR CONST. per rappresentare le DIPENDENZIE DUE SEZIE

Suppose x and y are binary variables such that:

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

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

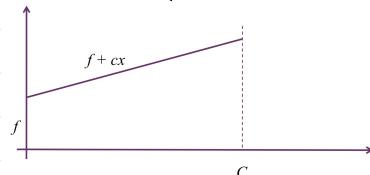
This can be expressed by the (linear) constraint

$$x - y \leq 0$$

MODELLORE FIXED COST

- Vogliamo minimizzare i fixed cost se un servizio è attivato

$$\min h(x) = \begin{cases} f + cx & \text{if } 0 < x \leq C \\ 0 & \text{if } x = 0 \end{cases} \quad \left\{ \begin{array}{l} \text{funzione non lineare} \\ \text{fatturazione manuale} \end{array} \right.$$



1- Aggiungiamo una var. bin. extra $y = "19 \text{ servizio è attivato?}"$

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

$$\Rightarrow \min y + Cx$$

$$x - Cy \leq 0 \rightarrow \text{Se } y=0 \text{ non ha senso rigore, quindi } \Rightarrow x \leq 0$$

$$\rightarrow \text{Se } y=1 \Rightarrow 0 \leq x \leq C$$

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

$$x \geq 0$$

\hookrightarrow si mette che se $y=1$ il minimo x è $x=0$, questa feasible solution va in contrasto com l'informazione di y , ma una OPT. SOL. avrà sempre $y=0$ se $x=0$