NETWORK FLOWS

II° SEMESTRE

2021

9/03

INTRODUCTION

NETWORK FLOWS
↓
APPROX. ALGOS
↓
Tool: GUROBI 9.1
Book: NETWORK FLOWS

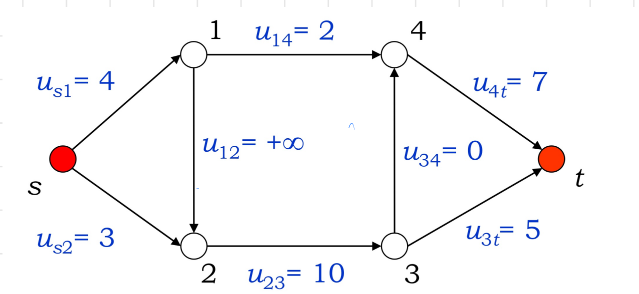
NETWORK OPTIMIZATION
↓
NP-HARD PROBLEMS
↓
Tool: NetworkX (Py. LIBRARY)
Book: INTEGER PROGRAMMING (L. WOLSEY, EDITION 1)

MIDTERMS
- 1- END MARCH
- 2- END APRIL
- 3- END MAY

NETWORK DESIGN

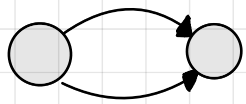
DIRECTED GRAPH · $G(N, A)$ con due nodi speciali s (SOURCE) e t (SINK)

CAPACITY · $u_{ij} \in [0, +\infty)$ è la capacità associata a $(i,j) \in A$



ASSUNZIONI

(1) G non ha un PATH DIRETTO (da s a t) di soli archi con $u_{ij} = \infty$

(2) G non ha ARCHI PARALLELI 

(3) Posso aggiungere archi con $u_{ij} = 0$ come mi pare

PATH PACKING PROBLEM

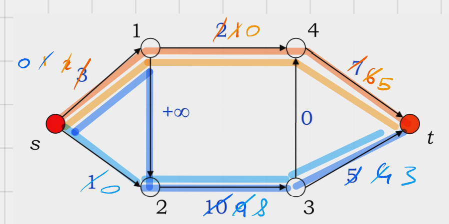
given... · $G(N, A)$, un capacity vector $u_{ij} \in \mathbb{Z}^{|A|}$

...find · $P = \{P_1, P_2, \_, P_k\}$ famiglia di SIMPLE DIPATH t/c

(1) ogni $(i,j) \in A$ fa parte di al più u_{ij} DIPATH

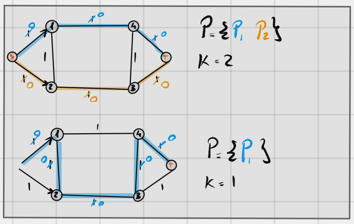
(2) k sia MASSIMIZZATO

ES



$\# P_1 = (s, 1, 4, t)$
$\# P_2 =$ "
$\# P_3 = (s, 1, 2, 3, t)$
" $P_4 = (s, 2, 3, t)$

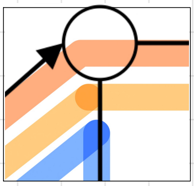
- Possiamo certificare l'ottimalità della soluzione?
  ↳ No, il greedy non è detto funzioni.



ILP (Flow Formulation)

VARs.
- X_{ij} occorrenze di (i,j) ed nei path in P

✱ (Trivial) OBS
- $\forall v \neq s, t \Rightarrow$ il #Path entranti è lo stesso di quelli uscenti



NOTA: Ignoriamo archi entranti in s

$$\max \sum_{j:(s,j)\in A} x_{sj} - \cancel{\sum_{j:(j,s)\in A} x_{js}}$$

s.t.

✱ BALANCE CONSTRAINT
$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0 \quad \forall i \in N \setminus \{s,t\}$$

CAPACITY CONSTRAINT
$$0 \le x_{ij} \le u_{ij}, \forall (i,j) \in A$$

INTEGRALITY REQ
$$x_{ij} \text{ Integer}, \quad \forall (i,j) \in A$$

(s,t)-FLOW
- Vettore X che soddisfa il BALANCE CONSTR.

FEASIBLE FLOW
- FLOW che soddisfa il CAPACITY CONSTR

NET FLOW IN v
- È il termine: $f_x(v) = \sum_{j:(v,j)\in A} x_{vj} - \sum_{j:(j,v)\in A} x_{jv}$

FLOW VALUE IN G
- $\delta_x(v)$

---

i Problemi sono equivalenti?

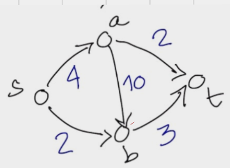
- Un FLOW ci dice solo le occorrenze di ogni arco, ma non ci mostra i PATH.

(trivial) ⇒
- Dati i path $P = \{P_1 \_ P_k\}$ posso facilmente costruire la soluzione al problema FF

(to proof) ⇐
- Il contrario è vero? → Dimostrandolo proviamo l'equivalenza tra i due problemi (PATH PACKING e FF)

12/03

(⇒) es



$P_1 = \{s,a,t\}$
$P_2 = \{s,a,t\}$
$P_3 = \{s,a,b,t\}$
$P_4 = \{s,b,t\}$

$k = 4$

Feasible solution to the path packing problem

⇒ Contruct FF solution:

$X_{sa} = 3$ $X_{at} = 2$ $X_{ab} = 0$
$X_{sb} = 1$ $X_{bt} = 2$

· Puoi controllare che soddisfa BALANCE e CAPACITY CONSTR.

Demostriamo (⇐)

DECOMPOSITION THEOREM

Decomposition theorem

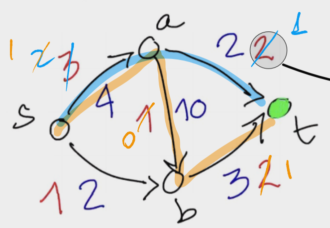
Given a graph $G=(N,A)$, vector of capacities $u \in \mathbb{Z}_+^{|A|}$ there exists a family $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ of k (s,t) paths such that each arc $(i,j) \in A$ is used at most u_{ij} times by the paths in $\mathcal{P}$, if and only if there exists an INTEGRAL FEASIBLE (s,t) flow of value k

↳ Proof

es.

Suppose you have a feasible solution to FF. Is it always possible to build a family of k (s,t)-paths?

$(k=4)$



(1) scegliamo un arco con $X_{ij} \geq 1$

Dato che $X_{at} \geq 1$, dal Balance constraint sappiamo che c'è almeno un arco entrante

(2) Prendiamo un arco (i,a) con $X_{ia} \geq 1$

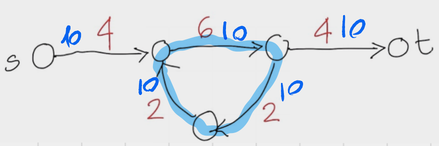
In questo caso prendiamo (s,a) e abbiamo il primo path

$P_1 = \{s,a,t\}$

$P_2 = \{s,a,b,t\}$

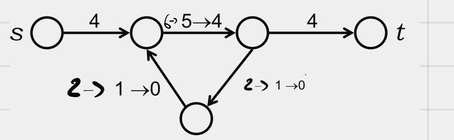
es.

Ritorna CYCLIC FEAS. FLOW



⇒

RETURN ACYCLIC FEAS. FLOW



• È possibile, dato X, FEAS FLOW, possiamo trasformarlo in una aciclica?

CYCLIC TO ACYCLIC

- Trasformiamo x feasible flow in x' feas e acyclic flow
  - (1) Identifichiamo il ciclo
  - (2) sottraiamo 1 a ogni x_{ij} sul ciclo finché almeno un x_{ij} non diventa $\emptyset$

ALGO

- (1) TRASFORMA x feas flow in x' aciclico
- (2) Partendo da t:
  - selezioniamo un arco con $x_{it} > 0$
  - continuiamo finché s non è nel path
- (3) Ripeti (2) finché puoi

CUT OF GRAPH

- Dato $G=(N,A)$, un TAGLIO è un insieme di archi:

$$\delta(R) = \{(v,w) \mid (v,w) \in A,\ v \in R,\ w \in \bar{R}\},\ R \subseteq N$$

NOTA

- IN un grafo diretto
  - (i,j) t/c $i \in R$ e $j \in \bar{R}$ è in $\delta(R)$
  - " $i \in \bar{R}$ e $j \in R$ è in $\delta(\bar{R})$

CAPACITY OF CUT

- $u(\delta(R)) = \sum_{(i,j) \in \delta(R)} u_{ij}$

Theorem WEAK DUALITY

- Per ogni (s,t)-cut $\delta(R)$ e ogni feas. flow x allora

$$x(\delta(R)) - x(\delta(\bar{R})) = f_x(s)$$

↳ PROOF

- Preso R, insieme che definisce un (s,t)-CUT
- Considera TUTTI i BALANCE CONSTRAINT associati a $v \in R, v \neq s$

$$\boxed{\underbrace{\sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ji}}_{LHS} = \underbrace{0}_{RHS}}$$

- Dimostreremo il teorema dividendo gli archi in insiemi

6

ARCHI INTERNI

(1) $\forall (v,w) \mid v,w \in R$ e $v \neq s \Rightarrow x_{vw}$ NON appare in LHS del BAL. CON

(2) $\forall (v,w) \mid v,w \notin R \Rightarrow x_{vw}$ NON appare in LHS

ARCHI $R \to \bar{R}$ e $\bar{R} \to R$

(3) Per ogni $(v,w) \mid v \in R, w \notin R \Rightarrow x_{vw}$ appare in LHS con coefficiente +1

(4) Per ogni $(v,w) \mid v \notin R, w \in R \Rightarrow x_{vw}$ appare in LHS con coefficiente -1

ARCHI DA S e IN S

(5) Per ogni $(s,v) \mid v \in R \Rightarrow x_{sv}$ appare con coefficiente -1

(6) Per ogni $(v,s) \mid v \in R \Rightarrow x_{sv}$ appare con coefficiente -1

- RAGGRUPPANDO le vars che soddisfano (3),(4) otteniamo:

$$x(\delta(R)) - x(\delta(\bar{R}))$$

- RAGGRUPPANDO " " (5),(6) invece

$$-f_x(s)$$

$\Rightarrow$ In conclusione quindi:

$$\text{LHS} = x(\delta(R)) - x(\delta(\bar{R})) - f_x(s)$$

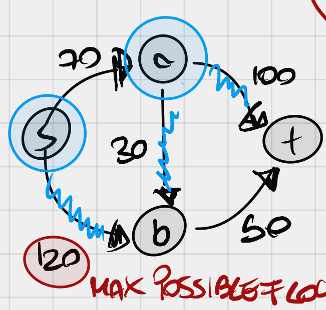
Corollary

- $\forall (s,t)$-CUT $\delta(R)$ e $\forall (s,t)$-FLOW x FEASIBLE

$$\boxed{f_x(s) \leq u(\delta(R))}$$

ovviamente a interessa il CUT $\delta(R)$ con minima capacità

esempio



120 COINCIDONO 120

- vediamo i tagli e le loro capacità

$R = \{s\} \Rightarrow u(\delta(R)) = 190$

$R = \{s,a\} \Rightarrow u(\delta(R)) = 250$

$R = \{s,a,b\} \Rightarrow$ " $= 150$

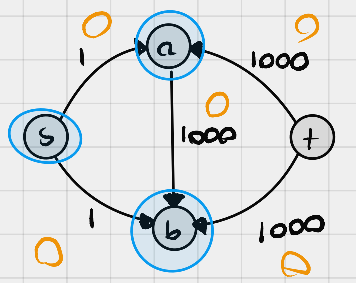
$R = \{s,b\} \Rightarrow$ " $= 120$

LOWER CAPACITY

OPT. CERTIFICATE

• Con questo teorema possiamo trarre un CERTIFICATO DI OTTIMALITÀ, ovvero, grazie alla WEAK DUALITY: una soluzione ottima del MIN-CAPACITY-CUT coincide con una ottima del MAX-FLOW

esempio



MAX (s,t)-FLOW
• sicuro 0 è un flow valid
MIN Capacity (s,t)-cut
• $R=\{s\} \to u(\delta(R))=2$
• $R=\{s,a\} \to$ " $=101$
• $R=\{s,b\} \to$ " $=1$
$R=\{s,a,b\} \to$ " $=0$
NON ARCHI DA R A $\bar{R}$

Proof

• Dal WEAK DUALITY TH. sappiamo che

$$x(\delta(R)) - x(\delta(\bar{R})) = f_x(s)$$

• Dalla definizione

$$u(\delta(R)) \geq x(\delta(R))$$

$$\Rightarrow u(\delta(R)) \geq x(\delta(R)) - \cancel{x(\delta(\bar{R}))} \geq f_x(s)$$

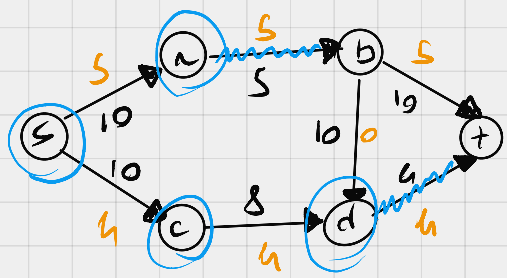
STRONG DUALITY TH.
[FORD e FULKERSON 1956
KOTZIG]

• Se $G=(N,A)$ ammette (s,t) maximum flow allora

$$\boxed{\max\{f_x(s) : x \text{ is a feasible } (s,t)\text{-flow}\} = \min\{u(\delta(R)) : \delta(R) \text{ is an } (s,t)\text{-cut}\}}$$

(ci dice che il max flow ha sempre valore UGUALE al min capacity cut)

esempio



• Troviamo un FLOW AMMISS.
$x = 9$

• Troviamo il min CAP. CUT

$R = \{s,a,c,b\}$

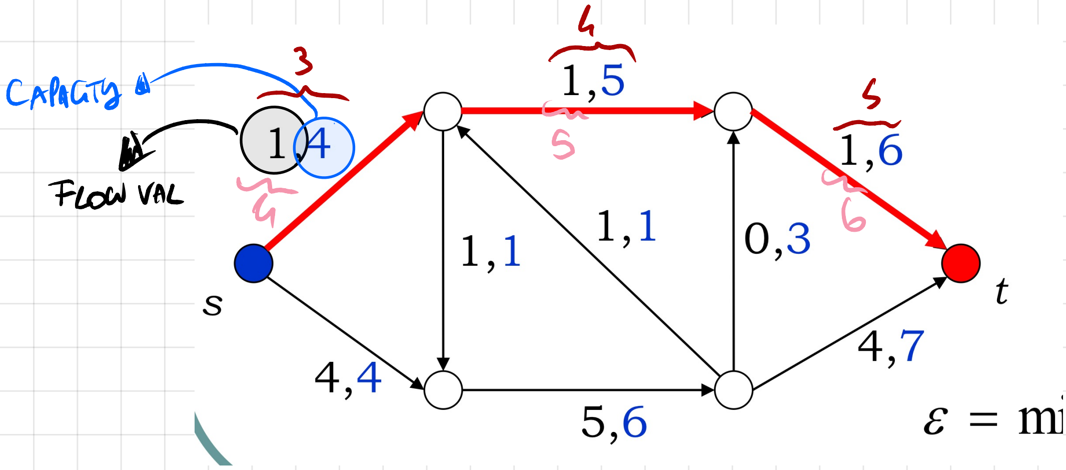
$\delta(R) = \{(a,b),(d,t)\}$

$u(\delta(R)) = 4+5 = 9$

$\Rightarrow 9=9 \Rightarrow$ il flow è ottimo!

↳ Proof

- SE IL FLOW VAL È uguale alla CAPACITY, l'arco È SATURED
- se non è saturated possiamo incrementare il flow?



- SI, NON ROMPIAMO il CAPACITY CONSTRAINT, ma il BALANCE?