19/04/2023

* ANALISI DI SENSITIVITÀ

REMINDER:

concetto di prezzi ombra = soluz ottima del problema duale, misurano il contributo dato alla FO a fronte di Δb variazione dei termini noti

⇒ quanto posso far variare b senza che il contributo dei λ^* non cambi?

⇒ di sicuro finchè che non cambia la soluzione di base
(perché $c^t x = \lambda^t b = c_B^t B^{-1} \cdot b$)
quindi finché rimango sullo stesso VERTICE

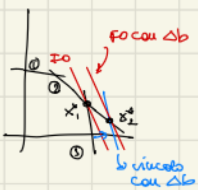
i parametri che potrebbero variare sono c^t e b (A pure ma non lo trattiamo)

* ANALIZZIAMO VARIAZIONI DEI PARAMETRI b CHE MANTENGANO INALTERATA LA SOLUZIONE DI BASE OTTIMALE *
↓
STABILITÀ, ROBUSTEZZA RISPETTO ALLE PERTURBAZIONI DEI DATI

intervalli più grandi ⇒ significa soluzione più robusta

Partiamo con un problema di PL già risolto.

NB. FO rimane ~~inalterata~~ non in termini di variabili, ma in termini di variabili x_B ed x_N (= quali vincoli hanno l'intersezione ottimale)



entrambi sono trovati da ②∩③

Definisco, dato un problema, le condizioni necessarie e sufficienti perché B rimanga la base ottima:

$b \to b + \Delta b$ $\quad c_B^t \to c_B^t + \Delta c_B^t$ $\quad c_N^t \to c_N^t + \Delta c_N^t$

NOTA: varia sempre un solo b_i alla volta

interesse di calcolo di questi intervalli:

- a volte i parametri dipendono dal tempo o altro esterno ⇒ uso stime di valori che sono corrette in intervalli di valori
- se i parametri non variano potrei comunque cercare entro quali limiti può avere senso aumentare la (es:) cap produttiva, capire quanto al max mi conviene spendere per potenziare la produz ⇒

1. da $Ax \geq b$

2. da $\lambda^t A \leq c^t$

Per il teorema sulle condizioni di ottimalità

x^* è sol base ottima se $\exists \lambda^* \in \mathbb{R}^m$ tale che:

1. $x_B^* = B^{-1} b \geq 0$ (x^* è amm. per il primale)

2. $r^t := c^t - c_B^t B^{-1} A \geq 0$ (λ^* è amm. per il duale)

CCR per x_N

3. $\lambda^{*t} = c_B^t B^{-1}$ (ortogonalità)

posso ricondurre l'ortogonalità alla amm. del duale

=> riconduco l'analisi di sensitività allo studio delle variazioni dei dati iniziali per cui (1) e (2) rimangono verificate

*VARIAZIONE DEI TERMINI NOTI $(b + \Delta b)$

Sia $\min \; c^t x$

$Ax \geq b$

$x \geq 0$

dove b è il vettore termini noti e Δb è la sua variazione che cerchiamo

=> condizioni per l'ottimalità di B diventano

1. $B^{-1}(b + \Delta b) \geq 0$ $(x^* \geq 0)$
2. $c^t - c_B^t B^{-1} A \geq 0$ (invariata)

=> quando vario un termine noto, la base rimane ottima se e solo se

$$B^{-1} b \geq -B^{-1} \Delta b$$

m disuguaglianze

m incognite Δb_i $i = 1 \ldots m$

⇓

$P \subseteq \mathbb{R}^m$ con vettori Δb per cui B non cambia

NB. variazioni di b influenzano $\Rightarrow$ coordinate di x^*_B

$\downarrow$

valore ottimo $f(x^*) = z^*$

$$c^t_B B^{-1} b \rightarrow \underbrace{c^t_B B^{-1}}_{\lambda^*} (b + \Delta b)$$

$$\Delta z := (c^t_B B^{-1}) \Delta b =$$
$$= \lambda^{*t} \Delta b =$$
$$= \sum_{i}^{m} \lambda^*_i \Delta b_i$$

prezzi ombra

* le informazioni che cerco per l'analisi di sensitività sono:

x_B	x_N	B	B^{-1}	$B^{-1}b$
c^t_B	c^t_N	$c^t_B B^{-1}$		

dal tableau ottimale

dal tableau iniziale

prezzi ombra λ^*

ccR (slack)

inizio:

$c^t_N - c^t_B B^{-1} N \geq 0$

$-\lambda^{*t} \geq 0$

$\Rightarrow$ calcolo con: $B^{-1}\Delta b \geq -\underline{B^{-1}b}$

var di base nella sol ottima

$$\begin{cases} B^{-1}\Delta b_i \geq -B^{-1} b_i \\ \dots \\ B^{-1}\Delta b_m \geq - \end{cases}$$

calcolo Δb_1 ponendo tutto $\Delta b_i = 0$

NB. $\Delta b = 0$ è una soluzione ammissibile

*VARIAZIONE DEI COSTI DELLE VAR. FUORI BASE

Sia $\min c^t x$
$Ax \geq b$
$x \geq 0$

con Δc_N^t il vettore variazione di c_N^t e definisco:

$\bar{r}^t :=$ CCR prima Δc_N^t

$\tilde{r}^t :=$ CCR post Δc_N^t

$\Rightarrow$ condizioni di ottimalità per B diventano:

1. $B^{-1} b \geq 0$ (invariato)

2. $\tilde{r}^t = [0^t ; (c_N^t + \Delta c_N^t) - c_B^t B^{-1} N] \geq 0^t$ (CCR fuori base)

$\Rightarrow$ quando vario ai CCR di x_N, la base rimane ottima se e solo se

$$\tilde{r}_N^t = c_N^t - c_B^t B^{-1} N + \Delta c_N^t = \bar{r}_N^t + \Delta c_N^t \geq 0$$

$$\Rightarrow \Delta c_N^t \geq -\bar{r}_N^t$$

ottengo n-m disuguaglianze, indipendenti

$\Delta c_j \geq -\bar{r}_j$

per ogni variabile $j \in \Phi_N$

NB_ $\bar{r}_j \geq 0$ è il massimo decremento del costo c_j per cui la base B rimane ottima

$\Rightarrow$ oltre $\bar{r}_j$ otterrei $\tilde{r}_j < 0$ quindi B non sarebbe più ottima ($\exists$ variabili convenienti da pivotare)

unica eccezione: soluz degenere

24/04/2023 FILE: simplesso duale

DEF: chiamo **ALGORITMO DUALE** un qualsiasi algoritmo che iteri a partire da una soluzione ammissibile per il Duale puntando alla condizione di ammissibilità del Primale ($\equiv$ che è l'ottimalità di D)

* SITUAZIONE IN CUI SI USA:

Ho già risolto un problema Primale, ho determinato x^* e ci vengono imposti dei nuovi vincoli $\alpha^T x \leq \alpha_0$

Prima cosa porto in forma standard e aggiungo la slack:

$$a^T x + x_{n+1} = \alpha_0$$

Possono succedere delle cose

x^* rimane ammissibile?

Se $\boxed{\text{Sì}}$ allora x^* rimane anche ottimo. Perché soddisfa tutti i vincoli precedenti ottimalmente + quello nuovo.

(E se portano ad una soluzione di sol ottima multipla?) me ne accorgerei da $x_{n+1} = 0$

Se $\boxed{\text{no}}$ la variabile $x_{n+1} < 0$ $\Rightarrow$ quindi mi accorgo che la soluzione precedente viola il nuovo vincolo.

Devo ri-ottimizzare.

Secondo la teoria della dualità, il vertice corrispettivo di x^* nel Duale sarà ammissibile e quindi posso pensare ad iterare il simplesso da lì. (fino all'ammissibilità (e ottimalità) di P)

x^* rimane ottimo?

Se $\boxed{\text{Sì}}$, ho già finito.

Se $\boxed{\text{No}}$, ricomincio a iterare il simplesso primale.

$\boxed{\text{NB}}$

Simplesso primale vs. Simplesso duale

Simplesso primale:

1. Inizia con una base B **ammissibile per il primale**: $\mathbf{x}_B = B^{-1}\mathbf{b} \geq \mathbf{0}$ all'ottimo P
2. Operazioni di pivoting che mantengono l'ammissibilità per il problema primale.
3. Termine: base B ammissibile anche per il duale.
   $\mathbf{c}^T - \mathbf{c}_B^T B^{-1} A \geq \mathbf{0}$ (coefficienti di costo ridotto ≥ 0). all'ottimo P

Simplesso duale:

1. Inizia con una base B **ammissibile per il duale** $\boldsymbol{\lambda}^T = \mathbf{c}_B^T B^{-1}$ tale che: $\boldsymbol{\lambda}^T A \leq \mathbf{c}^T$. dall'ottimo vecchio di P
2. Le operazioni di pivoting mantengono l'ammissibilità per il problema duale (D-ammissibilità).
3. Termine: base B ammissibile anche per il primale. $B^{-1}b \geq 0$ all'ottimo di D

se P di min, il simp duale fa aumentare la FO

Osservazione: nel simplesso duale scelgo il pivot in modo da
→ ottenere l'ammiss. primale
→ mantenere l'ammiss. duale

tecniche di modellizz. A | 03/05/2023

PLI

* STRATEGIE MODELLIZZAZIONE PLMI

Nella PL una buona formulazione prevede poche variabili e vincoli. La complessità è comunque in m ed n, per quanto buona può essere la formulazione.

Nella PLI una buona formulazione corrisponde ad un rilassato stringente ed è CRUCIALE perché potrebbe ridurre la difficoltà del problema ~~ed il tempo.~~

caso più semplice di PLI

SCELTA BINARIA

Il loro impiego più importante è modellizzare (codificare) la scelta tra due alternative.
Ad esempio ipotizzo un insieme di item con un certo valore e peso: solitamente questi problemi hanno vincoli di ricerca di un sottoinsieme con un valore max e che rispetti un peso limite

* PROBLEMI DI KNAPSACK (NP-hard)

Idea: zaino con una capienza massima e va riempito di oggetti

insieme di oggetti: $N = \{1 \dots n\}$ item: v_i valore, p_i peso $\quad x_j \in \{0,1\} \; \forall j \in N$

capienza zaino: c ← unica risorsa limite

obiettivo: cerco sottoinsieme ottimale $S \subseteq N$ che rispetti la capienza e massimizzi il valore degli oggetti

il modello di questo tipo di problema sarà:

$$\boxed{\begin{array}{ll} \text{FO:} & \max \sum_{j=1}^{n} v_j \cdot x_j \\ \text{VINCOLO:} & \sum_{j=1}^{n} p_j \cdot x_j \leq c \end{array}}$$

soluzione in uscita: vettore di 0 e 1

$x = (0, 1 \dots 0)_n$ per la selezione di n oggetti

Knapsack:
- proprietà combinatorie
- sottoprobl. in grani di ottimizzazione discreta

es: budgeting, caricamento mezzi, cutting stock, aste combinatorie

Quante combinaz. possibili?

2^n → item; 2 → scelta binaria

foglie all'albero delle combinazioni

⇒ la numerosità esplode velocem.

esempio: le aste

- hanno offerte tipo "offro x € per tot oggetti"
- ho un max di oggetti : 20 bottiglie
- devo max il profitto

$$\max \sum_i v_j x_j$$

$$\sum_i p_j x_j \leq 20$$

$$\forall p_j : p_j \leq 20$$

serve anche $\sum_i p_j > C$
altrimenti scelgo tutte le offerte

soluzione "buona" per i tempi che impiega

- SOLUZIONE EURISTICA (NON-OTTIMA)
  scelgo il rapporto prezzo/unità migliore

- SOLUZIONE OTTIMA
  cerco tutte le 2^n soluzioni e confronto
  ⇒ DISPENDIOSA

OSSERV: lavoro con quantità intere

- IDEA: scelgo un ordinamento per gli item tra cui scegliere e lo impongo a priori
  ⇒ inizio ad inserire secondo l'ordine e mi fermo una volta che arrivo a capacità massima (se ho oggetti che non ci stanno skippo)

$O(n \cdot \log n)$

⇒ SOLUZIONI GREEDY, non ottimali (algoritmi miopi)

hanno un worst case terribile

CRITERI DI ORDINAMENTO:

Si possono costruire esempi ad-hoc per dimostrare che la soluz euristica DIVERGE da quella ottimale.

Elenco i criteri e dimostro:

A. valore: $v_1 \geq v_2 \geq \dots \geq v_n$

Assumo l'istanza $c = 100$ con $n = 101$ item

$v_1 = 10 \quad v_2 = 9 \dots v_n = 9$

$p_1 = 100 \quad p_2 = 1 \dots p_n = 1$

- EURISTICA: inserisco l'item 1 con valore 10 ma che riempie completamente
- OTTIMA: prendo i 100 item con peso unitario $\Rightarrow$ valore totale sarà 900

* però una differenza tra 10 e 100 non è drammatica...

faccio tendere $c \to \infty$ così ci stanno più item unitari

$\Rightarrow$ valore sol ottima $\to \infty$ DIVERGE sempre più da 10

errore algoritmo A

$$\frac{z_A}{z^*} = \frac{10}{900} = \frac{1}{9}$$

errore % $\to \infty$ con $c \to \infty$

B. costo: $p_1 \leq p_2 \leq \dots \leq p_n$

ESERCIZIO: CASO PEGGIORE

istanza opposta: $c = 100$

$v_1 = 1 \dots v_k = 1 \quad \boxed{v_{k+1} = 100}$

$p_1 = 1 \quad p_k = 1 \quad p_{k+1} = 10$

- EURISTICA: prendo 100 oggetti dal peso unitario

$$z_B = 1 \cdot 100 = 100 \qquad \frac{c}{p_i}$$

- OTTIMA: seleziono 10 oggetti con valore 100

$$z^* = 10 \cdot 100 = 1000 \qquad \frac{c}{p_i}$$

$$\frac{z_B}{z^*} = \frac{100}{1000} = \frac{1}{10}$$

~~con $c \to \infty$~~

~~oppure~~ $v_{k+1} \to \infty$

l'errore esplode

C. rapporto valore/peso: $\frac{v_1}{p_1} \geq \frac{v_2}{p_2} \geq \dots \geq \frac{v_n}{p_n}$

Assumo un'istanza: n oggetti e capacità $c = kn$ con $k > 0 \in \mathbb{Z}$

- 1 oggetto: $p_{max} = c$ $\quad v_{max} = c-1 \rightarrow \frac{v_{max}}{p_{max}} < 1$
- $n-1$ oggetti: $p_j = 1$ $\quad v_j = 1 \rightarrow \frac{v_j}{p_j} = 1$

- EURISTICA: seleziona $n-1$ oggetti con valore e peso unitario

$$z_c = (n-1) \cdot 1 = n-1$$

- OTTIMA: seleziona un solo oggetto

$$z^* = \underset{\underset{c-1}{\parallel}}{v_{max}} = c-1 = kn-1$$

STIMA DELL'ERRORE:

$z^* = kn-1$

$z_c = n-1$

$$\frac{z^*}{z_c} = \frac{kn-1}{n-1} = \frac{k(n-1)+k-1}{n-1} = k + \frac{k-1}{n-1} > k$$

$\Rightarrow$ con $k \rightarrow \infty$ $\quad z^*$ è infinitamente più grande di z_c

OSSERVAZIONE: se prendo come soluzione il max tra la soluzione trovata dall'alg. greedy C ed il profitto dell'item a valore max quindi:

$\max\{z_c, v_{max}\}$ ottengo un algoritmo H tale che

garantisce un errore minimo $\quad \frac{z_H}{z^*} \geq \frac{1}{2}$

$$z_{greedy} = \max\{z_c, p_{max}\}$$

* COME FUNZIONA GUROBI CON KNAPSACK?

Essendo un problema di PLI devo fare il passaggio PLI → PL e passare da var binarie $\in \{0;1\}$ => var continue $\in [0;1]$

applico il rilassamento continuo
↓
questo fa un calcolo a formula chiusa, mi fornisce un upper bound

- avendo n variabili pongo n vincoli: $0 \leq x \leq 1$
- più il vincolo dello zaino: $\sum_j p_j x_j \leq c$

1. gurobi ordina secondo l'algoritmo C: $\frac{v_j}{p_j} \geq \frac{v_{j+1}}{p_{j+1}}$

2. analizzo in ordine decrescente e seleziono:
   partendo dal rapporto migliore, inserisce oggetti finché incontra l'ultimo item S che non ci sta, è l'item critico

$1 \ldots\ldots S \ldots\ldots n$

$x_j = 1 \quad \forall j = 1 \ldots s-1$ $\qquad$ $x_j = 0 \quad \forall j = s+1 \ldots n$

3. la soluzione ottima prende una parte frazionaria di S critico per riempire

ottimo del rilass continuo

$$x_s = \left(c - \sum_1^{s-1} p_j \right)$$

soluzione a numeri tutti interi fino ad S-1

p_s??
controlla che cazzo ho scritto

NB. se S riempie perfettamente lo zaino senza bisogno di parti frazionarie
=> il rilass. continuo è già la sol. OTTIMA INTERA.

SELEZIONE DA UN INSIEME

Dato un insieme di variabili binarie x_j associate ognuna ad un item j di un insieme T di dimensione finita, posso imporre:

$\sum_{j \in T} x_j \geq 1$ "scelgo almeno un item"

$\sum_{j \in T} x_j = 1$ "scelgo esattam. un item"

$\sum_{j \in T} x_j \leq 1$ "scelgo al massimo un item"

PROBLEMA DI SET COVERING (NP)

Esempio applicativo: minimizzare le stazioni da aprire per garantire la copertura del territorio

Dati:

$T = \{t_1, t_2 \dots t_n\}$ territorio diviso in regioni t_i

$F = \{F_1, F_2 \dots F_n : F_j \subseteq T\}$ stazioni

$F_j \subseteq T$ = stazione j vista come INSIEME di regioni

$\Rightarrow$ cerco $S \subseteq \{\text{impianti}\}$, una collezione di sottoinsiemi di regione

SOLUZIONE: S è $\bigcup_{j \in S} F_j = T$ (copertura totale)

$\exists$ regioni che possono essere coperte da + stazioni, importante: ALMENO UNO!!

* COSTRUISCO UN MODELLO:

righe $\rightarrow$ regioni

colonne $\rightarrow$ impianti

matrice:

$$\begin{array}{c|c} & \begin{matrix} F_1 & \dots & & F_n \end{matrix} \\ \begin{matrix} t_1 \\ \vdots \\ t_n \end{matrix} & \begin{bmatrix} 1 & 0 & \dots & 0 \\ \dots & & & \\ 1 & & & \end{bmatrix} \end{array}$$

$a_{ij} = \begin{cases} 0 \\ 1 \end{cases}$ "F_i copre o no t_j?"

variabili decisionali:

x_i, $i = 1 \dots n$ $\begin{cases} 0 \\ 1 \end{cases}$ attivazione dell'impianto j

FO? il set covering vuole una copertura ALMENO TOTALE

$$\min\ c^t x = \sum_i c_{ij} x_j \quad \text{con } x_j \in \{0,1\}, \ j \in N$$

"pesi" degli impianti

vincoli
- tutti quanti le regioni: $A\underline{x} \geq 1$ vettore

=> ma cosa rappresenta il prodotto $A\underline{x}$?

$$\begin{bmatrix} 1 & 1 & \dots & 0 \\ 0 & & & \\ \vdots & & & \\ 1 & & & \end{bmatrix} \cdot \begin{bmatrix} F_i \end{bmatrix} = \begin{bmatrix} F_1 & F_2 & \dots & 0 \\ F_i & & & \\ \vdots & & & \end{bmatrix}$$

prendendo un vincolo alla volta trovo gli impianti necessari ad aprire ogni singola regione

$\sum_i x_k$ tale che x_k sono tutti gli impianti per una certa regione

$\sum_i x_k \geq 1$

così sono sicuro di coprire tutto

SET COVERING: MATRICIALE

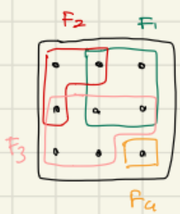
$$\min\ 1^t x$$
$$A\underline{x} \geq 1$$
$$x \in \{0,1\}^n$$

=> il rilassamento continuo calcola un lower bound mentre possiamo considerare upper bound la prima soluzione ammissibile:

→ apro tutti gli impianti

soluzione banale, prob NP easy

$$\bigcup_{j \in N} F_j = T$$

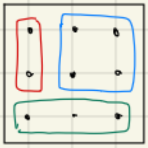


← esempio di soluz ammissibile

relax → lower bound

sol. iniziale → UP ma difficile!

* SET PARTITIONING "non puoi sovrapporre gli impianti"



es. soluzione ammissibile

$\min \ \mathbb{1}^t \underline{x}$

$A\underline{x} = \mathbb{1}$ → ogni regione ha esattamente una stazione

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

* NB: $\cup F_j = T$ ma con $F_j \cap F_i = \emptyset$ ← ∀ coppia di staz = insieme di regioni, non si devono intersecare

$(A\underline{x} = \mathbb{1})$

questa condizione forte rende il problema difficile

* SET PACKING "al massimo copro una volta sola"

soluzione ammissibile

$\max \ \mathbb{1}^t x$

$Ax \le \mathbb{1}$

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

relax → upper bound

sol iniziale → LB ma è facile

* NB: non impongo vincoli sulle unioni ma solo

$F_j \cap F_i = \emptyset \quad (Ax \le \mathbb{1})$

=> condizione banalmente soddisfatta (prendo una sola stazione)

È un problema facile

→ DECISIONI DIPENDENTI

Questo concetto è rappresentabile dalle scelte binarie:

"la decisione A (variab x) può essere presa ($x=1$) solo se la decisione B (variab y) è stata presa ($y=1$)"

Scenario: x_i decide se aprire l'impianto produttivo i $x_i \in \{0,1\}$

→ subordino questa scelta a "esistenza di una strada che porta all'impianto" rappresentato da $y \in \{0,1\}$

$x=1$ — $y=1$
$x=0$ — $y=1$ o 0

$\Rightarrow$ $\boxed{x \leq y}$

* Una variabile $x \geq 0$ può assumere valori strettamente positivi solo se una determinata decisione (variab y) è stata presa ($y=1$)

$\Rightarrow$ $\boxed{x \leq My}$

impegno che l'impianto produca $\Leftrightarrow$ $y=1$ è aperto

ma la prod x non è binaria
⇒ mi appoggio a un valore molto grande M

- $y=0$ (decisione non presa) impianto chiuso
  $x \leq 0 \Rightarrow x=0$ variab x può essere solo 0, non produce
- $y=1$ (decisione presa) impianto aperto
  $x \leq M \Rightarrow x$ libera limitata superiormente

M ad esempio sarà la

NB se non produco ($x=0$) anche y sarà forzata a zero

con M troppo grande, rischio l'instabilità numerica

* invece per forzare $y=0$ quando $x=0$

es: soluz ottima prevede di non produrre $\bar{e}_i$, quindi impianto chiuso

=> è un compito della F.O.

$$\min \sum_i K y_i$$

- se nel vincolo appare $x=1$ => $y=1$ forzato dal vincolo
- se nel vincolo appare $x=0$ => y è libera (indifferente) ma sappiamo bene che se non produco, non apro l'impianto

=> la F.O. porta $y=0$

* come vengono espressi i vincoli strutturali?

$$x_t \leq M_t \delta_t$$

semplice, la prod non supera l'upper bound (richiesto)

$x_t = 0 \rightarrow \delta_t = 0$ (dalla FO)

$x_t \neq 0 \rightarrow \delta_t = 1$

$\delta_t = 1 \rightarrow x_t \leq M_t$

M_t non deve essere un numero elevato scelto a caso per (vedi ragionamento instabilità numerica CRITERIO C knapsack)

colpa del simplex e delle sue matrici (se hanno dim numeri diversi, saltano fuori determinanti ed inverse molto approssimati)

$$\Rightarrow \quad M_t = \sum_{t}^{T} r_s$$

copro la domanda da adesso $(t=1)$ fino alla fine del periodo produttivo (T)

di solito x PLANT LOCATION: M_t = capacità magazzino

il vincolo di bilanciamento scorte è espresso

$$y_t = y_{t-1} + x_r - r_t \qquad \forall t$$

$$\vdots$$

$$y_1 = \underset{=0}{y_0} + \underbrace{x_1 - r_1}$$

qnt di prodotto venduto nel periodo t

NB. È NECESSARIA UNA GIACENZA A T=0 ALTRIMENTI NON POSSO PRODURRE A T=1

collego i costi setup con le giacenze tramite la quantità x_i

---

Modello dynamic lot-size (Wagner-Whitin)

$$v = \min \sum_{t=1}^{T} (f\delta_t + c_t x_t + h y_t)$$

min costi

soggetto a

$$y_t = y_{t-1} + x_t - r_t \quad t = 1, \ldots, T,$$

vincolo bilancio scorte

$$x_t \leq M_t \delta_t \quad t = 1, \ldots, T,$$

vincolo apertura impianti prod

$$x_t, y_t \geq 0 \quad t = 1, \ldots, T,$$

$$\delta_t \in \{0, 1\} \quad t = 1, \ldots, T.$$

NB:

* senza vincolo SET-UP: problema PL (formula Wagner-Whitin)
* con vincolo SET-UP: PLI, problema complesso

FILE: tecniche modellizzazione B

* VINCOLI DISGIUNTIVI: CASO CON DUE VINCOLI

IP: sia $\underline{x}$ un vettore di variabili decisionali non negative. Siano:

$$a^t \underline{x} \geq b \quad (1)$$

$$c^t \underline{x} \geq d \quad (2)$$

con a e c a componenti non negative

TH: solo uno dei due vincoli è attivo (= influisce sulla sol ottimale)

ATTIVO = soddisfatto da sol trovata   NON ATTIVO = sempre soddisfatto (RIDONDANTE)

* moltiplico per una variabile decisionale binaria!

$$a^t \underline{x} \geq yb$$

$$c^t \underline{x} \geq \boxed{(1-y)}\, d$$

complementare di y

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

- funzionamento:

$\boxed{y=0}$

$a^t \underline{x} \geq 0$ ① sempre verificato $\forall x\ (\geq 0)$

$c^t \underline{x} \geq d$ ② vincolo ATTIVO grazie alla modellizzazione

$\boxed{\text{NB}}$ imporre a^t e $c^t \geq 0$ è ESSENZIALE poiché la var binaria funzioni ad attivare/disattivare i vincoli a turno

se ∃ anche un elemento < 0 salta tutto perché $a^t \underline{x} \geq 0$ non sarebbe soddisfatto $\forall \underline{x} \geq 0$
⇒ influenza la soluzione ottimale

* VINCOLI DISGIUNTI : GENERALIZZAZIONE

COEFF NON NEGATIVI : siano dati m vincoli $a_i^T x \geq b_i$ dove $a_i \geq 0$

=> richiedo che siano soddisfatti esattamente K vincoli:

$$a_i^T \mathbf{x} \geq b_i y_i, \qquad i = 1, \dots, m$$

$$\sum_{i=1}^{m} y_i = k,$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, m.$$

COEFF QUALSIASI : siano dati m vincoli $a_i^T x \geq b_i$ dove le componenti di a_i sono libere di diventare negative:

ci permette di non perdere soluzioni

$$a_i^T \mathbf{x} \geq b_i - M_i(1 - y_i), \qquad i = 1, \dots, m$$

$$\sum_{i=1}^{m} y_i = k,$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, m.$$

- $y_i = 1$ => vincolo ATTIVO, grazie solo $a_i^T x \geq b_i$
- $y_i = 0$ => il vincolo diventa $a_i^T x \geq b_i - M_i$ che per un ragionevole $M_i >> 0$ sarà sempre verificato
  => ridondante

* PROBLEMA DI SEQUENCING (a macchina singola)

Suppongo di avere una macchina che deve svolgere più job in sequenza senza la possibilità di svolgerne più di uno in contemporanea.

$x_j + a_j \leq d_j$ ogni job va finito prima della scadenza

$x_j \geq 0$ ogni job inizia dopo l'istante zero (o ae)

$x_j + a_j \leq t$ nessun job finisce oltre l'orizzonte Tmp

$x_j + a_j \leq x_i$ la precedenza

* ed il non-parallelismo? $x_j + a_j \leq x_i$ oppure $x_i + a_i \leq x_j$ =>

$$x_i + a_i \leq x_j + M y_{ij}$$
$$x_j + a_j \leq x_i + M(1 - y_{ij})$$

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

- $y_{ij} = 1$ => $x_i + a_i \leq x_j + \underline{M}$ SUPERFLUO
  $x_j + a_j \leq x_i$ i segue j
- $y_{ij} = 0$ => $x_i + a_i \leq x_j$ j segue i
  $x_j + a_j \leq x_i + \underline{M}$ SUPERFLUO

TRUCCO (ese): • trasformo tutti i vincoli $\leq$ in vincoli $\geq$
• porto le variabili a SX e costanti a DX
=> scopro se sono disgiunte $a, c \geq 0$ o a, c liberi

FORMULAZIONE FINALE:

$$\min t$$

$$x_j + a_j \leq d_j \qquad j = 1, \ldots, n$$ ← rispetto scadenza

impongo t fine di tutto → $$x_j + a_j \leq t \qquad j = 1, \ldots, n$$

$[x_i + a_i \leq x_j] \rightarrow$ se ci sono vincoli di precedenza

$$My_{ij} + (x_j - x_i) \geq a_i \quad \forall\, i \neq j$$
$$M(1 - y_{ij}) + (x_i - x_j) \geq a_j \quad \forall\, i \neq j$$
] non parallelismo

$$x_j \geq 0 \quad \forall\, j$$
$$y_{ij} \in \{0,1\} \quad \forall\, i \neq j$$

* VARIABILI CON DOMINIO DISCRETO NUMERABILE

SOS1: SPECIAL ORDER SET TYPE 1

Una variabile x assume solo valori da un insieme $C \subseteq \mathbb{R}$ di cardinalità finita

$A = \{a_1 \dots a_2 \dots a_m\}$ variab. binaria : valore → 1:1

$$x = \sum_{j=1}^{m} a_j y_j$$ variab continua

$$\sum_{j=1}^{m} y_j = 1$$ impongo di SCEGLIERE dallo spazio di valori solo 1

$$y_j \in \{0,1\} \quad \forall j$$

creds...

* FUNZIONI LINEARI A TRATTI

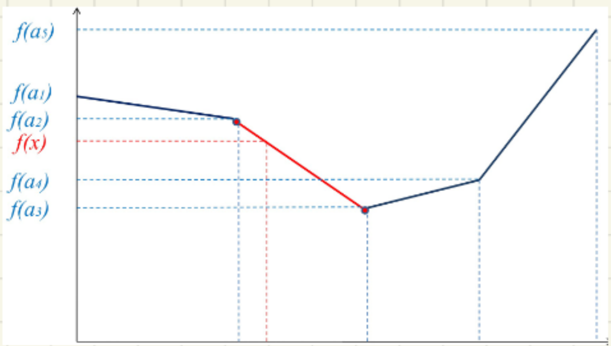
"L'idea è avere le var x che possano assumere un valore all'interno di diversi intervalli"

SOS2: SPECIAL ORDER SET TYPE 2

Modellizzare una spezzata le cui coordinate si possono scrivere $(a_i, f(a_i))$

Al massimo 2 variabili binarie di un insieme dato possono essere selezionate e queste devono essere consecutive

OK COSA VUOL DIRE 14/05
OK HO CAPITO ✓ 15/05



si parla di generare valori all'interno di intervalli di cui ne conosco gli estremi
⇒ comb. convessa!

* L'idea è sempre "generare" il valore della variabile continua x

dove seleziono un a_i (intervallo $\overline{a_i a_{j+1}}$) che individua un

$f(a_i) \in \overline{f(a_j) f(a_{j+1})} \equiv x$ valore generato

FUNZ. LINEARI A TRATTI:
non è né concava né convessa (globalmente)
ma lo sono i suoi tratti presi singolarmente.

FORMULAZIONE: solitamente in questi problemi vengono forniti i punti di passaggio [gli $f(a_i)$ dove la funz cambia definizione]

=> osservo che con n punti trovo n−1 intervalli => solo un intervallo alla volta è attivo => n−1 variabili binarie

y_i indica l'unico intervallo a cui a_i appartiene

$y=1$ ↓ λ_0 e λ_1 attivi ↓ intervallo 1 selezionato

• λ_i sono variabili decisionali ~~(ricominciamo gli a_i)~~ e rappresentano gli estremi degli intervalli => un intervallo ha 2 estremi

=> y_i deve vincolare i valori di λ_i e $\lambda_{i+1} \in [0,1]$, che saranno i coefficienti della combinazione convessa per trovare x

esprimo questa logica con:

se $y_1=0$ => $\lambda_1 \le 0$ ma dal modello $\forall \lambda_i \ge 0$ → => $\lambda_1=0$ $\boxed{OK}$

• se $y_1=0$ => $\lambda_1 \le 1$ (oltre che $\lambda_1 \ge 0$ implicito)

ma allora il legame diventa

$$\boxed{\lambda_i \le y_i}$$

per il primo e l'ultimo

• però λ_2 è l'estremo DX di y_1 e SX di y_2, come ne gestisco l'attivazione?

=> $\lambda_2 \le y_1 + y_2 \le 1$

$y_i\, y_j\ \forall i \ne j$ sono mutualmente esclusive questa somma varrà al max 1 in ogni caso

$y_1=0 \quad y_2=1$

$\Rightarrow \lambda_2 \le y_2 = 1$ ATTIVO (2)

$y_1=1 \quad y_2=0$

$\Rightarrow \lambda_2 \le y_1 = 1$ ATTIVO (1)

per tutti gli altri

$$\boxed{\lambda_i \le y_{i-1} + y_i \le 1}$$

STO SCEGLIENDO UN VALORE $f(a_i)$! NON GENERO!

VERSIONE I

$$f(x) = \sum_i^n \lambda_i f(a_i) \quad \text{con } \sum \lambda_i = 1$$

y_i fa scattare i λ_i che generano in un certo intervallo

$$\underbrace{\lambda_1 \le y_1}_{\text{primo}}, \quad \lambda_i \le y_{i-1} + y_i \quad \forall i = 2 \dots n, \quad \underbrace{\lambda_n \le y_{n-1}}_{\text{ultimo}} \quad \text{con } \sum_i^{n-1} y_i = 1$$

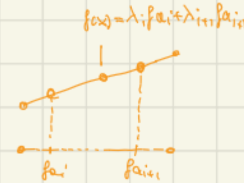
$\lambda_i \ge 0 \quad i = 1 \dots n$

$y_i \in \{0,1\} \quad i = 1 \dots n-1$

NB: questa versione non contiene il MECCANISMO CHIAVE del SOS di 2° tipo

intervallo n−1 regolato da a_{n-1} e a_n
↓ $y_{n-2}; y_{n-1}$ ↓ y_{n-1}

VERSIONE FINALE



$$\boxed{\begin{array}{ll} f(x) = \sum_{i}^{n} \lambda_i f(a_i) \quad \text{con} \quad \sum_i \lambda_i = 1 & \\ \lambda_i \leq y_i & \forall i = 1 \dots n \\ \sum_{i}^{n} y_i \leq 2 & \\ y_i + y_j \leq 1 & \forall i = 1 \dots n-2,\ j = i+2 \dots n \\ & y_i \in \{0,1\} \quad i = 1 \dots n \end{array}}$$

→ mi permette di scegliere y_k consecutive (estremi quindi dello stesso intervallo)
⇒ impone che se y_k non sono consecutivi, non possono essere attivi assieme (≤ 1)

osservazione: ciò che serve per mettere in piedi questo modello è
- un set di variabili binarie
- vincolo per imporre la selezione di al max 2 variabili
- vincolo per selezionare solo variabili binarie consecutive

10/05/2023

PREPROCESSING

Sono tutte quelle tecniche che vengono applicate prima dell'ottimizzazione, ovvero di operazioni che possono essere compiute per migliorare una data formulazione matematica

worst case: per instabilità numerica trovo valori completamente sbagliati

BOUND STRINGENTI

Si applica quando in un problema PLI appare un vincolo nella forma: $x_j \leq M y_j$

=> se sono in grado di fissare M in maniera intelligente, posso avere una convergenza all'ottimo più veloce!

esempio:

MODELLO 1

$M = 100$

MODELLO 2

in seguito ad un analisi interna scopro che x_j hanno come UB 30

=> $M = 30$

* essendo nella PLI, per risolvere i modelli si agisce così:
  - si calcola il rilassamento continuo
  - si cerca l'ottimo

=> cerco la soluzione ottimale stringendo sulle coordinate il più possibile

=> se il rilassamento è molto vicino al PLI allora convergo molto più velocemente

è qui che cambia tutto! nei termini di PLI può non fare la differenza, ma nel continuo sì!

* continuo coi due modelli:

FO: $\min \sum_j f \cdot y_j$

①

$x_j \leq 100 y_j$

$y_j \geq \frac{x_j}{100}$

②

$x_j \leq 30 y_j$

$y_j \geq \frac{x_j}{30}$

con $0 \leq y_j \leq 1$

* ipotesi ad-hoc: all'ottimo $x_j = 30$

①

$y_j^{PL} = \frac{30}{100} = \frac{3}{10}$

il relax continuo è frazionario

⇓

serviranno ulteriori iterazioni per stringermi attorno all'ottimo

②

$y_j^{PL} = \frac{30}{30} = 1$

valore intero relax
⇒ ottimo globale

problema di min + con $y \geq$ (qnt frazion) allora il min valore di y è proprio quella quantità!

↓

scrivo $y_j = \frac{\text{ottimo}}{M}$

* COME SCOPRO QUESTO UB A PRIORI?

Ad esempio un UB può essere la capacità impianto, la domanda del mercato
⇒ se non ho queste informazioni diventa un problema EURISTICO di stima OPPURE posso crearmi un altro modello:

individuo la variabile di cui devo stimare il valore max e la sostituisco nella FUNZIONE OBIETTIVO

⇒ la soluz ottima mi fornirà così l'UB (rilassamento)

↓

così sono sicuro di ottenere un UB in tempi polinomiali caso problema min

FIXING VARIABILI

Sfrutta delle tecniche di probing: stabilisce a priori il valore ottimo di alcune variabili

Siano:

- z_{PL} valore ottimo calcolato dal rilassamento continuo (PLI minimo)
- z_H valore di una soluz ammissibile qualsiasi del PLI
- $GAP = z_{PL} - z_H$
- $x_j^{PL} = 0$ valore assunto dalla variabile x_j nella sol. ottima del rilassam. continuo e c_j il suo CCR

CONDIZIONE:

$c_j > GAP \Rightarrow x_j^* = 0$ nella sol ottima (variabile sarà fuori base) e quindi può essere eliminata dal problema

non ho capito questa cosa

dall'ottimo torno indietro => fo peggiora

X_B X_N

0 ≥ 0

CCR ≥ 0

- sono all'ottimo con una $x_j \in X_N$

forzo il simplesso a inserire $x_j \neq 0$

=> fo peggiora di val$(x_j) \cdot CCR_j$

ex: $x_j = 1$

fo += 7 · 1
↑
peggiora

con $CCR_j = 7$

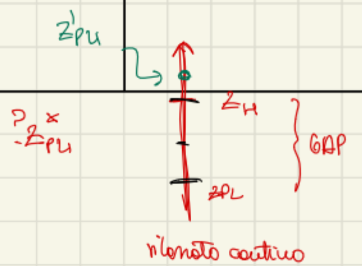
cosa significa questa cosa
↓
variabili vengono arrotondate?

* prendo un CCR ed immagino di forzare in una sol ottima con var va a positiva ⇒ no nel relax è possib che molte var passino da 0 a positivo

- in base porto x_j : $CCR_j > GAP$ (UB e LB)

=> ottengo un relax sporco (forzato) dove ogni var è libera tranne quella che nel diventare pos è schizzata fuori

=> ottengo un rilassamento sopra z_H che sarà superiore alle altre



tutte le var con CCR ≥ GAP nel rilassato non vanno toccate sennò trovo solo sol peggiori di z_H (UB di euristica)
↓
perché banale

- se $CCR_j > GAP$ finisco sopra z_H e non ho senso spostare le var x_j dallo zero che assumono all'inizio → le scarto e non le tocco durante la risoluzione.

DISUGUAGLIANZE LOGICHE

Bonoto sul semplificare insiemi di vincoli osservando il comportamento del problema.
=> l'idea è trovare una riformulazione che renda almeno un vincolo ridondante

V1: $3x_2 + 2x_3 \geq 2$

senza perdere soluzioni intere posso affermare che:

$x_2 + x_3 \geq 1$

V2: $2x_1 - 2x_2 + 6x_3 \leq 5$

* trasformo tutti i coefficienti in positivi
(con var binarie uso i complementi)

$2x_1 - 2(1 - x_2') + 6x_3 \leq 5$

=> $2x_1 + 2x_2' + 6x_3 \leq 5 + 2$

NB: $x_2' = 1 \wedge x_3 = 1$ non soddisfa il vincolo!

$x_2' + x_3 \leq 1$ limito ad avere al max 1 attivo

$1 - x_2 + x_3 \leq 1 \Rightarrow x_3 \leq x_2$

ma $x_2 + x_3 \geq 1$ da V1 => [risulta essere ridondante una volta fissato x_2]

=> x_2 non può essere 0 perché allora anche x_3 sarebbe 0 ma questo viola il primo vincolo

=> $\boxed{x_2 = 1}$

QUI ALCUNI ES CHE FACCIO SU UN ALTRO FILE

15/05/2023 teoremi + es. Trasporto

FILE: TUM FILE: piani di Taglio

TOTALE UNIMODULARITÀ

Si consideri il seguente problema di PLI in forma standard:

$$z_{PLI} := \min \; c^t x$$
$$Ax = b$$
$$x \geq 0 \text{ intero} \qquad \text{con } A, b \text{ interi}$$

* condizioni per cui una generica soluzione x^* ha componenti frazionarie?

Sia B la base associata di x^*:

$$x^* = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} \begin{array}{l} \rightarrow \text{sol sistema} \\ \rightarrow \text{componenti } x \; x_N \end{array}$$

$\Rightarrow$ x^* ha componenti frazionarie solo se anche $B^{-1}b$ non è intero

* osservazione:

Questo lo vediamo dal calcolo della matrice inversa

$$B^{-1} = \frac{1}{\det(B)} \begin{bmatrix} a_{11} & \\ & a_{mn} \end{bmatrix}^t$$ dove $a_{ij} = (-1)^{i+j} \cdot \det(M_{ij})$ ed M_{ij} è la sottomatrice da B eliminando la riga i e la colonna j

OK questo è vero ed è comodo. ma quando non è vero? non mi da info sull'interezza, può esserlo comunque

$\Rightarrow$ B è intero $\Rightarrow$ a_{ij} $\forall ij$ interi $\Rightarrow$ B^{-1} intera se $\det(B) = \pm 1$

CONDIZ. SUFFICIENTE ma non necessaria per l'interezza di $B^{-1}b$

PERCHÉ?

questo non lo ho capito $\rightarrow$

Dato una matrice A contenente una base B con $\det(B) \neq 1$ esiste sempre un problema di PL del tipo $\{\min \; c^t x : Ax = b, x \geq 0\}$ forma standard con c e b opportuni la cui soluzione ottima coincide con un vertice fraz.

B e b possono essere non interi ma può dividerli OK e moltiplicarli dando INTERI $\Rightarrow$ è caso, quindi COND. NON NECESSARIA

$\frac{+B_c}{|B|}$

SUMMARY:

- $B^{-1}b$ non intero $\Rightarrow$ x^* non intero
- B^{-1} intera se $\det(B) = \pm 1$
- B^{-1} intera $\Rightarrow$ $B^{-1}b$ intera
  $\not\Leftarrow$

× DEF: UNIMODULARITÀ

Una matrice intera A di dimensione $m \times n$ con $m \leq n$ si dice unimodulare se per ogni sua sottomatrice quadrata $m \times m$ B vale:

$$\det(B) \in \{\pm 1, 0\}$$

= ovvero le sottomat invertibili hanno det ±1 e le altre m×m quadrate hanno det nullo

th nello scritto

• TEOREMA 1:

Sia A unimodulare e b intero.

Allora il poliedro continuo $P = \{\underline{x} \geq 0 : A\underline{x} = \underline{b}\}$ ha solo vertici interi.

P è regione di un problema a vincoli con =

DIM:

• Sia x^* un qualunque vertice di P

$\Rightarrow$ ∃ una base B di A tale che $x^* = (B^{-1} \cdot b \,;\, 0)$

• B è una sottomatrice quadrata di A di ordine m e non singolare

$\Rightarrow$ $\det(B) = 1$

perché è base con componenti canoniche

$\Rightarrow$ interezza di B^{-1}

$\Rightarrow$ interezza di $B^{-1} \cdot b$ $\forall b$ intero (da ipotesi)

♣

con i vincoli di disug. A' è diverso per la presenza di $\bar{s}$ quindi il ragionamento diventa

* **MAT UNIMODULARE CASO FORMA CANONICA** → posto alla TUM

PLI in forma: $\{\min\ c^T x : Ax \geq b,\ x \geq 0\}$

⇓ trasformazione

$\{\min\ c^T x : Ax - Is = b, x \geq 0, s \geq 0\}$

con s variabili di surplus. Nuova mat dei vincoli $A' := [A, -I]$

* In generale ogni sottomatrice $m \times m$ di A' si può ottenere selezionando K colonne da A e $m-k$ da $-I$ $(0 \leq k \leq m)$

$k=0 \Rightarrow$ 0 da A, k da I — nessuno in Q

$k=m \Rightarrow$ k da A, 0 da I — tutte in Q

costruisco:

$$\begin{array}{cc} & A \\ B = & \left[\begin{array}{c|c} -I'_{m-k} & F \\ \hline 0 & Q \end{array}\right] \end{array}$$

$k \times k$

colonne da $-I$ colonne da A

$I' :=$ mat identità ordine $m-k$

⇒ la matrice B è quindi una triangolare superiore

⇒ il suo determinante si calcola come: $\det(B) = \pm\det(Q)$

riduzione della difficoltà del problema di calcolo

$\times k$ $\det(-I)$ è sempre ± 1 → moltiplico a $\det Q$

A' è unimodulare $\Leftrightarrow \det(Q) \in \{\pm 1; 0\}$ con Q ogni sottomatrice quadrata Q di A di un qualunque ordine.

deriva dal fatto che le k colonne selezionabili da A nella costruzione di B sono arbitrarie

* DEF: **TOTALE UNIMODULARITÀ**

Una matrice intera A di dimensione $m \times n$ si dice totalmente unimodulare se $\det(Q) \in \{\pm 1; 0\}$ per ogni sottomatrice quadrata Q, di qualunque ordine.

⇒ essendo una dimensione valida anche $k=1$ è richiesto che tutti gli elementi di A siano $\in \{\pm 1; 0\}$

da portare all'orale

* TEOREMA 2:

ipotesi

Sia A una matrice totalmente unimodulare e b intero.

$\Rightarrow$ il poliedro $P := \{ x \geq 0 : Ax \geq b \}$ ha solo vertici interi.

DIM:

valore delle surplus all'ottimo

prob canonico →

prob standard →

• Sia x^* un vertice qualunque del poliedro P e $s^* := Ax^* - b$

Dimostriamo che (x^*, s^*) è vertice di $P' := \{ (x,s) \geq 0 : Ax - s = b \}$

cerco l'assurdo

$\Rightarrow$ altrimenti se così non fosse:

$\exists$ due punti $(x^1 s^1)$ $(x^2 s^2)$ distinti $\in P'$ tali che:

$$(x^*, s^*) = \lambda (x^1 s^1) + (1-\lambda)(x^2 s^2) \quad \text{con} \quad 0 < \lambda < 1$$

tutti i punti di P permettono di individuare punti in P' ok GRZ

$P := \{ x \geq 0 : Ax \geq b \}$
$Ax - b \geq 0$

• Inoltre:

$s^1 = Ax^1 - b \geq 0$

$\Rightarrow$ x^1 e x^2 appartengono a P

$s^2 = Ax^2 - b \geq 0$

trovo che questo vale $\Rightarrow$

perché un vertice non è combinazione di altri punti

• Inoltre: $(x^1 s^1) \neq (x^2 s^2) \Rightarrow$ $x^* = \lambda x^1 + (1-\lambda) x^2$ non potrebbe essere un vertice di P.

da ipotesi assurda $x^1 \neq x^2$

Ora sono certo che $(x^* s^*)$ sia un vertice valido. (di P').

A' cost vincoli per P'

$\Rightarrow$ poiché A è TUM, $A' := [A, -I]$ è unimodulare

$\Rightarrow$ per il teorema 1 $(x^* s^*)$ è intero e quindi anche x^*

A' unimod + b intero

♣

* OSSERVAZIONE: ora conosco i criteri e teoremi per definire l'unimodularità ed i suoi legami con l'interezza dei vertici

=> come verifico una matrice A come TUM? ↗ enumero tutte le sottomatrici ↘ scopro condizioni semplici

CONDIZIONI PER L'UNIMODULARITÀ TOTALE

se parlo di elementi => ordine = 1 => TUM

(I)

banale, dalla def di TUM

C. Necessaria: $a_{ij} \in \{-1, 0, 1\}$ per ogni elemento (≡ sottomat) di A stessa.

(II)

C. Sufficiente:

condiz necessaria è implicita (è in ipotesi)

* TEOREMA 3

Sia A una matrice con $a_{ij} \in \{-1, 0, 1\}$. A è TUM se valgono:

1. ogni colonna di A NON HA PIÙ DI DUE ELEMENTI NULLI

partizione := "classifico" le righe in due insiemi

2. $\exists$ una partizione (I_1, I_2) delle RIGHE di A tale che ogni colonna con esattamente due elementi $\neq 0$ abbia questi su due ~~colonne~~ righe appartenenti a INSIEMI I_1 e I_2 DIVERSI se e solo se sono CONCORDI in segno

(III)

PROPOSIZIONE 1:

La matrice A è TUM se e solo se:

- A^t è TUM
- A' ottenuta da A permutando e/o cambiando segno ad alcune colonne e/o righe è anch'essa TUM
- le matrici:

$$\left[\begin{array}{c|c} \pm 1 & \\ 0 & A \\ \vdots & \\ 0 & \end{array}\right] \text{ e } \left[\begin{array}{c|c} 0 & \\ \vdots & A \\ 0 & \end{array}\right] \text{ sono TUM}$$

NB: l'ultimo punto nella proposizione mi da alcune regole importanti:

Se esiste una colonna che contiene $\begin{cases} \text{solo UN } a_{ij} = \pm 1 & \text{un elem. non nullo} \\ a_{ij} = 0 \ \forall i,j & \text{tutto nullo} \end{cases}$

$\Rightarrow$ questa colonna può essere scartata perché non inficia sulla condizione di TUM

× nell'eliminare colonne (righe) potrei implicare l'eliminazione di righe (colonne) perché gli ho portato via un valore / l'unico valore non nullo (effetto cascata)

è un problema di flusso, e quelli avranno un impostaz. simile in termini di TUM

PROBLEMA DEL TRASPORTO: un caso di mat A unimod e vertici interi

$$\min \sum_i^m \sum_j^n c_{ij} x_{ij} \quad \text{con} \quad \sum_j^n x_{ij} \leq s_i \ \text{(offerta)} \qquad \sum_i^m x_{ij} \geq d_j \ \text{(domanda)} \qquad \text{e} \quad 0 \leq x_{ij} \leq q_{ij}$$

Cambio i segni ai vincoli e trasformo in canonica $\{\min c^T x : Ax \geq b,\ x \geq 0\}$

$\Rightarrow$ esce una matriciona allucinante

offerta

qui non ho solo un elemento $\neq 0$ ma pure domanda

Ausiliari

	x_{11}	x_{12}	$\cdots$	x_{ij}	$x_{i,j+1}$	$\cdots$	$x_{m,n-1}$	$x_{m,n}$		$\mathbf{b}$
1	-1	-1	$\cdots$	0	0	$\cdots$	0	0		$-s_1$
$\vdots$	0	0	$\cdots$	0	0	$\cdots$	0	0		$\vdots$
i	$\vdots$	$\vdots$	$\cdots$	-1	-1	$\cdots$	$\vdots$	$\vdots$		$-s_i$
$\vdots$	0	0	$\cdots$	0	0	$\cdots$	0	0		$\vdots$
m	0	0	$\cdots$	0	0	$\cdots$	-1	-1		$-s_m$
1	1	0	$\cdots$	0	0	$\cdots$	0	0		d_1
2	0	1	$\cdots$	0	0	$\cdots$	0	0		d_2
$\vdots$	0	0	$\cdots$	0	0	$\cdots$	0	0		$\vdots$
j	$\vdots$	$\vdots$	$\cdots$	1	0	$\cdots$	$\vdots$	$\vdots$	$\mathbf{b}=$	d_j
$j+1$	$\vdots$	$\vdots$	$\cdots$	0	1	$\cdots$	$\vdots$	$\vdots$		d_{j+1}
$\vdots$	0	0	$\cdots$	0	0	$\cdots$	0	0		$\vdots$
$n-1$	0	0	$\cdots$	0	0	$\cdots$	1	0		d_{n-1}
n	0	0	$\cdots$	0	0	$\cdots$	0	1		d_n
$(1,1)$	~~-1~~	~~0~~	~~$\cdots$~~	~~0~~	~~0~~	~~$\cdots$~~	~~0~~	~~0~~		$-q_{1,1}$
$\vdots$	0	0	$\cdots$	0	0	$\cdots$	0	0		$\vdots$
(i,j)	0	0	$\cdots$	-1	0	$\cdots$	0	0		$-q_{i,j}$
$\vdots$	0	0	$\cdots$	0	0	$\cdots$	0	0		$\vdots$
(m,n)	0	0	$\cdots$	0	0	$\cdots$	0	-1		$-q_{m,n}$

ultime m×n righe: hanno solo 1 valore $\neq 0$

OSSERVAZIONI:

- ogni colonna ha 3 elem $\neq$ da 0
- tutti gli a_{ij} sono $\{-1, 0, 1\}$

QUINDI:

$\Rightarrow$ dimostro TUM con:

- elimino le ultime m×n righe P4
- partizione delle righe:
  $I_1 = \{1 \dots m, m+1 \dots m+n\}$
  $I_2 = \emptyset$ TH 3

no forse NON HO CAPITO

* PIANI DI TAGLIO   nuovo FILE: CuttingPlanes

CLASSIFICAZIONE ALGORITMI

- Alg. ESATTI:
  1. trovano l'ottimo globale ma hanno tempi potenzialm. esponenziali

si accontenta ma c'è un certificato di bontà (stima % errore)

- Alg. APPROSSIMATI:
  1. non si garantisce l'ottimo, ma una ~~soluz ammissibile~~ ottimo locale in tempi polinomiali
     => è una soluzione "buona", rispetto al worst case
  2. il valore della soluzione ha un bound garantito dall'ottimo

ottimo locale più o meno buono

- Alg. EURISTICI:
  1. soluzione non ottima e non garantisce la bontà
     => punto a buone performance in tempi brevi

ALGORITMI ESATTI DI PLI => funzionano con:
1. calcola il rilassato
2. itera simplex a x^*
3. trova il vertice
   - se fraz continua
   - se intero si ferma

↓

PIANI DI TAGLIO

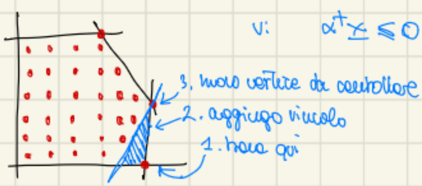
considero:

$$\min c^T x$$
$$Ax = b$$
$$x \geq 0$$

sia $P := \{x \geq 0 : Ax = b\}$ il poliedro comune (limitato e non vuoto) associato al rilassamento continuo di PLI

CASO IN CUI TROVO UN VERTICE FRAZIONARIO
=> INTRODUCO UN PIANO DI TAGLIO

Aggiungo al problema un vincolo nuovo che esclude la sol OTTIMALE appena trovata ed un'area a questo vertice vicina. (un'area di soluzioni frazionarie $x\bar{o}$!)



v: $\alpha^T \underline{x} \leq 0$

NB il vincolo non deve far perdere soluzioni intere!! deve solo stringere la soluzione ammissibile

* osservazione: i PIANI DI TAGLIO non cercano il convex hull...
rilogliamo la regione più stretta possibile attorno alla soluzione ottima del rilassato, almeno finché x^* non è intero

PROBLEMA DI SEPARAZIONE DI x^* DA X

* l'introduzione di questo vincolo è praticamente la ricerca di un vincolo soddisfatto da tutti i punti $\in X$ tranne x^* selezionato
=> un problema di ricerca della non ammissibilità (roba da dualità)

questa def non da altre info su x^*
=> noi sappiamo che è la solut ottima del rilassato

* DEF: PIANO DI TAGLIO

Dato $x^* \in P$ si dice piano di taglio una disuguaglianza $\alpha^T \underline{x} \leq \alpha_0$ tale che

1. $\alpha^T \underline{x} \leq \alpha_0 \qquad \forall x \in X := P \cap \mathbb{Z}^n$
2. $\alpha^T x^* > \alpha_0$ inammissibilità → escludo l'ottimo frazionario

OSSERVAZIONE: sarà necessario introdurre un numero esponenziale di tagli ($\equiv$ vincoli) prima di convergere! (TROPPI!)

CLASSIFICAZIONE TAGLI

studiamo solo due di questi
• zero half cuts
• gomory cuts

• tagli general purpose: derivano da considerazioni logico-matematiche generiche (tipo arrotondamenti, troncamenti, implicazioni etc...)

• tagli specifici: derivano dall'analisi poliedro della specifica istanza di un modello

NB. è ruolo dell'esperto saper introdurre tagli efficienti: né gurobi né gli algoritmi conoscono il significato semantico dei modelli e le loro parti

* TAGLI ZERO-HALF

Tecnica poco efficiente, si basa sul TRONCAMENTO.
Si cerca di ricondursi ad avere una disuguaglianza così:

- a SX variabili intere e coefficienti interi
- a DX qualsiasi valore frazionario

$\Rightarrow$ posso arrotondare il membro di DX per difetto (TAGLIO PER TRONCAMENTO)

NB. di solito questi termini noti frazionari non sono trovati così a caso nei problemi di PLI $\Rightarrow$ li introduciamo noi tramite somme di vincoli, divisione per un fattore comune ... combinaz lineari di vincoli pre-esistenti.

* TAGLIO DI GOMORY

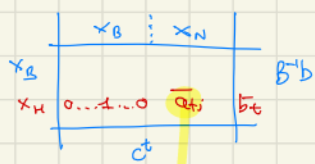
Idea: generare un taglio (vincolo) utilizzando le informazioni associate alla base B della soluzione di base ottima x^* del relax continuo corrente

Generazione: se x^* è frazionaria (o ha almeno una componente non intera)
$\Rightarrow$ si genera il vincolo aggiuntivo

SITUAZIONE INIZIALE: tableau ottimo (ho già $\overset{\text{non intero}}{\overset{\uparrow}{x^*}}$ del rilassato)

tableau di x^*

$x_H \in \underline{x}_B$



Esiste almeno un valore x^*_h non frazionario nella base: considero la sua variabile relativa, x_h

Nella base, x_H avrà componenti nulle tranne una $=1$ in corrispondenza della colonna di sé stessa

le componenti relative le variabili fuori base sono $\overline{a}_{tj}$

Φ: insieme delle variabili fuori base

Riscrivo la riga di x_H:

eq valida per punti di P (poliedro del rilassato) $\leftarrow$

$$\underbrace{x_h}_{\text{le variabili sono intere}} + \sum_{j \in \Phi} \overline{a}_{tj}\, \underbrace{x_j} = \underbrace{\overline{b}_t}_{\text{frazionario}} \equiv x^*_h$$

NB. chiamo "t" la RIGA GENERATRICE DEL TAGLIO

SUMMARY:

$$\underbrace{\alpha^t}_{\text{interi}}\ \underbrace{\underline{x}}_{\text{intero}} \leq \underbrace{\alpha_0}_{\text{frazionari}}$$

lo sostituisco con

$$\alpha^t \underline{x} \leq \lfloor \alpha_0 \rfloor$$

$\bar{b}_t \in \mathbb{R}$

(troncamento)

* procedimento secondo Gomory:

SX interi DX frazionari

Formulo un vincolo: $x_h + \sum_{j \in \Phi} \bar{a}_{tj} x_j \le \bar{b}_t$ (1)

So che posso scrivere: $x_h + \sum_{\Phi} \lfloor \bar{a}_{tj} \rfloor \cdot x_j \le b_t$ => quindi varrà anche troncando $\bar{b}_t$

Forma intera: $x_h + \sum \lfloor \bar{a}_{tj} \rfloor \cdot x_j \le \lfloor \bar{b}_t \rfloor$ (2)

NON HO CAPITO

Per costruzione questo vincolo è violato da x^*, ma valido $\forall$ altro elemento di X.

Poiché $x^*_j = 0$ $\forall j \in \Phi$ e $\bar{b}_t = x^*_h$ frazionario: $x_h + \sum_{\Phi} \lfloor \bar{a}_{tj} \rfloor x_j \big|_{x=x^*} = x^*_h = \bar{b}_t > \lfloor \bar{b}_t \rfloor$

Ma la forma frazionaria? Scrivo (1)−(2)

$$\sum_{j \in \Phi} (\bar{a}_{tj} - \lfloor \bar{a}_{tj} \rfloor) \ge \bar{b}_t - \lfloor \bar{b}_t \rfloor$$

trasformandolo in un $\ge$ vado a tagliare (??)

$$\sum_{j \in \Phi} \varphi(\bar{a}_{tj}) x_j \ge \varphi(\bar{b}_t)$$ con $r \in \mathbb{R}$ e $\varphi(r) := r - \lfloor r \rfloor \ge 0$ parte frazionaria

non credo di aver capito

surplus aggiunto per la STANDARD

Forma STANDARD: $$-\sum_{j \in \Phi} \varphi(\bar{a}_{tj}) x_j + s = -\varphi(\bar{b}_t) \qquad s \ge 0$$

scarto fraz $\bar{a}_t$ — scarto fraz di $\bar{b}_t$

↓

* il vincolo del taglio di Gomory in forma standard può essere aggiunto al tableau corrente (ottimo rilassato x^*) tramite l'orlatura

=> mantiene l'ammissibilità duale richiesta per poter applicare il simplex duale

NB: gomory implica l'utilizzo del duale perché devono esistere elementi $\bar{a}^*_{ij}$ negativi o nulli

(dalla th della dualità questo indica l'ammissibilità del duale => ma D non amm. sarebbe P non ammissibile => quindi condizione necessaria!)

↳ sufficiente? (BOH)

Trasformiamo il taglio di Gomory in forma standard:

$$-\sum_{j \in \Phi} \varphi(\bar{a}_{tj}) x_j + s = -\varphi(\bar{b}_t), \ s \ge 0$$

Questo vincolo può essere facilmente aggiunto al tableau corrente (operazione di **orlatura**) mantenendo l'ammissibilità duale richiesta per l'applicazione del simplesso duale. "termine noto negativo" dimostro come?

* **RISULTATO DA GOMORY**

È dimostrabile che il Taglio di Gomory termina per n° iterazioni finite e ricade in uno di questi casi:

I. nessun termine noto è frazionario ⇒ la soluz ottima del rilassato è la soluz ottima del PLI

che vuol dire riga positiva?

II. ∃ un termine noto frazionario ma la sua riga nel tableau mostra SOLO ELEMENTI POSITIVI O NULLI

⇒ dalla th duale il problema duale è illimitato → primale impossibile

⇒ PLI di partenza non ha soluzione

II.I Altro caso di impossibilità: l'ultimo taglio di Gomory ha svuotato la regione ammissibile

PROBLEMATICHE DI GOMORY

1. serve n° esponenziale di tagli
2. tailing off, sempre meno efficace ⊕ VEDI PAGINA DOPO
3. possibile instabilità numerica
4. non garantisce sol ottima ⇒ in caso la ho solo alla fine

casi impossibili scritti qua sopra II. x

PLI → rilasso → PL → duale illimitato → $\boxed{\text{Primale impossibile}}$ → PLI

22/05/2023

PROBLEMATICA NUMERO ② : TAILING OFF

Il taglio di Gomory è "PROFONDO", oltre all'ottimo continuo attualmente selezionato, elimina una regione frazionaria limitrofa CONSISTENTE

=> questo è MOLTO EFFICACE all'inizio, dopo un po' i calcoli si fanno sempre più "fini", finché i tagli non iniziano ad eliminare solo degli ε di regione ammissibile continua

=> rappresenta l'inefficienza del metodo

INFATTI IL CUTTING PLANES NON È IL PIÙ USATO PER CONVERGERE (vedi Branch & Cut)

RIPRENDIAMO E TERMINIAMO IL TAGLIO DI GOMORY...

- Alg. ESATTI:
  1. trovano l'ottimo globale ma hanno $\boxed{\text{Tempi potenzialm. esponenziali}}$

=> questo abbiamo detto nell'introdurre i piani di taglio (applicati agli alg. esatti)

riprendo il discorso...

- **RISOLUZIONE ESATTA DI UN PROBLEMA DI PLI**

  1. Risolvo il rilassamento (problema PL ricavato dal PLI rimuovendo i vincoli di interezza)

  2. se la z^*_{PL} soluzione del rilassato è <u>GIÀ INTERA</u> → <u>ho finito</u>

caso nostro

se la z^*_{PL} invece è frazionaria ovvero $\exists$ almeno una variabile in base x_B che assume valore frazionario

=> 3. Scelgo una riga ($\equiv$ var di base) <u>generatrice del taglio</u>

NB. per avere maggiore probabilità di convergenza

=> scelgo la variabile con parte frazionaria ($\varphi(x_i)$) maggiore tra tutte ($\equiv$ taglio "di più")

NB. il simplesso duale è alla base di tutti i metodi esatti

22/05/2023 riprendo il taglio di Gomory

QUESTA È LA ROBA CHE NON MI È MOLTO CHIARA

taglio di Gomory

risoluzione esatta di un problema di PLI

1. risolvi il rilassamento (rimuovo vincoli di interezza)
   - se sol intera ⇒ ho finito
   - se sol frazionaria ⇒ ∃ almeno una var $\in x_B$ con valore frazionario
2. scelgo una riga (una var di base) per generare il taglio

probabilità maggiore convergenza:

scelgo la var con parte frazionaria maggiore

$x_1 = 1{,}2 \quad 0.5$ (V)

$x_2 = {,}15 \quad 0.20$

prendo la riga di x_1

→ prendo variab in base e cerco la generica riga e la arrotondo per difetto

$$\underset{\substack{\uparrow \\ \text{unica var} \\ \text{con coeff } 1 \\ B}}{x_h} + \sum_{j \in N} \underbrace{\bar{a}_{tj} x_j}_{\text{var } N} = \bar{b}_t \quad (= x^*_h)$$

modellizzazioni:

$$x_h + \sum \lfloor a_{tj} \rfloor x_j \leq \bar{b}_t$$

$$x_h + \sum \lfloor a_{tj} \rfloor x_j \leq \lfloor b_t \rfloor$$

trovo tutti i coeff interi
↓
FORMA INTERA DEL TAGLIO DI GOMORY

COSA VUOL DIRE AH

NEL CASO IN CUI IO TROVI UNA RIGA CON $\bar{b}_t$ FRAZ MA TUTTI I COEFFICIENTI INTERI

⇒ USO LA FORMA INTERA DEL TAGLIO DI GOMORY

* cioè se io volessi la forma frazionaria COSA VUOL DIRE

=

$x_h + \sum a_{tj} x_j \leq b_t$ potremo

avendo tolto al primo qualcosa di più piccolo

$x_h + \sum \lfloor a_{tj} \rfloor x_j \leq \lfloor b_t \rfloor$ modell. intero

$\lfloor 1.7 \rfloor = 1$

$\Rightarrow \sum_j (\bar{a}_{tj} - \lfloor \bar{a}_{tj} \rfloor) x_j \geq b_t - \lfloor b_t \rfloor$ cambia il predicato

tolgo a SX una parte più piccola e a DX una più grande quindi cambia il predicato

* PERCHÉ FORZA TAT FRAZ E NON INTERA? dopo

FILE: Piani di taglio

Gomory ci da $\sum \varphi(\bar{a}_{tj}) x_j \geq \varphi(\bar{b}_t)$

$\Rightarrow$ la porto in forma standard e creo il tableau

$\downarrow$

così elimino la soluz appena trovata

vedi pag 12

vedi file ↑

---

finito gomory . . . (V)

OK FRA

vediamo parte sulla PLI... branch & bound sempre 22/05... qui sopra ci trascriverò l'esercizio di Gomory

Il metodo di Gomory è "superato"... ha comunque diverse controindicazioni.
Il B&B è il principale piano di taglio usato dai solver, è sempre un algoritmo esatto quindi prima o poi se converge all'ottimo, è globale.
Anche questo ha complessità esponenziale

schema di enumerazione completa MA non è un metodo forza bruta faccio valutaz. in modo da non dover esplorare tutta la regione

* BRANCH & BOUND

DEF generale

È un algoritmo esatto, una metodologia di enumerazione implicita basata sul principio DIVIDE ET IMPERA

ho la garanzia che esistano max num finito di vertici... (esponenz. finito!)

enumero con intelligenza, non li conto davvero tutti non analizzo tutte le soluzioni... ottimizzo questa ricerca!

spezzo il prob OG in problemi + piccoli e ne risolvo il rilass.

=> ricondurre la risoluzione di un problema di PLI alla risoluzione di sottoproblemi di PLI ma più semplici

IDEA DI BASE: si calcola l'ottimo, se ∃ variabili frazionarie si generano due problemi usando due vincoli del tipo $x_i \leq \lfloor x_i^* \rfloor$, $x_i \geq \lceil x_i^* \rceil$

NB. l'insieme delle soluzioni intere non cambia, è sempre quello.
si riduce solo la regione ammiss frazionaria
=> le soluz intere finiscono PARTIZIONATE, tra i due problemi "figli"

osservazione: richiede tempi esponenziali perché genera un num exp di sottoproblemi

BRANCHING: partiziono in sottoproblemi

BOUNDING: soluz dei sottoprobl rilassati permettono di PRUNARE altri problemi

* In generale... si ha

PLI		RILASS.
$\min c^t x$	$\Rightarrow$	$\min c^t x$
$Ax = b$		$x \in P$:
$x \geq 0$ intere		$P = \{x \geq 0 \ Ax \geq b\}$ $x \in \mathbb{R}$

* siano $x^{(0)}$ e $z^{(0)} = c^t x^{(0)}$ la soluzione ottima del rilassato continuo P
  * $\to x^{(0)} \in \mathbb{Z}$ FINE
  * $\to$ altrimenti $\exists i : x_i \in \mathbb{R} \Rightarrow$ da PLI_0 $\to$ PLI_1, PLI_2 sfruttando delle...

OPERAZIONI DI BRANCHING

Ad ogni soluzione ottima non intera, seleziono una variabile frazionaria x_i per generare due sottoproblemi: $x_i \leq \lfloor x^*_{i\,PL} \rfloor$

$x_i \geq \lceil x^*_{i\,PL} \rceil$

$$z^*_{PLI} = \min\{c^t x : x \in S^{(0)}\} = \min\{z^{(1)}, z^{(2)}\}$$ con $z^{(i)} = \min\{c^t x : x \in S^{(i)}\}$ $i = 1,2$

* le soluzioni intere sono:

$$X^{(0)} = \{x \in \mathbb{Z}^n : Ax = b ; x \geq 0\}$$

figlio ① $\to X^{(1)} = \{x \in \mathbb{Z}^n : Ax = b ; x \geq 0, \ x_i \leq \lfloor x_i^{(0)} \rfloor\}$

figlio ② $\to X^{(2)} = \{x \in \mathbb{Z}^n : Ax = b ; x \geq 0, \ x_i \geq \lceil x_i^{(0)} \rceil\}$

valgono sempre i seguenti:

$X^{(0)} = X^{(1)} \cup X^{(2)}$ metodo corretto $\to$ "non perde soluz intere"

$X^{(1)} \cap X^{(2)} = \emptyset$ metodo efficace $\to$ "partiziona in maniera vantaggiosa"

RISPOSTA A "quando smetto di branchare?"

CRITERI DI FATHOMING

Voglio evitare di esplorare tutti i figli quando è possibile.
Qui alcuni criteri di decisione:

1. rilassamento continuo è impossibile

I vincoli iniziali $Ax=b$, $x \geq 0$ sono incompatibili con i vincoli di branching relativi al percorso per arrivare al nodo corrente.

=> se spezzo e do origine a una regione vuota mi fermo
=> come fa a succedere? i vincoli hanno semipiani opposti quindi intersez nulla => nodo chiuso

2. rilassamento continuo da una soluzione intera

Ho già trovato l'ottimo! Sono apposto
=> se spezzoni ancora, non havrei nulla di migliore

3. regola di bounding

Termino la suddivisione in sottoproblemi sfruttando info da altre parti dell'albero
=> risolvendo il rilassamento continuo, scopro che è peggiore della miglior soluz intera che ho già trovato

▶ CRITERI DI BOUNDING

Suppongo di aver trovato x^e soluz ammissibile intera: $c^t x^e \geq z^*_{PLI}$

Considero un nodo k qualsiasi:

soluz intere per nodo k: $X_k = P_k \cap \mathbb{Z}^n$ (soluz $\mathbb{R}$ per nodo k (rilass continuo); punti interi del problema)

$$z_k = \min\{c^t x : x \in P_k\}$$
$$\leq$$
$$\min\{c^t x : x \in X_k\}$$

=> $\boxed{z_k \geq c^t x^e}$ allora è inutile proseguire il branching da questo nodo k trovato

---

in P_2 ho il rilass che mi da $z^* \in \mathbb{Z}$ $z^* = 71 \equiv z^*_{PL}$ Lower Bound x tutto l'albero

=> la segno e continuo ad analizzare altri nodi

in P_3 il rilassam mi da $w^* \in \mathbb{Z}$ $w^* = 83.5$

=> il sotto albero che genera ha come sol di partenza 83.5 PEGGIORE di 71

e questo essendo la prima soluz è un LOWER BOUND al piuttosto

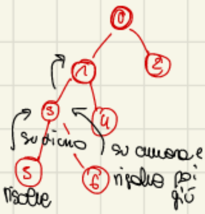
mi metto ad analizzare il

la soluz del relax continuo del padre è un lower bound per i figli

VEDI ESEMPIO NEL FILE: BranchandBound

STRATEGIE DI ESPLORAZIONE: CRITERI DI BACKTRACKING

se branch risolve TUTTO il sottoalbero di un nodo prima di passare al fratello



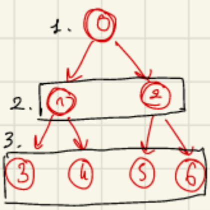
1. **Tecnica Depth First (ricerca in profondità)** → backtracking per [illegible] problemi
   Dato un nodo padre si considera immediatamente il primo dei suoi due figli, finché una delle condizioni di fathoming non costringe l'algoritmo a risalire di livello.

sfrutta la ricorsività dei rilassati P_i? come genero da $P_0 \to x^*_{(0)} \to$ genero vincolo
⇒ quindi sfrutterò il tableau ottimo già ed elimino un vincolo < torna

Vantaggi: arriva velocemente a soluzioni ammissibili; pochi nodi aperti in memoria; implementazione più semplice perché sfrutta la ricorsività del linguaggio;

P_i

Svantaggi: in caso di scelta errata si ritorna indietro solo dopo aver attraversato tutto il sotto-albero sbagliato.



2. **Tecnica Breadth First (ricerca in ampiezza)**
   Si espande interamente ogni livello dell'albero. Si risolvono **tutti** i nodi di un livello e solo successivamente si passa al livello successivo, considerando i figli dei nodi appena elaborati.

chiude i livelli prima
non scende in profondità

Vantaggi: visita l'albero in maniera bilanciata;

Svantaggi: molti nodi aperti in memoria; difficile aggiornamento della soluzione imperante (ottimo corrente).

3. **Tecnica Best-Bound (ricerca sulla limitazione)**

Si sceglie il nodo non ancora elaborato con maggior potenziale (più promettente) guardando il valore del rilassamento continuo (problema di minimo). Due possibili varianti:

AMMISSIBILITÀ

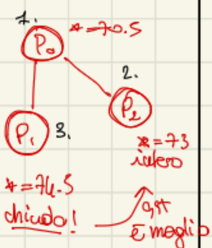
- **Best-Bound Feasibility** (*promettente* = valore maggiore del rilassamento continuo) scelgo un intero ammiss seguo il sottoprobl con ottimo >

OTTIMALITÀ

- **Best-Bound Optimality** (*promettente* = valore minore del rilassamento continuo) se voglio prima la sol ottima, e vabbè seguo la minore

Vantaggi: vengono analizzati mediamente meno nodi; aggiornamento della soluzione imperante veloce.

Svantaggi: molti nodi aperti in memoria (soprattutto optimality).



OSSERVAZIONE: ho in mano un ottimo intero > del rilass ottimo di un nodo ricavato in un modo qualsiasi

⇒ a logica potrei fermarmi

ma se continuo posso trovare ottimi multipli chiudendo i nodi

(ex. una soluz per la produz di un prodotto coinvolge linee vecchie, ma lo stesso la posso ricavare usando delle linee nuove)

ogni padre è LOWER BOUND per i problemi figli

STRATEGIE DI ACCELERAZIONE (prob di min)

Il bounding è tanto più efficace quanto più sono buone le soluzioni ammissibili (UPPER BOUND) trovate DURANTE LA RICERCA, e quanto è più STRINGENTE il rilassamento continuo (LOWER BOUND) nei nodi.
Quindi:

1. inizializzo BAB con una soluz euristica: così ho una soluzione intera con cui fare bounding fin da subito
2. applicare procedimenti euristici durante l'esplorazione dell'albero con una certa frequenza
3. ricorrere a una formulaz stringente del PLI: con un rilassato migliore posso chiudere prima il problema

così è più stringente il LOWER BOUND

OSSERVAZIONI SUL METODO:

- Si possono combinare tra loro più tecniche di esplorazione dell'albero oppure implementarne di più complesse (backtracking);
- è spesso importante decidere su quale variabile fare branching, nel caso ce ne sia più di una frazionaria (random, quella a parte frazionaria maggiore, quella a parte frazionaria più prossima a 1/2, etc.);
- si possono utilizzare rilassamenti diversi da quello continuo (esempio: rilassamento Lagrangiano);
- nel caso di problemi puramente binari, il branching degenera nel fixing di una variabile a 0 o a 1 (eliminandola di fatto dal sottoproblema);
  un figlio ottiene 0 e uno 1 così ho fissato la x; a un val intero
- si possono utilizzare tecniche di branching più complesse (vincoli di branching che coinvolgono più di una variabile, strong branching).
  solo informativo

SCEGLIERE VAR:
random: a caso
greedy: parte fraz maggiore
migliore: parte fraz vicino a 1/2
che cazzo vor di

vabbeh da qua poi c'è l'esempio de l'esercizio (scritto sul quadernetto)

29|05|2023 FILE: 10 grafi

* TEORIA DEI GRAFI

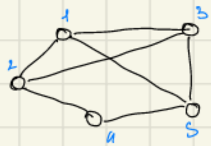
VERTICI ADIACENTI: legati da un lato

- Un po' di definizioni

caso particolare

Definizione: grafo non orientato

Un *grafo non orientato* (o indiretto) è una coppia $G = (V, E)$ in cui V è un insieme finito di n vertici $\mathbf{V} = \{v_1, v_2, \cdots, v_n\}$ ed $\mathbf{E} = \{e_1, e_2, \cdots, e_m\}$ è un insieme finito di *m lati o spigoli*, dove ogni lato $e_k = (v_i, v_j)$ è formato da una coppia non ordinata di vertici distinti v_i, v_j detti estremi del lato.



- Indichiamo Γ l'insieme dei vertici adiacenti (star) ad un vertice dato. totalità dei vicini

$\Gamma(v_1) = \{v_2 ; v_3\}$

- I lati adiacenti sono quelli che condividono un vertice ad esempio (v_1, v_2) (v_1, v_3)

GRADO

Definizione: grado di un vertice

Il numero di lati incidenti in un vertice v viene detto *grado* del vertice e si indica con $\deg(v) = |\Gamma(v)|$.

cardinalità dell'insieme dei vertici adiacenti a v_i

= il numero di lati che in v_i incidono

COMPLETO

Definizione: grafo completo

Un grafo si dice *completo* se contiene tutti i possibili lati cioè se tutte le coppie di vertici sono adiacenti $E = \{(v_i, v_j) : v_i, v_j \in V, v_i \leq v_j\}$.

- ∃ tutti i lati possibili, generabili
- è completamente magliato

OSSERVAZIONE: qual è il legame lati-vertici?

sia n il numero dei vertici e sia G un grafo COMPLETO

∃ allora $\binom{n}{2}$ lati (ovvero combinazioni di vincoli a due a due) (senza ordine!)

• ∀ grafo possiamo andare a definire...
CAMMINI

CAMMINO E CICLO

Definizione: cammino

Dato un grafo indiretto $G = (V, E)$, si dice *cammino* tra i vertici v_1 (inizio) e v_{k+1} (fine) una sequenza $\{e_1, e_2, \cdots, e_k\} \subseteq E$ di lati tali che ciascuno è adiacente al successivo ossia $e_1 = (v_1, v_2),\ e_2 = (v_2, v_3), \cdots,\ e_k = (v_k, v_{k+1})$.
Un cammino è detto **ciclo** se $v_1 = v_{k+1}$.
inizia dove termina

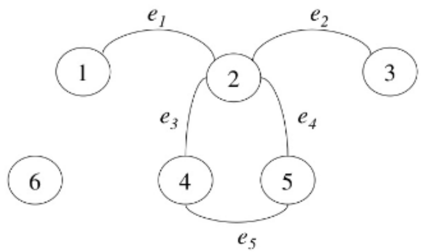


Figura: Grafo non connesso: esempio di cammino.

semplice: passo una volta per lato

elementare passo una volta per vertice

Definizione: cammino semplice e cammino elementare

Un cammino si dice **semplice** se non usa lo stesso lato più di una volta, mentre è **elementare** se non usa mai due volte uno stesso vertice.

VERTICI E GRAFI CONNESSI

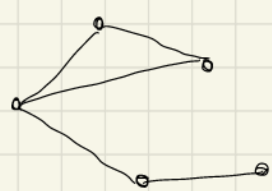
Definizione: vertici connessi

Due vertici v_i e v_j si dicono *connessi* se esiste almeno un cammino che li collega.

Definizione: grafo connesso

Un grafo si dice *connesso* se ogni coppia di vertici è connessa.

un grafo non è connesso se ∃ almeno un vertice non raggiungibile dagli altri



grafo connesso non completo

SOTTOGRAFO E ALBERO DI SUPPORTO

Definizione: sottografo

Un grafo $G' = (V', E')$ è detto *sottografo* di $G = (V, E)$ se $V' \subseteq V$ e $E' \subseteq E$. Un sottografo G' è detto *grafo parziale* di $G = (V, E)$ se $V' = V$.

SOTTOGRAFO parte dei vertici e lati

SOTTOGRAFO PARZIALE: tutti i vertici e parte di lati

non orientato → il cammino minimo invece sì

Definizione: albero di supporto

Dato un grafo non orientato $G = (V, E)$ si dice *albero di supporto* (o albero ricoprente) per G il grafo parziale $G'(V, E')$ connesso e privo di cicli.

NB. i problemi di ricerca dell'albero di supporto a costo minimo sono FACILI

trovo un albero di supporto se, dato un grafo connesso, rimuovo un lato (e privo di cicli)

* QUANTI ALBERI DI SUPPORTO ESISTONO?

sia G con n vertici (V) e m lati (E). Un AdS contiene tutto V e non E, è connesso e privo di cicli.

⇒ In quanti modi posso avere un grafo connesso e privo di cicli? (senza usare tutti i lati)

∃ 2 tipi di problemi per la TH dei grafi:

- albero di supporto (non orientato)
- cammino minimo (orientato)

usati nei problemi di accoppiamento e assegnazione di missioni

↓

se ho +2 sottoinsiemi allora anche prob del trasporto

Definizione: grafo bipartito

Un grafo si dice *bipartito* se esiste una partizione dell'insieme dei suoi vertici V in due sottoinsiemi disgiunti V_1 e V_2 tale che ogni lato del grafo abbia un estremo in V_1 e l'altro in V_2.

OSSERVAZ: i grafi bipartiti sono privi di cicli dispari e se un grafo contiene cicli dispari, non è bipartito

* PROPRIETÀ ELEMENTARI DEI GRAFI

$\boxed{\text{TH 1}}$

Teorema 1

Dato un grafo $G = (V, E)$ con m lati si ha che

$$\sum_{v \in V} \underbrace{deg(v)}_{\text{grado dei vertici}} = 2\overset{\text{num lati}}{m}$$

la somma di tutti i gradi è uguale a $2 \cdot m$

Corollario

In ogni grafo il numero di vertici di grado dispari è pari.

DIM SEMPLICE ED IMMEDIATA:

posso riscrivere $\sum_v deg(v) = 2m$

OSSERVAZ: chiamo V_p vertici grado pari, V_d vertici grado dispari $\Rightarrow \sum_{V_p} deg(v) + \sum_{V_d} deg(v) = 2m$

$\Rightarrow$ $2m$ è numero pari, ma la sommatoria dei gradi dei vertici con grado pari è SICURAMENTE pari

$\Rightarrow$ per forza allora la sommatoria dei gradi dispari dovrà essere anch'essa un numero pari

vertice su cui insistono 5 lati:

$\Rightarrow$ il numero di questi è pari

* PROBLEMI HAMILTONIANI

esempio: raccolta rifiuti con cassonetti

Definizione: cammini e cicli hamiltoniani

Un cammino, o un ciclo, si dice **hamiltoniano** se visita tutti i vertici del grafo una e una sola volta. Un grafo che contenga almeno un $\boxed{\text{ciclo}}$ hamiltoniano è detto **grafo hamiltoniano**.

proprietà hamiltoniana : il ciclo o cammino visita TUTTI i vertici UNA E UNA SOLA volta.

↓

grafo vuole un ciclo hamiltoniano!

esistono di solito delle condiz. semplici per verificare la proprietà di un grafo

Grafi hamiltoniani ⇒ non vale per i grafi in cui cerco cicli hamiltoniani

caso grafo con + di 3 vertici

Non sono note condizioni "semplici" per verificare se un dato grafo $G = (V, E)$ con $|V| \geq 3$ sia hamiltoniano oppure no. Banalmente:

necessaria

1. condizione necessaria (ma non sufficiente): G deve essere connesso, con vertici tutti di grado ≥ 2;

sufficiente

2. condizione sufficiente (ma non necessaria): ogni vertice del grafo ha grado maggiore o uguale a $|V|/2$ (Teorema di Dirac).

NB. un grafo completamente magliato è di sicuro hamiltoniano

CONDIZIONE SOLO SUFFICIENTE

EXTRA: TH DI DIRAC È CONOSCIUTO ANCHE COME IL TH DI ORE

Un grafo è hamiltoniano se comunque io prendo due vertici v_i, v_j non adiacenti: [$(v_i, v_j) \notin E$] vale che:

$$\underbrace{deg(v_i)}_{\geq n/2} + \underbrace{deg(v_j)}_{\geq n/2} \geq n \quad (n \to |V|)$$

⇒ il grado di ciascun vertice deve essere almeno uguale a $\frac{n}{2}$

dirac è il caso particolare infatti dice che ogni vertice deve avere grado $\geq \frac{|V|}{2}$, ma valendo per entrambi

v_i e v_j ho $\geq \frac{|V|}{2} + \frac{|V|}{2} = |V|$

⇒ ritrovo ORE

VARIANTE SUI LATI

ex: raccolta rifiuti porta a porta
↓
ogni casa non è più un vertice come i consueti dei prob hamiltoniani
⇒ ho le case sulle vie che considero lati da percorrere

ciclo a costo minimo euleriano è FACILE

* PROBLEMI EULERIANI

cammino: sequenza di lati adiacenti
ciclo: cammino chiuso

Definizione: cammini e cicli euleriani

Un cammino, o un ciclo, si dice **euleriano** se attraversa tutti i lati del grafo una e una sola volta. Un grafo che contenga almeno un ciclo euleriano è detto **grafo euleriano**.

ma questo problema è facile e computazion. diverso da quello hamiltoniano

Grafi euleriani

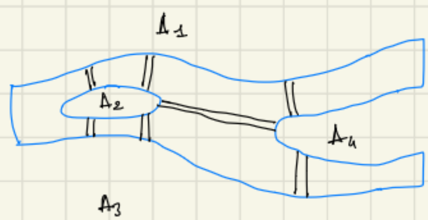
Eulero ha individuato la condizione necessaria e sufficiente affinché un grafo non orientato sia euleriano:

Teorema di Eulero (grafi indiretti): Un grafo indiretto $G = (V, E)$ è euleriano se e solo se è connesso con vertici tutti di grado pari.

↳ condizione necessaria e sufficiente per i grafi non orientati

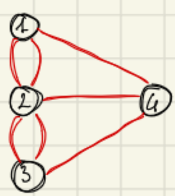
*uno stesso problema modellizzato in due modi diversi può essere risolto in maniera diversa (raccolta spazzatura – Euleriano vs Hamiltoniano)

DILEMMA DEI PONTI DI KÖNIGSBERG



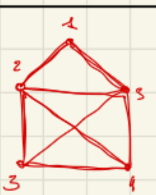
Ci sono 7 ponti in totale.

* È possibile per chi abita in una Ai qualsiasi, attraversare tutti i ponti una ed una sola volta e poi tornare a casa?
EULERO: no, non è un grafo euleriano



∃ vertici con grado dispari ⇒ TUTTI 4 vertici con grado dispari

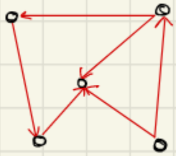
↑ multigrafo: ∃ per due stessi vertici più di un lato



→ va in contrasto al TH di eulero?
$\Gamma(v_1, v_2, v_5)$ = pari
$\Gamma(v_3, v_4)$ = dispari → 2 vertici con grado dispari (sempre in num pari)

⇒ NON È EULERIANO!

in realtà non ho seguito un ciclo euleriano, non sono rientrato nello stesso nodo
⇓
∃ solo un CAMMINO EULERIANO
devo partire da un vertice con grado dispari, anche il nodo finale dovrà avere grado dispari
(∃ un cammino euleriano ∀ coppia a grado disp)



GRAFO ORIENTATO

non si parla più di connessione ma raggiungibilità

Definizione: grafo orientato

Un *grafo orientato* (diretto) è una coppia $G = (V, A)$ in cui $V = \{v_1, v_2, \cdots, v_n\}$ è un insieme finito di n nodi ed A è un insieme finito di m archi, dove ogni arco (v_i, v_j) è una coppia ordinata di nodi.

Definizione: nodi raggiungibili

Un nodo v_j è **raggiungibile** dal nodo v_i se esiste almeno un cammino (orientato) da v_i a v_j.

Definizione: grafo connesso

Un grafo orientato G si dice **connesso** se il grafo ottenuto ignorando il verso di percorrenza degli archi è connesso.

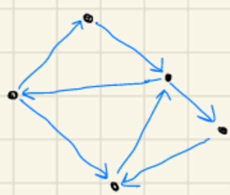
Definizione: gradi di un nodo

Ogni nodo v del grafo ha:

- un insieme di nodi successori (**forward star** $\Gamma^+(v)$) e un **grado uscente** ($deg^+(v) = |\Gamma^+(v)|$) che rappresenta il numero di **archi uscenti** da v;
- un insieme di nodi predecessori (**backward star** $\Gamma^-(v)$) un **grado entrante** ($deg^-(v) = |\Gamma^-(v)|$) che rappresenta il numero di archi **entranti** in v.
- $deg(v) = deg^-(v) + deg^+(v)$.

Definizione: grafo fortemente connesso

Un grafo orientato G si dice **fortemente connesso** se comunque presa una coppia di nodi esiste un cammino orientato che li collega (ogni nodo è raggiungibile da qualsiasi altro nodo del grafo).



fort connesso

non fort connesso

✱ RAPPRESENTAZIONE DI GRAFI MEDIANTE MATRICI

$$D = |V| \overbrace{\begin{bmatrix} 1 \dots 0 \dots 1 \, 0 \\ \\ \\ \end{bmatrix}}^{|E|}$$

$d_{ij} = \exists$ un vertice i su cui incide il lato j

MATRICE INCIDENZA VERTICI-LATI (GRAFO INDIRETTO)

Definizione: matrice di incidenza per un grafo non orientato

La *matrice di incidenza vertici-lati* D di un grafo non orientato $G = (V, E)$ è la matrice $|V| \times |E|$ con elementi

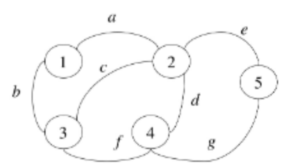
$$d_{ij} = \begin{cases} 1 & \text{se il } j-\text{esimo lato è incidente nel vertice } i, \\ 0 & \text{altrimenti.} \end{cases}$$

m colonne (lati)
×
n righe (nodi)

NB. Ogni riga ha tanti 1 quanto il suo grado

NB. Ogni colonna i (lato) avrà due elementi "1", essendo gli estremi del lato i

ESEMPIO:



lati - i

	a	b	c	d	e	f	g	
1	1	1	0	0	0	0	0	= 2
2	1	0	1	1	1	0	0	= 4
3	0	1	1	0	0	1	0	= 3
4	0	0	0	1	0	1	1	= 3
5	0	0	0	0	1	0	1	= 2

nodi - i

$\Rightarrow \sum_i d_i = 10 = 2 \cdot 5 = 2m$ ✓

$\sum_i \deg(v_i) = 2m \quad = \sum_i d_i$

$$\begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

Il grafo in figura è bipartito? Un grafo è bipartito se e solo se non contiene cicli dispari. Poiché il grafo contiene il ciclo dispari $\{a, c, b\}$ non è bipartito.

- Qual è il determinante della matrice di incidenza associata a questi tre lati? = det nullo
- La matrice di incidenza vertici-lati è $\boxed{\text{TUM}}$ se e solo se il grafo è bipartito.

=

- grafo bipartito <=> matrice TUM

↓

∃ partizione vertici (questa è rotta dalla presenza di cicli dispari) => ∃ partizione di righe (tutti elementi non nulli sono concordi quindi le righe vanno in insiemi diversi)

GRAFO BIPARTITO È CONDIZIONE NECESSARIA E SUFFICIENTE a differenza del grafo non orientato, ho sempre le colonne con 2 elementi $\neq 0$, ma stavolta sono discordi

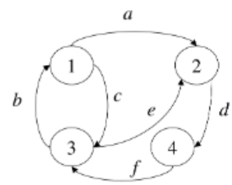
MATRICE INCIDENZA NODI ARCHI (grafo orientato)

Definizione: matrice di incidenza per un grafo orientato

La *matrice di incidenza nodi-archi* D per un grafo orientato $G = (V, A)$ è la matrice $|V| \times |A|$ con elementi

$$d_{ij} = \begin{cases} +1 & \text{se il } j\text{-esimo arco è uscente dal vertice } i, \\ -1 & \text{se il } j\text{-esimo arco è entrante nel vertice } i, \\ 0 & \text{altrimenti.} \end{cases}$$

ESEMPIO



	a	b	c	d	e	f
1	1	-1	1	0	0	0
2	-1	0	0	1	-1	0
3	0	1	-1	0	1	-1
4	0	0	0	-1	0	1

NB. le matrici di incidenza nodi-archi sono TUM

$\Rightarrow$ grazie alle colonne con 2 elementi non nulli e discordi con la partizione di righe esiste e la cond. sufficiente è banalmente confermata

$\Rightarrow$ il problema è facile

sia le righe
che le colonne
sono i vertici

MATRICI DI ADIACENZA

Sono quadrate. $m=n$, sia per grafo orientato che indiretto.

Definizione: matrice di adiacenza

La **matrice di adiacenza** Q per un grafo non orientato G=(V,E) o orientato G=(V,A) è la matrice $|V| \times |V|$ con elementi

$$q_{ij} = \begin{cases} 1 & \text{se } (i,j) \in E \text{ (oppure } A\text{)}, \\ 0 & \text{altrimenti.} \end{cases}$$

NB. nel caso di mat di adiacenza per grafi non orientati ho matrici SIMMETRICHE (è ovvio, se $\exists\, a_{ij} = 1$ allora anche $a_{ji} = 1$)

* nei grafi orientati questa costruzione non è banale, di solito non è simmetrica.

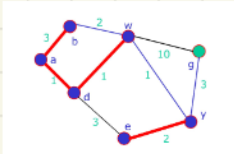
GRAFI PESATI: si parte da una matrice di adiacenza

$$\begin{cases} 1 \rightarrow \text{"peso" dell'arco} \\ 0 \rightarrow \infty \quad \text{non c'è la connessione} \end{cases}$$

05/08/2023 FILE: minimum_spanning_tree

* CONCETTI SUGLI ALBERI

connesso
privo di cicli
⇓ sufficiente
albero



Foresta con due alberi

1. Un albero in un grafo G=(V,E) è un sottografo connesso di G contenente k vertici e k-1 lati
2. Un albero è detto di supporto o massimale se k=n ovvero se è un grafo parziale contenente tutti i vertici di G.
3. In ogni albero ogni nodo avente grado 1 è detto foglia
4. Un grafo le cui componenti connesse sono alberi è detto foresta (= grafo aciclico formato dall'unione di alberi disgiunti)

PROPRIETÁ ALBERI

Teorema

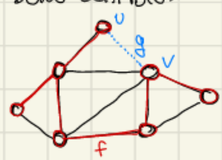
Valgono le seguenti proprietà:

DEF ALBERO:
1. un grafo è un albero se e solo se è connesso e privo di cicli;
2. ogni albero contenente più di un vertice contiene almeno una foglia;
3. in un albero ogni coppia distinta di vertici è connessa da uno e un solo cammino;

DEF CICLO FONDAMENTALE:
4. se ad un albero si aggiunge un lato il grafo risultante contiene uno e un solo ciclo detto *ciclo fondamentale*.

se aggiungo "f" all'albero trovo un ciclo
⇒ il suo ciclo fondamentale

TH DELLO SCAMBIO:



▶ se tengo il lato con cui ho trovato il ciclo fondamentale e ne scarto un altro non collegato ad una foglia, trovo un altro albero

* per il MST ∃ una doppia modellizzazione... prima il contesto comune

questo tipo di modellizzazione si usa per i problemi di reti

▶ **MST**: problema dell'albero a costo minimo

Sia $G=(V,E)$ un grafo non orientato, con $|V|=n$ ed $|E|=m$.
Ogni lato $e \in E$ è associato ad un costo c_e.

⇒ il problema MST trova l'albero ricoprente $G_T=(V,T)$ ossia un grafo connesso e privo di cicli contenente tutti i nodi del grafo, avente costo minimo.

notazione:

$$E(S) = \{ e=(i,j) \mid i \in S, j \in S \}$$
$$\delta(S) = \{ e=(i,j) \mid i \in S, j \in V \setminus S \}$$

* **MST (1)**

Caratteristiche:
→ 1. grafo aciclico
→ 2. grafo contenente n-1 lati

$$(MST1) \quad \min \sum_{e \in E} c_e x_e$$
$$\sum_{e \in E} x_e = n-1$$
$$\sum_{e \in E(S)} x_e \le |S|-1 \quad \forall S \subset V : S \neq \emptyset \quad (2)$$
$$x_e \ge 0 \text{ intero}, e \in E$$

OSSERVAZIONE: i vincoli (2) sono i subtour-elimination e sono in numero esponenziale

* MST (2)

Caratteristiche: → 1. grafo connesso
↘ 2. grafo contenente n-1 lati

$$(MST2)\quad \min \sum_{e \in E} c_e x_e$$

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

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subset V : S \neq \emptyset \quad (S)$$

$$x_e \geq 0 \text{ intero}, e \in E$$

OSSERVAZIONE: i vincoli (S) sono i connectivity constraints e sono in numero esponenziale

OSSERVAZIONI:

- le variabili x_e hanno significato <=> hanno dominio binario {0,1}
  Ma comunque MST1, MST2 li definisco intero
  Perché? Boh forse è perché è un rilassamento
- MST ha tempi polinomiali => FACILE

ALGORITMO DI KRUSKAL

- algoritmo greedy, usa scelte con criterio locale e non le riconsidera dopo

- algoritmo esatto, complessità O(m·log n) con struttura dati UNION FIND

Algoritmo di Kruskal:

$T := \emptyset$
ordina E per costi non decrescenti in una lista ordinata L;
repeat
  individua il primo lato e (a costo minimo) appartenente a L
  $L := L \setminus \{e\}$ — passa tutti i lati
  if $T \cup \{e\}$ non ha cicli **then** — butta via quelli che generano cicli
    $T := T \cup \{e\}$
  end if
until $|T| = n - 1$ — si ferma quando ne ha selezionati (n-1)

KRUSKAL INVECE EVITA I CICLI

usa il concetto di connettività

ALGORITMO DI PRIM

"Parte da un vertice e costruisce una componente"

Alternativo a Kruskal basato sul concetto di selezione del lato a costo minimo appartenente ad un insieme δ(S) (tutti i lati che collegano un insieme di vertici dati, al resto del mondo) detto taglio dove S è un sottoinsieme di nodi inizialmente a cardinalità 1 che cresce di ogni passo fino all'inserimento di n-1 lati.

31/05/2023

- ESERCITAZIONE 4 : fine

- ESERCITAZ 5

5/06/2023

vetrina ..z mannini

Kruskal e Prim sono esercizi facili

minzo: algoritmo di Prim

FOE: commesso viaggiatore

poi FOO: shortest Path facciamo solo parte A la parte B è per l'orale

CONCLUSIONE VEDIAMO UN TE

come si risolve?

descrizione (con definizioni)

gennaio 2022,

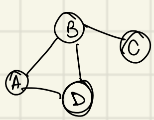
→ introduci il TSP → ciclo hamiltoniano con costo minimo ⇒ grafo hamiltoniano

100% ha soluzione!

> dai esempio di grafo non orientato, connesso con almeno 4 lati 4 vertici e senza soluzione

vengo la COND. NECESSARIA

→ ogni nodo ha grado ≥ 2



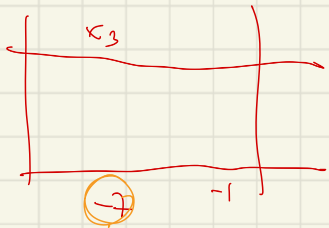
grazie!

legame fra termini noti (archi) e soluzione degenere

Vero o Falso:

1. amiss nuova sol base dopo pivot, garantita da:

=> entrata in base da CCR minore



migliora di -7 (x unità) la FO

(ottimalità)

STABILISCO

CCR migliore => PIÙ PROMETTENTE -> DIREZIONE SIMPLEX

l'ammissibilità è data dalla uscente => rapporti minimi

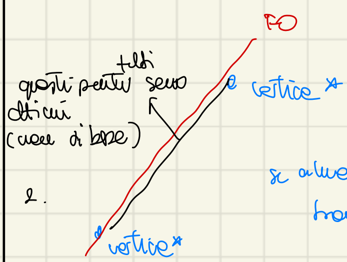
da il valore che avrei in base

la variab con rapporto non minore manda alcune var a valore NEGATIVO

esco dalla reg. ammissibile

bland non bypassa ammissibilità

2\.



se almeno 2...

trovo un segmento

3. due fasi: prob inammissibile?

dopo prima fase var aus ... rimangono in base con valore $\neq 0$

condiz. dequene quo non riguarda l'inammissibilità

questo riguarda x la seconda base

4. P^* relax

Q^* duale → se inammissibile quindi P^* inammissibile

dalla PL, Q^d inammissib ⇒ P^* illim o inammissibile

⇒ devi fare la con...

DUALITÀ FORTE:

P^* inamm ⇒ Q^d inamm

P^* illim ⇒ Q^d inamm o illim

minuto 1:00:00