Tipologie esercizi

↳ ① Modelli
② Program. lineare - simplesso
③ Program. lineare - risol. grafica
④ Program. lineare - dualità
⑤ Grafi - albero copertura
⑥ Grafi - albero cammini minimi → 6.A Spt-aciclico/ordinam. topologico; 6.B Dijkstra; 6.C Bellman-Ford
⑦ Grafi - Flusso massimo
⑧ PLI - tagli di Gomory
⑨ PLI - branch and bound
⑩ PLI - branch and bound zaino

② Simplesso

- Dato un problema (PL), riportarlo in forma standard:

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

TRASFORMAZIONI
max → - min
$x_i \leq 0 \rightarrow -x_i \geq 0$
vincolo $\leq K$ → aggiungo $+s_i$ var slack, $s_i \geq 0$, vincolo =
vincolo $\geq K$ → sottraggo $-s_i$ var slack, $s_i \geq 0$, vincolo =

Se ho $I \subset A$, ho una base di partenza $\mathcal{B} = \{s_1, s_2\}$ $A_B = I$ (di solito)

Se non ho sottomatrice I in A uso fase 1 per trovare base ammissibile (vedi sotto)

- ALGORITMO

↳ dato $\mathcal{B}$, calcolo $x_B = A_B^{-1} b$, c_N^T, c_B^T, z^*
A_B, A_B^{-1}, $A_B^{-1} A_N$, $r_N^T = c_N^T - c_B^T A_B^{-1} A_N$ costi ridotti

↳ $r_N \geq 0 \leadsto x_B, z^*$ ottimi, TERMINO

↳ altrimenti scelgo il primo i tc $r_N[i] < 0$ ⇒ entra la var. associata a i (BLAND 1977)

dividdo x_B con $(A_B^{-1} A_N)_i$ componente per componente

- $\exists\, \theta_{min} > 0$, esce var. associata a θ_i, entra var. con valore θ_{min} REITERO
- tutti $\theta \leq 0$, ottimo illimitato

NB inversa 2x2

$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ $A^{-1} = \frac{1}{\det A} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

- FASE 1

risolvo con simplesso

min $w_1 + w_2$, aggiungo al primo vincolo w_1, al secondo w_2 (non avrò mai più di 2 vincoli in esame)

$w_i \geq 0$, se trovo soluzione ottima tc w_i fuori base $\forall i$ ⟶ ho $\mathcal{B}$ per simplesso

, se sol. ottima ha w_i in base ⟶ (PL) non ammette soluzione

③ Risol. grafica

Dato (PL) in qualsiasi forma, disegno sul piano le zone ammissibili

se min → - gradiente

se max → + gradiente

→ Traccio la perpendicolare al gradiente e traslo nella direz. di ± gradiente (+ max, - min) fino a quando interseco ultimo vertice

→ Alternativamente, traccio le normali ai vincoli in ogni vertice, se gradiente c cono delle normali ⇒ ho ottimo

④ Dualità

Dato (PL) costruisco duale:

max → min, min → max

(PL) min/max $c^T x$
$Ax \gtreqless b$
$x \gtreqless 0$

↳ ∀ vincolo di PL c'è y_i duale il cui dominio è determinato da vincolo
∀ variabile di PL c'è vincolo duale il cui segno è determinato da dominio variabile
↳ Coeff. di f. obiettivo: vettore colonna b del (PL) per var duali
↳ termini noti vincoli duali: c^T del problema (PL)
Coeff. di y_i nel vincolo duale associato a var k-esima di (PL): colonna k-esima di A

→ SEGNI

(PL)		(duale)		(PL)		(duale)	
max	→	min		min	→	max	
var	→	vinc	uguali	var	→	vinc	opposti
vinc	→	var	opposti	vinc	→	var	uguali

se x_i libera → vincolo con =
se vincolo = → y_i libera

→ SCARTI COMPLEMENTARI

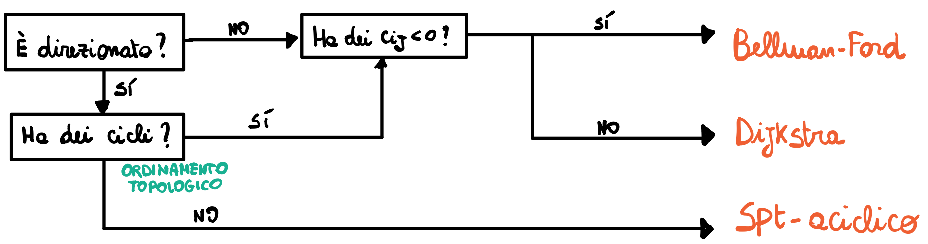
$\begin{cases} (\text{vincolo PL})\ y_i = 0 \\ (\text{vincolo duale})\ x_i = 0 \end{cases}$ data $\bar{x}$, se sostituendo nel sistema trovo $\bar{y}$ ammissibile, allora $\bar{x}$ ottimo

⑤ Albero di copertura
(DI COSTO MINIMO)

Dato grafo non direzionato, uso algoritmo Kruskal: |T| numero di nodi, c_{ij} costo
Ordino (i,j) crescentemente per c_{ij}, aggiungo uno a uno solo se non formano ciclo,
mi fermo a |T|-1 archi aggiunti

⑥ Albero dei cammini minimi (r = radice)

∃ modello??



- ORDINAMENTO TOPOLOGICO
  Dato T, cerco nodo senza predecessore, se ∃ gli assegno etichetta progressiva, lo rimuovo da |T|, REITERO
  se ∄ allora c'è un ciclo
- SPT-ACICLICO
  Esploro nodi nell'ordine dell'ordinam. topologico. Inizializzo d[r] = 0, P[r] = r e tutti gli altri d[i] = M P[i] = 1
  Esamino FS(i), se c_{ij} + d[i] < d[j] aggiorno d[j] = c_{ij} + d[i] e P[j] = i
  Al termine costruisco albero coi predecessori

- Dijkstra
  Inizializzo come SPT. Esploro partendo da r.
  Esamino FS(i) per nodi j non estratti (fissati), aggiungo j a lista Q e aggiorno se c_{ij} + d[i] < d[j] d[j] = c_{ij} + d[i] e P[j] = i. Fisso i e d[i]
  Estraggo i da lista t.c. d[i] è il minimo tra gli i ∈ lista REITERO
  IMPORTANTE
  Quando lista Q = ∅ termino.

- Bellman-Ford
  Inizializzo come SPT. K[i] = 0 ∀i, K[r] = 1 → n entrate in lista Q. Parto da r
  Esamino FS(i), se c_{ij} + d[i] < d[j] aggiorno come Dijkstra e se j ∉ Q lo aggiungo e incremento K[j] di 1 (solo se ho aggiornato d[j] altrimenti NON entra in lista!!)
  Se K[j] ≥ |T| c'è ciclo negativo, altrimenti REITERO (estraggo i da Q a caso)
  Se Q = ∅ concludo.

MODELLO PL FLUSSO

⑦ Flusso massimo

Dato albero direzionato, capacità e condizione iniziale:

Costruisco grafo residuale, cerco cammino aumentante da s a t

Modifico le capacità di $\min_{(i,j)} \lambda$ del grafo → se concorde incremento
→ se discorde decremento

REITERO, se ∄ cammino aumentante v = max flow

Cerco taglio N_s, N_t tc archi in avanti saturi
archi indietro scarichi → $v(N_s,N_t) = v$ = max flow = mincut

MODELLO PLI

⑧ Tagli di Gomory

Dato (PLI), risolvo rilassamento (PL) via metodo grafico.

Se trovo ottimo intero concludo.

Altrimenti riporto il problema in forma standard:

→ scrivo i vincoli ciascuno dipendente da una sola variabile in base, con coefficiente 1

→ oppure ricavo il sistema sopra con $A_B^{-1}A_N$ coefficienti, x_B termini noti e le variabili in base all'inizio

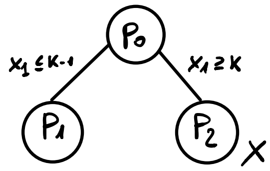
USO PARTE INTERA (floor) sui coeff del sistema trovato, sostituisco = con ≤, sono le eq. dei tagli

NB $\lfloor -\frac{1}{2} \rfloor = -1$ $\lfloor \frac{1}{2} \rfloor = 0$

Se i tagli contengono var. di slack s_i sostituisco i vincoli (della forma standard) per ottenere equazioni in x_i (disegnabili nel piano x_1, x_2)

Risolvo con i tagli per via grafica → potrei non trovare soluzione!? Controllo

⑨ Branch and bound (binario)



Dato (PLI), risolvo rilassamento (PL) via metodo grafico.

Se trovo ottimo intero concludo

altrimenti ottimo con sol. non intera: stima = UB (max) o LB (min)

applico branch su una delle variabili frazionarie

Risolvo il (PLI) rilassato con le condizioni del branch, aggiorno la stima per i nuovi nodi, REITERO

Posso chiudere nodo i se → trovo ottimo intero per P_i $LB_i = UB_i$ = stima
→ ∃j tc → $LB_j \geq UB_i$ (max)
→ $UB_j \leq LB_i$ (min)
→ regione ammissibile vuota.

NB ∀ branch la stima deve peggiorare o rimanere uguale
(aumentare LB per minimo
diminuire UB per massimo)

NB se mi dà già un albero di branch con LB e UB già calcolati e mi chiede intervallo più stringente per ottimo: max maggior LB ≤ OTTIMO ≤ maggior UB di nodi ancora aperti *

min minor LB di nodi ancora aperti * ≤ OTTIMO ≤ minor UB

* o nodi genitori con figli con UB/LB non ancora calcolati

⑩ Branch and bound zaino (binario)

MODELLO ZAINO