Ricerca Operativa

Francesco Bonistalli, Luca Minuti,
Tommaso Pellegrini, Marco Lari,
Niccolò Franchi

March 2023

Indice

List of Theorems

List of Theorems

Parte I

PL

Capitolo 1

Teoria della PL

1.1 Lezione 1

Glossario 1 Soluzione ammissibile: soluzione che rispetta i vincoli □

Glossario 2 Soluzione ottima: è la soluzione che tra quelle ammissibili mi da il massimo guadagno (def alternativa: non c'è nessun altro punto ammissibile migliore di lui) □

Nota 1: *per noi i modelli matematici sono solo di* $=$, $\geq$, $\leq$

Glossario 3 Regione ammissibile: insieme di tutte le soluzioni ammissibili □

Glossario 4 Problema di programmazione lineare primale standard:

$$\begin{cases} max\ c^T x \\ Ax \leq b \end{cases}$$

- programmazione perché lo passiamo a un software
- lineare perché tutti i vincoli e funzioni obiettivo sono funzioni lineari
- primale perché: vedi Glossario 5
- standard perché: vedi Glossario 6

in particolare:

- c_n è il vettore che rappresenta la funzione obiettivo
- $A_{m \times n}$ è la matrice che rappresenta i vincoli
- b_m

dove m e n sono rispettivamente il numero di vincoli e il numero di variabili □

Nota 2: *ogni vincolo è un iperpiano (in* $\mathbb{R}^2$ *quindi rette) (perché le funzioni sono lineari)*

Nota 3: *non usiamo* $>$, $<$ *perché tolgono pezzi di bordo dove ci sono sempre (ammesso che esistano) le soluzioni ottime*

Glossario 5 Primale standard: caso particolare di un problema di PL in cui si utilizzano solo vincoli di $\leq$ e solo *max* □

Glossario 6 Standard: si dicono standard tutti i modelli particolari di PL che sono invece "generali". In altre parole:

- ogni problema di PL lo si può ricondurre in maniera equivalente a un formato standard
- se so risolvere un problema di PL standard li so risolvere tutti
- è una semplificazione che non lede la generalità della trattazione

□

Nota 4 : linprog**:** *è la function di optimization che risolve i problemi di PL*

Per far sì di avere tutti e solo vincoli di minore uguale si può procedere in questo modo:

- se ho un vincolo di $\geq$ moltiplico per -1
- se ho un $=$ (es. $x_1 + x_2 = 0$) costruisco il sistema $\begin{cases} x_1 + x_2 \leq 0 \\ x_1 + x_2 \geq 0 \end{cases}$

Nota 5 : Osservazione**:** *Ogni problema di PL può essere equivalentemente scritto in formato primale standard:*

$$min\ c^T x = -max\ (-c^T x)$$

ovvero: il punto di minimo di una certa funzione è lo stesso del punto di massimo della sua opposta.

Problema 1 Problema di produzione: problema che segue il formato:

$$\begin{cases} max\ cx \\ Ax \leq b \\ x \geq 0 \end{cases}$$

□

1.2 Lezione 2

Definizione 1 Poliedro: 2 definizioni:

- definizione matematica: intersezione [1] di un un numero finito di semi-spazi chiusi

- definizione in PL: l'insieme delle x appartenenti a R alla n che rispettano $Ax \leq b$

□

Definizione 2 Linee di isoguadagno (o di isocosto): sono le linee del tipo: *funzione obiettivo = valore*, ovvero tutte le linee (se in $\mathbb{R}^2$ altrimenti iperpiani) ortogonali al vettore c:

$$L(v) = \{x \in \mathbb{R}^n : c^T x = v\} \text{ } con \text{ } v \in \mathbb{R}$$

□

Nota 6: $c = \nabla(funzione\,obiettivo)$ *(∇ = direzione di massima crescita della funzione)*

1.2.1 Proprietà elementari di un problema di PL

1. risoluzione geometrica per $n = 2$

2. minimi e massimi

3. non ci sono mai soluzioni ottime interne: la soluzione ottima di un problema di P è sempre sul bordo

4. la soluzione ottima non è necessariamente unica

5. non esiste sempre la soluzione ottima (esistono poliedri illimitati)

6. un poliedro può essere vuoto: ad esempio se ci sono troppi vincoli molto restrittivi

1.2.2 Proprietà matematiche dei poliedri

Prop. Combinazioni convesse: un punto y di $\mathbb{R}^n$ si dice combinazione convessa di $x^1, ..., x^k \in \mathbb{R}^n$ se esistono dei coeffcienti $\lambda_1, ..., \lambda_k$ tali che:

$$y = \sum_{i=1}^{k} \lambda_i x_i \text{ } con \text{ } \lambda_i \in [0, 1] \; \forall i \text{ } e \text{ } \sum_{i=1}^{k} \lambda_i = 1$$

□

Definizione 3 conv: $conv\{x^1, ..., x^k\}$ è l'insieme delle combinazioni convesse □

[1] quindi tutti i vincoli devono essere rispettati

Prop. l'insieme conv di $x^1, ..., x^k$ è il più piccolo insieme convesso che contiene i punti $x^1, ..., x^k$. □

Prop. Combinazioni coniche: Dati $x^1, ..., x^k \in \mathbb{R}^n$ un punto $y \in \mathbb{R}^n$ si dice combinazione conica dei punti $x^1, ..., x^k$ se $\exists \lambda,\ i = 1, .., k$ e $\lambda_i \geq 0$:

$$y = \lambda_1 x_1 + ... + \lambda_k x_k$$

□

e viene indicato con

Definizione 4 cono: $cono\{x^1, ..., x^k\}$ □

Teorema 1 Teorema di Weyl (rappresentazione dei poliedri): sia dato un poliedro $P = \{Ax \leq b\}$ con $A_{m \times n},\ b_m$ esistono un sottoinsieme finito $V = \{v^1, ..., v^k\} \subset P$ ed un insieme finito $E = \{e^1, ..., e^p\}$ eventualmente anche vuoti tali che:

$$P = conv(V) + cono(E)$$

□

Nota 7: *V sono punti del poliedro.*

1.3 Lezione 3

Definizione 5 Problemi di PL: vedi Glossario 4 □

Esempio 1 problema di assegnamento**:** Qual'è il costo minimo dell'azienda?

ingegneri ↓ / progetti →	1	2	3	4
1	6	8	12	20
2	7	10	14	18
3	9	11	12	14
4	8	9	13	19

Tabella 1.1: matrice contenente i costi dell'azienda

Vincolo: ogni ingegnere fa esattamente 1 progetto.
Si tratta di un **problema di assegnamento** (=problema di minimo).

Problema 2 Problema di assegnamento: ha n^2 variabili □

La soluzione banale sarebbe cercare il minimo per ogni colonna (ottenendo quindi (1,1)(1,2)(3,3)(3,4)) ma risulta non ammissibile in quanto non rispetta i vincoli. Si seguono quindi i passi:

1. varibili decisionali
2. funzione obbiettivo

3. vincoli

La matrice contiene i costi è la funzione obbiettivo, ho quindi bisogno di 16 variabili (siamo in $\mathbb{R}^{16}$)

$$x_{ij} = \begin{cases} 0 & i \text{"}nonfa\text{"}\ j \\ 1 & i \text{"}fa\text{"}\ j \end{cases}$$

Il vettore di variabile decisionale è quindi:

$$x = (x_{11}x_{12}x_{13}x_{14}|x_{21}x_{22}x_{23}x_{24}|x_{31}x_{32}x_{33}x_{34}|x_{41}x_{42}x_{43}x_{44})$$

un esempio di soluzione ammissibile quindi è $x = (1000|0100|0010|0001)$

$$\begin{cases} min\ \ 6x_{11} + 10x_{22} + 12x_{33} + 19x_{44} = cx & \\ x_{11} + x_{12} + \cdots + x_{14} & = 1 \\ x_{21} + \ldots & = 1 \\ \ldots & = 1 \\ x_{41} + \cdots + x_{44} & = 1 \\ x_{11} + x_{21} + x_{31} + x_{41} & = 1 \\ x_{12} + x_{22} + x_{32} + x_{42} & = 1 \\ \ldots & = 1 \\ x_{14} + \cdots + x_{44} & = 1 \\ 0 \leq x_{ij} \leq 1 \Rightarrow x \geq 0\ \ (= 16\ \ vincoli) & \end{cases}$$

Altro esempio di soluzione ammissibile:

$$x = \left(\frac{1}{2}\frac{1}{2}00|\frac{1}{2}\frac{1}{2}00|0010|0001\right)$$

In questo caso si parla di **Assegnamento cooperativo** (per escludere il caso cooperativo andrebbe aggiunto il vincolo $x_{ij} \in \mathbb{Z}$)

Nota 8: *i problemi di PL dove si lavora solo con interi si dicono di PLI*

La soluzione si ottiene dunque costruendo

```
>> c=[16]
>> A=[32x16]
>> b=[32]
```

Nota 9: *se il programma non da una risposta ci sono 2 possibilità: o il poliedro è vuoto o fa $+\infty$ (o $-\infty$ se il problema è di minimo)*

□

1.3.1 Utilizzo di linprog

Definizione 6 Formato di linprog: formato standard del tipo:

$$\begin{cases} min\ cx \\ Ax \leq b \\ A_{eq}x = b_{eq} \\ LB \leq x \leq UB \end{cases}$$

dove:

- LB sta per lower bound: è il confine inferiore e ha lo stesso numero di componenti di x
- UB sta per upper bound: è il confine superiore e ha lo stesso numeri di componenti di x

□

L'utilizzo della funzione linprog in matlab è in questa forma:

```
>> c=[ ]
>> A=[ ]
>> b=[ ]
>> LB=[ ]
>> UB=[ ]
>> Aeq=[ ]
>> beq=[ ]
>> linprog(c, A, b, Aeq, beq, LB, UB)
```

1.4 Lezione 4

Dal Teorema 1 (di Weyl) :

1.4.1 Proprietà del Teorema di Weyl

Proprietà 1

Prop. Se esistono vertici, V lo posso scegliere come l'insieme dei vertici. □

In altre parole: ***i vertici li devo mettere e bastano***. Infatti se ne aggiungo altri v non cambia.

Nota 10: *dato P devo sapere A,b ; V,E ; d,S*

Proprietà 2

Prop. se P è limitato:

$$E = \{(0,0)\} \Rightarrow cono\{(0,0)\} = 0 \Rightarrow conv(V) + cono(E) = conv(V)$$

□

Proprietà 3

Dati i poliedri senza vertici come:

Definizione 7 Poliedri senza vertici: sono i poliedri che contengono rette al loro interno □

Prop. i poliedri senza vertici sono pochissimi, posso quindi elencarli:

- rette
- fasce
- semipiani

□

Ad esempio una strisca è un poliedro senza vertici.

1.4.2 Espressione di Weyl (e conseguenze)

Algoritmo I.1 Algoritmo risolutivo della PL: dati c, A e b eseguire i passi:

1. $(A,\ b) \rightarrow (V,\ E)$
2. ce^j altrimenti (se $ce^j \leq 0$) calcolo cv^r

□

Nota 11: *se c'è un elemento di E in cui vale* $ce^j \geq 0$ *allora s.o.*=+∞

1.4.3 Teorema fondamentale della PL

Teorema 2 Teorema fondamentale della PL: Dato un (P) (vedi Glossario 4) si può (e si deve) verificare uno solo di questi casi:

1. $P = \emptyset$ (=poliedro vuoto ->troppi vincoli)$(V = \emptyset,\ E = \emptyset)$
2. $(P) = +\infty \rightarrow$ troppi pochi vincoli
3. $\exists$ soluzione ottima data da $v^r \in V :\ max_{i=1,\ldots,k} cv^i = cv^r$

□

1.5 Lezione 5

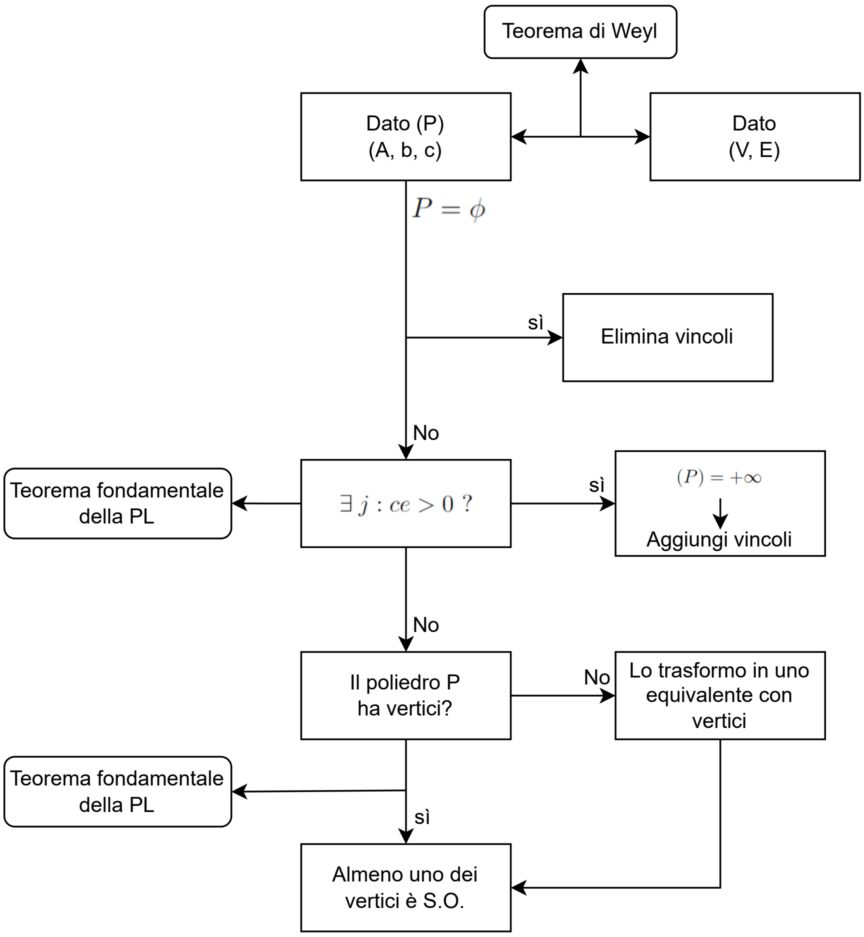


Figura 1.1: schema riassuntivo degli argomenti precedenti

Definizione 8 Vertice: un punto y appartenente a P si dice vertice di P se <u>non</u> si può esprimere come combinazione convessa propria di punti P diversi da y.
Con propria si intende: i λ_i della combinazione convessa $\neq 1\ \forall i$ □

Nota 12 : Definizione geometrica di vertice**:** *un punto P è un vertice se non sta all'interno di un segmento appartenente al poliedro. Questa definizione non ci risulta utile in quanto non riesco a insegnarla a un algoritmo e non so "vederla" in* $\mathbb{R}^n$ *con n>2.*

Ci serve quindi la definizione algebrica che dati A e b ci permette di calcolare i vertici.

Definizione 9 Soluzione di base di un poliedro: data $A_{m\times n}$ *con* $m > n$ e con $rango(A) = n$ e data una base B (ovvero un sottoinsieme degli indici di riga di dimensione n qualunque tale che la matrice che si ottiene da tale estrazione sia inveritbile):

$$(S.B.) = \begin{cases} B \subset \{1, ..., m\} \\ |B| = n \\ A_B \; sia \; invertibile \; (det \neq 0) \\ \overline{x} \; soluzione \; del \; sistema \\ A_B\overline{x} = b_B \end{cases}$$

$\square$

Esempio 2:

$$\begin{cases} 3x_1 + 2x_2 \leq 8 \\ 4x_1 + 5x_2 \leq 13 \\ -x_1 \leq 0 \\ -x_2 \leq 0 \end{cases}$$

con base $B = \{1, 4\}$ si ha $A_B = \begin{pmatrix} 3 & 2 \\ 0 & -1 \end{pmatrix}$ e $b_B = \begin{pmatrix} 8 \\ 0 \end{pmatrix}$, la soluzione del sistema è quindi $\overline{x} = \begin{pmatrix} \frac{8}{3} \\ 0 \end{pmatrix}$ di cui ne verifico l'essere ammissibile sostituendola nel sistema iniziale $\square$

Teorema 3 Caratterizzazione dei vertici: un punto $v \in P$ è un vertice di $P \iff$ esso è una soluzine di base ammissibile $\square$

$$b_N - A_N A_B^{-1} b_B \geq 0 \longrightarrow b_N - A_N\overline{x} \geq 0 \longrightarrow A_N\overline{x} \leq b_N$$

dove N indica gli indici non di base.

Definizione 10 Soluzione di base degenere: una soluzione di base si dice degenere se rispetta uno dei vincoli non di base con l'= $\square$

Nota 13: *in altre parole "il vincolo (retta) passa sopra al vertice"*

1.5.1 Come trattare i poliedri senza vertici

Se un poliedro ($Ax \leq b$) ha il vincolo $x \geq 0$ non ha rette (trovandoci nel primo quadrante)

Quando questo vincolo è però assente si può utilizzare un "trucco":

Esempio 3: dato:

$$\begin{cases} 3x_1 + 5x_2 \leq 61 \\ 4x_1 + 6x_2 \leq 51 \\ x_1 \geq 0 \end{cases}$$

poichè ogni numero è differenza di due numeri positivi posso scrivere:

$x = x' - x''$ e ottenere quindi: $\begin{cases} 3x_1 + 5x_2' - 5x_2'' \leq 61 \\ 4x_1 + 6x_2' - 6x_2'' \leq 51 \\ x_1, x_2', x_2'' \geq 0 \end{cases}$ □

Nota 14: *nel 99% dei casi i poliedri hanno sempre i vertici, in caso contrario si può usare questo trucco*

1.5.2 Teoria della dualità

Dato un (P) in formato primale standard $\left(\begin{cases} max\ c^T x \\ Ax \leq b \end{cases}\right)$, associamo ad esso un problema detto **duale standard** definito come:

$$(D) = \begin{cases} min\ yb \\ y^T A = c \\ y \geq 0 \end{cases}$$

dove $\begin{cases} y^T A = c \\ y \geq 0 \end{cases}$ è detto **poliedro duale**

1.6 Lezione 6

$$(P) = \begin{cases} max\ c^T x \\ Ax \leq b \end{cases} \longleftrightarrow (D) = \begin{cases} min\ y^T b \\ y^T A = c \\ y \geq 0 \end{cases}$$

Nota 15: *il duale del duale è il primale*

Trasformazioni che servono a portare un generico (P) a un formato standard:

1. $min \longleftrightarrow max$
2. $\leq \longleftrightarrow \geq$
3. $= \longrightarrow \leq$
4. vincolo di segno: $x \geq 0$ (sdoppiamento)
5. $\leq \longrightarrow =$ (spiegato nell'esempio 4)

Esempio 4 trasformazione 5**:** data $3x_1 + 5x_2 \leq 7$ diventa $\begin{cases} 3x_1 + 5x_2 = 7 \\ S \geq 0 \end{cases}$
dove $S = 7 - 3x_1 - 5x_2$ è detta **variabile di scarto** □

Teorema 4 Teorema della dualità forte (strong duality):

$$se \;\; P, D \neq \phi \Longrightarrow max_{x \in P} \; c^T x = min_{y \in D} \; y^T b$$

ovvero: i valori ottimi coincidono.
Quindi se ho una coppia di soluzioni ammissibili tali che $c\overline{x} = \overline{y}b$ ho la soluzione ottima. □

1.6.1 Vertici D

Esempio 5:

$$\begin{cases} yA = c \\ y \geq 0 \end{cases} \quad \begin{cases} 3y_1 + 5y_2 + 6y_3 = 11 \\ y4_1 + 3y_2 + 6y_3 = 9 \\ y \geq 0 \end{cases}$$

Nota 16: *è un duale standard perché ho tutti $=$ e tutte le variabili $\geq$ e $n > m$*

Data una base $B \subset \{1, ..., n\}$ dove $1, ..., m$ sono gli indici di base I e dove $N = \{I \backslash B\}$ e dove $|B| = n$ si ha che:

$$A = \begin{pmatrix} A_B \\ A_N \end{pmatrix} \longrightarrow \begin{pmatrix} y_B & y_N \end{pmatrix} \begin{pmatrix} A_B \\ A_N \end{pmatrix} = c \longrightarrow y_B A_B + y_N A_N = c$$

si pone poi $y_N = 0$ e si ottiene $y_B = cA_B^{-1}$. La soluzione di base duale sarà quindi $\overline{y} = (cA_B^{-1}, 0)$

□

Teorema bis Caratterizzazione dei vertici: dal Teorema 3: $\overline{y} \in D$ è un vertice di D $\iff$ $\overline{y}$ è soluzione di base ammissibile. □

1.7 Lezione 7

1.7.1 Conseguenze del teorema della dualità

Dato il Teorema 4 che può essere scritto anche come:

$$Se \; \exists \; \overline{x} \in P, \; \overline{y} \in D \;\; c\overline{x} = \overline{y}b \;\; allora \; \overline{x} \; e' \; ottimo \; per \; (P), \; \overline{y} \; e' \; ottimo \; per \; (D)$$

mi serve un test di ottimalità:

Esempio 6 Conto**:** data B base ammissibile di A_B si hanno:

- $\overline{x} = A_B^{-1} b_B$
- $\overline{y} = (cA_B^{-1}, 0)$

quindi:

- $c\overline{x} = c(A_B^{-1} b_B) = (cA_B^{-1}) b_B$

- $\overline{y}b = \begin{pmatrix} y_B & y_N \end{pmatrix} \begin{pmatrix} b_B \\ b_N \end{pmatrix} = \overline{y}b_B + \overline{y}_N b_B$ che con $\overline{y}_N = 0$ diventa

$$\overline{y}b = (cA_B^{-1})b_B$$

$\square$

Si arriva dunque ad affermare che:

Prop. dato $\overline{x}$ vertice di P (con base B):

$$se\ \ cA_B^{-1} \geq 0 \Longrightarrow allora\ sono\ all'\ ottimo$$

$\square$

Prop. Supponiamo di avere una base:

$$B \longrightarrow A_B \longrightarrow \overline{y} \in D\ vertice\ con\ \ \overline{y} = (cA_B^{-1}, 0)$$

per verificare quindi che $\overline{x} = A_B^{-1}b_B$ sia soluzione ammissibile deve rispettare la condizione: $b_N - A_N A_B^{-1} b_b \geq 0$ (dal Teorema 3) $\square$

1.7.2 Categorie di basi

Le basi si possono dividere in 4 categorie:

1. generano soluzioni ammissibili per P e D (basi ottime)
2. generano soluzioni ammissibili per P e non ammissibili per D
3. generano soluzioni non ammissibili per P e ammissibili per D
4. generano soluzioni non ammissibili per P e D

1.7.3 Problema di trasporto

Problema 3 Problema di trasporto: problema del tipo: *data la tabella*

	1	2	3	
A	21	18	17	100
B	24	19	14	90
	50	60	60	

Tabella 1.2: 1, 2, 3 sono i destinatari delle consegne; A e B i depositi che effettuano le consegne; 100 e 90 le quantità massime trasportabili dal rispettivo deposito; 50, 60, 60 le quantità richieste dai destinatari. Infine i valori nella tabella sono i costi di trasporto per unità.

qual'è il costo ***minimo*** *di trasporto al giorno?* $\square$

Le variabili decisionali da addotare sono del tipo x_{ij} dove i è il deposito e j il destinatario. Il modello risolutivo adoperato è:

$$\begin{cases} min\ 21x_{A1} + 18x_{A2} + ... + 14x_{B3} \\ x_{A1} + x_{A2} + x_{A3} \leq 100 \\ x_{B1} + x_{B2} + x_{B3} \leq 90 \\ x_{A1} + x_{B1} = 50 \\ x_{A2} + x_{B2} = 60 \\ x_{A3} + x_{B3} = 60 \\ x \geq 0 \end{cases}$$

Nota 17: *Questo modello posso trattarlo sia come un primale che come un duale*

Nota 18: *I vincoli con l'= si sostituiscono con >=, perché tanto devo trovare il minimo quindi quello che lo rspetta con l'=.*

Nota 19: *Il problema potrebbe non avere soluzione se la domanda totale supera l'offerta.*

Modello:

$$\begin{cases} min\ \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \\ \sum_{i=1}^{m} x_{ij} \geq r_J \\ \sum_{j=1}^{n} x_{ij} \leq b_i \\ x \geq 0 \end{cases}$$

Capitolo 2

Algorimo del simplesso

2.1 Lezione 8 (algoritmo del simplesso)

L'idea dell'algoritmo (creato da *g. B. Dantzig*) è che, partendo da un vertice B non ottimo, detto *starting point*, si cerca di andare di vertice in vertice adiacente con funzione obbiettivo maggiore (in quanto per il Teorema 3 essi contengono soluzioni di base ammissibili). Tuttavia, essendo questo procedere complesso, si preferisce lavorare sulle basi e quindi andare di base in base:

$$Base \rightarrow Base_{new}$$

Ci muoviamo quindi sugli spigoli (dato che non essendo all'ottimo la funzione in almeno una delle direzione cresce). Quindi si passa da una base a un'altra base adiacente.

Idea di fondo: abbandono un vincolo e un altro vincolo entra in base

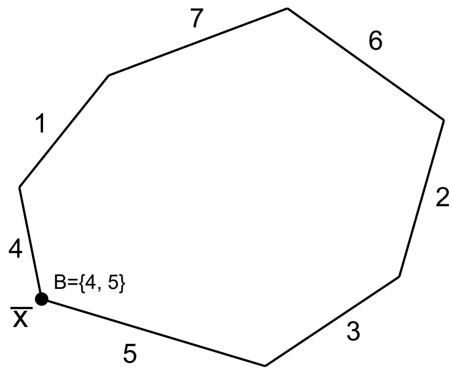
Dato quindi:

Definizione 11: $W = -A_B^{-1}$ matrice $n \times n$ dove W^i indica la i-esima colonna di W □

e avendo escluso $y_B \geq 0$ (altrimeti risulterebbe ottima la soluzione) considero l'eqazione degli spigoli che ha forma

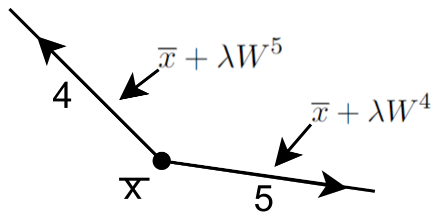
$$\overline{x} + \lambda W^i$$

con $\lambda \geq 0$ e che è quindi un'equazione parametrica di una semiretta.
Prendiamo quindi come esempio il seguente poliedro:



2.1.1 Passo 1

Partendo quindi dalla base $b = \{4, 5\}$ posso procedere in 2 direzioni (essendo in $\mathbb{R}^2$) come illustrato in figura:



2.1.2 Passo 2

Procediamo inizialmente in una direzione:

$$c(\overline{x} + \lambda W^i) = c\overline{x} + c\lambda W^i = c\overline{x} + \lambda(cW^i)$$

dove $cW^i = c(-A_B^{-1}) = -\overline{y_i}$ in cui $\overline{y}$ è la soluzione di base duale definita come $\overline{y} := \begin{pmatrix} \overline{y}_B \\ \overline{y}_N \end{pmatrix}$ con $\overline{y}_N = 0$.

2.1.3 Passo 3

Abbiamo però prima escluso il fatto che $\overline{y_i}$ sia ≥ 0 (altrimenti saremmo già all'ottimo), ci troviamo quindi nel caso in cui $\overline{y_i} < 0$ e quindi $c\overline{x} - \lambda \underbrace{\overline{y_i}}_{<0} > c\overline{x}$:

si ha che quindi la funzione obiettivo cresce spostandosi da $\overline{x}$ in direzione W^h con:

Definizione 12 Indice uscente: $h = \{i \in B : \overline{y}_i < 0\}$ □

Se ci sono più direzioni negative, per evitare cicli:

Definizione 13 Regola (anticiclo di Bland):

$$h = min\{i \in B : \overline{y}_i < 0\}$$

□

2.1.4 Passo 4

Ricerchiamo ora l'indice entrante k:

$$A(\overline{x} + \lambda W^h) \leq b \Longrightarrow A_i\overline{x} + \lambda A_i W^h \leq b_i$$

in cui $\leq b$ mi assicura di rimanere dentro il poliedro. Abbiamo quindi 2 casi:

1. con $i \in B$ allora $A_i W^h$ può essere:
   - 0 se $i \in B \backslash \{h\}$
   - -1 se $i = h$

   e quindi i vincoli di base $A_i x \leq b_i$ sono sempre soddisfatti $\forall \lambda$ con $i \in B$
2. con $i \in N$ allora $A_i W^h$ può essere:
   - ≤ 0 e quindi non mi da problemi in quanto tutta la semiretta $x(\lambda)$ è contenuta nel poliedro e quindi $cx(\lambda)$ tende a $+\infty$
   - > 0 allora $A_i\overline{x} + \lambda \underbrace{A_i W^h}_{>0} \leq b_i$ e quindi la soluzione è ammissibile se $0 \leq \lambda \leq min \frac{b_i - A_i\overline{x}}{A_i W^h}$

Nota 20: *λ rappresenta di quanto mi sposto dal vertice, se $\lambda = 0$ siamo in un vertice degenere*

L'indice entrante lo scegliamo quindi (seguendo la regola di Bland) come

$$k = \{min\ i : min \frac{b_i - A_i\overline{x}}{A_i W^h}\}$$

dove il rapporto $\frac{b_i - A_i\overline{x}}{A_i W^h}$ prende il nome di r_i. Possiamo allora scrivere l'indice entrante come:

$$k = \min_i \{min\ r_i\}$$

Si aggiorna dunque la base: $B = B \backslash \{h\} \cup \{k\}$ e si torna al passo 2.

Teorema 5: l'algoritmo del simplesso è corretto (test di ottimalità) e termina in un numero finito di passi (poiché il numero di vertici è finito) □

2.1.5 Sintesi simplesso primale

1. Trovo una base B che genera una soluzione di base primale ammissibile.
2. Calcolo la soluzione di base primale $\overline{x} = A_B^{-1} b_B$ e la soluzione di base duale $\overline{y} = (cA_B^{-1}, 0)$.
3. se $\overline{y}_B \geq 0$ stop: sono all'ottimo.

   Altrimenti calcolo l'indice uscente

   $$h = min\{i \in B : \overline{y}_i < 0\}$$

   e pongo $W = -A_B^{-1}$ con W^h che indica la h-esima colonna di W.
4. se $A_i W^h \leq 0 \ \forall i \in N$ allora stop: (P) ha valore ottimo $+\infty$ e (D) è vuoto.

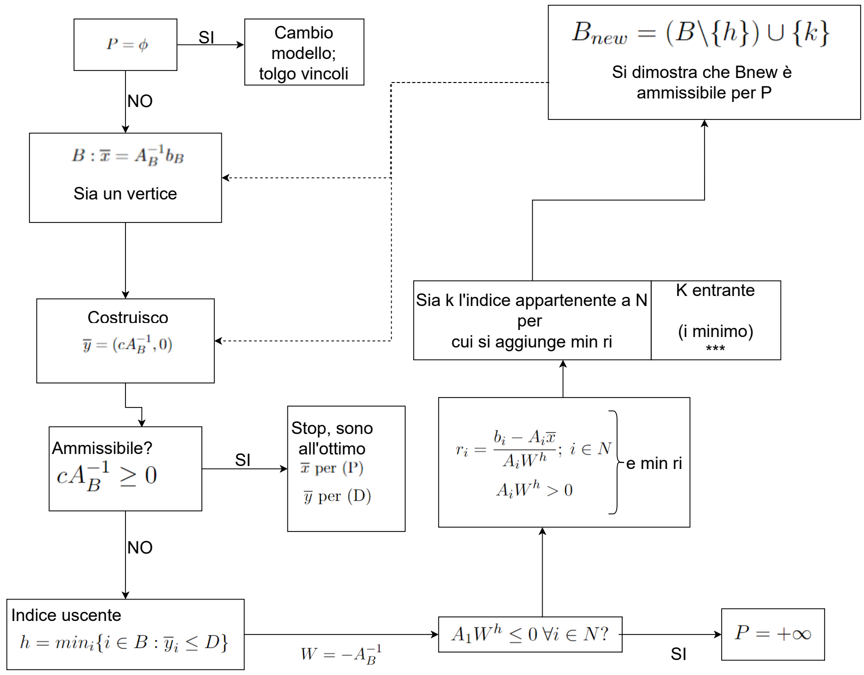
   Altrimenti calcolo l'indice entrante:

   $$k = \{min\ i : min \frac{b_i - A_i \overline{x}}{A_i W^h}\}$$

   e aggiorno la base come: $B = B \backslash \{h\} \cup \{k\}$.

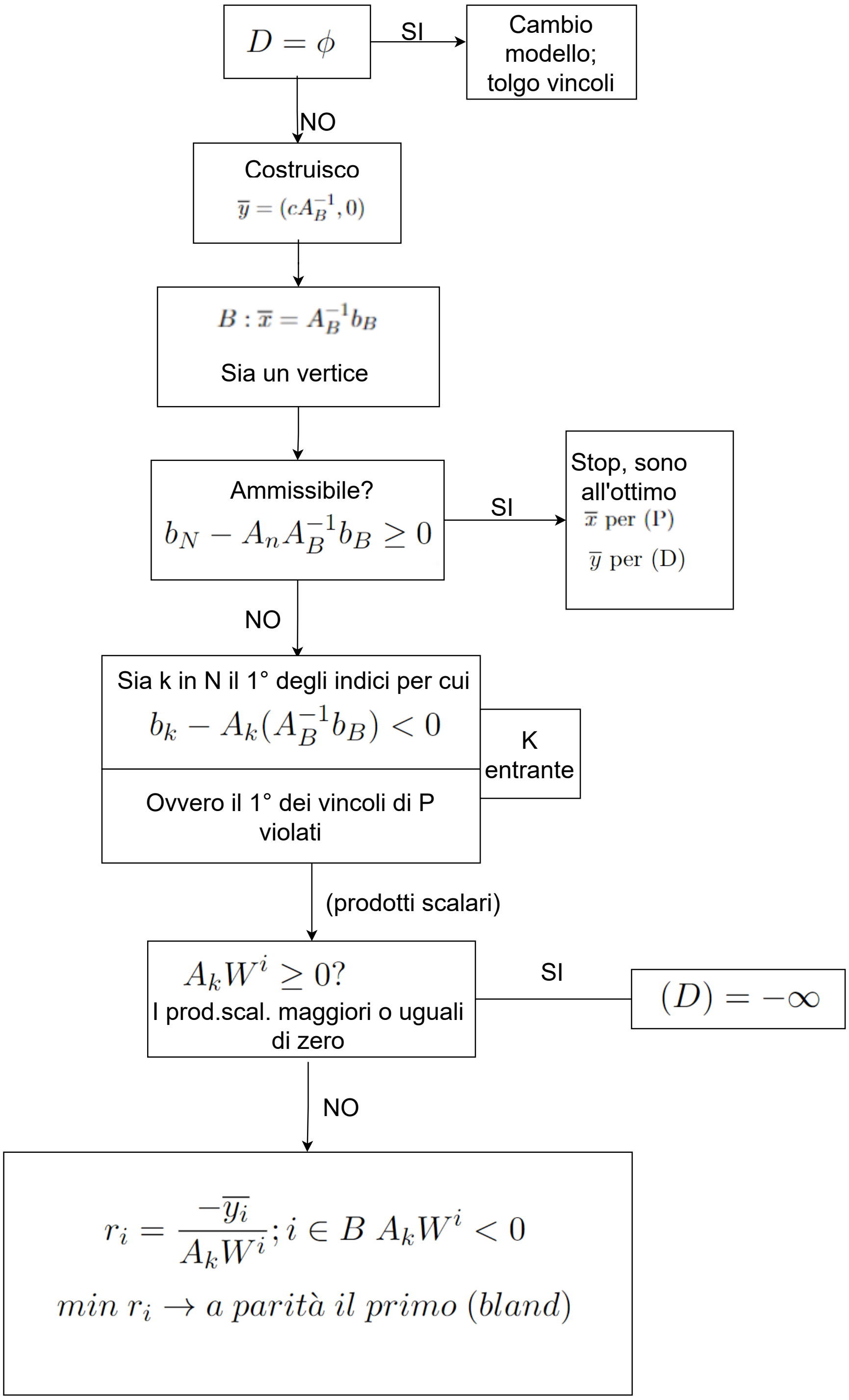
2.2 Lezione 9 (Algoritmo del simplesso polinomiale e duale)

2.2.1 Simplesso polinomiale



Teorema 6 Teorema di Bland: nei vertici degeneri se si usano le regole anticiclo di Bland non ci sarà loop. □

2.2.2 Simplesso duale



2.2.3 Sintesi simplesso duale

1. Trovo una base B che genera una soluzione di base primale ammissibile.
2. Calcolo la soluzione di base primale $\overline{x} = A_B^{-1} b_B$ e la soluzione di base duale $\overline{y} = (cA_B^{-1}, 0)$.
3. se $b_N - A_N \overline{x} \geq 0$ stop sono all'ottimo.

   Altrimenti calcolo l'indice entrante:

   $$k = min\{i \in N : b_i - A_i \overline{x} < 0\}$$

   (primo bland) e pongo $W = -A_B^{-1}$ con W^i che indica la i-esima colonna di W.
4. se $A_k W^i \geq 0 \ \ \forall i \in B$ stop: (D) ha valore ottimo $-\infty$ e (P) è vuoto.

   Altrimenti calcolo l'indice uscente:

   $$h = \{min \ \ i \in B : min \frac{\overline{y}_i}{-A_k W^i}\}$$

   (secondo bland) e aggiorno la base come: $B = B \backslash \{h\} \cup \{k\}$.

2.3 Lezione 10 (poliedro vuoto)

Nota 21: *trattiamo solo il caso duale: se ho un primale mi basta cambiare il formato*

La domanda che ci poniamo è: $P = \phi$? Per rispondere costruiamo il:

Definizione 14 Duale ausiliario:

$$\begin{cases} yA = c \\ y \geq 0 \end{cases} \longrightarrow \begin{cases} min \ \varepsilon_1 + ... + \varepsilon_n \\ y^T A + \varepsilon I = c \\ y, \varepsilon \geq 0 \end{cases}$$

in cui i ε sono variabili positive fittizie, con coefficiente 1, che ci sono utili per la risoluzione del problema, e I è la matrice identità. □

Nota 22: *Posso usare questo problema per trovare una base ammissibile di (D).*

Esempio 7: dato

$$\begin{cases} 3y_1 + 5y_2 + 6y_3 + 7y_4 = 9 \\ 2y_1 + 5y_2 + 7y_3 + 8y_4 = 11 \\ y_i \geq 0 \end{cases}$$

il suo (DA) (ovvero il suo problema duale ausiliario) è

$$\begin{cases} min\, \varepsilon_1 + \varepsilon_2 \\ 3y_1 + 5y_2 + 6y_3 + 7y_4 + \varepsilon_1 = 9 \\ 2y_1 + 5y_2 + 7y_3 + 8y_4 + \varepsilon_2 = 11 \\ y_i, \varepsilon_i \geq 0 \end{cases}$$

dove il poliedro DA è

$$\begin{cases} 3y_1 + 5y_2 + 6y_3 + 7y_4 + \varepsilon_1 = 9 \\ 2y_1 + 5y_2 + 7y_3 + 8y_4 + \varepsilon_2 = 11 \\ y_i, \varepsilon_i \geq 0 \end{cases}$$

e V_{DA} è l'ottimo del problema ausiliario □

Teorema 7: .

- se $V_{DA} > 0 \Longrightarrow D = \phi$ che possiamo scrivere anche come

$$V_A D > 0 \iff D = \phi$$

- se $V_{DA} = 0 \Longrightarrow D \neq \phi$

□

Nota 23 : Osservazioni**:** *ci sono 2 osservazioni importanti necessarie per capire la validità del teorema:*

- V_{DA} *non può essere <0 perché è somma di variabili positive*
- *DA non può essere vuoto perché la soluzione* $(y, \varepsilon) = (0, c)$ *è sempre ammissibile (anche nel caso di* $c < 0$ *infatti basta moltiplicare preventivamente ed eventualmente per* -1 *essendoci solo equazioni)*

Tuttavia, dopo aver visto che non può essere vuoto, manca l'esistenza di un vertice di partenza per poter applicare il simplesso duale per trovare V_{DA} ovvero l'ottimo. Questo può essere però sempre individuato in $(0, c)$ che risulta anche essere di base.

Esempio 8: dall'esempio Esempio 7 ... □

Lezione 11: esercitazione

Parte II

P.L.I.

Capitolo 3

Teoria della PLI

3.1 Lezione 12 (Problema di P.L.I.)

Un problema di programmazione lineare intera agisce con numeri interi.

Problema 4 Problema di caricamento (knapsack problem): Questione di riempimento di uno spazio limitato con entità di diverso valore e magnitudine.
Possono essere descritti come:

$$\begin{cases} max\ cx \\ Ax \leq b \\ x \in N \end{cases}$$

□

Esempio 9: Caricamento di una stiva

v	7	8	11	14	16	18
p	13	16	19	20	27	28

Tabella 3.1: Sei oggetti identificati dalla coppia Valore Peso

I sei oggetti devono essere stivati in un contenitore con una capienza $P = 52$, cercando di avere la combinazione più valente. Un esempio di soluzione ammissibile è:

$$(1\ 1\ 1\ 0\ 0\ 0) \rightarrow v = 26$$

Il modello matematico che otteniamo è:

$$\begin{cases} max\ 7x_1 + 8x_2 + 11x_3 + 14x_4 + 16x_5 + 18x_6 \\ 13x_1 + 16x_2 + 19x_3 + 20x_4 + 27x_5 + 28x_6 \leq 52 \\ x \in \{0, 1\} \end{cases}$$

L'ultimo vincolo ci ricorda il problema di assegnamento, infatti la valenza delle variabili è: $x_i = \begin{cases} 1 \rightarrow presente \\ 0 \rightarrow non\ presente \end{cases}$ e il numero di soluzioni ammissibili è finito.

Questo è il problema del riempimento dello zaino, il quale può essere di due tipi:

- zaino binario: $\begin{cases} ... \\ x \in \{0, 1\} \end{cases}$
- zaino intero: $\begin{cases} ... \\ x \in \mathbf{Z} \end{cases}$ che a sua volta possono essere:
  - indivisibile: e quindi un problema di PLI
  - divisibile: nel caso in cui si vada a rilassare[1] il vincolo sostituendolo con il vincolo: $x_i \geq 0$ e quindi un problema di PL. Anche lo zaino binario può essere divisibile nel caso in cui si vada a rilassare il vincolo con x appartenente a [0,1].

□

Definizione 15 Valutazione superiore e inferiore (Vs e Vi): il rilassato continuo di un problema di PLI è una valutazione superiore se è un problema di massimo, è invece una valutazione inferiore se è un problema di minimo. Con $\tilde{x} = soluzione\ ammissibile \rightarrow c\tilde{x} = V_I$ per max, mentre $c\tilde{x} = V_S$ per min

$$V_I \leq V_{zi} \leq V_S$$

□

Nota 24: *la PL si risolve. La PLI quando molto complessa si stima (dal basso e dalll'alto)*

Nota 25: *le stime devono essere facili (dal momento che ne devo fare molte)*

[1]rilassare un problema significa sostiuire il vincolo intero con uno divisibile, il Rilassamento continuo si ottiene sostituendo ai vincoli di integralità delle variabili dei semplici vincoli di non negatività, oppure eliminandoli del tutto a seconda della struttura del problema; un rilassamento continuo può essere facilmente risolto con le tecniche di soluzione della PL

3.1.1 Costruzione di Vi/Vs per il problema dello zaino intero

Dall'Esempio 9 costruisco il duale del rilassato continuo (RC):

$$\begin{cases} min\ 52y_1 + 0y_2 + ... + 0y_7 \\ 13y_1 - y_2 = 7 \\ 16y_1 - y_3 = 8 \\ 19y_1 - y_4 = 11 \\ 20y_1 - y_5 = 14 \\ 27y_1 - y_6 = 16 \\ 28y_1 - y_7 = 18 \\ y \geq 0 \end{cases} \longrightarrow \begin{cases} min\ 52y \\ 13y \geq 7 \\ 16y \geq 8 \\ 19y \geq 11 \\ 20y \geq 14 \\ 27y \geq 16 \\ 28y \geq 18 \\ y \geq 0 \end{cases}$$

il passaggio descritto dalla freccia è reso possibile dal fatto che i coefficienti delle $y \neq 1$ siano tutti a 0 nella funzione obbiettivo, perdiamo quindi tutte le y_i con $i \neq 1$ e ci ritroviamo con una solita incognita. Andando a calcolare tutte le disequazioni descritte come $y \geq \frac{v_i}{p_i}$ risulta che il più alto rapporto $\frac{v_i}{p_i}$ è quello con $i = 4$ e valore 0.7 che quindi, moltiplicato per 52, da la soluzione ottima del duale del rilassato continuo ovvero:

$$V_{RC} = 52 \cdot \frac{14}{20} = V_S$$

e dove $\overline{x} = (0, 0, 0, \frac{52}{20}, 0, 0)$ è la soluzione ottima del RC e in cui $\frac{52}{20} = 2.6$ che arrotondato per difetto è 2 e che da il minimo numero di volte in cui posso mettere l'oggetto in stiva.

3.1.2 Algoritmi

Continuando a trattare l'RC dello zaino intero:

Algoritmo II.1 Algortimo dei rendimenti (VS): dati i rendimenti definiti come $\frac{v_i}{p_i}$ (ovvero il valore rispetto al peso), calcolo tutti i rendimenti, prendo il più alto e saturo lo zaino con il bene con il rendimento più alto:

$$V_{RC} = P\Big(\frac{v_i}{p_i}\Big)_{max} \longrightarrow V_{RC} = P\frac{v_k}{p_k} = v_k\frac{P}{p_k}$$

L'algoritmo da quindi la soluzione ottima di un problema di PL. □

Nota 26: *tramite l'esempio precedente si dimostra come l'algoritmo 1 è corretto e trova la soluzione ottima del RC*

Algoritmo II.2 (VI): Questo algoritmo genera una soluzione ammissibile greedy: Calcolo i rendimenti, prendo il bene con rendimento massimo, ne metto il più possibile e calcolo lo spazio rimasto e poi procedo. □

Nota 27: *questo algoritmo genera una soluzione ammissibile greedy*

Esempio 10: ripendendo l'esempio precendente: $x_4 = 2$, $v_4 = 14$ e $p_4 = 20$. Si ha quindi $S_{rimasto} = 52 - (20 \cdot 2) = 12$ e $V_I = 14 \cdot 2 = 28$ che si traduce in:

$$28 \leq V_{zi} \leq 52 \cdot \frac{14}{20} \longrightarrow 28 \leq V_{zi} \leq 36,4 \longrightarrow 28 \leq V_{zi} \leq 36$$

$\square$

Nota 28: *in generale possiamo considerare la PL a dati interi (senza ledere l'ipotesi di poter avere dati razionali)*

Nota 29: *caso particolare:* ***quando il gap tra*** V_I ***e*** V_S ***è*** 0 ***allora sono all'ottimo*** *(*V_I *è il vero valore ottimo perché* V_S *non è ammissibile)*

3.1.3 Utilizzo di intlinprog

```
>> intlinprog(c, INT, A, b, Aeq, beq, LB, UB)
```

in cui INT ha forma del tipo:

```
>> INT=[1 2 3 4 5 6]
```

dove $1, 2, ..., 6$ indicano le variabili che devono essere intere

3.2 Lezione 14 20/03

3.2.1 Problema dello zaino

Caso binario

$$\begin{cases} max\ v^T x \\ p^T x \leq P \\ x \in \{0, 1\} \end{cases}$$

Rilassato continuo:

$$\begin{cases} max\ v^T x \\ p^T x \leq P \\ 0 \leq x \leq 1 \end{cases}$$

Caso intero

$$\begin{cases} max\ v^T x \\ p^T x \leq P \\ x \in \mathbf{Z} \end{cases}$$

Trattiamo di seguito il caso binario.

3.2.2 Algoritmi dei rendimenti

- Algoritmo per V_S (non greedy):
  1. calcolo i rendimenti r_i
  2. ordino in modo decrescente
  3. prendo il max e lo metto a 1 nello zaino
  4. calcolo la capacità residua
  5. ripeto fino a che l'ultimo non lo metto come frazione
- Algoritmo per V_I (greedy):
  1. calcolo i rendimenti r_i
  2. prendo il max e metto a 1
  3. calcolo la capacità residua
  4. vedo se il secondo ci entra, poi il terzo ecc.

Nota 30: *Se V_I diverso da Vs, l'errore si trova facendo $(V_S - V_I)/V_I$ in percentuale.*

3.3 Lezione 15 21/03

3.3.1 Equivalenza tra PL e PLI

Teorema 8: dato $v = \max\limits_{x \in S} c^T x$ e $v^* = \max\limits_{x \in convS} c^T x$ (dove il primo è un problema di PLI e il secondo uno di PL) i valori v e v^* coincidono. □

Nota 31: *convS è la regione ammissibile del problema di PL equivalente a quello di PLI*

3.3.2 Algoritmi di riduzione del gap

Definizione 16 Disuguaglianza valida: la disequazione $\gamma^T x \leq \gamma_0$ si dice disuguaglianza valida (DV) per l'insieme S se

$$\gamma^T x \leq \gamma_0, \forall x \in S$$

□

Definizione 17 Piano di taglio: data $\gamma^T x \leq \gamma_0$ DV essa si dice piano di taglio se $\gamma^T x_{RC} > \gamma_0$ (dove x_{RC} è l'ottimo dell'RC). □

Definizione 18 Piani di taglio di Gomory: dato:

$$\begin{cases} min/max\ c^T x \\ A^T x = b \\ x \geq 0 \\ x \in \mathbf{Z}^n \end{cases}$$

con:

- x_{RC} ottimo del rilassato continuo
- $A = (A_B, A_N)$ con B base ottima
- $\tilde{A} = A_B^{-1} A_N$
- $a \in \mathbb{R}$
- $\{a\} = a - \lfloor a \rfloor$ (parte frazionaria di a)

e sia r una parte frazionaria di x_{RC}

□

Teorema 9:

$$\sum_{j \in N} \{\tilde{a}_{rj}\} x_j \geq \{(x_{RC})_r\}$$

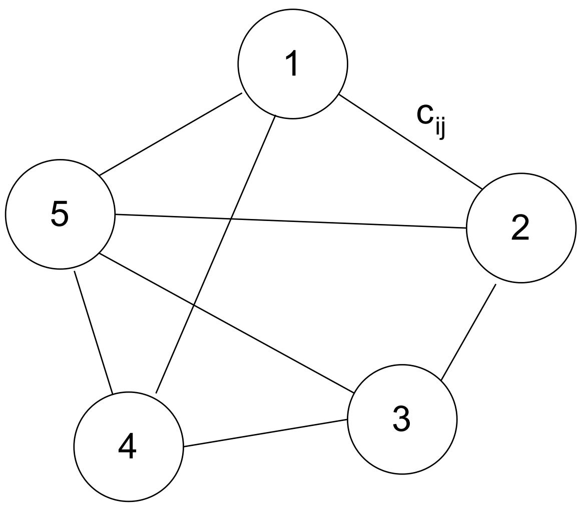
è un piano di taglio □

Criterio di stop-Gomory: quando l'ottimo di RC viene a componenti intere.

Nota 32: *Gomory termina in un numero finito di passi.*

3.4 Lezione 16 22/03 Problema del commesso viaggiatore (TSP)

Problema 5 TSP: trovare il ciclo di costo minimo che passa 1 sola volta per tutti i nodi



nodi $N = \{1,2,3,4,5\}$,con archi $A \subseteq N \times N$ $A = \{(1,2),(2,3)...\}$ e costo degli archi c_{ij}.

TSP asimmetrico: se c'è un $c_{ij} \neq c_{ji}$ (quando la matrice dei costi non è simmetrica)

TSP completo: se gli archi sono tutti $n^2 - n$

$\square$

3.4.1 Problema del ciclo Hamiltoniano di costo minimo

Problema: trovare un ciclo da un nodo a se stesso passando per tutti i nodi.

Nota 33: *le soluzioni ammissibili sono* $(n-1)!$

Considerando quindi la variabile $x_{ij} = \begin{cases} 1 & \text{se l'arco lo percorro} \\ 0 & \text{se non percorro l'arco} \end{cases}$ il modello matematico per il problema TSP avrà forma (nel nostro caso):

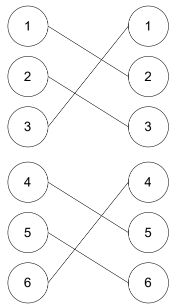
$$\begin{cases} min\ c_{11}x_{11} + c_{12}x_{12} + ... + c_{55}x_{55} \\ x_{11} + x_{12} + x_{13} + x_{14} + x_{15} = 1 \\ x_{21} + x_{22} + x_{23} + x_{24} + x_{25} = 1 \\ x_{31} + x_{32} + x_{33} + x_{34} + x_{35} = 1 \\ x_{41} + x_{42} + x_{43} + x_{44} + x_{45} = 1 \\ x_{51} + x_{52} + x_{53} + x_{54} + x_{55} = 1 \\ x_{11} + x_{21} + x_{31} + x_{41} + x_{51} = 1 \\ ... \\ x_{15} + x_{25} + x_{35} + x_{45} + x_{55} = 1 \\ x \geq 0 \end{cases}$$

Risulta quindi che: Ciclo Hamiltoniano $\subseteq$ assegnamento. Tuttavia, dal modello appena proposto, per far si che questo assegnamento risulti essere anche un ciclo Hamiltoniano bisogna aggiungere qualche vincolo. Il modello finale sarà quindi:

Modello TSP asimmetrco:

$$\begin{cases} min\ c^T x \\ \sum_{i=1}^{n} x_{ij} = 1 \ \ \forall j \\ \sum_{j=1}^{n} x_{ij} = 1 \ \ \forall i \\ \sum_{i \in S, j \notin S} x_{ij} \geq 1 \ \ \forall S \subseteq N \\ x_{ij} \in \{0, 1\} \end{cases}$$

I vincoli di connessione sono 2 alla n -2n -2. Mentre i primi due insiemi di vincoli stabiliscono che ogni nodo abbia un solo arco entrante e un solo arco uscente, il terzo insieme (detto dei **vincoli di connessione**) stabilisce che il ciclo Hamiltoniano abbia almeno un arco uscente da ogni sottoinsieme non vuoto di nodi in modo da evitare cicli orientati che non passano da tutti i nodi come mostrato ad esempio in figura:



3.4.2 Calcolo di VI

Risolvo l'assegnamento di costo minimo (tolgo i vincoli di connessione)

3.5 Lezione 17 24/03

3.5.1 Algoritmo delle toppe

Algoritmo II.3 Algortimo delle toppe (VS): Si devono ricongiungere 2 cicli disgiunti, si cancella un arco di un ciclo e un arco di un altro. Si sostituiscono (i,j) e (k,l) con (i,l) e (k,j)

- somma costi (i,l) e (k,j) - somma costi (i,j) e (k,l) = gap

Tolgo gli archi più costosi dei sottocicli □

3.5.2 TSP simmetrico

Questo caso si verifica quando $c_{ij} = c_{ji}$ (matrice dei costi simmetrica).
Modello:

$$\begin{cases} min\ c^T x \\ \sum_{i<j} x_{ij} + \sum_{i>j} x_{ij} = 2 \quad \text{(detti vincoli di grado)} \\ \text{vincoli di connessione} \\ x_{ij} \in \{0,1\} \end{cases}$$

Nota 34: *le soluzioni ammissibili sono* $\frac{(n-1)!}{2}$

3.5.3 Calcolo VS

	2	3	4	5
1	22	28	33	36
2	/	41	42	44
3	/	/	38	39
4	/	/	/	27

Algoritmo II.4 Algoritmo del nodo più vicino: Seguendo il percorso $1 - 2 - 3 - 4 - 5 - 1$ si ha $V_S = 164$ □

3.6 Lezione 18 27/03

3.6.1 Ricerca di VI

dal modello:

$$\begin{cases} min\ c^T x \\ \text{vincoli di grado} \\ \text{vincoli di connessione} \\ x_{ij} \in \{0,1\} \end{cases}$$

si lasciano tutti i vincoli di connessione mentre si cancellano tutti i vincoli di grado tranne quello di un nodo k:

$$\sum_{i<k} x_{ik} + \sum_{k<i} x_{ki} = 2$$

Esempio 11: per $k = 3$ e $n = 6$ diventa: $x_{13} + x_{23} + x_{34} + x_{35} + x_{36} = 2$ □

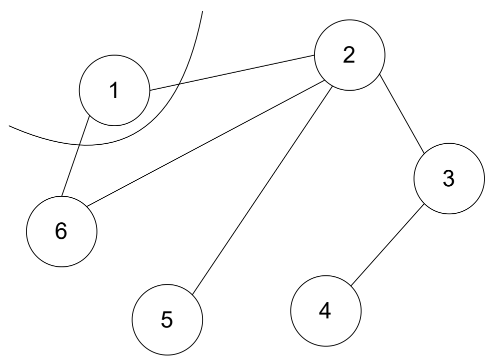


Figura 3.1: esempio per k=1

Nota 35: *La struttura che mi garantisce il "connesso"*[2] *in un grafo è l'albero di copertura*[3]

Il problema che si ottiene eliminando i vincoli di grado tranne quello del nodo k si chiama **k-albero** di costo minimo.

3.6.2 Costruzione del k-albero

1. si isola il nodo k
2. si costruisce l'albero di copertura di costo minimo sugli altri nodi (tramite l'algoritmo di kruskal[4])
3. si collega k per rispettare il vincolo di grado aggiungendo 2 archi di costo minimo

V_I si ottiene quindi sommando i costi dei rami che formano il k-albero di costo minimo.

3.6.3 Bin packing

Problema 6 Bin packing: Dato il peso dei software illustrato in tabella:

P	20	31	33	41	44	46	48	51
	1	2	3	4	5	6	7	8

e disponendo di dischi rigidi di capienza pari a 90 trovare il minimo numero di dischi per contenere tutti i software. □

Calcolo di VI

Dato che $\sum_{i=1}^{n} P_i = 328$ si ha che $\frac{328}{90} = 3,64$ (ovvero la soluzione ottima del RC) che arrotondato per eccesso è $= 4 = V_I$

[2] Un grafo non orientato è connesso se esiste un cammino fa due nodi qualsiasi del grafo.

[3] Insieme di componenti connesse (sottografi connessi) massimali (nessun nodo è è connesso ad un'altra componente connessa) acicliche

[4] algoritmo per trovare il minimo albero di copertura (quello in cui la somma dei pesi degli archi è minima):

(a) Ordina gli archi del grafo in ordine crescente

(b) Scorri l'elenco ordinato degli archi:
per ogni arco a
if (a connette due componenti non connesse) {
scegli a;
unifica le componenti;
}

Calcolo di VS

Algoritmo II.5 Next fit decreasing: Esamina gli oggetti in ordine di peso decrescente. A partire dal primo sw (nel nostro caso il numero 8) si inserisce in un disco e si calcola lo spazio residuo, se il sw successivo (ovvero il numero 7) ci entra si inserisce e si calcola nuovamente lo spazio rimasto e così via. Nel caso in cui lo spazio rimasto non fosse sufficiente si apre un nuovo disco e si ripete l'algoritmo. Infine il numero di contenitori utilizzati sarà V_S. □

Algoritmo II.6 First fit decreasing: Esamina gli oggetti in ordine di peso decrescente. Una volta inserito il primo sw (il numero 8) e calcolato lo spazio rimasto si scorrono tutti gli altri sw fino a trovarne uno che rientra in tale spazio rimasto e così via. Quando lo spazio di un disco viene esaurito se ne apre uno nuovo e si ripete l'algoritmo. □

Algoritmo II.7 Best fit decreasing: Esamina gli oggetti in ordine di peso decrescente. Tra tutti i contenitori usati che possono contenere un oggetto, scegli quello con la minima capacitá residua. Se nessuno di essi puó contenerlo, assegna l'oggetto ad un nuovo contenitore. □

3.7 Lezione 20 31/03 Bin Packing

Prendiamo ora un nuovo esempio del Problema 6:

P							
7	11	14	15	18	21	27	51

e consideriamo dei contenitori di capacità $C = 35$. Il peso totale è dunque 113 e di conseguenza $V_I = \frac{113}{35} = 3,23 = 4$.

Per scrivere il modello ci serviamo quindi della variabile decisionale

$$x_{ij} = \begin{cases} 0 \\ 1 \end{cases}$$

in cui i è il contenitore e j è l'oggetto. Il primo vincolo che ci viene in mente è:

$$\sum_{i=1}^{m} x_{ij} = 1 \ \ \forall j$$

ovvero il vincolo di semi assegnamento (=ogni oggetto lo devo mettere una volta sola). Non si conosce però m ovvero il numero di contenitori: ci serve quindi un'altra variabile:

$$\underset{i=1,\dots,m}{y_i} = \begin{cases} 0 \\ 1 \end{cases}$$

che identifica il prendere o il non prendere un certo contenitore. Riscriviamo quindi la funzione obbiettivo e la sommatoria con il caso peggiore:

$$min\ y_1 + ... + y_n$$

$$\sum_{i=1}^{n} x_{ij} = 1 \ \ \forall j = 1, ..., n$$

aggiungo il vincolo: $\sum_{j=1}^{n} p_j x_{ij} \leq cy_i$ ovvero "prendi meno contenitori possibili"

Modello:

$$\begin{cases} min \ y_1 + ... + y_m \\ \sum_{i=1}^{m} x_{ij} = 1 \ \ \forall j = 1, ..., n \\ \sum_{j=1}^{n} p_j x_{ij} \leq cy_i \\ x_{ij}, y_i \in \{0, 1\} \end{cases}$$

Osservazione: prima di scrivere il modello (visto che n non conviene) mi trovo una soluzione ammissibile (con metodo greedy) per cambiare nel modello la n in m nella funzione obbiettivo e nel primo vincolo[5]

Nota 36: *questo problema ha* $\underbrace{m \times n}_{x_{ij}} + \underbrace{m}_{y_i}$ *variabili*

[5]modifica già effettuata negli appunti

Capitolo 4

Metodo del Branch and Bound

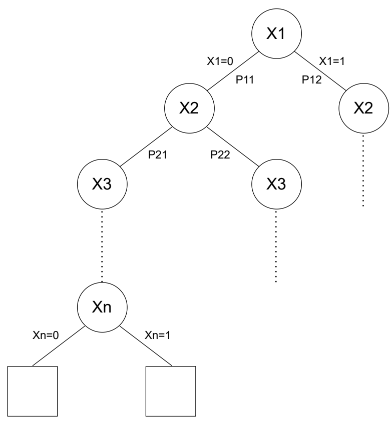
4.1 Lezione 19 28/03

4.1.1 Metodo Branch and Bound

Questo metodo si basa sul concetto dell'albero dell'enumerazione totale.

Caso del minimo (binario:)

$$\begin{cases} min\ c^T x \\ Ax \leq b \\ x \in \{0,1\}^n \end{cases}$$



Nota 37: *x1, x2, ... sono in ordine per comodità: potrebbero essere anche in posizioni diverse.*

Nota 38: *l'albero ha 2^n foglie e quindi soluzioni: una di queste è quella ottima ma non sappiamo quali di queste sono ammissibili.*

Per applicare questo metodo si presuppone che ci siano delle regole di Bound:

$$[V_I, V_S]$$

Per accertarci che ci siano devo avere una regola di Branch: *do un nome agli elementi su ogni piano:* P_{ij}. Si tratta quindi di un problema di ottimizzazione combinatoria; ad esempio:

$$(P_{24})\begin{cases} min\ cx \\ Ax \leq b \\ x \in \{0,1\}^n \\ x_1 = 1, x_2 = 1 \end{cases}$$

Teoria: se sapessi risolvere tutti i P_{ij} (minimi) mi basterebbe prendere il minimo tra essi, è però troppo complicato. Ci serve quindi una regola del tipo: "qui sotto non c'è nulla di meglio di $\overline{x}$".

Nota 39: *tagliare i rami più in alto sarebbe ottimo per semplificare il problema*

Entra quindi in gioco la regola di Branch (=taglio).

4.1.2 Regole di taglio

1. $P_{ij} = \emptyset$ (è una regola poco utile a meno che non sia di facile implementazione, ad esempio con TSP)
2. se $V_I(P_{ij}) \geq V_S(P)$
3. se $V_I(P_{ij}) < V_S(P)$ ma è ammissibile allora aggiorno $V_S(P) = V_I(P_{ij})$ e taglio

C'è anche il branch and bound per problemi di massimo (zaino):

1. $P_{ij} = \emptyset$
2. se $V_S(P_{ij}) \leq V_I(P)$
3. se $V_S(P_{ij}) > V_I(P)$ ma è ammissibile allora aggiorno $V_I(P) = V_S(P_{ij})$ e taglio

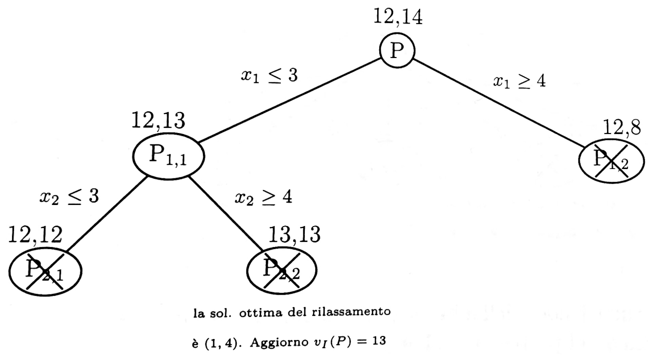
Esempio 12 esempio dal libro**:** Dato il problema:

$$\begin{cases} max\ x_1 + 3x_2 \\ x_1 + 5x_2 \leq 21 \\ 8x_1 + 2x_2 \leq 35 \\ x \geq 0 \\ x \in \mathbf{Z} \end{cases}$$

La soluzione ottima del rilassamento continuo è $\left(\frac{7}{2}, \frac{7}{2}\right)$ per cui $v_S(P) = 14$. Arrotondando per difetto le componenti di tale soluzione otteniamo la soluzione ammissibile $(3,3)$ che fornisce $v_I(P) = 12$. Iniziamo l'esplorazione dell'albero di enumerazione totale istanziando la variabile x_1 che ha un valore frazionario ($x_{11} = 3.5$) nella soluzione ottima del rilassamento continuo e distinguiamo due casi: $x_1 \leq 3$ oppure $x_1 \geq 4$ in modo che la soluzione $\left(\frac{7}{2}, \frac{7}{2}\right)$ non sia ottima per nessuno dei rilassamenti continui dei nodi $P_{1,1}$ e $P_{1,2}$ al primo livello. Vicino ad ogni nodo $P_{i,j}$ indichiamo nell'ordine i valori $v_I(P)$ e $v_S(P_{i,j})$.

La soluzione ottima del rilassamento continuo di $P_{1,1}$ è $(3, \frac{18}{5})$ di valore 13.8, quindi $v_S(P_{1,1}) = 13 > 12 = v_I(P)$, pertanto non possiamo fare una visita implicita di $P_{1,1}$ che rimane quindi aperto. La soluzione ottima del rilassamento continuo di $P_{1,2}$ è $(4, \frac{3}{2}, 0)$ di valore 8.5, quindi $v_S(P_{1,2}) = 8 < 12 = v_I(P)$, pertanto è possibile fare una visita implicita di $P_{1,2}$, ossia chiudere il nodo. Dal nodo $P_{1,1}$ istanziamo la variabile x_2 che ha valore frazionario nella soluzione ottima del rilassamento continuo di $P_{1,1}$.

La soluzione ottima del rilassamento di $P_{2,1}$ è $(3,3)$, quindi $v_S(P_{2,1}) = 12 = v_I(P)$, pertanto chiudiamo il nodo $P_{2,1}$. La soluzione ottima del rilassamento di $P_{2,2}$ è $(1,4)$, quindi $v_S(P_{2,2}) = 13 > 12 = v_I(P)$. Poiché tale soluzione è ammissibile aggiorniamo la vecchia soluzione ammissibile $(3,3)$ con la nuova $(1,4)$, poniamo $v(P) = 13$ e chiudiamo il nodo $P_{2,2}$.



□

Capitolo 5

Problema di copertura

5.1 Lezione 21 03/04

	1	2	3	4	5
1	20	18	16	14	26
2	11	15	22	24	10
3	23	13	14	15	12
4	14	15	18	22	26
5	15	19	13	24	25
6	30	28	10	16	14
7	20	22	15	19	23
8	21	20	16	10	10
9	9	11	12	14	13

Problema 7 Problema di copertura: Trovare la copertura minima. Nella tabella le righe (1-9) sono i quartieri (m), le colonne (1-5) sono le sedi delle asl (n) (dove c'è l'ambulanza ad esempio) e i valori nella tabella sono il tempo di viaggio. Il vincolo è che il tempo tassativo di percorrenza deve essere inferiore o uguale ai 15 minuti (bisogna garantire copertura a tutti i quartieri) □

Osservazioni di riduzione:

- osservazione 0: riduzione della matrice A (matrice degli elementi nella tabella $m \times n$): porto a 0 e 1 dove c'è un valore ≤ 15, si ottiene così una matrice binaria:

	1	2	3	4	5
1	0	0	0	1	0
2	1	1	0	0	1
3	0	1	1	1	1
4	1	1	0	0	0
5	1	0	1	0	0
6	0	0	1	0	1
7	0	0	1	0	0
8	0	0	0	1	1
9	1	1	1	1	1

- osservazione 1: c'è una riga tutti 0? il problema va risolto in un'altra maniera (o semplicemente la elimino)
- osservazione 2: c'è una riga tutti 1? si toglie in quanto "sempre buona"
- osservazione 3: riga con un solo 1? (ad esempio la riga 1 dove solo $x_4 = 1$) cancello tutte le righe dove $x_4 = 1$:

	1	2	3	5
2	1	1	0	1
4	1	1	0	0
5	1	0	1	0
6	0	0	1	1
7	0	0	1	0

- aggiungiamo fattore del costo dell'asl C_i:

	C1	C2	C3	C4	C5
	1	2	3	4	5
1	0	0	0	1	0

- osservazione 4: supponiamo che 2 zone siano nella seguente situazione:

1	2
1	1
1	1
0	0
0	0
1	0
0	0
1	0

ci sono degli 1 nella stessa posizione, in più una ha degli 1 che l'altra non ha (situazione di dominanza): se non ho i costi o questi sono del tipo $C_2 \geq C_1$ posso eliminare la colonna 2

5.1.1 Modello

Modello copertura minima:

$$\begin{cases} min\ c_1x_1 + ... + c_nx_n = cx \\ Ax \geq p \\ x_i \in \{0, 1\} \end{cases}$$

Nota 40: *p è un vettore di tutti 1.*

5.1.2 Algoritmi (VS)

Esempio 13: come scelgo un servizio che apre in più quartieri rispetto a quello che costa meno? (concetto del rendimento)

Ci:	7	8	5	3
	1	2	3	4
1	1	1	0	0
2	0	1	0	1
3	1	1	1	0
4	0	1	1	1
5	1	0	0	1
6	0	1	0	1
7	1	1	1	0
	4	6	3	4

Tabella 5.1: la prima riga sono i costi C_i mentre l'ultima il numero di quartieri coperto da ogni colonna (asl)

Algorimo 1: apro prima il servizio che costa meno: $x_4 = 1$:

- cancello colonna 4
- cancello righe 2, 4, 5, 6

$x_3 = 1$

- cacello colonna 3
- cancello righe 3, 7

e infine $x_1 = 1$; si ha quindi $x = (1, 0, 1, 1)$ quindi $V_S = 7 + 5 + 3 = 15$

Algoritmo 2: apro prima quello con più quartieri: $x_2 = 1$:

- cancello colonna 2
- cancello tutte le righe tralle la 5

$x_4 = 1$ (anche $x_1 = 1$ ma ha costo maggiore); si ha quindi $x = (0, 1, 0, 1)$ e quindi $V_S = 11$. Tra le due V_S scelgo la seconda in quanto migliore.

$\square$

Algoritmo II.8 Algoritmo di Chvatal (VS): calcolo il costo unitario (ovvero il rendimento) e scelgo prima il minore:

Esempio 14: dall'esempio precedente:

$$x_1 = \frac{7}{4},\ \ x_2 = \frac{8}{6},\ \ x_3 = \frac{5}{3},\ \ x_4 = \frac{3}{4}$$

scelgo quindi x_4 e cancello e rimangono le righe 1, 3, 7. Scelgo poi x_1 (dopo aver prima ricalcolato i costi unitari) e cancello tutte le righe rimanenti e quindi $x = (1, 0, 0, 1)$ $\square$

$\square$

Nota 41: *questo algoritmo, a differenza di quello di Kruskal, non trova sempre l'ottimo*

5.2 Lezione 22 04/04 Copertura massima

h		1	2	3	4
8000	1	1	0	1	0
5000	2	1	0	1	1
6000	3	1	1	1	0
7000	4	0	1	1	0
3500	5	1	0	0	1
4000	6	0	1	0	0
5000	7	0	1	0	1

Tabella 5.2: Gli h_i sono la popolazione di ogni quartiere, $j = 1, ..., 4$ sono i siti in cui le sedi possono aprire.

Consideriamo che una sede può aprire al massimo in due quartieri: $P = 2$ (da $P \leq 2$). Utilizziamo poi una variabile per i servizi:

$$x_j = \begin{cases} 0 \\ 1 \end{cases} \quad j = 1, ..., 4$$

e una variabile per quali quartieri sono serviti

$$z_i = \begin{cases} 0 \\ 1 \end{cases} \quad i = 1, ..., 7$$

Modello: il modello che quindi si ottiene è:

$$\begin{cases} max\ h_1z_1 + ... + h_7z_7 = hz \\ \sum_{j=1}^{4} x_j = P \\ \sum_{j=1}^{4} c_{ij}x_j \geq z_i \ \ \forall i \\ x, z \in \{0, 1\} \end{cases}$$

Nota 42: *la seconda sommatoria va a formare la matrice Ax*

Nota 43: *nel modello il vincolo di budget cerca di portare le x a 0, la funzione obbiettivo invece cerca di portare le z a 1*

5.2.1 Calcolo di VI e VS

La V_I si fa tramite un approccio greedy, la V_S invece tramite il rilassato continuo.

Calcolo VI

Si guarda la funzione obbiettivo (nel nostro caso tiene conto di h fortemente). Sommo quindi per ogni colonna gli h:

$$22500 \mid 22000 \mid 26000 \mid 13500$$

Il maggiore è quindi x_3, dunque lo pongo $= 1$. La matrice ridotta che si ottiene è

$$\begin{array}{c|ccc|} & 1 & 2 & 4 \\ \hline 5 & 1 & 0 & 1 \\ 6 & 0 & 1 & 0 \\ 7 & 0 & 1 & 1 \\ \hline \end{array}$$

Faccio nuovamente la somma (la precedente non è più valida)

$$3500 \mid 9000 \mid 8500$$

risulta quindi che x_2 è il maggiore. Riduco nuovamente e ottengo

$$x = (0, 1, 1, 0)$$

quindi $V_I = 26000 + 9000 = 35000$ (mi fermo perchè il vincolo di budget mi imponeva di aprire 2 sedi).
La soluzione ammissibile che ho trovato dunque è:

$$(0, 1, 1, 0|1, 1, 1, 1, 0, 1, 1)$$

con la funzione obbiettivo:

$$c = (0, 0, 0, 0|h_1, h_2, h_3, h_4, h_5, h_6, h_7)$$

Nota 44 : Attenzione!!!**:** *a intlinprog va passata col - (in quanto è di minimo)*

Parte III

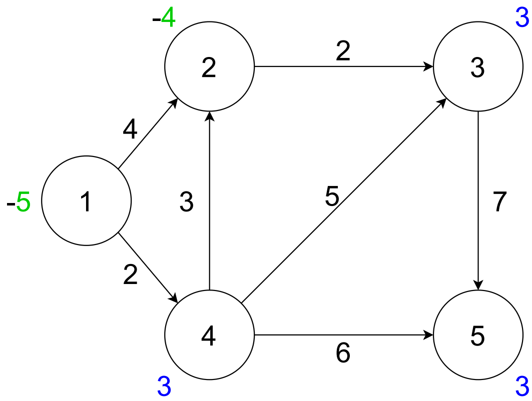
PL su reti

Capitolo 6

Reti non capacitate

6.1 Lezione 23 18/04

Studiamo il modello del flusso di distribuzione sulla rete.



La rete sopra raffiguarata è composta da n nodi $N = \{1, 2, 3, 4, 5\}$ e m archi $A \subseteq N \times N$ orientati $\big((4,3) \neq (3,4)\big)$. La rete è inoltre incompleta: mancano degli archi.

I numeri in verde rappresentano il flusso prodotto da un nodo, quelli blu, al contrario, il flusso richiesto: questi flussi prendono il nome di **bilanci dei nodi** ($b_i \;\; i = 1, ..., n$). In nero invece sono descritti i costi degli archi: più precisamente i costi unitari di gestione degli archi (c_{ij}).

Problema 8 Flusso di costo minimo: cerchiamo un flusso sulla rete, a costo minimo, che rispetti le richieste e le produzioni di flusso. □

Nota 45: *nel nostro caso* $5 + 4 = 3 + 3 + 3$ *e quindi la rete è bilanciata*

Utilizziamo il segno $-$ per evidenziare le sorgenti:

Definizione 19 Rete bilanciata: se $\sum b_i = 0$ la rete si dice bilanciata. □

6.1.1 Reti sbilanciate

Le reti sbilanciate possono essere di due tipi:

- se $\sum b_i > 0$ (ovvero più richiesta dell'offerta), e in assenza di priorità di flusso da parte di certi nodi richiedenti, il problema è irrisolvibile.
- se $\sum b_i < 0$ (ovvero più offerta della richiesta) si inventa un pozzo fittizio a costo 0 dove si mandano gli eccessi.

6.1.2 Reti bilanciate

Per iniziare a studiare il modello ci serviamo della variabile decisionale

$$x_{ij}, \quad (i,j) \in A$$

Nota 46: *una soluzione potrebbe non essere ammissibile se non c'è un cammino tra nodi produttori e nodi richiedenti.*

Esempio 15 Esempio di soluzione ammissibile**:** (soluzione costruita a tentativi)

$$x = (0, 5, 4, 1, 0, 0, 2)$$

in cui le componenti di x sono le quantità di flusso che passano dall'arco ij. Il costo di questa soluzione è $C = 37$ ed è dato da:

$$\sum (c_{ij} \cdot \text{unità di flusso che passano})$$

□

Costruzione del modello

Dall'esempio iniziale:

$$\begin{cases} min\ cx = min\ 4x_{12} + 2x_{14} + 2x_{23} + ... + 6x_{45} \\ x_{45} + x_{35} = 3 \\ x_{14} - x_{42} - x_{43} - x_{45} = 3 \\ x_{23} + x_{43} - x_{35} = 3 \\ x_{12} + x_{42} - x_{23} = -4 \\ -x_{12} - x_{14} = -5 \\ x_{ij} \geq 0 \end{cases}$$

Nota 47: *Le equazioni sono dette "equazioni di bilancio ai nodi", il vincolo di positività è necessario in quanto flussi negativi non hanno senso.*

Nota 48: *Il fatto che serva il vincolo di interezza dipende dal problema*

Matrice E: Le colonne della matrice sono rispettivamente gli archi:

$$12, 14, 23, 35, 42, 43, 45$$

mentre le righe sono i nodi da 1 a 5.

$$E = \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{12} \\ x_{14} \\ x_{23} \\ x_{35} \\ x_{42} \\ x_{43} \\ x_{45} \end{pmatrix} = -x_{12} - x_{14} ...$$

Le equazioni di bilancio ai nodi sono quindi date dal prodotto tra E e la variabile decisionale:

$$Ex = b$$

Modello: il modello finale sarà quindi:

$$\begin{cases} min\ cx \\ Ex = b \\ x \geq 0 \end{cases}$$

Nota 49: *il modello non è altro che un duale standard (che quindi posso risolvere tramite il simplesso)*

6.1.3 PL

Se voglio vedere la PL devo avere *archi* $\times$ *nodi*:

$$\begin{cases} min\ cx \\ x^T E^T = b^T \\ x \geq 0 \end{cases}$$

Guardando la matrice E si nota che ogni colonna ha uno e un solo 1 e uno e un solo -1 con tutti gli altri elementi 0; E^T è quindi linearmente dipendente e il rango non può essere massimo. Non ci sono quindi matrici di base invertibili, e di conseguenza non ci sono basi: questo esclude la possibilità di applicare il simplesso.

6.1.4 Utilizzo degli alberi di copertura

Definizione 20 Albero di copertura: insieme di archi che connette tutte la rete senza creare cicli. □

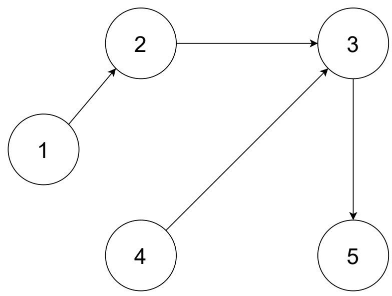


Figura 6.1: esempio di albero di copertura T dell'esempio iniziale

Nota 50: *l'albero di copertura ha dimensione* $n-1$

La matrice E relativa alla Definizione 6.1.4 è:

$$E = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Essendo, come visto in precedenza, linearmente dipendente, una riga è ridondante e la posso quindi eliminare (ad esempio la prima):

$$E = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

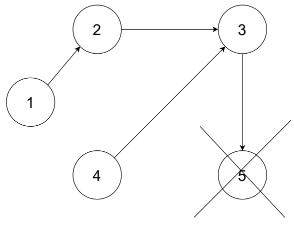
ottenendo quindi una matrice $(n-1) \times m$

Visita posticipata per foglie

A partire da una foglia si costruisce la matrice eliminando dopo ogni passaggio la foglia appena considerata:

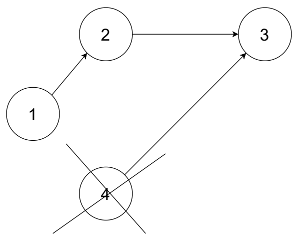
Esempio 16: Esempio dal nostro albero di copertura partendo dal nodo 5 (foglia):

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$



Prendiamo poi la il nodo 4 (essendo foglia):

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$$



e così via fino a ottenere:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Si ottiene quindi una matrice triangolare inferiore sulla cui diagonale non ci sono 0 (per via del fatto che si vanno sempre a prendere le foglie a ogni passaggio) e quindi il rango sarà sempre $n-1$... □

Teorema 10 Caratterizzazione delle basi: Gli alberi di copertura sono tutte e solo le basi:

$$\text{Alberi di copertura} \iff \text{Basi}$$

□

6.2 Lezione 24 19/04

La matrice E può essere scritta come nodi×(archi di base, archi non di base):

$$\left(\underbrace{E_T}_{albero} \quad , \quad \underbrace{E_L}_{archi\ vuoti} \right)$$

Nota 51 : Flusso di base amm. degenere**:** *un flusso di base ammissibile è degenere una delle componenti di x_T è 0.*

6.2.1 Duale del problema

Il duale del problema (che essendo esso stesso un duale sarà quindi un primale) è:

$$\begin{cases} min\ cx \\ Ex = b \\ x \geq 0 \end{cases} \xrightarrow{\pi} \begin{cases} max\ \pi b \\ \pi E \leq c \\ \pi \in \mathbb{R}^n \end{cases}$$

con $\pi E \leq c$ che può essere riscritto come: $\pi(E_T, E_L) \leq c$ che può essere riscritto come:

$$\begin{cases} \pi E_T \leq c_T \\ \pi E_L \leq c_L \end{cases}$$

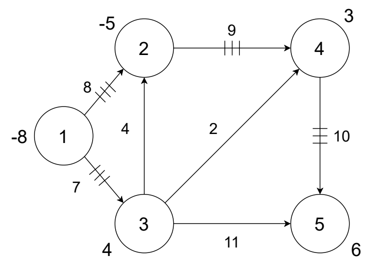
dove $\pi E_T = c_T \longrightarrow -\pi_i + \pi_j = c_{ij} \ \ \forall (i,j) \in T$ e dove

$$-\pi_i + \pi_j \leq c_{ij} \ \ \forall (i,j) \in L$$

è la condizione di ammissibilità.

Potenziali ai nodi della rete

Esempio 17: prendiamo l'albero:



Con la visita posticipata per foglie ottengo la soluzione ammissibile:

$$x = (4, 4, 9, \underbrace{0, 0, 0}_{x_L}, 6)$$

Essendo una equazione ridondate (e quindi superflua) posso porre: un $\pi_i = 0$, ad esempio $\pi_1 = 0$; a partire da esso calcolo dunque tutte le altre differenze di potenziale legate ai nodi di un certo arco ($-\pi_i + \pi_j = c_{ij}$). Ottengo quindi:

$$\pi = (0, 8, 7, 17, 27)$$

dove π ha componenti π_i ($con\ i = 1, ..., 5$) e prende il nome di **potenziale ai nodi della rete**.

Per verificare di essere all'ottimo devo quindi verificare tutti i

$$-\pi_i + \pi_j \leq c_{ij} \ \ \forall (i,j) \in L$$

Non siamo quindi all'ottimo perché non si verifica $-7 + 17 \leq 2$

□

Teorema 11 Teorema di Bellman: Sia T un albero che genera un flusso ammissibile, se

$$-\pi_i + \pi_j \leq c_{ij} \;\; \forall (i,j) \in L$$

allora sono all'ottimo. □

Definizione 21 Costo ridotto dell'arco:

$$C_{ij}^{\pi} = c_{ij} + \pi_i - \pi_j$$

□

Prop. sugli archi di base $C_{ij}^{\pi} = 0$, che è equivalente a dire che $-\pi_i + \pi_j = c_{ij} \;\; \forall (i,j) \in T$ □

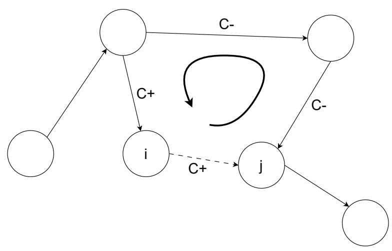
Nota 52: *La condizione di ammissibilità (e ottimalità per Bellman) diventa quindi* $C_{ij}^{\pi} \geq 0$

6.3 Lezione 25 21/04 Simplesso su reti

Dato un albero di copertura che genera un flusso di base ammissibile, supponiamo che:

$$\exists (i,j) \in L : \; C_{ij}^{\pi} < 0$$

ovvero che c'è un vincolo violato e quindi devo fare il cambio di base.



Nota 53: *senza l'arco tratteggiato si ha l'albero di copertura*

L'arco (i,j) è l'arco entrante ovvero il primo che viola Bellman.

Tecnica di rottura del ciclo: (vedi immagine sopra)

1. si orienta il ciclo in verso concorde all'arco entrante

2. si dividono gli archi in concordi e discordi al verso del ciclo, nominandoli rispettivamente con C^+ e C^-.

3. adopero un aggiornamento del flusso con un $\theta > 0$:

$$x(\theta) = \begin{cases} \overline{x}_{ij} + \theta \;\; (i,j) \in C^+ \\ \overline{x}_{ij} - \theta \;\; (i,j) \in C^- \\ \overline{x}_{ij} \;\; altrove \end{cases}$$

Teorema 12:

$$cx(\theta) = c\overline{x} + \theta C^{\pi}_{ij}$$

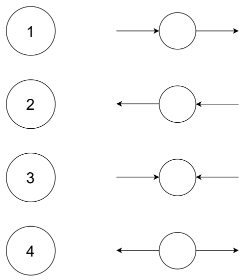
ovvero la nuova funzione obbiettivo è uguale alla vecchia + theta volte il costo ridotto dell'arco entrante. □

Nota 54: *Nel Teorema 12* $\theta \geq 0$ *e* $C^{\pi}_{ij} < 0$ *per ipotesi*

Nota 55: *sull'arco* (i, j) *inizialmente c'è flusso 0 perché non di base*

Più aumento θ più il Teorema 12 "viene bene", però aumentando θ c'è il rischio che la soluzione non sia più ammissibile: per verificare che lo sia devo:

- verificare i bilanci: essendo gli archi (rispetto ai nodi del ciclo) di solo 4 tipi possibili:

  se applico $+\theta$ o $-\theta$ agli archi il bilanciamento sui nodi non cambia

- verificare la positività del flusso:
  - nei casi di C^+ è sempre verificata
  - nei casi di C^- no:

  Regola per scegliere theta ad ogni passo:

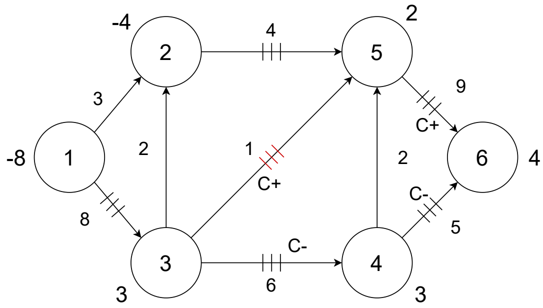
  $$\theta = \min_{(i,j) \in C^-} \overline{x}_{ij}$$

  Utilizzando il minimo ad ogni passo svuoto un arco e quindi interrompo il ciclo che c'era utilizzando come arco uscente quello con il minimo $\overline{x}$ tra quelli di C^-; ottengo dunque un nuovo flusso di base ma di costo inferiore.

Nota 56: *se* $\theta = 0$ *caso degenere.*

Nota 57: *se il ciclo ha solo* C^+ *allora il costo minimo è* $-\infty$

Esempio 18: La prima soluzione ammissibile la trovo col duale ausiliario (in questo caso essendo semplice il problema si può anche fare a occhio) L'arco (2,5) è in base.



la soluzione che quindi si trova è:

$$x = (0, 8, 4, 0, 5, 0, 0, 2, 2)$$

da cui

$$\pi = (0, 3, 8, 14, 10, 19)$$

Calcoliamo quindi i costi ridotti:

- $C_{12}^{\pi} = 0$ (possibile caso degenere)
- $C_{32}^{\pi} = 7$
- $C_{35}^{\pi} = -1$ è negativo quindi $(3, 5)$ è l'arco entrante (in rosso)

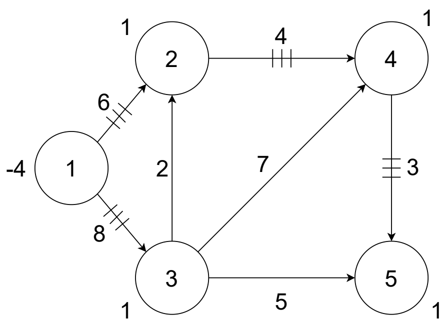
Scelgo quindi tra i C^- quello con la $\overline{x}$ minore: prendo quindi $(4, 6)$ che ha $\overline{x} = 2$. Aggiorno quindi il flusso (con le regole descritte prima) e ottengo:

$$x = (0, 8, 4, 0, 3, 2, 0, 0, 4)$$

□

Nota 58 : Domanda**:** *quando un flusso di base è ammissibile? quando rispetta la positività (in quanto i bilanci sono già rispettati)*

Problema 9 Cammini minimi (shortest paths): Per convenzione l'origine è il nodo 1, la fine un nodo k.



Il modello rimane:

$$\begin{cases} min\ cx \\ Ex = b \\ x \geq 0 \end{cases}$$

in cui nelle equazioni di bilancio b è definito come:

$$\begin{cases} -(n-1)\ \ origine \\ 1\ \ altrove \end{cases}$$

Dall'esempio in figura l'albero dei cammini minimi è:

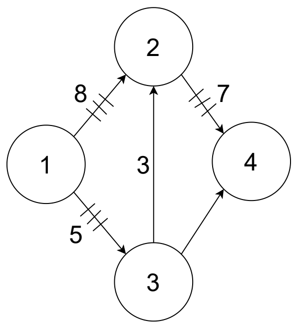
$$x = (3, 1, 2, 0, 0, 0, 1)\ \ costo = 37$$

dove x è trovato tramite il contare dei discendenti di un certo nodo (ad esempio il nodo 1 ha 3 discendenti dal lato dell'arco $(1, 2)$)

□

Nota 59: *x va trovato con simplesso duale nei casi complessi*

Esempio 19: .



è ottimo? per vederlo calcolo i potenziali:

$$\pi = (0, 8, 5, 15)$$

calcolo i costi ridotti

- $C^{\pi}_{32} = 0$
- $C^{\pi}_{34} = -8$ è negativo quindi l'albero in figura non è ottimo e $(3, 4)$ è l'arco entrante

In questo caso l'uscente non può essere altro che l'unico altro arco che incide sul nodo 4, ovvero l'arco $(2, 4)$ (non serve qui controllare i C^{-}); questo perché nei cammini minimi un solo arco può essere entrante in un nodo. Il flusso sarà quindi:

$$x = (2, 1, 1, 0, 0)$$

in cui le singole componenti risultano semplici da calcolare in quanto basta contare il numero di figli di un nodo (per via del fatto che i bilanci non all'origine sono tutti a 1) □

Capitolo 7

Reti capacitate

7.1 Lezione 26 24/04

Fino ad ora abbiamo trattato i problemi di flusso di costo minimo su reti non capacitive. In realtà i problemi di flusso hanno un altro vincolo: la capcacità, ovvero la quantità di flusso che può al massimo passare per un arco. La definiamo come:

$$u \in \mathbb{R}^m$$

Il **modello** aggiornato con la capacità risulterà quindi essere:

$$\begin{cases} min\ cx \\ Ex = b \\ 0 \leq x \leq u \end{cases}$$

Questo modello nella sua forma risulta non essere né duale né primale: devo scegliere a quale modello PL ricondurmi. Scegliamo il duale e introduciamo la variabile di scarto:

$$w \in R^m$$

che indica quanto flusso ancora può passare da quell'arco:

$$\begin{cases} min\ cx \\ Ex = b \\ Ix + Iw = u \\ x, w \geq 0 \end{cases}$$

dove i due vincoli $Ex = b$ e $Ix + Iw = u$ possono essere espressi tramite il calcolo matriciale come:

$$\begin{pmatrix} x^T & w^T \end{pmatrix} \cdot \begin{pmatrix} E^T & I \\ 0 & I \end{pmatrix} = \begin{pmatrix} b \\ u \end{pmatrix}$$

Per risolvere questo problema partiamo dalle basi utilizzando:

Definizione 22 Tecnica di tripartizione degli archi: dividiamo gli archi in 3 blocchi (T, L, U) la cui unione sia A (ovvero tutti gli archi della rete)

Nota 60: *in T ci va un albero di copertura, in U archi da saturare, in L gli archi vuoti.*

□

Esempio 20: Dividiamo gli archi in T (nero), U (rosso), L (blu):

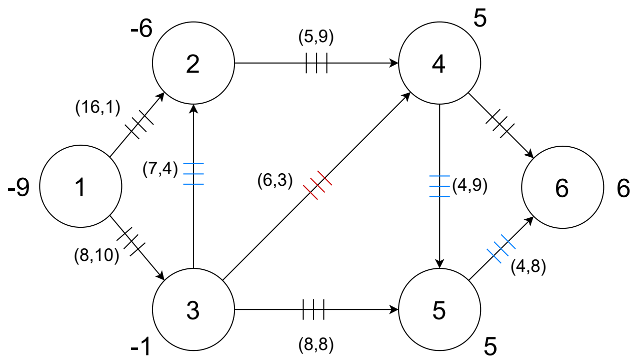
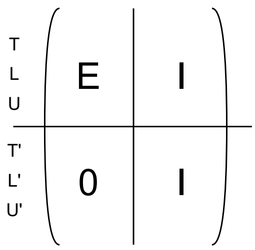


Figura 7.1: le coppie di numeri sugli archi sono del tipo (c_{ij}, u_{ij})

La matrice avrà forma del tipo:



dove le righe con l'indice primo sono quelle di w □

Teorema 13 Teorema di caratterizzazione delle basi: se scelgo gli indici $TUT'L'$ ho una base e viceversa:

$$\text{indici } \; TUT'L' \iff \text{indici di base}$$

□

Riprendendo quindi l'Esempio 20 il flusso di base risulta essere:

$$(2, 7, 8, 0, 3, 5, 0, 6, 0 \mid -1, 3, 1, 4, 0, 3, 9, 4, 8)$$

che è stato costruito nel seguente modo:

1. Per gli archi in L si mettono a 0 tutti i flussi e al massimo gli scarti

2. Per gli archi i n U si mettono al massimo i flussi e a 0 gli scarti
3. Si procede poi per visita posticipata per foglie

Essendo, nel nostro esempio, $w_{12} < 0$ allora la tripartizione non è ammissibile.

Nota 61: *se $u_{12} = 2$ allora $w_{12} = 0$ ovvero avremmo avuto una tripartizione ammissibile degenere in quanto un arco in T è saturo, se invece la capacità fosse stata 3 sarebbe stata non degenere*

Nota 62: *la cardinalità delle soluzioni di base è $m + n - 1$ ($n - 1$ sono sono la base data dall'albero di copertura mentre m sono le ω non a 0 più le U)*

7.1.1 Tabella riassuntiva (dal libro)

	Flusso di base	Potenziale di base
Ammissibile	$\forall(i,j) \in T$ si ha $0 \leq x_{ij} \leq u_{ij}$	$\forall(i,j) \in L$ si ha $C^{\pi}_{ij} \geq 0$ $\forall(i,j) \in U$ si ha $C^{\pi}_{ij} \leq 0$
Non ammissibile	$\exists\,(i,j) \in T$ tale che $x_{ij} < 0$ oppure $\exists(i,j) \in T$ tale che $x_{ij} > u_{ij}$	$\exists\,(i,j) \in L$ tale che $C^{\pi}_{ij} < 0$ oppure $\exists\,(i,j) \in U$ tale che $C^{\pi}_{ij} > 0$
Degenere	$\exists\,(i,j) \in T$ tale che $x_{ij} = 0$ oppure $\exists\,(i,j) \in T$ tale che $x_{ij} = u_{ij}$	$\exists\,(i,j) \in L$ tale che $C^{\pi}_{ij} = 0$ oppure $\exists\,(i,j) \in U$ tale che $C^{\pi}_{ij} = 0$
Non degenere	$\forall(i,j) \in T$ si ha $x_{ij} \neq 0$ e $x_{ij} \neq u_{ij}$	$\forall(i,j) \in L$ si ha $C^{\pi}_{ij} \neq 0$ $\forall(i,j) \in U$ si ha $C^{\pi}_{ij} \neq 0$

Capitolo 8

Flusso massimo

8.1 Lezione 27 26/04

Dal modello del Problema 9 su reti capacitate:

$$\begin{cases} min\ cx \\ Ex = b \\ 0 \leq x \leq u \end{cases}$$

il problema del potenziale avrà come variabili π e μ e quindi il modello sarà:

$$\begin{cases} max\ \pi b + \mu u \\ \begin{pmatrix} E^T & I \\ 0 & I \end{pmatrix} \begin{pmatrix} \pi \\ \mu \end{pmatrix} \leq \begin{pmatrix} c \\ 0 \end{pmatrix} \end{cases} \longrightarrow \begin{cases} max\ \pi b + \mu u \\ E^T \pi + I\mu \leq c \\ \mu \leq 0 \end{cases}$$

Da cui il sistema di base (considerando gli indici $TUT'L'$) è:

$$\begin{cases} E_T^T \pi + \mu_T = c_T \\ E_U^T \pi + \mu_U = c_U \\ \mu_{T'} = 0 \\ \mu_{L'} = 0 \end{cases}$$

La relativa soluzione del sistema è quindi:

$$(\pi, 0, 0, \mu_U) = (\pi, 0, 0, c_U - E_U^T \pi)$$

Per trovare la soluzione mi basta quindi calcolare π da $\pi E_T^T = c_T$
Nota 63: *è lo stesso delle reti non capacitate.*

8.1.1 Soluzioni non di base

Le soluzioni non di base (ovvero quelle LU') sono:

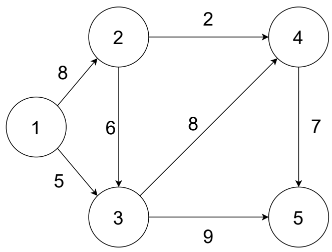
$$\begin{cases} \pi E_L \leq c_L \\ \mu_U \leq 0 \end{cases} \longrightarrow$$

$$\begin{cases} -\pi_i + \pi_j \leq c_{ij} \longrightarrow C^{\pi}_{ij} \geq 0 \;\; \forall (i,j) \in L \\ c_U - \pi E_U \leq 0 \rightarrow c_U \leq \pi E_U \rightarrow c_{ij} \leq -\pi_i + \pi_j \rightarrow C^{\pi}_{ij} \leq 0 \;\; \forall (i,j) \in U \end{cases}$$

Nota 64: *quando è quindi ammissibile il potenziale? quando i costi ridotti di L sono positivi mentre quelli di U sono negativi*

Osservazione: come il determinante degli alberi di copertura il determinante della matrice di base fa ± 1 quindi posso applicare il teorema dell'interezza

Problema 10 Flusso massimo (max flow): Sugli archi sono indicate le portate massime



Per convenzione l'origine del flusso è sempre il nodo 1, la destinazione invece nel nostro caso è il nodo 5. Un esempio di soluzione ammissibile è:

$$x = (2, 5, 0, 2, 0, 5, 2)$$

□

Modello:

$$\begin{cases} max\ v \\ Ex = b \rightarrow b = \begin{cases} 0 \;\; i \neq 1,5 \\ v \;\; i = 5 \\ -v \;\; i = 1 \end{cases} \\ 0 \leq x \leq u \\ v \geq 0 \end{cases}$$

dove v è la quantità di flusso uscente dall'origine.

8.2 Lezione 28 28/04

8.2.1 Teoremi di Bellman

Teorema 14 Teorema di Bellman iii: data una tripartizione che genera un flusso di base ammissibile[1] se:

$$C^{\pi}_{ij} \geq 0 \ \ \forall (i,j) \in L$$

$$C^{\pi}_{ij} \leq 0 \ \ \forall (i,j) \in U$$

allora (T, L, U) è ottima. □

Teorema 15 Teorema di bellman iv: data T, L, U che genera un potenziale ammissibile, se:

$$0 \leq \overline{x}_{ij} \leq u_{ij} \ \ \forall (i,j)$$

allora T, L, U è ottima. □

Teorema 16 Teorema di Bellman ii: data T, L, U che genera un potenziale ammissibile, se:

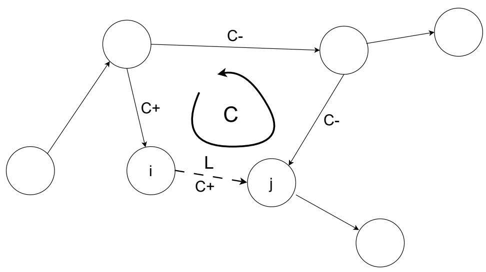
$$\overline{x}_{ij} \geq 0$$

allora T, L, U è ottima. □

8.2.2 Condizioni di Bellman non verificate

Supponiamo che le condizioni di Bellman non siano verificate: devo quindi fare un cambio di base (parte il simplesso ora per reti capacitate). Si distinguono 2 casi:

1. se $\exists (i,j) \in L \ : \ C^{\pi}_{ij} < 0$ esso è l'arco entrante in T



Nota 65 : Ripasso**:** *come già visto:*

$$x(\theta) = \begin{cases} \overline{x}_{ij} + \theta \ \ (i,j) \in C^{+} \\ \overline{x}_{ij} - \theta \ \ (i,j) \in C^{-} \\ \overline{x}_{ij} \ \ altrove \end{cases}$$

[1] i flussi di base rispettano sempre i bilanci

e la funzione obbiettivo aggiornata:

$$cx(\theta) = c\overline{x} + \theta C_{ij}^{\pi}$$

Devo quindi verificare:

- i bilanci b_i: sono sempre verificati $\forall\theta$ (vedi sezione 6.3)
- $0 \le x \le u$:

$$\theta^{-} = \min_{(i,j)\in C^{-}} \{\overline{x}_{ij}\}$$

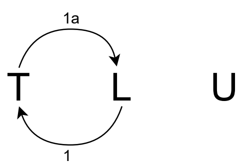
$$\theta^{+} = \min_{(i,j)\in C^{+}} \{u_{ij} - \overline{x}_{ij}\}$$

$$\theta = min\{\theta^{-}, \theta^{+}\}$$

Come si decide l'arco uscente? ci sono 2 casi:

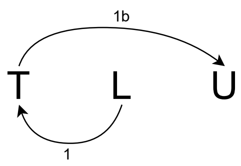
$$\begin{cases} \theta = \theta^{-} \\ \theta = \theta^{+} \end{cases}$$

(a) se $\theta = \theta^{-}$ allora un arco in C^{-} si svuota



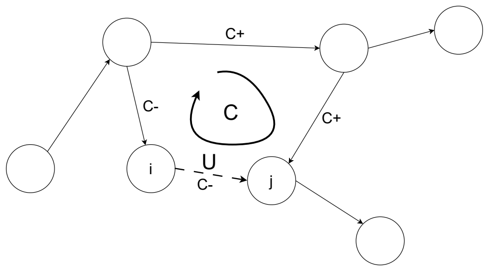
Nota 66: *in questo caso non ci può essere loop perché l'arco entrante è in C^{+} mentre quello uscente in C^{-}*

(b) se $\theta = \theta^{+}$ allora un arco in C^{+} si satura



Nota 67: *in questo caso l'arco che va in T può essere quello che esce, T quindi non cambia ma è la tripartizione che lo fa (la base è cambiata)*

2. se $\exists(i,j) \in U\ :\ C_{ij}^{\pi} > 0$ esso è l'arco entrante in T



Nota 68: *in questo caso il ciclo lo oriento in verso discorde a (i,j)*

Si fa quindi di nuovo l'update del flusso:

$$x(\theta) = \begin{cases} \overline{x}_{ij} + \theta \ \ (i,j) \in C^+ \\ \overline{x}_{ij} - \theta \ \ (i,j) \in C^- \\ \overline{x}_{ij} \ \ altrove \end{cases}$$

e della funzione obbiettivo:

$$cx(\theta) = c\overline{x} - \theta C^{\pi}_{ij}$$

Devo quindi verificare:

- i bilanci: che sono verificati $\forall\theta$
- $0 \leq x \leq u$:

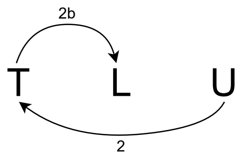
$$\theta^- = \min_{(i,j)\in C^-} \{\overline{x}_{ij}\}$$

$$\theta^+ = \min_{(i,j)\in C^+} \{u_{ij} - \overline{x}_{ij}\}$$

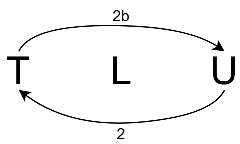
$$\theta = min\{\theta^-, \theta^+\}$$

Per trovare dunque l'arco uscente ci sono 2 casi:

(a) se $\theta = \theta^-$ allora un arco in C^- si svuota



(b) se $\theta = \theta^+$ allora un arco in C^+ si satura

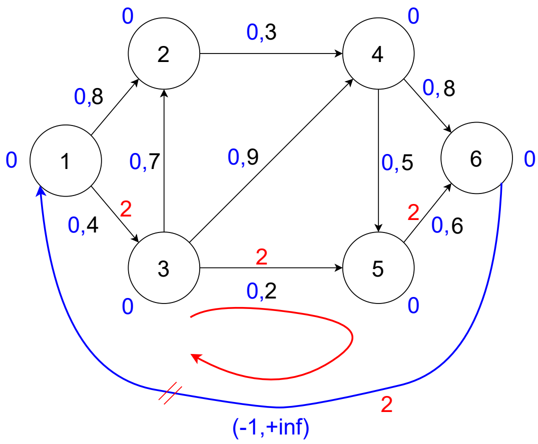


8.3 Lezione 29 02/05

Il modello per il flusso massimo che abbiamo quindi trovato è:

$$\begin{cases} max\ v \\ Ex = b \rightarrow b = \begin{cases} 0 & i \neq t, s \\ v & i = t \\ -v & i = s \end{cases} \\ 0 \leq x \leq u \\ v \geq 0 \end{cases}$$

dove s è il nodo sorgente mentre t il nodo destinazione.



Creo il modello di **flusso minimo** per interpretare il **flusso massimo** (nell'immagine in blu):

- collego il nodo destinatario alla sorgente con costo -1
- metto i bilanci a 0
- metto i costi a 0

La funzione obbiettivo (nel nostro esempio) è:

$$-x_{61} \text{ e quindi } cx = -1$$

una soluzione ammissibile sarà quindi:

$$x = (1, 0, 1, 0, 0, 0, 0, 1, 0, 1)$$

Domanda: dato un flusso come faccio a vedere se è o no di base?

Algoritmo: gli archi nè vuoti nè saturi stanno in T. Questi archi formano un ciclo?

- se sì allora non sono di base
- se no devo formare un albero di copertura:

Esempio 21: dato

$$(5, 0, 3, 0, 2, 4)$$

con 5, 3, 2 in T. se l'ultimo arco che devo mettere in T è uno tra quelli a 0 quello a 4 va in U mentre l'altro 0 in L. Se 4 va in T entrambi gli 0 vanno in L. □

Esempio 22: Dalla rete iniziale: esempio di soluzione:

$$x = 0 \ \text{(vettore)}$$

in:

- T ci vanno: 12, 13, 24, 35, 56
- L ci vanno: 32, 34, 45, 46, 61

il potenziale quindi è:

$$\pi = 0$$

calcolando i costi ridotti: $C^{\pi}_{61} < 0$ allora (6,1) è l'arco entrante in base. Individuiamo ora il ciclo (in rosso):

$$C^- = \phi \rightarrow \theta^- = \{+\infty\}$$

$$C^+ = \{13, 35, 56, 61\} \rightarrow \theta^+ = \min_{C^+}\{u_{ij} - x_{ij}\} = \min_{C^+}\{4, 2, 6, M\}$$

si esegue l'update del flusso:

$$x_{new} = (0, 2, 0, 0, 0, 2, 0, 0, 2, 2)$$

arco uscente (ovvero l'arco che mi da il theta), ovvero l'arco (3,5), va in U

$$(3, 5) \in U$$

potenziali nuovi (trovati da $-\pi_6 + \pi_1 = c_{61}$):

$$\pi = (0, 0, 0, 0, 1, 1)$$

calcolo costi ridotti:

$$...C^{\pi}_{45} < 0$$

(4,5) è quindi l'arco entrante.

$$\theta^+ = min\{8, 3, 5, 4, +\infty)$$

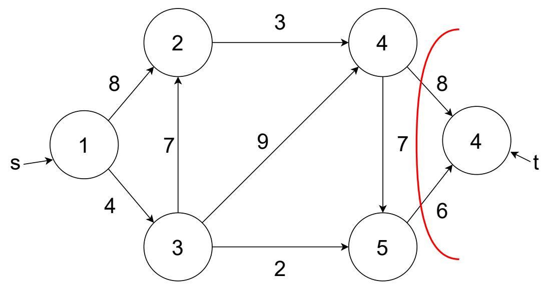
3 è il minimo e quindi l'arco uscente è 24. Update di flusso:

$$x_{new2} = (3, 2, 3, 0, 0, 2, 3, 0, 5, 5)$$

Nota 69: *al primo passo ho "mandato" 2 unità (attraverso 35 che poi è uscito dalla base), al secondo passo ho "mandato" 3 unità: in totale 5 unità sono state mandate (alla fine del passo due).*

□

8.3.1 Taglio della rete



Definizione 23 Taglio della rete: è una partizione dei nodi in due blocchi tali che:

$$s \in N_s \;\; e \;\; t \in N_t \;\; \text{tali che:} \;\; N = N_s \cup N_t \; e \; N_s \cap N_t = \phi$$

□

Nota 70: *gli altri nodi (che non siano t o s) hanno* 2^n *possibilità diverse di essere suddivisi negli insiemi*

Per ogni taglio esiste il suo insieme degli archi diretti;

Definizione 24 Archi diretti: si dicono archi diretti:

$$\underset{(i,j)}{A^+} \in (N_s, N_t)$$

con $i \in N_s$ e $j \in N_t$. □

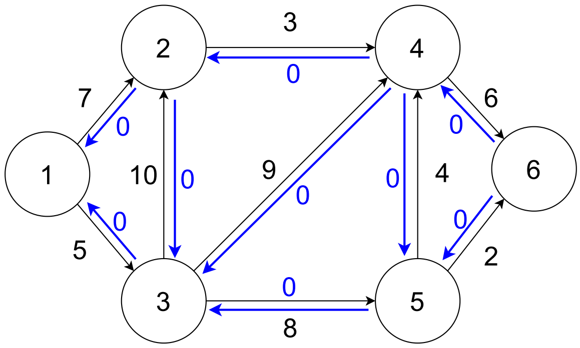
Definizione 25 Capacità di un taglio:

$$u(N_s, N_t) = \sum_{(i,j) \in A^+} u_{ij}$$

□

Teorema 17 Max flow / min cut: Il flusso massimo della rete è uguale al taglio di capacità minima. □

Algoritmo III.1 Algoritmo Ford-Fulkerson: prendiamo una rete:



Costruzione di G_{RES} (grafo residuo, in blu):

- stessi nodi
- per ogni arco (i, j) metto in più l'arco (fittizio) (j, i)
- su ogni arco (fittizio) metto le capacità residue iniziali uguali a 0
- sugli altri archi (veri) le capacità residue sono pari alla capacità

Definizione 26 Cammino aumentante sul grafo residuo: è un cammino orientato nel grafo residuo da s a t (con capacità residua positiva) □

Definizione 27 Capacità aumentante: è uguale alla minima capacità residua δ degli archi che fanno parte del cammino aumentante. □

1. **Passo 0** $\overline{x} = 0$
2. **Passo 1** cerca un C_{aum} su G_{res}, se non lo trovo STOP ottimo. Altrimenti passo 2.
3. **Passo 2** aggiorno:

$$\begin{cases} \overline{x}_{ij} = \overline{x}_{ij} + \delta \quad con \ (i, j) \in C_{aum} \\ \overline{x}_{ij} = \overline{x}_{ij} \quad con \ (i, j) \notin C_{aum} \end{cases}$$

e aggiorno:

$$r_{ij} = u_{ij} - x_{ij} + x_{ji}$$

e vai al passo 1.

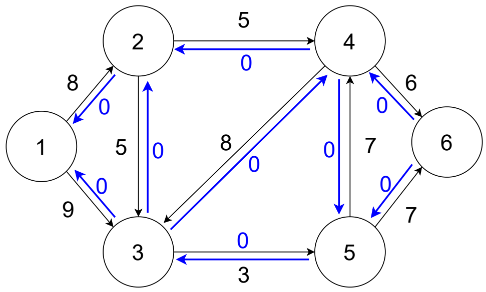
□

Teorema 18 Teorema di Ford-Fulkerson: l'algoritmo di F.F. è corretto e termina in un numero finito di passi. □

8.4 Lezione 30 03/05

8.4.1 Ripasso ultima lezione

Rete:



Algoritmo di Ford Fulkerson: si basa su 4 elementi:

- G_{RES}
- $r_{ij} = u_{ij} - x_{ij} + x_{ji}$ ovvero la capacità residua (sugli archi veri corrisponde inizialmente con la portata).
- C_{aum} su G_{RES}: cammino da s a t con archi con $r_{ij} > 0$
- δ

Definizione 28 Delta di un cammino aumentante:

$$\delta_{C_{aum}} = \min_{(i,j) \in C_{aum}} r_{ij}$$

□

Passi dell'algoritmo (dopo aver posto $x = 0$):

1. cerco C_{aum}, se non lo trovo $\overline{x}$ è ottimo; altrimenti passo al passo 2
2. aggiorno
$$\overline{x}_{ij} = \begin{cases} \overline{x}_{ij} + \delta & con \ \ (i,j) \in C_{aum} \\ \overline{x}_{ij} & con \ \ (i,j) \notin C_{aum} \end{cases}$$
aggiorno r_{ij} e vai al passo 1.

Teorema di F.F.: l'algoritmo di F.F. è corretto e termina in un numero finito di passi.

8.4.2 Come si trova Caum

Algoritmo III.2 Algoritmo di Edmons-Karp (della croce): Si crea una croce a due colonne del tipo:

dove, a partire da s, aggiungo nella colonna sinistra i nodi che visito mentre in quella di destra a quali essi sono collegati direttamente con $r_{ij} > 0$ (senza ripeterli dopo averli scritti la prima volta nella colonna) fino ad arrivare a t.

Nota 71: *L'ordine in cui visito i nodi è quello in cui sono inseriti nella colonna di destra.*

Dal nostro esempio mostrato nell'immagine la croce sarà:

1	2
	3
2	4
3	5
4	6

il C_{aum} sarà quindi:

$$1 \rightarrow 2 \rightarrow 4 \rightarrow 6$$

Degli archi in C_{aum} si prende il minimo degli r_{ij} per avere δ (nel nostro caso $\delta = 5$). Il flusso aggiornato sarà dunque:

$$x = (5, 0, 0, 5, 0, 0, 0, 5, 0)$$

e $v = 5$. Si aggiornano quindi le capacità seguendo la regola:

$$r_{ij} + r_{ji} = u_{ij}$$

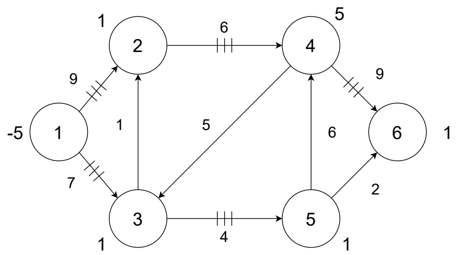
e poi si ripete l'algoritmo fino allo **STOP: quando nella colonna di destra non ci sono più elementi (e non sono quindi arrivato a t) non c'è C_{aum} e quindi sono all'ottimo** □

Nel nostro caso l'algoritmo termina al terzo passo con la croce:

1	2
	3
2	/
3	/

Questo produce $N_S = \{1,2,3\}$

8.5 Lezione 31 05/05



8.5.1 Cammini minimi

Cammino minimo: albero orientato di costo minimo. Si trova con l'algoritmo di Dijkstra.

Algoritmo III.3 Dijkstra: si costruiscono 2 vettori:

- **etichette**, identificato con $\pi \in \mathbb{R}^n$, che nel nostro esempio è:

$$\pi = (0, +\infty, +\infty, +\infty, +\infty, +\infty)$$

- **predecessori**:

$$P = (1, \phi, \phi, \phi, \phi, \phi)$$

Si procede quindi con i seguenti passi:

1. si estrae da N il nodo di etichetta minima, (i), e si toglie da N, questo fintanto che $N \neq \phi$.
2. $\forall j \in FS(i)$, dove FS è la stella uscente, si controlla se:

$$\pi_j > \pi_i + c_{ij}$$

   - se **sì** aggiorno:

$$\begin{cases} \pi_j = \pi_i + c_{ij} \\ p_j = i \end{cases}$$

   - **altrimenti** si prosegue e si va al passo 1.

□

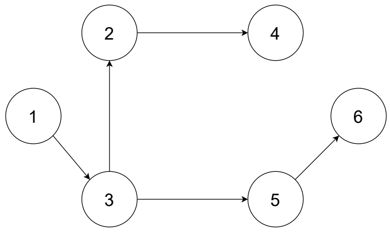
Nel nostro esempio alla fine dell'algoritmo si ha:

$$\pi = (0, 8, 7, 14, 11, 13)$$

$$P = (1, 3, 1, 2, 3, 5)$$

Per ricavare il cammino minimo devo quindi guardare a P e procedere nel seguente modo: *il predecessore di 1 è 1, il predecessore di 2 è 3, quello di 3 è 1 ecc...*

Il cammino minimo che si ottiene è quindi:



π invece rappresenta la lunghezza del cammino dal nodo 1 al nodo i (es. 7 è la lunghezza del cammino da 1 a 3), le etichette non sono quindi altro che i potenziali.

8.5.2 Problemi vuoti su reti

Un problema su reti può essere vuoto nei seguenti casi:

1. $b_i < 0$ su un nodo senza archi uscenti
2. $b_i > 0$ su un nodo senza archi entranti
3. per via delle capacità

In generale il flusso massimo deve essere uguale alla somma dei bilanci dei nodi sorgente per non essere vuoto.

8.5.3 Cammini minimi con vincolo di budget

Dati archi:

$$(c, l)$$

problema di trovare il cammino di lunghezza minima di costo non superiore a C.

Modello:

$$\begin{cases} min\ lx \\ Ex = b \\ x \geq 0 \\ cx \leq C \\ x \in \mathbf{Z}^n \end{cases}$$

È un problema di PLI quindi, sapendo VI e VS si potrebbe risolvere con il branch and bound:

- per la VI si fa il rilassamento cancellando il vincolo di budget. (= Dijkstra su l)
- per la VS: $\begin{cases} min\ cx \\ Ex = b \\ x \geq 0 \end{cases}$ (= Dijkstra su c), se $\leq C$ allora siamo all'ottimo.

Parte IV

PNL

Capitolo 9

Prerequisiti di Analisi 2

9.1 Lezione 32 09/05

Definizione 29 Gradiente:

$$\nabla f(x^0) = \left(\frac{\partial f(x^0)}{\partial x_1}, \frac{\partial f(x^0)}{\partial x_2}, \dots, \frac{\partial f(x^0)}{\partial x_n} \right)$$

□

Definizione 30 Concetto di restrizione: (nel campo delle derivate parziali) si restringe la derivata lungo una retta (direzione). □

Considerando il dominio D compatto (=chiuso e limitato):

Teorema 19 Teorema di Fermat:

$$\begin{cases} \overline{x} \text{ minimo locale} \\ \overline{x} \text{ interno a } D \end{cases} \Rightarrow \nabla f(\overline{x}) = 0$$

□

Teorema 20:

$$\text{se } f \in \mathcal{C} \Rightarrow \exists \text{ min assoluto}$$

□

Capitolo 10

Teoria della PNL

10.1 Lezione 33 10/05

Data $f \in \mathcal{C}^2$ la PNL si occupa di trovare

$$\min_{x \in D} f(x)$$

con D chiuso e limitato (ovvero compatto).

Nota 72: *il minimo che si va a cercare è assoluto: qui, a differenza di PL e PL su reti, oltre al minimo assoluto però ci possono essere i minimi locali.*

Dal tipo di problema si presenta la seguente situazione:

- date queste ipotesi il minimo esiste sicuramente
- $\nabla f(x) = 0$ è condizione necessaria per il minimo all'interno (interno + frontiera)
- $Hf(x) \geq 0$ è condizione necessaria all'interno se $\nabla f(x) = 0$
- $Hf(x) > 0$ è condizione sufficiente all'interno se $\nabla f(x) = 0$

Queste condizioni valgono però per i minimi locali: per trovare il minimo globale dovrei calcolarli tutti e prendere il minimo tra essi. Questo risulta però complesso e quindi bisogna cercare di lavorare con alcuni minimi locali; infatti tutti gli algoritmi ci garantiscono almeno un minimo locale (ma non tutti).

Questo almeno che non ci troviamo in classi particolari di funzioni:

10.1.1 Funzioni convesse

Definizione 31 Funzioni convesse: i grafici stanno sotto le corde:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) \quad \forall x, y \wedge \forall \lambda \in [0,1]$$

□

Nota 73: *se f è convessa allora $Hf(x) \geq 0 \; \forall x$*

10.1.2 Funzioni quadratiche

Definizione 32 Funzioni quadratiche: funzioni di secondo grado, ovvero più formalmente: funzione della forma:

$$f(x) = xQx + cx + d$$

dove $xQx =< x, Qx >$ con $Q \in \mathbb{R}^{n \times n}$, $c \in \mathbb{R}^n$ e $d \in \mathbb{R}$. $\square$

Nota 74: $\nabla f(x) = 2Qx$ *e* $Hf(x) = 2Q$

Esempio 23: data $3x_1^2 - x_1x_2$ ottengo: $Q = \begin{pmatrix} 3 & -\frac{1}{2} \\ -\frac{1}{2} & 0 \end{pmatrix}$ $\square$

Tra le funzioni quadratiche e le Q simmetriche c'è una perfetta corrispondenza biunivoca.

Esempio 24: dato $\begin{cases} max\ x_1^2 - 3x_1x_2 + 5x_2 \\ 3x_1 + 4x_2 \leq 7 \\ 2x_1 - x_2 = 8 \end{cases}$ si ha:

$$c = (0, -5) \text{ e } Q = \begin{pmatrix} -1 & \frac{3}{2} \\ \frac{3}{2} & 0 \end{pmatrix}$$

$\square$

10.1.3 Problema di programmazione quadratica

Modello:

$$\begin{cases} min\ xQx + cx + d \\ Ax \leq b \\ A_{eq}x = b_{eq} \\ LB \leq x \leq UB \end{cases}$$

: *d è una costante e quindi eliminabile*

Comando matlab: .

```
>> quadprog(Q, c, A, b, Aeq, beq, LB, UB)
```

Esempio 25:

$$f = -2x_1^3 - 3x_1^2x_2 + x_1^2 + x_1x_2 + x_2^2$$

$$\nabla f = (-6x_1^2 - 6x_1x_2 + 2x_1 + x_2,\ -3x_1^2 + x_1 + 2x_2)$$

Pongo quindi $\nabla f = 0$ tramite il sistema:

$$\begin{cases} -6x_1^2 - 6x_1x_2 + 2x_1 + x_2 = 0 \\ -3x_1^2 + x_1 + 2x_2 = 0 \end{cases}$$

che ha soluzione:

$$x = \begin{cases} (0,0) \\ (-\frac{1}{2}, \frac{5}{8}) \\ (\frac{1}{3}, 0) \end{cases}$$

Mi pongo quindi il quesito $Hf(x) \geq 0$?

$$Hf = \begin{pmatrix} -12x_1 - 6x_2 & -6x_1 + 1 \\ -6x_1 + 1 & 2 \end{pmatrix}$$

sostituisco quindi ogni punto soluzione nell'Hessiano e ne vado a studiare gli autovalori, ricavo quindi la seguente tabella:

	mL	**mG**	**ML**	**MG**	**sella**
(0,0)	si	?	no	no	no
(-1/2, 5/8)	no				si
(1/3, 0)					si

? è no. □

10.1.4 Restrizione a rette o semirette

Prop. f è convessa $\iff Hf \geq 0 \ \forall x$ □

10.2 Lezione 34 12/05

Consideriamo sempre $f \in \mathcal{C}^2$

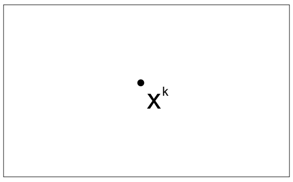
Definizione 33 Base: è la minimizzazione di una funzione su tutto $\mathbb{R}^n$:

$$\underset{x \in \mathbb{R}^n}{f(x)}$$

□

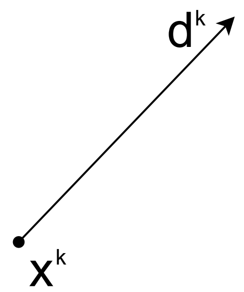
Nota 76: *gli esempi e gli esercizi che trattiamo sono con $n = 2$ variabili in quanto il salto si ha tra 1 e 2 e non tra 2 ed n variabili.*

Per tutti gli algoritmi risolutivi si parte da un punto:



Dove k indica il k-esimo passo dell'algoritmo.

Costruiamo la successione ricorsiva: $\{x^k\}$. per farlo si prende una direzione di discesa (ovvero dove f decrementa):

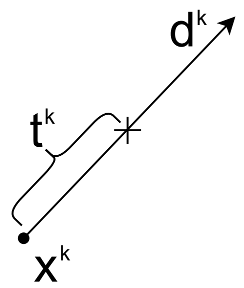


Definizione 34 Direzione di discesa:

$$\text{se } \exists \varepsilon > 0 \ : \ f(\overline{x} + t\overline{d}) < f(\overline{x}) \quad \forall t \in [0, \varepsilon]$$

□

In PNL questa direzione può però non essere sempre in discesa ma solo per un certo tratto:



Devo trovare fino a che punto la funzione decrementa per poi fermarmi. Lo si fa tramite la succesione ricorsiva:

$$x^{k+1} = x^k + t_k d^k$$

Concetto di parametrizzazione delle curve: in particolare ci interessiamo solo di rette, semirette e segmenti in quanto diventano funzioni di 1 variabile:

$$x^k + td^k$$

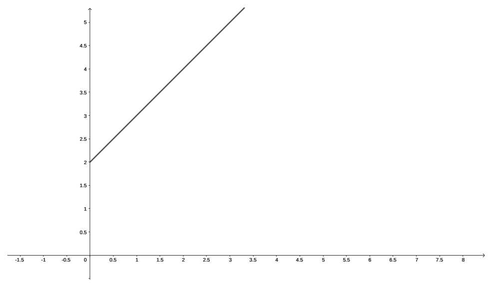
Esempio 26: per la bisettrice la forma parametrica è:

$$(0, 0) + t(1, 1)$$

□

Esempio 27: Data:

$$\begin{cases} x_1 = t \\ x_2 = 3t + 2 \end{cases}$$



$\square$

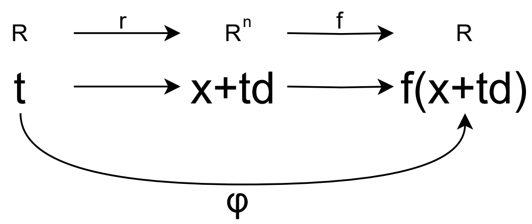
Per trovare quindi fino a che punto la funzione è in discesa costruisco la restrizione, ovvero il profilo della funzione lungo una retta $x^k + td^k$:

$$\varphi(t) = f(x^k + td^k)$$

Dove d^k è la direzione di discesa. Per trovarla bisogna verificare che $\varphi(t)$ sia decrescente (cosa che posso fare essendo n=1). La verifica si fa per $t = 0$, ovvero vicino a x^k:

$$\varphi'(0) < 0$$

Definizione 35 Funzione phi: $\varphi : \mathbb{R} \to \mathbb{R}$



dove r è la funzione restrizione $\square$

Definizione 36 Jacobiano della composta:

$$< \nabla f(x + td), d >$$

$\square$

Se considero $\varphi(t)$ ho che:

$$\varphi'(t) = \nabla f(x + td) \cdot d$$

$$\varphi'(0) = \nabla f(x) \cdot d$$

quindi $\nabla f(x) \cdot d < 0$ è condizione sufficiente per essere decrescente.

Teorema 21 Teorema 1 PNL: se vale $\nabla f(x) \cdot d < 0$ allora d è direzione di discesa. □

Nota 77: *per il massimo:* $\nabla f(x) \cdot d > 0$

Nota 78: *se è* $= 0$ *lascio perdere: non so cosa succede (può essere sia crescente che decrescente).*

Definizione 37 Metodo del gradiente:

$$d = -\nabla f(x)$$

in quanto direzione sicuramente negativa. L'equazione diventa quindi:

$$\nabla f(x) \cdot (-\nabla f(x)) = -||\nabla f(x)||^2 < 0$$

Se fa 0 allora siamo in un punto stazionario: mi fermo. □

Nota 79: *tutti gli algoritmi di PNL vanno alla ricerca di punti stazionari.*

Nota 80: *Cerco i punti dove* $\nabla f = 0$ *e se* f *è convessa allora siamo nel minimo assoluto.*

10.2.1 Di quanto mi sposto lungo la direzione?

Definizione 38 Punto di minimo della restrizione:

$$t_k \in \underset{t \geq 0}{argmin}\, \varphi(t)$$

□

Risolvo quindi questo problema:

$$min\, \varphi(t)$$

Esempio 28: Fare un passo data: $f(x) = x_1^2 + 3x_2^2 + 5x_1$ con $d = (1,1)$ e $x = (0,0)$:

$$\nabla f = (2x_1 + 5, 6x_2)$$

quindi $\nabla f(0) = (5,0) \to (5,0)(1,1) = 5$ quindi d non è direzione di discesa. Se adopero ora il metodo del gradiente ho che:

$$x^{k+1} = (0,0) + t_k(-5,0)$$

Devo ora calcolare $\varphi(t)$:

$$\varphi(t) = f((0,0) + t(-5,0)) = f(\underset{x_1}{-5t}, \underset{x_2}{0})$$

sostituisco in f: $25t^2 - 25t$ e quindi il minimo è $= t_k = \frac{1}{2}$. Sostituisco il t_k nel passo della successione x^{k+1} e ottengo:

$$x^{k+1} = (-\frac{5}{2}, 0)$$

□

10.3 Lezione 35 15/05

Sempre data $f \in \mathcal{C}^2$ con

$$\min_{x \in \mathbb{R}^n} f(x)$$

per determinare se esiste il minimo posso sfruttare il concetto di funzioni coercive:

Definizione 39 Funzioni coercive: f è coerciva se

$$\lim_{||x|| \to +\infty} f(x) = +\infty$$

ovvero se "sale" in tutte le direzioni. □

Prop. .

- se f è coerciva allora esiste il minimo.
- se f è "anticoerciva" (ovvero se il limite tende a $-\infty$) allora esiste il massimo.

□

Per calcolare quale sia il minimo bisogna risolvere il sistema $\nabla f(x) = 0$. Il sistema ha dimensioni $n \times n$ ed essendo formato da derivate prime lo so risolvere per le quadratiche.
$\Rightarrow$ sistema $xQx + cx \to \nabla f(x) = 2Qx + c = 0$
Considero ancora la successione di ricorrenza $\{x^k\}$ definita come $x^{k+1} = x^k + t_k d^k$

Teorema 22 "Convergenza" del metodo del gradiente: Data una funzione coerciva e $\{x^k\}$ costruita con il metodo del gradiente con ricerca esatta: ogni punto di accumulazione[1] è stazionario. □

Quindi se mi accorgo che la norma $||x^{k+1} - x^k|| \to 0$ (ma vale lo stesso anche con altre norme) allora ho trovato una sottosuccessione tendente a 0, quindi ho un punto stazionario $\Rightarrow \nabla f = 0$.

Teorema 23 Teorema 1 bis: Data una funzione coerciva e convessa, data $\{x^k\}$ costruita con il metodo del gradiente:

$$x^k \to x^* \;\; e \;\; x^* \text{ minimo assoluto}$$

□

[1] se una sottosuccesione converge in un punto esso è un punto di accumulazione.

Questi metodi (come tutti quelli in PNL) hanno però il problema che non terminano in un numero finito di passi.

10.3.1 Criteri di stop

Utilizzando dei criteri di stop si ha un numero finito di passi:

1. tempo: dopo tot tempo interrompo l'algoritmo. Per verificare che la soluzione ottima ottenuta sia "abbastanza ottima" guardo se $||\nabla f(x^{k+1})||$ è abbastanza piccolo (oppure con gli autovalori dell'Hessiano).
2. il numero di passi
3. si controlla $||x^{k+1} - x^k||$ per qualche punto di seguito: se abbastanza piccolo posso determinare la convergenza.
4. $|f(x^{k+1}) - f(x^k)|$ se abbastanza piccolo

Teorema 24 Teorema 1 ter: Data una funzione coerciva, convessa e quadratica: allora il metodo del gradiente con ricerca esatta converge al minimo assoluto in un numero finito di passi. □

Ricapitolando: Per risolvere un problema vincolato bisogna avere:

1. una condizione necessaria
2. un algoritmo iterativo
3. processo di feedback

10.3.2 Insiemi

Nota 81 : Importante**:** *Nel nostro caso il dominio D è* ***sempre*** *chiuso in quanto ci sono vincoli di $\leq$ e $=$ (che sono condizione sufficiente per un insieme chiuso)*

Definizione 40 Definizione di limitato: D è limitato se

$$\exists R > 0 \ : D \subset B(R)$$

□

Definizione 41 Regolare: ... □

10.4 Lezione 36 16/05

Nota 82: *consideriamo sempre che per noi i domini sono chiusi*

Problema di PNL scritto in forma compatta: dati:

- $f : \mathbb{R}^n \to \mathbb{R}$

- $g : \mathbb{R}^n \to \mathbb{R}^m$
- $h : \mathbb{R}^n \to \mathbb{R}^p$
- $f, g, h \in \mathcal{C}^1$

$$\begin{cases} min\ f(x) \\ g(x) \leq 0 \\ h(x) = 0 \end{cases}$$

con $D = \{x \in \mathbb{R}^n : g(x) \leq 0,\ h(x) = 0\}$
Teorema 25 LKKT (Lagrange-Karush–Kuhn–Tucker): sia D regolare e sia $\overline{x}$ minimo locale di (P), allora:

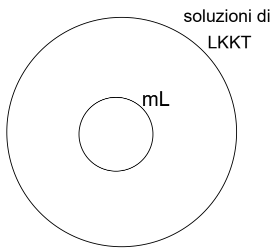
$$\exists\ (\overline{\lambda}_1, ..., \overline{\lambda}_n) = \overline{\lambda} \in \mathbb{R}^m \text{ con } \overline{\lambda} \geq 0,\ (\overline{\mu}_1, ..., \overline{\mu}_n) = \overline{\mu} \in \mathbb{R}^p$$

tali che:

$$\begin{cases} \nabla f(\overline{x}) + \sum_{i=1}^{m} \overline{\lambda}_i \nabla g_i(\overline{x}) + \sum_{j=1}^{p} \overline{\mu}_j \nabla h_j(\overline{x}) = 0 \\ \overline{\lambda} g(\overline{x}) = 0 \\ h(\overline{x}) = 0 \end{cases}$$

con $g_i(\overline{x}) \leq 0 \forall i$
Nota 83: *il sistema prende il nome di sistema LKKT.*

□



Essendo quindi i minimi locali contenuti tra le soluzioni di LKKT (vedi immagine) abbiamo costruito dunque una condizione necessaria.
Nota 84: *nel sistema ho:*

- $\overline{\lambda} g(\overline{x}) = 0 \Rightarrow \overline{\lambda}_1 g_1(\overline{x}) + ... + \overline{\lambda}_m g_m(\overline{x}) = 0 \Rightarrow \overline{\lambda}_i g_i(\overline{x}) = 0\ \ \forall i$
- $h(\overline{x}) = 0 \Rightarrow h_j(\overline{x}) = 0\ \ \forall j$

quindi il sistema è quadrato:

$$\begin{cases} m + n + p\ \ \textit{variabili} \\ m + n + p\ \ \textit{equazioni} \end{cases}$$

Teorema 26 LKKT (per il massimo): sia D regolare e sia $\overline{x}$ massimo locale di (P), allora:

$$\exists\ (\overline{\lambda}_1, ..., \overline{\lambda}_n) = \overline{\lambda} \in \mathbb{R}^m \text{ con } \overline{\lambda} \leq 0,\ (\overline{\mu}_1, ..., \overline{\mu}_n) = \overline{\mu} \in \mathbb{R}^p$$

tali che:

$$\begin{cases} \nabla f(\overline{x}) + \sum_{i=1}^{m} \overline{\lambda}_i \nabla g_i(\overline{x}) + \sum_{j=1}^{p} \overline{\mu}_j \nabla h_j(\overline{x}) = 0 \\ \overline{\lambda} g(\overline{x}) = 0 \\ h(\overline{x}) = 0 \end{cases}$$

con $g_i(\overline{x}) \leq 0\ \ \forall i$ □

10.4.1 Calcolo delle soluzioni di LKKT

Devo trovare una terna $(\overline{x}, \overline{\lambda}, \overline{\mu}) \in \mathbb{R}^{n+m+p}$:

- se $\overline{\lambda} \neq 0\ \ \forall \lambda_i$:
  - se tutti $\overline{\lambda} \geq 0$ posso avere mL o mG o sella
  - se tutti $\overline{\lambda} \leq 0$ posso avere ML o MG o sella
  - se $\overline{\lambda}$ discordi allora ho una sella
- se tutti $\overline{\lambda} = 0$ allora: mL o mG o ML o MG o sella

10.4.2 Classificazione soluzioni LKKT

Per Weierstrass almeno 2 soluzioni del sistema LKKT ci sono (massimo e minimo) se il dominio è limitato.

Esempio 29: .

1. $(\underbrace{1, 2}_{x}, \underbrace{3, 4}_{\lambda}, \underbrace{7}_{\mu})$
2. $(-1, 3, -4, 5, 2)$
3. $(-1, -6, -3, -2, 1)$

Le soluzioni sono classificabili come: 1 possibile minimo ($\lambda > 0$), 2 sella (λ discordi), 3 possibile massimo ($\lambda < 0$).

Se si aggiunge una quarta soluzione: $(3, 1, 1, 1, 1)$ questa diventa il minimo: andando a sostituire la x nella f ha un valore minore della soluzione 1 che quindi non è più il minimo. □

Nota 85: *Quando devo risolvere un esercizio devo verificare che D sia:*

- *regolare: per poter risolvere LKKT*
- *convesso (prossimo argomento)*
- *chiuso: sempre*
- *limitato: bisogna controllare il singolo caso*

10.5 Lezione 37 17/05

Definizione 42 Vincoli attivi in x segnato:

$$A(\overline{x}) = \{i : g_i(\overline{x}) = 0\} \cup \{j : h_j(\overline{x}) = 0\}$$

□

Come vedere se D è **regolare**: (3 modi)

1. vincoli g_i e h_j lineari (D poliedri)
2. Condizione di Slater: g_i convesse $\forall i$ oppure h_j lineari $\forall j$ e $\exists\, \hat{x} : g_i(\hat{x}) < 0 \;\; \forall i$ (ovvero se esiste un punto interno)
3. Condizione di Mangasarian:

$$\begin{cases} (\nabla g_i(\overline{x}),\ \nabla h_j(\overline{x})) \text{ linearmente indipendenti} \\ i, j \in A(\overline{x}) \;\; \forall \overline{x} \end{cases}$$

Nota 86: *Per verificare l'indipendenza lineare guardo il rango (se matrice quadrata guardo il determinante)*

Definizione 43 Problema convesso: Un problema si dice convesso se:

$$\begin{cases} f \text{ è convessa} \\ D \text{ è convesso} \end{cases}$$

□

Definizione 44 Problema concavo: Un problema si dice concavo se:

$$\begin{cases} f \text{ è concava} \\ D \text{ è convesso} \end{cases}$$

Nota 87: *D concavo non esiste.*

□

Teorema 27 Condizione sufficiente di primo ordine (per il minimo): sia:

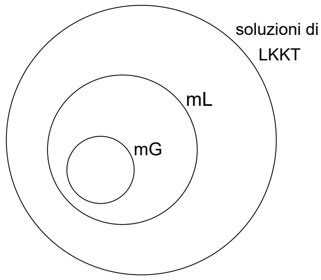
- (P) un problema **convesso** e regolare
- $(\overline{x}, \overline{\lambda}, \overline{\mu})$ una soluzione di LKKT

con

$$\overline{\lambda} \geq 0$$

allora $\overline{x}$ è **minimo globale**. □

Nota 88: *Questo teorema è quindi condizione sufficiente, mentre LKKT era condizione necessaria: ho quindi una condizione necessaria e sufficiente.*



Teorema 28 Condizione sufficiente di primo ordine (per il massimo): sia:

- (P) un problema **concavo** e regolare
- $(\overline{x}, \overline{\lambda}, \overline{\mu})$ una soluzione di LKKT

con

$$\overline{\lambda} \leq 0$$

allora $\overline{x}$ è **massimo globale**. □

Nota 89: *Per i teoremi non ci sono quindi, rispettivamente, altri minimi locali e altri massimi locali diversi da quelli globali.*

10.5.1 Dominio convesso

Definizione 45 D convesso: se le g_i sono convesse e le h_j sono lineari allora D è convesso. □

10.5.2 Metodo della ricerca locale

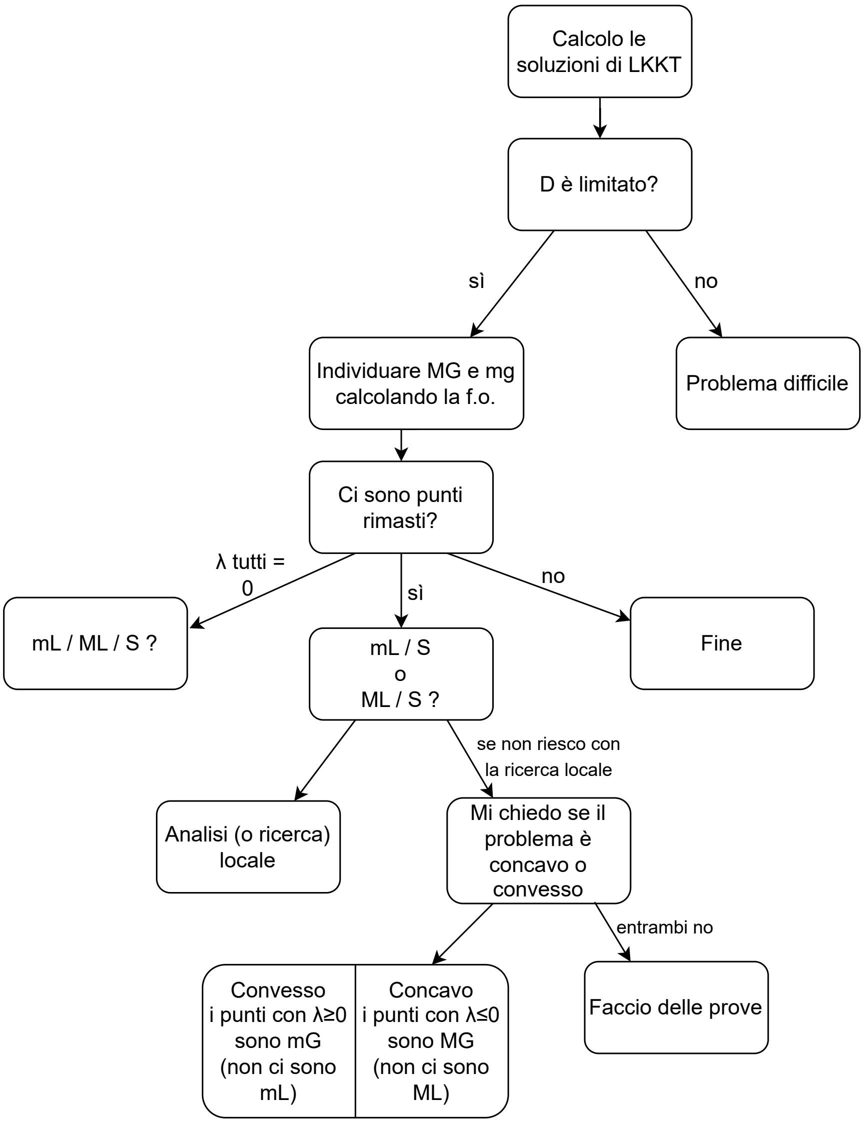
Se ci sono dei punti rimasti da analizzare, che non hanno tutti i moltiplicatori (λ) uguali a 0, possono essere, a seconda del segno, minimi locali o selle oppure massimi locali o selle. Per discriminare la scelta bisogna parametrizzare la funzione lungo i bordi individuando correttamente il dominio da studiare (con variazione ε nelle direzioni possibili) e studiarne il comportamento:

- se tutte le parametrizzazioni sono concordi il punto è quindi un mL o ML, altrimenti è un punto di sella.

10.5.3 Note sugli esercizi

Nota 90: *Per verificare che, in un D non limitato, esista un massimo (o un minimo) globale posso guardare una restrizione e farne il limite (essendo in n=1): se il limite fa ∞ posso escludere l'esistenza del massimo globale (o del minimo a seconda dei casi).*

10.5.4 Schema riassuntivo



Capitolo 11

Algoritmi per la pnl

11.1 Lezione 38 19/05 Algoritmi per la PNL

Nota 91: *gli algoritmi che trattiamo sono su poliedri chiusi e limitati*

11.1.1 Nozioni preliminari

Dato un problema:

$$(P)\begin{cases} max/min\ f(x) \\ Ax \leq b \end{cases}$$

con $f : \mathbb{R}^n \to \mathbb{R}$.

Teorema 29: Dato il problema (P) (non vuoto) con f **concava**, il **minimo globale** di (P) viene raggiunto in un vertice. □

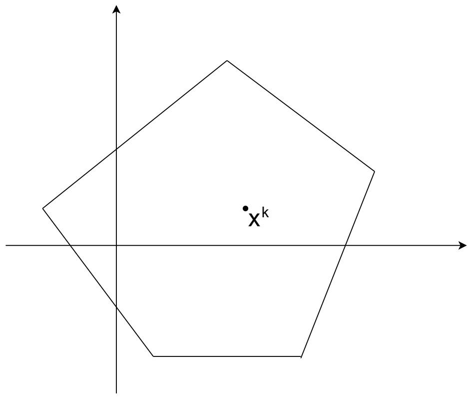
Teorema 30: Dato il problema (P) (non vuoto) con f **convessa**, il **massimo globale** di (P) viene raggiunto in un vertice. □

Nota 92: *questi due teoremi vanno di pari passo con i due sulle "condizioni sufficienti di primo ordine" e aiutano a risolvere i problemi:*

- *se f concava dalle "condizioni sufficienti di primo ordine" ho informazioni sul massimo, da Teorema 29 ho informazioni sul minimo.*
- *se f convessa dalle "condizioni sufficienti di primo ordine" ho informazioni sul minimo, da Teorema 30 ho informazioni sul massimo.*

11.1.2 Algoritmo 1

Algoritmo IV.1 Algoritmo di Frank-Wolfe: 1. Prendiamo un poliedro e un punto iniziale ammissibile $x^k = x^0 \in P$



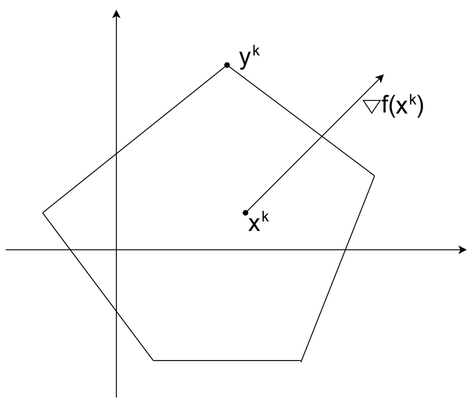
Nota 93: *Siamo nel caso in cui non so risolvere LKKT e quindi costruisco la successione di ricorrenza* $x^{k+1} = x^k + t_k d^k$

2. prendiamo il gradiente e risolviamo questo problema detto "**problema linearizzato**"):

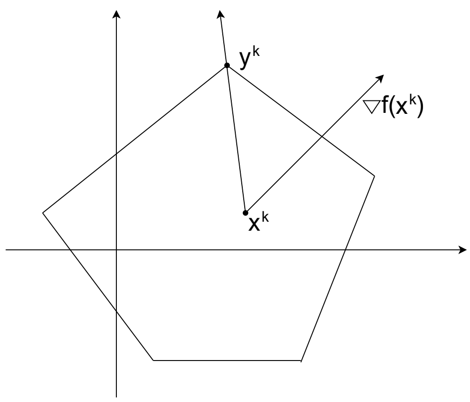
$$PL(x^k) \begin{cases} max\ \nabla f(x^k)x \\ Ax \leq b \end{cases}$$

Nota 94: *posso definire* $\nabla f(x^k)$ *come funzione obbiettivo in quanto è un vettore (non una funzione).*

L'ottimo di questo problema (che è un problema di PL) è y^k, che è un vertice.



3. Prendo quindi la direzione $d^k = y^k - x^k$



Nota 95: *faccio questa scelta, invece di quella di prendere* d^k *come* $\nabla f(x^k)$, *perché così posso sapere dove esco dal poliedro (oltre il vertice esco dal poliedro), cosa che non potrei sapere usando il gradiente come direzione: in questo modo so come scegliere* t_k.

Nota 96: *posso fare questa scelta perché tra* d^k *e* ∇ *c'è un angolo* $< \frac{\pi}{2}$ *e quindi so che comunque quella è una direzione crescente.*

t_k lo scelgo quindi come:

$$t_k \in \underset{t\in[0,1]}{argmax}\ \varphi(t)$$

dove $\varphi(t) = f(x^k + t(y^k - x^k))$

4. Pongo quindi x^{k+1} come:

$$x^{k+1} = x^k + t_k d^k$$

Criterio di stop dell'algoritmo: quando: $y^k = x^k$. ovvero quando non ho più una direzione crescente (di discesa nel caso di min).

Nota 97: *per il minimo il problema linearizzato è:*

$$PL(x^k) \begin{cases} min \nabla f(x^k)x \\ x \in P \end{cases}$$

e:

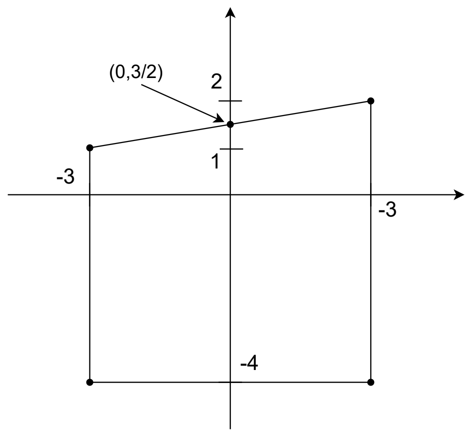
$$t_k \in argmin\ \varphi(t)$$

□

Esempio 30: Data: $f = 2x_1^2 - 2x_1x_2 - 6x_1 + 5x_2$ e i vertici:

$$(-3,-4)\ \ (-3,1)\ \ (3,-4)\ \ (3,2)$$

Trovare il massimo dato: $x^k = (0, \frac{3}{2})$



Svolgere 1 passo di F-W.
Bisogna trovare $\nabla f(0, \frac{3}{2})$ quindi:

$$\nabla f = (4x_1 - 2x_2 - 6\ ,\ -2x_1 + 5)$$

e dunque: $\nabla f(0, \frac{3}{2}) = (-9, 5)$.
Problema linearizzato:

$$\begin{cases} max\ -9x_1 + 5x_2 \\ x \in P \end{cases}$$

Osservazione: per trovare y^k posso:

- usare linprog (ma dovrei prima trovare tutte le rette che formano il poliedro)
- usare il disegno
- provare i vertici nella funzione obbiettivo

Ho quindi $y^k = (-3, 1)$.
Calcolo ora la restrizione:

$$\varphi(t) = f((0, \frac{3}{2}) + t((-3,1) - (0, \frac{3}{2}))) =$$

$$= f((0, \frac{3}{2}) + t(-3, -\frac{1}{2})) =$$

$$= f(0 - 3t\ ,\ \frac{3}{2} - \frac{1}{2}t)$$

sostituisco quindi x_1 e x_2 nell'equazione iniziale:

$$\varphi(t) = s(-3t)^2 - 2(-3t)(\frac{3}{2} - \frac{1}{2}t) - 6(-3t) + 5(\frac{3}{2} - \frac{1}{2}t) =$$

$$= 15t^2 + \frac{49}{2} + \alpha$$

dove α è il termine noto (non ci interessa ai nostri fini).
Per trovare argmax disegno la parabola e ottengo $t_k = 1$.
x^{k+1} è quindi $(-3, 1)$ □

11.1.3 Teoremi di Frank-Wolfe

Teorema 31 Teorema di Frank-Wolfe 1: la successione $\{x^k\}$ converge ad un punto stazionario, cioè ad una soluzione del sistema LKKT. □

Nota 98: *essendo un metodo di salita non è sicuramente un minimo, può essere:*

- *una sella*
- *un massimo locale (no se è concava)*
- *un massimo globale*

Teorema 32 Teorema di Frank-Wolfe 2: se f è convessa allora:

$$x^k \to x^*$$

con x^* minimo globale. □

Teorema 33 Teorema di Frank-Wolfe 3: se f è concava allora:

$$x^k \to x^*$$

con x^* massimo globale. □

Teorema 34 Teorema di Frank-Wolfe 4: se f è convessa e quadratica allora:

$$x^k \to x^*$$

con x^* minimo globale e termina in un numero finito di passi. □

Teorema 35 Teorema di Frank-Wolfe 5: se f è concava e quadratica allora:

$$x^k \to x^*$$

con x^* massimo globale e termina in un numero finito di passi. □

11.1.4 Note varie

Nota 99: *Il punto iniziale dell'algorimo si trova con il duale ausiliario.*

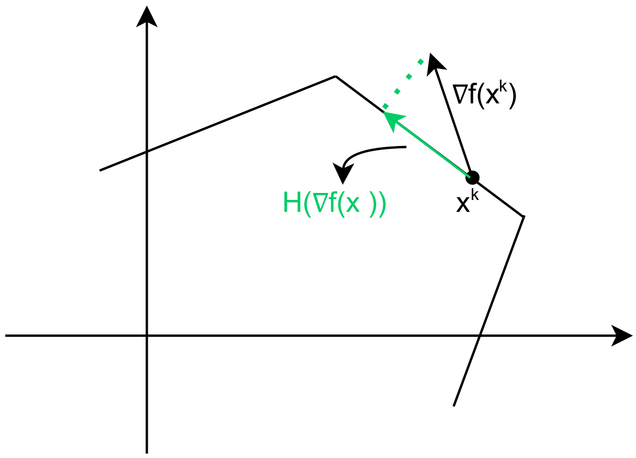
Nota 100: *sul bordo, il gradiente (per esempio) del minimo può non fare 0. (cosa che non accade all'interno)1*

11.2 Lezione 39 23/05 Metodo del gradiente proiettato

Il metodo del gradiente proiettato è il secondo algoritmo della PNL, come alternativa di quello di Frank-Wolfe.

Nota 101: *Anche questo metodo costruisce una successione del tipo* $x^{k+1} = x^k + t_k d^k$

Concetto di base: Quando il punto è sul bordo ma il gradiente punta fuori dal poliedro, se si seguisse questa direzione si uscirebbe: si sfrutta quindi la proiezione del gradiente (proiezione in forma chiusa).



La **proiezione** si fa **sui vincoli attivi in** $\overline{x}$ che sono gli:

$$i : A_i\overline{x} = b_i$$

Costruisco quindi la matrice M estratta da A e formata dai vincoli attivi (M è formata dalle righe i di A tali che i è un vincolo attivo)

Definizione 46 Matrice di proiezione: Proiezione ortogonale di un vettore "a" sulla varietà $Mx = 0$

$$H \triangleq I - M^T(MM^T)^{-1}M$$

con H matrice di proiezione □

Nota 102: *MM^T è invertibile sempre.*

Teorema 36: La proiezione ortogonale di "a" su M è data da Ha. □

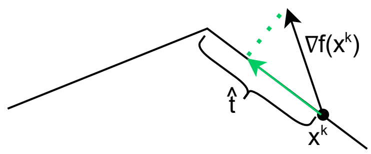
Nota 103: *Se x^k è interno $M = 0$ (non ci sono vincoli attivi e quindi $H = I$).*

Si distinguono ora due casi per determinare il punto dove la direzione proiettata incontra il bordo:

- con x^k **interno**: risolvo questo problema:

$$\begin{cases} max\ t \\ A(x^k + t\nabla f(x^k)) \leq b \\ t \geq 0 \text{ (sempre valido)} \end{cases}$$

che ha come soluzione il massimo spostamento $\hat{t}$



Nota 104: *è un problema di PL in 1 variabile*

Nota 105: *il poliedro è:* $(A\nabla f(x^k))t \leq b - Ax^k$ *dato da m disequazioni*

- con x^k **sul bordo**:

$$\begin{cases} max\ t \\ A(x^k + t_k d^k) \leq b \end{cases}$$

dove d^k indica la direzione che devo prendere, ovvero quella della proiezione che è:

$$d^k = H\nabla f(x^k)$$

Nota 106: *con* x^k *interno era* $H = I$

Una volta ottenuto $\hat{t}$ si trova il passo t_k come:

$$t_k = \underset{t\in[0,\hat{t}]}{argmax}\ \varphi(t)$$

con $\varphi(t) = f(x^k + td^k)$

Infine si aggiorna $x^{k+1} = x^k + t_k d^k$

Algoritmo IV.2 Metodo del gradiente proiettato: .

1. trovo la direzione
2. trovo di quanto mi sposto

□

Nota 107: *la direzione che abbiamo scelto è di salita perché l'angolo tra la proiezione e il gradiente è sicuramente* $< \frac{\pi}{2}$ *infatti scatta la condizione:*

$$se\ \ d^k \cdot \nabla f(x^k) > 0\ \ allora\ d_k\ \text{ è direzione di salita}$$

11.2.1 Caso d=0

Questo caso si può verificare in 2 occasioni:

- $\nabla = 0$
- proiezione = 0 (ovvero gradiente perpendicolare)

Se $d^k = 0$ allora bisogna calcolare questo vettore:

$$\lambda = -(MM^T)^{-1}M\nabla f(x^k)$$

e verificare:

- se $\lambda \leq 0$ allora stop, l'algoritmo termina.
- altrimenti sia i il primo indice di $\lambda > 0$, elimina questo indice e ripeti.

Nota 108: *λ è soluzione di LKKT*

11.2.2 Caso di minimo

Nel caso il problema sia di minimo bisogna apportare le seguenti modifiche all'algoritmo:

- $d^k = H(-\nabla f(x^k))$
- $t_K = \underset{t\in[0,\hat{t}_k]}{argmin}\ \varphi(t)$
- $\lambda \geq 0$ e "primo indice di $\lambda < 0$"