

### Sondaggio Mail Segreteria

Partecipa al sondaggio pubblicato sul Sito degli Studenti:

"Diversi studenti propongono che le mail inviate dalla segreteria contengano direttamente il testo dell'avviso e non un link al sito dti.unimi.it . Qual è la tua opinione?"  
(per votare dovete essere registrati)

[Torna alla pagina di Ricerca Operativa](#)

## :: Ricerca Operativa - Algoritmo del semplice ::

Alcune immagini di questa pagina sono prese dalle slide del prof [Giovanni Righini](#)

### Forma canonica di un problema di PL

Nella scorsa lezione abbiamo visto cosa sono i problemi di programmazione lineare e abbiamo anticipato che per darli in pasto all'algoritmo del semplice dobbiamo scriverli in **forma canonica**. Vediamo come ottenerla a partire dalla forma standard.

#### 1 Forma standard.

Notare che non importa se stiamo minimizzando o massimizzando, la funzione obiettivo è indicata con la notazione generica  $opt$

$$\begin{aligned} opt \quad z &= c^T x \\ s.t. \quad &\begin{cases} Ax = b \\ x \geq 0 \end{cases} \end{aligned}$$

#### 2 Scegliamo una base e dividiamo il problema in "parte in base" (indicata in verde) e "parte fuori base" (indicata in rosso). In altre parole dobbiamo dividere la matrice $A$ in due matrici $B$ ed $N$ , e il vettore $x$ in $x_B$ e $x_N$ .

$$\begin{aligned} opt \quad z &= c_B^T x_B + c_N^T x_N \\ s.t. \quad &\begin{cases} Bx_B + Nx_N = b \\ x_B \geq 0, x_N \geq 0 \end{cases} \quad \left( x_B = B^{-1}b - B^{-1}Nx_N \right) \end{aligned}$$

#### 3 Moltiplichiamo a sinistra per $B^{-1}$ .

$$\begin{aligned} opt \quad z &= c_B^T x_B + c_N^T x_N \\ s.t. \quad &\begin{cases} Ix_B + \bar{N}x_N = \bar{b} \\ x_B \geq 0, x_N \geq 0 \end{cases} \quad \begin{aligned} \bar{N} &= B^{-1}N \\ \bar{b} &= B^{-1}b \\ \left( x_B &= \bar{b} - \bar{N}x_N \right) \end{aligned} \end{aligned}$$

#### 4 Sostituiamo tutto in $z$ ottenendo finalmente la forma canonica.

$$\begin{aligned} opt \quad z &= \bar{z}_B + \bar{c}_N^T x_N \\ s.t. \quad &\begin{cases} Ix_B + \bar{N}x_N = \bar{b} \\ x_B \geq 0, x_N \geq 0 \end{cases} \quad \begin{aligned} \bar{z}_B &= c_B^T \bar{b} \\ \bar{c}_N^T &= c_N^T - c_B^T \bar{N} \end{aligned} \end{aligned}$$

A cosa serve tutto questo? A far vedere che una volta azzerate le variabili fuori base  $x_N$  ottengo che:  
 **$z = z_B$  e  $x_B = b$  (segnato).**

Avremo intuito che esistono tante forme canoniche quanti sono i modi in cui posso dividere le variabili di base da quelle fuori base, o analogamente, tante quanto sono le sue basi.

Formalmente, un problema è in forma canonica se e solo se:

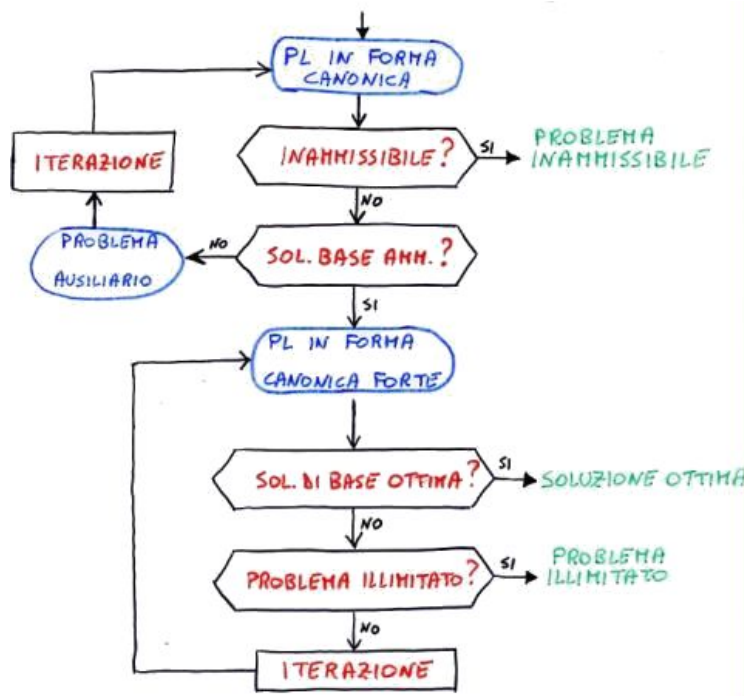
1. i coefficienti delle  $x_B$  nel sistema dei vincoli formano una matrice identità  $I$
2. le  $x_B$  non compaiono nella funzione obiettivo  $z$

Inoltre si dice che la forma canonica è **forte** se e solo se i termini noti ( $b$  segnato) sono non negativi, altrimenti è detta **debole**. Quindi se ho una forma canonica forte avrò una serie di soluzioni di base ammissibili, mentre se ho una forma canonica debole saranno non ammissibili.

### Algoritmo del semplice

L' **algoritmo del semplice** garantisce l'ottimalità attraverso un processo iterativo di miglioramento che si sposta da un vertice all'altro. Ciò vuol dire che ad ogni iterazione la soluzione o viene lasciata invariata o viene migliorata, e dato che il poliedro è convesso non si può che terminare nella soluzione ottima. Non è

polinomiale (nel caso peggiore ha complessità esponenziale), ma nella pratica - anche grazie a costanti raffinement - si dimostra più efficiente di questa classe di algoritmi. Ecco qui:



Affronteremo i vari passi via via nel corso del capitolo.

## Tableau

La struttura dati che rappresenta la forma canonica su cui lavora l'algoritmo del simplesso si chiama **tableau** ed è una matrice.

Facciamo un esempio di derivazione di un tableau da un problema già espresso in forma canonica. In particolare le variabili di slack in base ( $B = \{x_3, x_4, x_5\}$ ) saranno scritte in verde, mentre quelle fuori base ( $N = \{x_1, x_2\}$ ) in rosso.

$$\begin{aligned} \max z &= x_1 + 2x_2 \\ \text{s.t.} \quad & \begin{cases} -x_1 + 2x_2 + x_3 = 8 \\ x_1 + x_2 + x_4 = 10 \\ x_1 + x_5 = 7 \end{cases} \\ & x_1, x_2 \geq 0, \quad x_3, x_4, x_5 \geq 0 \end{aligned}$$
  

$$\begin{array}{c|cccccc} & \text{red } x_1 & \text{red } x_2 & \text{green } x_3 & \text{green } x_4 & \text{green } x_5 & \text{b} \\ \hline -z_B & 0 & 1 & 2 & 0 & 0 & 0 \\ 8 & -1 & 2 & 1 & 0 & 0 & 0 \\ 10 & 1 & 1 & 0 & 1 & 0 & 0 \\ 7 & 1 & 0 & 0 & 0 & 1 & 0 \\ \hline \bar{N} & \uparrow & \uparrow & & & & \\ \bar{I} & & & & & & \end{array}$$

Facciamo qualche considerazione sul tableau:

- nella posizione (0,0) si mette il termine noto della funzione obiettivo. Per convenzione il problema si esprime sempre in forma di minimo, quindi se sto massimizzando devo invertire i segni (come andrà fatto in questo caso)
- la riga 0 è quella dei **coefficienti di costo ridotto**, ovvero i coefficienti delle  $x$  che compaiono nella funzione obiettivo. Il perché del loro nome lo capiremo poi
- sulla colonna 0 si mettono i vari termini noti  $b$
- in tutte le altre righe si mettono i coefficienti delle  $x$  che compaiono nei vincoli

## Test di ottimalità

Guardando l'algoritmo del simplesso osserviamo che una volta posto il problema in forma canonica forte ci chiediamo se la soluzione è ottima o no. Come posso rispondere a questa domanda? Con il **test di ottimalità**.

Ricordiamo anzitutto che:

$$\text{opt } z = z_B + \bar{C}_N^T x_N$$

, dove i coefficienti di costo ridotto  $C_N$  indicano di quanto cambierebbe la funzione obiettivo  $z$  se il valore delle variabili fuori base  $x_N$  passasse da 0 a un numero positivo.

A questo punto possiamo distinguere due *test di ottimalità* a seconda della direzione di ottimizzazione:

- se stiamo massimizzando, allora siamo all'ottimo quando ho i coefficienti di costo ridotto minori o uguali a 0
- se stiamo minimizzando, allora siamo all'ottimo quando ho i coefficienti di costo ridotto maggiori o

uguali a 0

In entrambi i casi non sarà possibile migliorare ulteriormente la soluzione  $z$  corrente, il che significa che è quella ottima.

Ci ricordiamo il discorso sulla *degenerazione*? Avevamo detto che era una situazione critica ma non ne avevamo spiegato bene i motivi. Ora però sappiamo che dato che a una soluzione degenerare possono corrispondere tante basi diverse, le condizioni di ottimalità non valgono più, e quindi ci potremmo trovare nella situazione in cui siamo già all'ottimo ma non riusciamo ad accorgercene. In questi casi ci vorranno diverse iterazioni in più del necessario per rendersi conto che magari siamo alla soluzione ottima già da un po', ed è tutto tempo computazionale buttato alle ortiche.

## Pivoting

Passiamo alla parte divertente.

Come si fa a cambiare base? Come si passa da una forma canonica all'altra? A queste domande - peraltro equivalenti - si risponde con la tecnica del **pivoting**.

Facciamo un esempio commentato.

### 1 Forma iniziale.

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

### 2 Scegliere da una delle colonne fuori base un elemento positivo, che chiameremo **pivot**. Nel nostro caso quello sulla riga 1 e colonna 2.

Il perché deve essere positivo lo vedremo poi, per ora ci basti sapere che è un requisito per mantenere l'ammissibilità.

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

### 3 Dividere la riga ( $r$ ) che contiene il pivot per il valore del pivot stesso.

0	-1	-2	0	0	0
4	$-\frac{1}{2}$	1	$\frac{1}{2}$	0	0
10	1	1	0	1	0
7	1	0	0	0	1

### 4 Tenetevi forte: sottrarre ad ogni riga $i$ diversa da $r$ (quella del pivot) il valore della riga del pivot moltiplicata per l'elemento di $i$ sulla stessa colonna del pivot.

Facciamo prima con un esempio:

riga 0: si sottrae  $(-2) \cdot$  (riga del pivot)

riga 2: si sottrae  $(1) \cdot$  (riga del pivot)

riga 3: si sottrae  $(0) \cdot$  (riga del pivot)

8	-2	0	1	0	0
4	$-\frac{1}{2}$	1	$\frac{1}{2}$	0	0
6	$\frac{3}{2}$	0	$-\frac{1}{2}$	1	0
7	1	0	0	0	1

Come risultato avremo ancora una forma canonica, ma abbiamo fatto entrare in base la colonna del pivot e fatto uscire quella che aveva il valore 1 sulla sua riga (in corrispondenza delle variabili in base). Ovviamente non è questo l'unico risultato, ma possiamo osservare che la funzione obiettivo è migliorata:  $z_{\text{ott}} = -z_B = -8$ .

Ok, funziona, ma che significato hanno questa serie di operazioni? Rivediamo lo stesso esempio in un altro modo, così da poterne osservare meglio le corrispondenze.

->

$$\begin{aligned} \min z &= -x_1 - 2x_2 \\ \text{s.t.} \quad &\begin{cases} -x_1 + 2x_2 + x_3 = 8 \\ x_1 + x_2 + x_4 = 10 \\ x_1 + x_5 = 7 \\ x_i \geq 0 \end{cases} \end{aligned}$$

**Problema in forma canonica**

pivot  
->

$$\begin{cases} x_3 = 8 + x_1 - 2x_2 \\ x_4 = 10 - x_1 - x_2 \\ x_5 = 7 - x_1 \end{cases}$$

**Lo stesso problema  
riscritto esprimendo le  
variabili in base in  
funzione di quelle non in  
base**

Questo problema ha lo stesso tableau che abbiamo usato sopra come esempio. Se facciamo pivot sull'elemento in riga 1 e colonna 2, faremo entrare in base  $x_2$  e uscire  $x_3$ . Dovremo quindi aggiornare le equazioni del sistema, esprimendo stavolta  $x_2$ ,  $x_4$  e  $x_5$  in funzione di  $x_1$  e  $x_3$ . Ciò che otteniamo è riportato nella figura sotto.

$$\begin{cases} x_3 = 8 + x_1 - 2x_2 \\ x_4 = 10 - x_1 - x_2 \\ x_5 = 7 - x_1 \end{cases}$$

**Sistema prima del cambio  
basi**

->  
pivot  
->

$$\begin{cases} x_2 = 4 + \frac{1}{2}x_1 - \frac{1}{2}x_3 \\ x_4 = 6 - \frac{3}{2}x_1 + \frac{1}{2}x_3 \\ x_5 = 7 - x_1 \end{cases}$$

$$\min z = -8 - 2x_1 + x_3$$

**Sistema dopo il cambio basi**

Se proviamo a fare il tableau di questo nuovo sistema otteniamo esattamente quello che prima avevamo ricavato con la tecnica del pivoting.

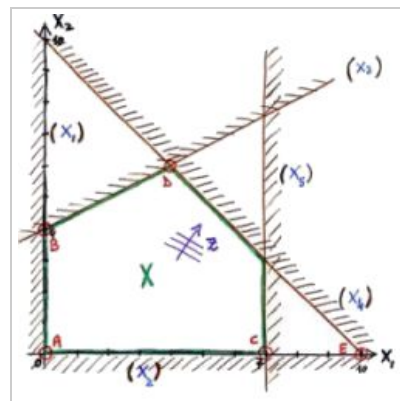
Per capire cosa succede da un punto di vista geometrico facciamo riferimento al grafico riportato a destra (che corrisponde al problema in esame).

Un iterazione di pivoting consiste nello spostarci da un vertice del poliedro (ad esempio A), che sappiamo corrispondere a una soluzione di base ammissibile, ad un altro vertice (ad esempio B).

Tutto qui? Tutto qui.

Si noti che quando diciamo che le variabili di base in A sono  $x_3$ ,  $x_4$  e  $x_5$ , il vertice corrispondente è quello dato dall'intersezione dei vincoli attivi, e quindi  $x_1$  e  $x_2$ .

Stesso ragionamento per B: le variabili di base sono  $x_2$ ,  $x_4$  e  $x_5$ , quindi il vertice corrispondente è dato dall'intersezione di  $x_1$  e  $x_3$ .



Ricapitolando, a quanto pare a seconda della scelta del pivot otteniamo una soluzione diversa, migliore o peggiore. Ma come facciamo a scegliere quello giusto? I prossimi due capitoli ci daranno una risposta.

### Scelta della riga del pivot

Attraverso la scelta della riga del pivot determiniamo la variabile di base che deve uscire, quindi dobbiamo fare in modo di garantire l'ammissibilità della soluzione. Andando per tentativi provando su tutte le righe posso verificare l'ammissibilità controllando se le coordinate (*termine noto riga pivot; coefficiente funzione obiettivo colonna pivot*) ottenute alla fine del pivoting individuano effettivamente un vertice del poliedro o se si trovano al di fuori della regione ammissibile.

Per fortuna esiste una regolina che ci evita l'imbarazzo di andare a tentoni, e consiste nello scegliere la riga - tra tutte quelle in cui il pivot è strettamente positivo - che minimizza il rapporto tra termine noto e pivot. Seguire questo metodo ci permetterà di spostarci sulla prima soluzione di base che incontriamo nella direzione scelta, il che implica la garanzia di ammissibilità.

### Scelta della colonna del pivot

Per quanto riguarda la scelta della colonna del pivot dobbiamo riprendere in considerazione il *test di ottimalità*, che ricordiamo ci permette di conoscere (grazie ai coefficienti di costo ridotto) di quanto varierebbe la funzione obiettivo se le variabili fuori base non valessero 0 ma un qualsiasi numero positivo. Grazie a questo test possiamo scoprire se esistono delle direzioni ammissibili di miglioramento della soluzione oppure no. In particolare, nella scelta della colonna del pivot potremo scegliere quella che ha il coefficiente di costo ridotto che fa fallire il test di ottimalità:

- valore positivo se il problema è di massimizzazione

- valore negativo se è di minimizzazione (il caso più frequente, dato che per convenzione si trasformano tutti i problemi in ottimizzazioni di questo tipo)

Nel nostro esempio (vedi figura accanto) entrambi i coefficienti di costo ridotto sono sbagliati, perché stiamo minimizzando e loro sono entrambi negativi.

Facendo riferimento all'aspetto grafico del problema, se faccio entrare in base la variabile  $x_1$  avrò uno spostamento della soluzione verso destra (verso c); mentre se faccio entrare in base  $x_2$ , la soluzione si sposta verso l'alto (verso A). In entrambi i casi la situazione migliora.

0	-1	-2	0	0	0
8	-1	2	1	0	0
10	1	1	0	1	0
7	1	0	0	0	1

Ma che fare se stiamo minimizzando ed abbiamo più di una colonna con costo ridotto negativo? Abbiamo diverse strategie:

- scegliere quella con minimo costo ridotto
- scegliere la prima da sinistra
- scegliere a caso
- utilizzare metodi di scelta come quello di Bland
- scegliere la colonna che produce il massimo miglioramento della soluzione

Ora sappiamo tutto quello che ci serve per individuare riga e colonna del pivot. Abbiamo inoltre capito che la scelta della riga garantisce l'ammissibilità della soluzione, mentre quella della colonna l'ottimalità.

## Test di illimitatezza

Concludiamo i nostri approfondimenti sulle varie fasi dell'algoritmo del simplesso andando a vedere il **test di illimitatezza**, ovvero il test grazie al quale l'algoritmo capisce che non esiste una soluzione ottima perché il problema è illimitato.

Il test di illimitatezza è molto semplice: se le colonne del tableau che hanno costo ridotto negativo (quelle che fanno fallire il test di ottimalità) non hanno nemmeno un elemento positivo, allora il problema è illimitato.

Quando l'algoritmo riconosce questa situazione si blocca ed esce dal processo risolutivo.

---

[Torna alla pagina di Ricerca Operativa](#)

Ultimo aggiornamento: March 29, 2009, at 08:12 PM