

[Torna alla pagina di Ricerca Operativa](#)

:: Ricerca Operativa - Programmazione lineare ::

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

Programmazione matematica

Breve introduzione. La *programmazione matematica* è uno dei metodi più importanti della Ricerca Operativa, l'altro è la simulazione. Si basano entrambi su modelli matematici, ma mentre quest'ultima procede per tentativi, la programmazione matematica si propone di calcolare direttamente la soluzione migliore per ottenere ciò che si è prefissata. I suoi elementi essenziali sono:

- *dati*, tutte le informazioni sul problema di cui disponiamo. In essi si concentra gran parte del lavoro, poiché "la loro raccolta e preparazione è un bagno di sangue" (sic)
- *variabili*, grandezze senza un valore già definito ma che dovremo calcolare noi. Rappresentano la nostra libertà decisionale, ovvero le alternative che abbiamo di fronte
- *vincoli*, equazioni e/o disequazioni che rappresentano ciò che non possiamo fare
- *funzione obiettivo*, il criterio in base al quale stabiliamo che una soluzione è migliore di un'altra. Non è detto che ce ne sia solo una

Per l'algoritmo risolutivo del problema la soluzione corrisponde a un assegnamento di valori alle variabili. Notare che questi valori possono essere ammissibili o non ammissibili a seconda che rispettino o meno i vincoli dati.

Nei problemi di esistenza il nostro unico scopo è trovare soluzioni ammissibili, mentre nei problemi di ottimizzazione vogliamo trovare una **soluzione ottima**, cioè quella che massimizza o minimizza la funzione obiettivo.

Bene, ora abbiamo tutto ciò che ci serve per cominciare a parlare della *programmazione lineare*.

Programmazione lineare (PL)

La **programmazione lineare (PL)** si occupa di problemi decisionali, e si ha quando sia la funzione obiettivo che i vincoli sono funzioni lineari delle variabili. Esistono quattro forme per rappresentare tali problemi, iniziamo da quella **generale**:

$$\begin{array}{l}
 \text{min } z = C^T x \\
 \text{s.t. } \left\{ \begin{array}{l}
 A_1 x \geq b_1 \\
 A_2 x \leq b_2 \\
 A x = e \\
 x' \geq 0 \\
 x'' \text{ libere}
 \end{array} \right.
 \end{array}$$

Gli altri elementi che non sono direttamente spiegati in figura sono:

- C^T , vettore di coefficienti trasposto, così da diventare un vettore riga
- x , vettore di variabili
- **s.t.**, sta per "subject to", cioè "soggetto a" quella serie di vincoli indicati alla sua destra
- A , matrice dei coefficienti delle variabili
- b , termini noti

Questa forma è la più grezza e generale possibile (da cui il nome), ed è opportuno applicarvi alcune correzioni. In primo luogo si osservi che avere vincoli di uguaglianza è come dire che ci sono gradi di libertà in meno, cioè è sempre possibile ricavare una variabile in funzione delle altre per poi sostituirla

ed eliminarla. Per questo motivo tutti i vincoli di uguaglianza di possono togliere, e ad ogni vincolo tolto corrisponde una variabile in meno.

In secondo luogo possiamo eliminare anche le variabili libere dato che possiamo vederle come differenza di variabili non negative. Se ad esempio abbiamo una variabile libera x , scrivendo che $x = y - z$ otteniamo un numero positivo se $y > z$ e un numero negativo se $z > y$. Riscrivendo in questo modo le variabili libere potremo gestire solo quelle non negative, guadagnando sicuramente in semplicità ma aggiungendo però una variabile al totale.

Dall'applicazione di queste correzioni si arriva alla **forma alle disuguaglianze**:

$$\text{max } w = c^T x$$

$$\text{s.t. } \begin{cases} Ax \leq b \\ x \geq 0 \end{cases}$$

Tutti i vincoli sono in forma di disequazioni, e per quanto riguarda il segno si adottano le seguenti convenzioni:

- se la funzione obiettivo chiede di massimizzare, si pongono i vincoli come \leq
- se la funzione obiettivo chiede di minimizzare, si pongono i vincoli come \geq

Interpretazione geometrica della PL

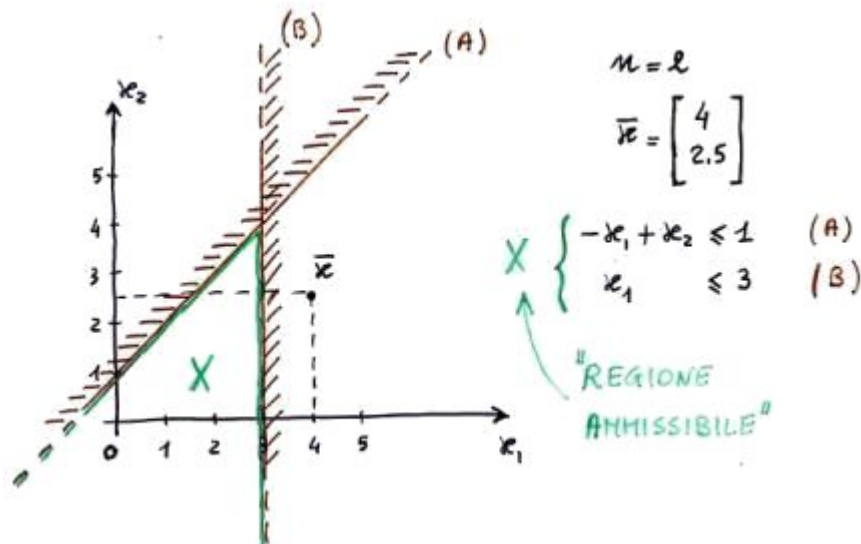
La programmazione lineare ha un'interpretazione geometrica, utile da conoscere poiché molto intuitiva: la soluzione di un problema di PL è un punto in uno spazio ad n dimensioni (R^n), dove n è il numero di variabili. Per punto si intende quell'insieme di valori che diamo alle variabili, quindi ha precise coordinate nello spazio; in particolare lo spazio è continuo (essendo definito in R), dunque le variabili possono assumere tutti i valori reali.

L'interpretazione geometrica di un vincolo di uguaglianza è un *iperpiano* (che per due dimensioni è una retta, per tre una superficie piatta, per quattro o più prende il nome generico di iperpiano), mentre un vincolo di disuguaglianza è un *semispazio* (l'insieme di tutti i punti che stanno da una certa parte rispetto a un iperpiano).

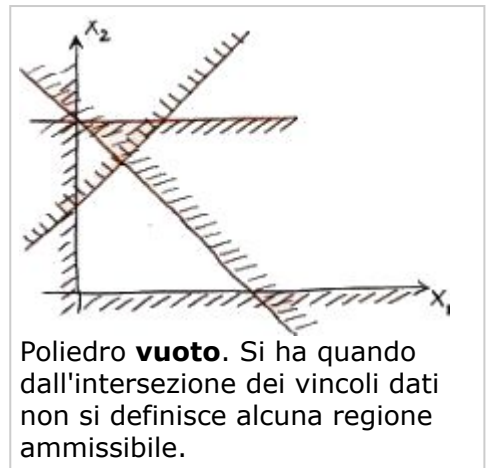
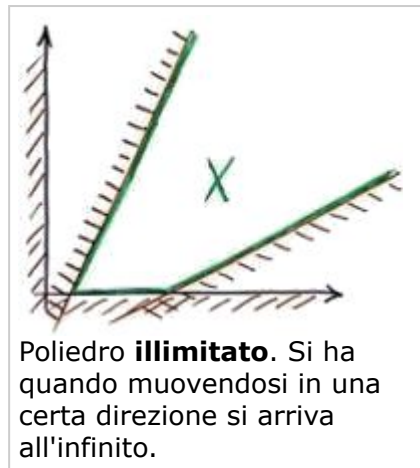
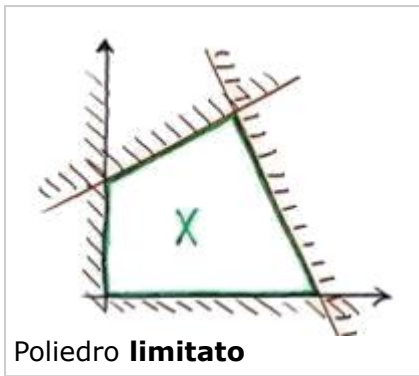
L'intero sistema dei vincoli (quindi sia di uguaglianza che di disuguaglianza) di un problema di programmazione lineare è dato dall'intersezione dei corrispondenti iperpiani e semispazi. Dato però che convertiranno tutti i vincoli di uguaglianza in vincoli di disuguaglianza, tutti gli iperpiani spariscono e ci rimangono semispazi, la cui intersezione è un **poliedro**.

I semispazi hanno un'altra caratteristica, ovvero sono *convessi*: presi due punti qualsiasi che appartengono al semispazio e collegandoli con un segmento, tutti i punti che fanno parte di quel segmento appartengono anch'essi allo stesso semispazio. Poiché l'intersezione di insiemi convessi è convessa, ne deduciamo che anche il poliedro è convesso.

Nella figura successiva saranno riassunti tutti i concetti visti finora, e in particolare chiameremo il poliedro **regione ammissibile**:



Vediamo alcuni casi particolari di poliedro:



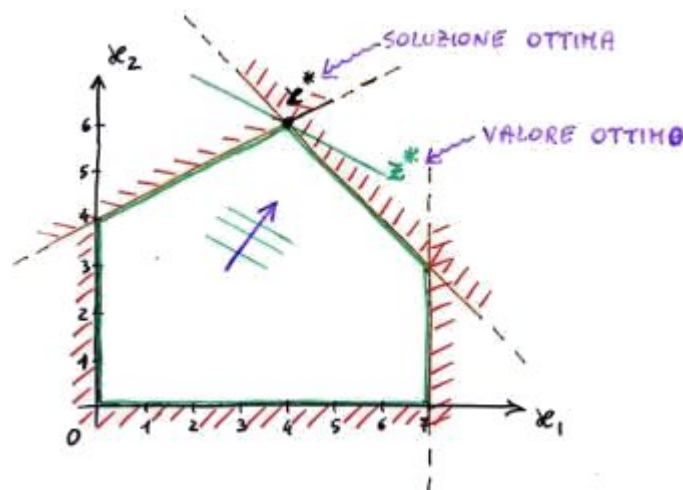
Un poliedro limitato e non vuoto è chiamato **politopo**.

La funzione obiettivo è rappresentata geometricamente con le **curve di livello** e la **direzione di ottimizzazione**. Le curve di livello sono l'insieme di quel luogo di punti che soddisfano l'equazione $C \cdot x = z$, quindi un insieme di rette (o nel caso generale, di interpiani). Perché parliamo di insieme e non di un'unica retta? Perché al variare del risultato z della funzione obiettivo, la C rimane invariata e quindi si ottiene una retta parallela alla precedente. Avremo in questo modo un fascio di rette parallele (le *curve di livello*) che si muovono da una parte verso la massimizzazione della funzione obiettivo, e dall'altro verso la minimizzazione. Questa direzione perpendicolare alle rette è chiamata *direzione di ottimizzazione*, ed è indicata graficamente con una freccia.

Abbiamo finalmente tutti gli strumenti e le conoscenze necessarie per rappresentare geometricamente un problema di programmazione lineare. Inoltre per la convessità del poliedro che rappresenta la regione ammissibile e per la linearità delle curve di livello che rappresentano la funzione obiettivo, possiamo affermare che la soluzione ottima:

- o non esiste, se il poliedro è vuoto
- o è illimitata, se il poliedro è illimitato nella direzione di ottimizzazione
- o è in un vertice del poliedro, se questo è un politopo

Vediamo un esempio riassuntivo:



Forma standard

Ritorniamo ora a vedere il problema da un punto di vista algebrico. La **forma standard** di un problema di programmazione lineare prevede la ritrasformazione di tutti i vincoli di disuguaglianza in vincoli di uguaglianza, e l'introduzione di nuove variabili. Tali variabili sono chiamate **variabili di scarto (slack)** o **di surplus** a seconda di come sono girati i vincoli. Iniziamo a considerare quelle di surplus facendo un esempio:

La figura a sinistra mostra il passaggio da forma alle disuguaglianze a forma standard.

$$\begin{aligned}
 \min \quad & z = c^T x \\
 \text{s.t.} \quad & \begin{cases} Ax \geq b & (m) \\ x \geq 0 & (n) \end{cases}
 \end{aligned}$$

↓

$$\begin{aligned}
 & \begin{cases} Ax - I\tilde{x} = b & (m) \\ x \geq 0 & (n) \\ \tilde{x} \geq 0 & (m) \end{cases}
 \end{aligned}$$

\tilde{x} VAR. DI SURPLUS

Notare la moltiplicazione delle variabili di surplus per una matrice identità I , operazione necessaria dato che abbiamo tante variabili di surplus quanti sono i vincoli.

Facciamo ora un esempio di utilizzo di variabili di scarto:

$$\begin{aligned}
 \max \quad & z = c^T x \\
 \text{s.t.} \quad & \begin{cases} Ax \leq b & (m) \\ x \geq 0 & (n) \end{cases}
 \end{aligned}$$

↓

$$\begin{aligned}
 & \begin{cases} Ax + I\hat{x} = b & (m) \\ x \geq 0 & (n) \\ \hat{x} \geq 0 & (m) \end{cases}
 \end{aligned}$$

\hat{x} VAR. DI SLACK

La figura a sinistra mostra il passaggio da forma alle disuguaglianze a forma standard.

Con questa forma invece di dire che $A \cdot x$ è minore di b , diremo che la quantità che devo sommare ad $A \cdot x$ perché sia uguale a b è $I \cdot \hat{x}$. Capiamo bene che è la stessa cosa.

Tirando le somme, passeremo da un problema con n variabili ed m vincoli di disuguaglianza a un problema con $n+m$ variabili ed m vincoli di uguaglianza. Quindi passando dalla forma alle disuguaglianze a quella standard abbiamo ottenuto un sistema di equazioni con sole condizioni di non negatività.

Breve inciso su una proprietà che si rivelerà fondamentale per i discorsi a venire: ad una variabile nulla corrisponde un *vincolo attivo*. Un vincolo si dice attivo quando, mentre stiamo ottimizzando, abbiamo raggiunto la frontiera della regione ammissibile ed il vincolo ci sta impedendo di superarla; in altre parole ci troviamo proprio sui suoi punti.

Riprendiamo il discorso di prima: con la forma standard abbiamo raggiunto un sistema ad m equazioni lineari ed $n+m$ incognite, a meno che non abbiamo vincoli ridondanti. Questo significa che avremo - se esiste - esattamente un'unica soluzione solo se riusciremo ad eliminare gli n gradi di libertà, ovvero la differenza tra il numero di variabili ed il numero di vincoli. Per riuscirci possiamo dividere le nostre variabili in due sottoinsiemi, dicendo che ci sono m variabili *in base* ed n *fuori base*. Anche la matrice dei vincoli A' (che ha una colonna per ogni variabile) può essere divisa in due sottomatrici, una con le variabili di base e l'altra per quelle fuori base; stesso discorso vale per il vettore x delle variabili.

$$A' = \left[\begin{array}{c|c} B & N \end{array} \right] \quad x = \left[\begin{array}{c} x_B \\ x_N \end{array} \right]$$

$\leftarrow m \rightarrow \leftarrow n \rightarrow$

\uparrow VAR. BASICHE
 \downarrow VAR. NON BASICHE

Possiamo dire che una **base** è un sottoinsieme delle colonne della matrice dei vincoli A' , linearmente indipendenti tra loro, che formano una matrice di dimensioni $m \times m$ non singolare chiamata **B**. Le variabili *fuori base* sono i rimanenti gradi di libertà.

Possiamo perciò riscrivere il sistema dei vincoli come: $\mathbf{B} \mathbf{x}_B + \mathbf{N} \mathbf{x}_N = \mathbf{b}$ ((colonne di base per variabili di base) + (colonne fuori base per variabili fuori base) = vettore dei termini noti)

Se riuscissimo a fissare il valori delle \mathbf{x}_N , e quindi di quegli n valori che ho in più, ciò che ci resta avrebbe una soluzione univocamente determinata. In altre parole, se tratto gli \mathbf{x}_N come dati del problema ottengo:

$$\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b} - \mathbf{B}^{-1} \mathbf{N} \mathbf{x}_N$$

Notare che nella parte destra dell'equazione ho solo dati!

Tutto ciò è molto bello, ma che valori scegliamo per \mathbf{x}_N ? Beh, intanto osserviamo che per ogni scelta che faremo staremo definendo un punto, dal momento che ciò che ci rimane del problema avrà esattamente una soluzione. Noi però stiamo cercando l'ottimalità, quindi non ci interessano tutti i punti ma solo quelli in cui gli n vincoli di disuguaglianza sono attivi, cioè i vertici del poliedro. Avevamo detto infatti che a ogni vincolo attivo corrisponde una variabile nulla, e quali sono quei punti in cui le variabili sono azzerate? Proprio i vertici del poliedro! Le soluzioni ottenute ponendo $\mathbf{x}_N = 0$ sono chiamate **soluzioni di base** e si ottengono come: $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b}$.

Attenzione però! Dobbiamo distinguere tra soluzioni di base *ammissibili*, quelle che si trovano sui vertici del poliedro, e soluzioni di base *non ammissibili*, che sono sì originate dall'intersezione di vincoli ma che si trovano al di fuori del poliedro. Graficamente è immediato distinguere una dalle altre, mentre a livello algebrico basta verificare che tutte le variabili di base associate a quella soluzione siano non negative.

Degenerazione

Consideriamo questo caso particolare: può capitare che per un vertice di un poliedro passino più di n vincoli, e quindi si avrebbe almeno un vincolo attivo in più rispetto a quelli che ho attivato io azzerando le variabili fuori base. Questa situazione in cui abbiamo una variabile di base nulla (proprio perché corrisponde a un vincolo attivo) si chiama **degenerazione**, e si dice che la soluzione è **degenere**.

Il problema di una soluzione degenerare è che possiamo ottenerla con basi diverse, perché avremo almeno una variabile in più rispetto a quelle fuori base che ha valore 0. Consideriamo ad esempio una soluzione che ha questo vettore delle $\mathbf{x} = [\mathbf{2} \ \mathbf{3} \ \mathbf{0} \ 0 \ 0]$, dove in grassetto sono indicate le variabili basiche; abbiamo almeno tre basi diverse che permettono di ottenere la soluzione, ovvero: $\{1,2,3\}$ $\{1,2,4\}$ $\{1,2,5\}$.

Quando parleremo dell'algoritmo del simplesso nelle [[prossime lezioni->RO-Algoritmo del simplesso] capiremo perché la degenerazione è cosa negativa e da evitare. Per ora ci basti sapere che comporta uno spreco di computazione.

Teorema fondamentale della PL

Quanto abbiamo appreso finora è più che sufficiente per enunciare il **teorema fondamentale della programmazione lineare**: dato un problema lineare in forma standard, quindi $\{\min \mathbf{C}^T \mathbf{x} : \mathbf{A} \mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$, con \mathbf{A} di rango m (quindi senza vincoli ridondanti):

1. se esiste una soluzione ammissibile (il poliedro non è vuoto) allora esiste anche una soluzione di base ammissibile (il poliedro deve avere almeno un vertice)
2. se esiste una soluzione ammissibile ottima (il poliedro non è illimitato) allora esiste anche una soluzione ammissibile ottima di base

Che conseguenze ha questo teorema? Che il problema lineare non impossibile o illimitato può essere risolto considerando solo le soluzioni di base, quindi solo un numero finito di punti che appartengono al poliedro; passiamo così da un problema continuo a uno di tipo di combinatorio, che pur essendo di dimensioni molto più contenute cresce comunque rapidamente all'aumentare di vincoli e variabili.

Soluzioni algoritmiche

Ora che sappiamo tutto ciò che dovevamo sapere sulla programmazione lineare, che algoritmo possiamo utilizzare per risolvere questo tipo di problemi? Noi studieremo l' *algoritmo del simplesso*, inventato da Dantzig nel 1947, sicuramente il più importante da un punto di vista storico anche se non l'unico.

Qualche anticipazione. L'algoritmo del simplesso 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 raffinamenti - si dimostra più efficiente di questa classe di algoritmi. Infine, un prerequisito che impareremo a conoscere nella prossima lezione è che il problema deve essere espresso in **forma canonica**. La spieghiamo ora? Ovviamente no, così aumenta la suspense. E che suspense.

[Torna alla pagina di Ricerca Operativa](#)

(Printable View of <http://www.swappa.it/wiki/Uni/RO-ProgrammazioneLineare>)