Indice

Capitolo 1
Complementi

1.1 Spazi vettoriali

Sia V un insieme di elementi dello stesso tipo. Siano poi definite su questo insieme (almeno) due operazioni: una **somma di vettori componente per componente** ed un **prodotto fra un vettore ed uno scalare**. La prima associa a ciascuna coppia di elementi $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ uno ed un solo elemento di V, indicato con $\boldsymbol{v}_1 + \boldsymbol{v}_2$, mentre la seconda associa associa ad ogni numero reale α ed ad ogni elemento $\boldsymbol{v} \in V$ uno ed un solo elemento di V, indicato con $\alpha \cdot \boldsymbol{v}$ o semplicemente con $\alpha\boldsymbol{v}$. Si supponga che l'insieme V possieda tutte le seguenti proprietá:

1. $\boldsymbol{v}_1 + \boldsymbol{v}_2 = \boldsymbol{v}_2 + \boldsymbol{v}_1 \ \forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in V$
2. $(\boldsymbol{v}_1 + \boldsymbol{v}_2) + \boldsymbol{v}_3 = \boldsymbol{v}_1 + (\boldsymbol{v}_2 + \boldsymbol{v}_3) \ \forall \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3 \in V$
3. $\exists 0 \in V t.c. \boldsymbol{v} + 0 = \boldsymbol{v} \ \forall \boldsymbol{v} \in V$
4. $\forall \boldsymbol{v} \in V \ \exists \ \text{-}\boldsymbol{v} \ t.c. \ \boldsymbol{v} + (\text{-}\boldsymbol{v}) = 0$
5. $(\alpha\beta)\boldsymbol{v} = \alpha(\beta\boldsymbol{v}) \ \forall \boldsymbol{v} \in V \wedge \forall \alpha, \beta \in \mathbb{R}$
6. $\alpha(\boldsymbol{v}_1 + \boldsymbol{v}_2) = \alpha\boldsymbol{v}_1 + \alpha\boldsymbol{v}_2 \ \forall \boldsymbol{v}_1, \boldsymbol{v}_2 \in V \wedge \forall \alpha \in \mathbb{R}$
7. $(\alpha + \beta)\boldsymbol{v} = \alpha\boldsymbol{v} + \beta\boldsymbol{v} \ \forall \boldsymbol{v} \in V \wedge \forall \alpha, \beta \in \mathbb{R}$
8. $1 \cdot \boldsymbol{v} = \boldsymbol{v} \ \forall \boldsymbol{v} \in V$

Se questo é vero, V é allora detto **spazio vettoriale** ed i suoi elementi prendono il nome di **vettori**. Dalle proprietá sopra citate é possibile immediatamente dedurne altre, fra le quali:

- $\boldsymbol{v}_1 + \boldsymbol{v}_2 = \boldsymbol{v}_2 + \boldsymbol{v}_3 \Rightarrow \boldsymbol{v}_1 = \boldsymbol{v}_3$
- $\exists! \boldsymbol{x} = \boldsymbol{v}_2 + (\text{-}\boldsymbol{v}_1) \ t.c. \ \boldsymbol{v}_1 + \boldsymbol{x} = \boldsymbol{v}_2$
- $0\boldsymbol{v} = 0 \ \forall \boldsymbol{v} \in V$
- $\alpha 0 = 0 \ \forall \alpha \in \mathbb{R}$
- $(-1)\boldsymbol{v} = \text{-}\boldsymbol{v} \ \forall \boldsymbol{v} \in V$

Uno spazio vettoriale puó essere contenuto in uno spazio vettoriale piú ampio: si parla in questo caso di **sottospazio vettoriale**. Dato V uno spazio vettoriale e S un sottoinsieme non vuoto di V ($S \subseteq V$), si ha che S é un sottospazio vettoriale di V se valgono (almeno) le seguenti due proprietá:

$$\boldsymbol{v}_1, \boldsymbol{v}_2 \in S \Rightarrow (\boldsymbol{v}_1 + \boldsymbol{v}_2) \in S \qquad \boldsymbol{v} \in S \Rightarrow (\alpha\boldsymbol{v}) \in S \ \forall \alpha \in \mathbb{R}$$

Ogni spazio vettoriale V ha sempre almeno due sottospazi vettoriali: sé stesso e 0. Il primo é detto **sottospazio improprio**, il secondo **sottospazio banale**.
Sia V uno spazio vettoriale qualsiasi e siano $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ dei vettori di V. Siano poi $a_1, a_2, ..., a_n$ dei coefficienti reali non necessariamente distinti. É detta **combinazione lineare** dei vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ ogni somma del tipo:

$$\sum_{i=1}^{n} a_i \boldsymbol{v}_i = a_1\boldsymbol{v}_1 + a_2\boldsymbol{v}_2 + ... + a_n\boldsymbol{v}_n$$

L'insieme di tutte le (infinite) combinazioni lineari dei vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ di V é un sottospazio vettoriale di V. In particolare, questo sottospazio vettoriale viene detto **sottospazio di V generato da** $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$, indicato con $[\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n]$ oppure con $\text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n\}$.
Un insieme di vettori di uno spazio vettoriale sono **linearmente dipendenti** se esiste una loro combinazione lineare nulla che abbia almeno un coefficiente non nullo. Un insieme di vettori di uno spazio vettoriale sono **linearmente indipendenti** se non sono linearmente dipendenti, ovvero se ogni loro combinazione lineare nulla ha per coefficienti solo valori nulli. Legate alle combinazioni lineari e alla dipendenza ed indipendenza lineare sono le seguenti proprietá degli spazi vettoriali:

- Se un vettore $\boldsymbol{v} \in V$ si puó esprimere come una combinazione lineare dei vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in V$, allora i vettori $\boldsymbol{v}, \boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ sono linearmente dipendenti. Allo stesso modo, se i vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in V$ sono linearmente dipendenti, allora almeno uno di essi puó essere espresso come una combinazione lineare dei rimanenti;
- Se un insieme di vettori contiene (almeno) un vettore nullo o (almeno) una coppia di vettori uguali, allora quei vettori sono linearmente dipendenti;
- Un vettore $\boldsymbol{v} \in V$ non nullo preso singolarmente é sempre linearmente indipendente. Questo perché se $\boldsymbol{v} \neq 0$, allora si ha $\alpha\boldsymbol{v} = 0$, con il coefficiente α non nullo;
- Sia S_1 un insieme di n vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in V$ linearmente dipendenti. Sia S_2 l'insieme di m vettori costruito aggiungendo ad S_1 i vettori $\boldsymbol{v}_{n+1}, \boldsymbol{v}_{n+2}, ..., \boldsymbol{v}_m \in V$, questi non necessariamente linearmente dipendenti. A prescindere da quali siano i vettori $\boldsymbol{v}_{n+1}, \boldsymbol{v}_{n+2}, ..., \boldsymbol{v}_m$, i vettori in S_2 sono linearmente dipendenti;

- Sia S_1 un insieme di n vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in V$ linearmente indipendenti. Sia S_2 l'insieme non vuoto costruito rimuovendo da S_1 un numero qualsiasi (anche nessuno) di vettori. I vettori in S_2 sono linearmente indipendenti;
- Se i vettori $\boldsymbol{w}, \boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in V$ sono linearmente dipendenti ma $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$ sono linearmente indipendenti, allora $\boldsymbol{w}$ é una combinazione lineare di $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$.

Uno spazio vettoriale V ha **dimensione finita** se esiste (almeno) un insieme di vettori $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n \in V$ che **generano** tutto V, cioé tali per cui ogni elemento di V puó essere espresso come combinazione lineare di $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n$. Se uno spazio vettoriale non ha dimensione finita, si dice allora che ha **dimensione infinita**.
Un insieme di vettori linearmente indipendenti che generano uno spazio vettoriale é detto **base** dello spazio vettoriale. Allo stesso modo, ogni vettore di uno spazio vettoriale puó essere espresso come combinazione lineare degli elementi di una sua qualsiasi base. Valgono le seguenti proprietá:

- Se uno spazio vettoriale non vuoto ha dimensione finita, allora ha sicuramente almeno una base;
- Se v é un vettore di uno spazio vettoriale espresso come combinazione lineare di una sua base, i coefficienti di questa combinazione lineare sono univoci. Questi prendono il nome di **coordinate** o **componenti**;
- Il numero di elementi di ogni base di ogni spazio vettoriale V é sempre lo stesso. Questa quantitá viene chiamata **dimensione** di V e viene indicata con $\dim(V)$;
- Un sottospazio vettoriale di V ha sempre dimensione minore o uguale a V. In particolare, se un sottospazio vettoriale di V ha la stessa dimensione di V, allora quel sottospazio é V stesso;
- Una qualsiasi base di un sottospazio V contiene il massimo numero di vettori linearmente indipendenti che é possibile avere in V. Viceversa, se un qualsiasi insieme di vettori di V linearmente indipendenti é formato da tanti elementi quanti $\dim(V)$, allora quell'insieme é una base per V.

Fra le basi di uno spazio vettoriale figura per importanza la **base canonica**, ovvero la base "piú semplice possibile" per quello spazio vettoriale. La definizione di base canonica differisce per ciascuno spazio vettoriale.

1.2 Matrici

Una **matrice** é un oggetto matematico bidimensionale, rappresentato nelle seguenti forme:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad A = \left(a_{ij}\right)\ i = 1, ..., m \wedge j = 1, \ , n$$

I numeri reali a_{ij} sono detti **elementi della matrice**, mentre i numeri interi i e j sono detti **indici**. I numeri $a_{i1}, a_{i2}, ..., a_{in}$ formano una **riga** di una matrice, mentre i numeri $a_{1j}, a_{2j}, ..., a_{mj}$ formano una **colonna** di una matrice. Il numero di righe e di colonne di una matrice é detto **ordine**, e si indica con $m \times n$. Due matrici $A = \left(a_{ij}\right)$ e $B = \left(b_{ij}\right)$ si dicono **uguali** se hanno lo stesso ordine e se $\left(a_{ij}\right) = \left(b_{ij}\right) \forall i, j$.

$$\begin{pmatrix} 3 & -1 & 0 & \pi \\ -\frac{3}{4} & 5 & \frac{2}{5} & 0 \\ 5 & -2 & 9 & \frac{1}{2} \end{pmatrix}$$

Una matrice che abbia $m = n$ é detta **matrice quadrata di ordine n** (o di ordine m). Gli elementi $a_{11}, a_{22}, ..., a_{nn}$ di una matrice quadrata sono detti **elementi diagonali** e costituiscono la **diagonale** della matrice.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \qquad A = \left(a_{ij}\right)\ i = 1, ..., n \wedge j = 1, ..., n$$

Una matrice é detta **matrice diagonale** se $\left(a_{ij}\right) = 0$ per $i \neq j$, ovvero se tutti gli elementi non diagonali sono nulli (gli elementi diagonali possono anche non essere nulli). Una matrice diagonale particolare é la cosiddetta **matrice identitá**, indicata anche con I_n, che ha la diagonale costituita da tutti e soli 1.

$$A = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix} \qquad I = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Una matrice i cui elementi sono tutti elementi nulli é detta **matrice nulla** e si indica con $A = (0)$ oppure $A = \left(0_{ij}\right)$.

$$A = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Data una matrice A, é detta **trasposta** di A la matrice A^t le cui colonne sono ordinatamente le righe di A.

$$A = \begin{pmatrix} 1 & 0 & 2 \\ \pi & 1 & 3 \\ 3 & -4 & -2 \end{pmatrix} \qquad A^t = \begin{pmatrix} 1 & \pi & 3 \\ 0 & 1 & -4 \\ -2 & 3 & -2 \end{pmatrix}$$

Una matrice formata da una sola riga e da un certo numero n di colonne (in altre parole, di ordine $1 \times n$) é anche detta **matrice riga**, mentre una matrice formata da una sola colonna e da un certo numero di righe (ordine $n \times 1$) é detta **matrice colonna**.

$$A = (a_{11} \;\; a_{12} \;\; \cdots \;\; a_{1n}) \qquad A = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix}$$

Una matrice A é detta **matrice a scala** se in ogni riga il numero di zeri dopo il primo elemento diverso da zero aumenta di riga in riga, fino ad avere eventualmente solo righe nulle (il primo elemento della prima riga puó essere nullo). I primi elementi non nulli di ciascuna riga sono detti **pivot**.

$$A = \begin{pmatrix} 0 & 4 & 1 & 5 & 2 \\ 0 & 0 & 6 & 1 & 9 \\ 0 & 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix} \qquad B = \begin{pmatrix} 2 & 1 & 3 \\ 0 & 4 & 2 \\ 0 & 0 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 3 & 5 & 0 \\ 0 & 9 & 0 & 1 \\ 0 & 0 & -4 & 2 \end{pmatrix}$$

Sulle matrici é possibile applicare operazioni algebriche. La **somma di matrici** genera, prese due matrici A e B, una **matrice somma** $A + B$ i cui elementi sono ottenuti sommando algebricamente gli elementi corrispondenti delle matrici A e B. La somma di matrici é esprimibile come $A + B = \left(a_{ij} + b_{ij}\right)$.

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix} \qquad A + B = \begin{pmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{pmatrix}$$

La somma tra due matrici é possibile solo se queste hanno lo stesso ordine. Si dice in questo caso che le due matrici sono **conformi per la somma**.

$$A = \begin{pmatrix} 3 & 4 & 1 \\ 2 & 5 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} -1 & 2 & 3 \\ 6 & 7 & -2 \end{pmatrix} \qquad A + B = \begin{pmatrix} 2 & 6 & 14 \\ 8 & 12 & -2 \end{pmatrix}$$

Data una matrice A, é detta **matrice opposta** di A la matrice $-A$ tale per cui $A + \left(-A\right) = \left(0_{ij}\right)$

$$A = \begin{matrix} 4 & -1 & 2 \\ 5 & 6 & 0 \end{matrix} \qquad -A = \begin{matrix} -4 & 1 & -2 \\ -5 & -6 & 0 \end{matrix} \qquad A + \left(-A\right) = \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$$

Il **prodotto di una matrice per uno scalare** genera, presi un numero reale k e una matrice A, la matrice kA ottenuta moltiplicando tutti gli elementi di A per il numero k. Possiamo scrivere il prodotto di una matrice per uno scalare come $kA = k(a_{ij})$.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad kA = \begin{pmatrix} ka_{11} & ka_{12} & \cdots & ka_{1n} \\ ka_{21} & ka_{22} & \cdots & ka_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ ka_{m1} & ka_{m2} & \cdots & ka_{mn} \end{pmatrix}$$

$$k = 2A = \begin{pmatrix} -2 & 3 & 1 \\ 4 & 5 & 6 \\ -3 & 1 & -4 \end{pmatrix} \qquad kA = \begin{pmatrix} -4 & 6 & 2 \\ 8 & 10 & 12 \\ -6 & 2 & -8 \end{pmatrix}$$

Il **prodotto tra matrici** (anche detto **prodotto righe per colonne**) genera, data una matrice A di ordine $m \times p$ e una matrice B di ordine $p \times n$, la **matrice prodotto** C i cui elementi si ottengono come segue:

$$C = (c_{ij}) = \sum_{i=1}^{p} a_{ik} \cdot b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + \ldots + a_{ip}b_{pj} \; i = 1, 2, \ldots, m \wedge j = 1, 2, \ldots, n$$

Se due matrici possono essere moltiplicate tra di loro, allora le due matrici sono dette **conformi per il prodotto**. Due matrici sono conformi per il prodotto se il numero di colonne della prima matrice é uguale al numero di righe della seconda matrice. La matrice risultante avrá numero di righe pari al numero di righe della prima matrice e numero di colonne pari al numero di colonne della seconda matrice.

$$A = \begin{pmatrix} -1 & 4 \\ 6 & 1 \end{pmatrix} \qquad B = \begin{pmatrix} 3 & 2 & -4 \\ 5 & 0 & 2 \end{pmatrix}$$

$$AB = \begin{pmatrix} -1 & 4 \\ 6 & 1 \end{pmatrix}\begin{pmatrix} 3 & 2 & -4 \\ 5 & 0 & 2 \end{pmatrix} = \begin{pmatrix} (-1)\cdot 3 + 4 \cdot 5 & (-1)\cdot 2 + 4 \cdot 0 & (-1)\cdot(-4) + 4 \cdot 2 \\ 6 \cdot 3 + 1 \cdot 5 & 6 \cdot 2 + 1 \cdot 0 & 6 \cdot (-4) + 1 \cdot 2 \end{pmatrix} = \begin{pmatrix} 17 & -2 & 12 \\ 23 & 12 & -22 \end{pmatrix}$$

L'operazione di prodotto tra matrici differisce dall'operazione di prodotto tra due numeri reali [1]:

- Il prodotto tra matrici non é commutativo. Date due matrici A e B, AB e BA non sono (sempre) uguali.
- Matrici diverse possono portare a prodotti di matrici di uguale risultato. Date tre matrici A, B e C, se si verifica $AB = AC$ non é necessariamente vero che B e C siano uguali.
- Il prodotto di due matrici non nulle puó essere nullo. Date due matrici A e B, entrambe non nulle, puó comunque verificarsi $AB = 0$.
- La matrice identitá e l'elemento neutro della moltiplicazione fra matrici, in quanto $AI = IA = A$ per qualsiasi matrice A.

Il **determinante** di una matrice é un particolare numero associato ad ogni matrice quadrata, indicato con $\det(A)$. Il calcolo del determinante di una matrice viene fatto per ricorsione. I casi base del procedimento si hanno per le matrici di ordine 1 e 2, il cui determinante é calcolabile in maniera immediata:

$$A = (a_{11}) \qquad \det(A) = a_{11}$$

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad \det(A) = a_{11}a_{22} - a_{21}a_{12}$$

Data una matrice quadrata A di ordine n, sia A^*_{ij} la matrice di ordine $n-1$ ottenuta a partire da A togliendo la i-esima riga e la j-esima colonna. Il determinante di A puó essere ottenuto in funzione del determinante di A^*_{ij} mediante la seguente equazione di ricorrenza:

1. É possibile vedere il prodotto fra due numeri reali come il caso particolare del prodotto fra due matrici quadrate di ordine 1.

$$det(A) = (-1)^{i+1} a_{i1} det(A^*_{i1}) + (-1)^{i+2} a_{i2} det(A^*_{i2}) + \ldots + (-1)^{i+n} a_{in} det(A^*_{in})$$

Questa formula é detta **sviluppo del determinante secondo la i-esima riga di A**. A prescindere da quale sia la riga scelta, il risultato che si ottiene é sempre lo stesso.

Il determinante di una matrice é uguale al determinante della relativa matrice trasposta. Pertanto, il determinante di una matrice puó essere sviluppato anche secondo le sue colonne.

Se una matrice ha (almeno) una riga/colonna composta interamente da zeri, il suo determinante é zero.

Dimostrazione. Sia A una matrice la cui i-esima riga/colonna é composta interamente da zeri. Sviluppando il determinante secondo tale riga/colonna, si ottiene:

$$\det(A) = (-1)^{i+1} \cdot 0 \cdot det(A^*_{i1}) + \ldots + (-1)^{i+n} \cdot 0 \cdot det(A^*_{in}) = 0 + \ldots + 0 = 0$$

Il determinante di una matrice é una funzione omogenea.

Dimostrazione. Osservando l'equazione di ricorrenza per il calcolo del determinante, il risultato é immediato:

$$\det(kA) = (-1)^{i+1} k a_{i1} det(A^*_{i1}) + (-1)^{i+2} k a_{i2} det(A^*_{i2}) + \ldots + (-1)^{i+n} k a_{in} det(A^*_{in}) = k\det(A)$$

Sviluppare il determinante di una matrice secondo le righe/colonne aventi degli zeri (se esistono) é vantaggioso, perché parte delle computazioni puó venire evitata.

$$\det\begin{pmatrix} -2 & 2 & -3 \\ -1 & 1 & 3 \\ 2 & 0 & -1 \end{pmatrix} = 2 \cdot (-1)^{3+1} \det\begin{pmatrix} 2 & -3 \\ 1 & 3 \end{pmatrix} + 0 \cdot (-1)^{3+2} \det\begin{pmatrix} -2 & -3 \\ -1 & 3 \end{pmatrix} + (-1) \cdot (-1)^{3+3} \det\begin{pmatrix} -2 & 2 \\ -1 & 1 \end{pmatrix} = 18$$

Il determinante é stato sviluppato secondo la terza riga, perché contiene uno zero.

Teorema di Binet. Date due matrici quadrate, il determinante della loro matrice prodotto é uguale al prodotto dei loro determinanti.

La nozione di **dipendenza lineare** e **indipendenza lineare** puó essere declinata nel caso specifico delle matrici. Data una matrice A, le righe/colonne $A_1, A_2, \ldots, A_m$ di A si dicono linearmente dipendenti se esistono dei coefficienti $k_1, k_2, \ldots, k_m \in \boldsymbol{R}$ non tutti nulli tali che:

$$k_1 A_1 + k_2 A_2 + \ldots + k_m A_m = (0_{ij})$$

Scegliendo come coefficienti $2, -1, 1$, é possibile mostrare come le righe (e quindi le colonne) della matrice presentata di seguito siano linearmente dipendenti.

$$A = \begin{pmatrix} 1 & -2 & 1 \\ 3 & 4 & 7 \\ 1 & 8 & 5 \end{pmatrix}$$

$$2A_1 - A_2 + A_3 = (2, -4, 2) - (3, 4, 7) + (1, 8, 5) = (0, 0, 0)$$

Se le righe/colonne di A non sono linearmente dipendenti, si dicono linearmente indipendenti. In particolare, le righe/colonne $A_1, A_2, ..., A_m$ della matrice A sono linearmente indipendenti se:

$$k_1 A_1 + k_2 A_2 + ... + k_m A_m = (0) \Rightarrow k_1 = 0, k_2 = 0, ..., k_m = 0$$

Se le righe/colonne di una matrice quadrata sono linearmente dipendenti, si dice che tale matrice é una **matrice singolare**. Viceversa, se le righe/colonne di una matrice quadrata sono linearmente indipendenti, si dice che tale matrice é una **matrice non singolare**.

Se una matrice é singolare, il determinante di tale matrice é zero, e viceversa.

Una riga/colonna A_i della matrice A é combinazione lineare delle altre righe/colonne se esistono $a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n \in \boldsymbol{R}$ non tutti nulli tali che:

$$A_i = a_1 A_1 + a_2 A_2 + ... + a_{i-1} A_{i-1} + a_{i+1} A_{i+1} + ... + a_n A_n$$

Se le righe/colonne di una matrice sono linearmente dipendenti, allora (almeno) una delle righe/colonne é esprimibile come combinazione lineare delle altre, e viceversa.

Dimostrazione. Se le righe/colonne di una matrice sono linearmente dipendenti, allora vale:

$$k_1 A_1 + ... + k_{i-1} A_{i-1} + k_i A_i + k_{i+1} A_{i+1} + ... + k_m A_m = (0_{ij})$$

Dove almeno uno dei coefficienti non é nullo, sia questo ad esempio k_i. Spostando $k_i A_i$ a destra e dividendo ambo i membri per k_i, si ottiene:

$$\frac{k_1}{k_i} A_1 + ... + \frac{k_{i-1}}{k_i} A_{i-1} + \frac{k_{i+1}}{k_i} A_{i+1} + ... + \frac{k_m}{k_i} A_m = A_i$$

Che é la definizione di combinazione lineare.

Il massimo numero di righe linearmente indipendenti di una matrice A é chiamato **rango per righe** di A. Analogamente, il massimo numero di colonne linearmente indipendenti di una matrice A é detto **rango per colonne** di A.

Presa una matrice generica A di ordine $m \times n$, é detto **minore di ordine r** una qualunque matrice quadrata ottenuta intersecando r colonne della matrice A e r righe della matrice A. É evidente dalla definizione come l'ordine di un qualsiasi minore di una matrice non possa essere maggiore del numero di righe o di colonne della matrice da cui é stato estratto.

La **caratteristica** di una matrice A, indicata con car(A), é l'ordine del minore di A che ha il maggior numero di righe/colonne avente il determinante diverso da zero. Essendo la caratteristica di una matrice l'ordine di un suo minore, dovrá necessariamente aversi che anche la caratteristica di una matrice non puó essere maggiore del numero di righe/colonne della matrice stessa.

$$A = \begin{pmatrix} 4 & 1 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 4 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 4 & 3 & 9 \\ 0 & -3 & -4 & 6 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- Il determinante di A é 32, pertanto il minore con determinante diverso da zero di A avente dimensione massima é sé stessa. Allora, $\text{car}(A) = 3$.
- B ha una riga composta interamente da zeri. Questo significa che qualsiasi minore di ordine 3 estratto da B avrá a sua volta una riga composta interamente da zeri, e pertanto avrá determinante pari a zero. Tuttavia, B possiede almeno un minore di ordine 2 con determinante diverso da zero, ad esempio quello estratto a partire dalle prime due righe e dalle prime due colonne (che ha determinante pari a -3). Allora, $\text{car}(B) = 2$.

La caratteristica di una matrice a scala é pari al numero dei suoi pivot.

Il rango per righe/colonne di una matrice é uguale alla sua caratteristica.

Vengono definite **mosse di Gauss** le tre operazioni riportate di seguito che possono essere eseguite una o piú volte sulle matrici:

- Permutare due righe/colonne;
- Moltiplicare una riga/colonna per un numero reale diverso da zero;
- Sommare ad una riga/colonna un'altra riga/colonna moltiplicata per un numero reale.

Applicando un qualsiasi numero di volte una o piú mosse di Gauss ad una matrice si ottiene una matrice con la stessa caratteristica della matrice originaria.

Essendo il calcolo della caratteristica di una matrice a scala molto semplice ed essendo la caratteristica invariante rispetto alle mosse di Gauss, per calcolare la caratteristica di una matrice generica é utile convertirla in una matrice a scala operando mosse di Gauss e calcolare la caratteristica sulla matrice risultante.

$$A = \begin{pmatrix} 1 & -1 & 3 & 2 \\ 3 & 2 & 7 & 6 \\ 1 & 4 & 1 & 2 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & -1 & 3 & 2 \\ 0 & 5 & -2 & 0 \\ 0 & 5 & -2 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & -1 & 3 & 2 \\ 0 & 5 & -2 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

La caratteristica della matrice generica A é pari a 2.

Applicando k volte la prima mossa di Gauss ad una matrice si ottiene una matrice il cui determinante é dato dal prodotto fra il determinante della matrice originaria e $(-1)^k$.

Applicando una o piú volte la terza mossa di Gauss ad una matrice si ottiene una matrice con lo stesso determinante della matrice originaria.

Viene detta **matrice inversa** della matrice quadrata A la matrice quadrata A^{-1} tale per cui valga la relazione $AA^{-1} = A^{-1}A = I$. Indicando con x_{ij} l'elemento della matrice inversa A^{-1} di riga i e colonna j, si ha:

$$x_{ij} = \frac{(-1)^{i+j}\det(A^*_{ji})}{\det(A)}$$

Dove A^*_{ji} indica la matrice A di partenza a cui é stata tolta la j-esima riga e la i-esima colonna.

Il determinante di una matrice é l'inverso del determinante della rispettiva matrice inversa.

Dimostrazione. Il teorema é una diretta conseguenza del teorema di Binet:

$$\det\left(AA^{-1}\right) = \det\left(A\right) \cdot \det\left(A^{-1}\right) = 1 \;\Rightarrow\; \det\left(A\right) = \frac{1}{\det(A^{-1})} \;\Rightarrow\; \det\left(A^{-1}\right) = \frac{1}{\det(A)}$$

Se una matrice é singolare, la sua matrice inversa non esiste.

Dimostrazione. Una matrice A é singolare se $\det(A) = 0$. Dato che per il teorema precedente $\det\left(A\right) = \frac{1}{\det(A^{-1})}$, se A é singolare si avrebbe $0 = \frac{1}{\det(A^{-1})}$, che é una equazione impossibile.

$$A = \begin{pmatrix} 2 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \qquad A^{-1} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{4} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$x_{11} = \frac{(-1)^{1+1}\det(A^*_{11})}{\det(A)} = \frac{1}{4} \qquad x_{12} = \frac{(-1)^{1+2}\det(A^*_{12})}{\det(A)} = \frac{1}{2} \qquad x_{13} = \frac{(-1)^{1+3}\det(A^*_{13})}{\det(A)} = -\frac{1}{2}$$

$$x_{21} = \frac{(-1)^{2+1}\det(A^*_{21})}{\det(A)} = -\frac{1}{4} \qquad x_{22} = \frac{(-1)^{2+2}\det(A^*_{22})}{\det(A)} = -\frac{1}{2} \qquad x_{23} = \frac{(-1)^{2+3}\det(A^*_{23})}{\det(A)} = \frac{1}{2}$$

$$x_{31} = \frac{(-1)^{3+1}\det(A^*_{31})}{\det(A)} = -\frac{1}{4} \qquad x_{32} = \frac{(-1)^{3+2}\det(A^*_{32})}{\det(A)} = \frac{1}{2} \qquad x_{33} = \frac{(-1)^{3+3}\det(A^*_{33})}{\det(A)} = \frac{1}{2}$$

Una matrice quadrata puó essere classificata sulla base del segno dei suoi autovalori:

- Se sono tutti strettamente positivi, si dice che la matrice é **definita positiva**;
- Se sono tutti strettamente negativi, si dice che la matrice é **definita negativa**;
- Se sono tutti positivi o nulli, si dice che la matrice é **semidefinita positiva**;
- Se sono tutti negativi o nulli, si dice che la matrice é **semidefinita negativa**;
- Altrimenti, si dice che la matrice é **indefinita**.

Una matrice quadrata a coefficienti interi é detta **unimodulare** se il suo determinante é 1 oppure -1. Una matrice (non necessariamente quadrata) é detta **totalmente unimodulare** se tutti i suoi minori sono una matrice unimodulari (determinante = ± 1) o matrici singolari (determinante = 0).

La generica matrice presentata di seguito é una matrice totalmente unimodulare. É inoltre facile verificare che la matrice identitá di un qualsiasi ordine é sempre una matrice totalmente unimodulare.

$$A = \begin{matrix} -1 & -1 & 0 & 0 & 0 & +1 \\ +1 & 0 & -1 & -1 & 0 & 0 \\ 0 & +1 & +1 & 0 & -1 & 0 \\ 0 & 0 & 0 & +1 & +1 & -1 \end{matrix}$$

1.3 Algebra lineare

É detta **equazione lineare** su $\mathbb{R}$ ogni equazione del tipo:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$$

$x_i, i = 1, 2, \ldots, n$ sono dette **variabili** o **incognite**. Se il numero di variabili é piccolo, queste vengono spesso indicate con $x, y, z, \ldots$ anziché $x_1, x_2, x_3, \ldots$. I numeri $a_i, i = 1, 2, \ldots, n$ sono invece chiamati **coefficienti** delle variabili x_i, mentre b é detto **termine noto** dell'equazione.
Una n-pla ordinata di numeri reali $(k_1, k_2, \ldots, k_n)$ é detta **soluzione** dell'equazione precedente se vale:

$$a_1k_1 + a_2k_2 + \ldots + a_nk_n = b$$

Una equazione lineare puó avere una, nessuna o infinite soluzioni.

1. L'equazione $3x = 5$ ha una ed una sola soluzione: $x = \frac{5}{3}$;
2. L'equazione $2x - y = 1$ ha infinite soluzioni: tutte le coppie di numeri reali $(k, 2k-1), k \in \mathbb{R}$;
3. L'equazione $0x = 1$ non ha nessuna soluzione: qualsiasi valore si sostituisca a x si ottiene sempre $0 = 1$, che é una equazione impossibile.

Un insieme di m equazioni in n incognite $x_1, x_2, \ldots, x_n$ considerate contemporaneamente viene chiamato **sistema lineare di m equazioni in n incognite**, o semplicemente **sistema lineare**. Un sistema lineare ha in genere la seguente forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \quad\quad \vdots \quad\quad \ddots \quad\quad \vdots = \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases} a_{ij}\ i = 1, \ldots, m\ j = 1, \ldots, n$$

In particolare, se i termini noti $b_i, i = 1, \ldots, m$ sono tutti nulli, il sistema é detto **sistema lineare omogeneo**.

$$\begin{cases} 2x + 3y - 5z = 1 \\ 4x + y - 2z = 3 \end{cases} \qquad \begin{cases} 2x + 3y - 2z = 0 \\ 4x + 3y - 2z = 0 \\ x + 2y - z = 0 \end{cases} \qquad \begin{cases} 2x + 2y = 1 \\ 4x + y = -1 \end{cases}$$

Una n-pla ordinata di numeri reali $(k_1, k_2, \ldots, k_n)$ é detta **soluzione** del sistema lineare se soddisfa contemporaneamente tutte le equazioni del sistema. Ciascuna soluzione del sistema é detta **soluzione particolare**, mentre l'insieme di tutte le soluzioni particolari é detta **soluzione generale**.
Risolvere un sistema significa trovare la soluzione generale del sistema: se questa esiste, il sistema é detto **consistente**, altrimenti é detto **inconsistente**. La soluzione generale puó essere costituita da una, nessuna o infinite soluzioni.

$$A = \begin{cases} x + y = 0 \\ 2x + 2y = 1 \end{cases} \Rightarrow \begin{cases} x = -y \\ 2(-y) + 2y = 1 \end{cases} \Rightarrow \begin{cases} x = -y \\ -2y + 2y = 1 \end{cases} \Rightarrow \begin{cases} x = -y \\ 0 = 1 \end{cases}$$

$$B = \begin{cases} 2x + y = 2 \\ 4x + 2y = 4 \end{cases} \Rightarrow \begin{cases} y = 2-2x \\ 4x + 2(2-2x) = 4 \end{cases} \Rightarrow \begin{cases} y = 2-2x \\ 4x + 4-4x = 4 \end{cases} \Rightarrow \begin{cases} y = 2-2x \\ 0 = 0 \end{cases}$$

$$C = \begin{cases} 3x + 2y = 4 \\ 5x + y = 1 \end{cases} \Rightarrow \begin{cases} y = 1-5x \\ 3x + 2(1-5x) = 4 \end{cases} \Rightarrow \begin{cases} y = 1-5x \\ 3x + 2-10x = 4 \end{cases} \Rightarrow \begin{cases} x = -\frac{2}{7} \\ y = \frac{17}{7} \end{cases}$$

Il sistema A non ha nessuna soluzione. Il sistema B ha infinite soluzioni nella forma $(k, 2-2k), k \in \mathbb{R}$. Il sistema C ha per unica soluzione $-2 / 7, 17 / 7$.

É possibile associare delle matrici ad un sistema di equazioni. In particolare, la matrice formata dai coefficienti del sistema lineare (disposti nello stesso ordine in cui il sistema lineare é riportato) é chiamata **matrice incompleta** del sistema lineare, indicata in genere con la lettera A.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

La matrice colonna formata dalle incognite di un sistema lineare viene in genere indicata con X, mentre la matrice colonna formata dai termini noti di un sistema lineare viene in genere indicata con B.

$$X = \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \qquad B = \begin{matrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{matrix}$$

La matrice formata giustapponendo alla matrice incompleta A di un sistema lineare e la matrice dei termini noti B viene chiamata **matrice completa**, e viene in genere indicata con (A, B).

$$\left(A, B\right) = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{pmatrix}$$

Data la struttura delle matrici A, B e X, é possibile notare come gli elementi di un sistema lineare possano venire rappresentati come prodotto $AX = B$ fra matrici:

$$X_1 A^1 + X_2 A^2 + \ldots + X_n A^n = B$$

$$\begin{matrix} 2x-y+z=4 \\ x+3y-4z=1 \\ -x+5y+6z=0 \end{matrix} \Rightarrow \begin{pmatrix} 2 & -1 & 1 \\ 1 & 3 & -4 \\ -1 & 5 & 6 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix} \Rightarrow \begin{pmatrix} 2x-y+z \\ x+3y-4z \\ -x+5y+6z \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 2 & -1 & 1 \\ 1 & 3 & -4 \\ -1 & 5 & 6 \end{pmatrix} \qquad X = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \qquad B = \begin{pmatrix} 4 \\ 1 \\ 0 \end{pmatrix}$$

Teorema di Rouché-Capelli. Un sistema lineare é consistente se e solo se la relativa matrice completa e la relativa matrice incompleta hanno la stessa caratteristica.

$$\begin{cases} x-y+2z=1 \\ 3x+y+3z=6 \\ x+3y-z=-1 \end{cases} \qquad A=\begin{pmatrix} 1 & -1 & 2 \\ 3 & 1 & 3 \\ 1 & 3 & -1 \end{pmatrix} \qquad \left(A,B\right)=\begin{pmatrix} 1 & -1 & 2 & 1 \\ 3 & 1 & 3 & 6 \\ 1 & 3 & -1 & -1 \end{pmatrix}$$

La caratteristica della matrice incompleta A é 2, mentre quella della matrice completa (A,B) é 3, quindi il sistema é inconsistente.

$$\begin{cases} x+3y-z=-2 \\ 4x+y+z=1 \\ 2x-5y+3z=5 \end{cases} \qquad A=\begin{pmatrix} 1 & 3 & -1 \\ 4 & 1 & 1 \\ 2 & -5 & 3 \end{pmatrix} \qquad \left(A,B\right)=\begin{pmatrix} 1 & 3 & -1 & -2 \\ 4 & 1 & 1 & 1 \\ 2 & -5 & 3 & 5 \end{pmatrix}$$

Sia la matrice incompleta A sia la matrice completa (A,B) hanno caratteristica 2, quindi il sistema é consistente.

Un sistema lineare omogeneo é sempre consistente.

Dimostrazione. La matrice completa di un sistema lineare omogeneo é ottenuta giustapponendo una colonna di soli zeri alla matrice incompleta, pertanto la caratteristica di tali matrici é necessariamente la stessa. Il teorema di Rouché-Capelli stabilisce allora che tale sistema avrá sempre almeno una soluzione. É facile verificare che tale soluzione é la soluzione nulla, ovvero quella formata esclusivamente da zeri.

Un sistema lineare é detto **determinato** se é costituito da tante equazioni quante sono le incognite che in queste compaiono. É invece detto **sovradeterminato** se ha piú equazioni che incognite. Infine, é detto **sottodeterminato** se ha piú incognite che equazioni.

Teorema di Cramer. Un sistema lineare determinato ha una ed una sola soluzione se e solo se la matrice incompleta associata al sistema non é singolare.

Sia dato un sistema lineare consistente e determinato/sovradeterminato. Questo avrá una sola soluzione se la caratteristica della relativa matrice incompleta coincide con il numero delle incognite, mentre avrá infinite soluzioni se la caratteristica della relativa matrice incompleta é inferiore al numero delle incognite.

Un sistema lineare sottodeterminato, se é consistente, ha sempre infinite soluzioni.

Sia data la matrice incompleta associata ad un sistema lineare. Applicando un qualsiasi numero di volte una o piú mosse di Gauss a tale matrice si ottiene una matrice incompleta associata ad un sistema lineare avente lo stesso insieme di soluzioni del sistema originario.

$$\begin{cases} x-y=2 \\ x-y-z=3 \\ x+y+z=6 \end{cases} \qquad \begin{pmatrix} 1 & -1 & 0 \\ 1 & -1 & -1 \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 6 \end{pmatrix}$$

Il sistema puó essere semplificato applicando mosse di Gauss alla relativa matrice completa:

$$\begin{pmatrix} 1 & -1 & 0 & 2 \\ 1 & -1 & -1 & 3 \\ 1 & 1 & 1 & 6 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & 1 & -1 & -1 \\ 1 & 1 & 1 & 6 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 2 & 1 & 4 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 3 & 6 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

Si sommi alla seconda riga la prima riga moltiplicata per -2.
Si sommi alla terza riga la prima riga moltiplicata per -1.
Si sommi alla terza riga la seconda riga moltiplicata per -2.
Si moltiplichi la terza riga per 1/3
Il sistema associato a tale matrice completa ha come sola ed unica soluzione $(3, 1, 2)$.

1.4 Gradiente ed Hessiano

Sia data una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$. Siano poi $\boldsymbol{v} \in \mathbb{R}$ un vettore non nullo e $\boldsymbol{x}_0 \in X^o$ un punto interno ad X. Si dice **derivata nella direzione $\boldsymbol{v}$** di f nel punto x_0 il valore, ammesso che esista, del seguente limite:

$$D_{\boldsymbol{v}} f(\boldsymbol{x}_0) = \frac{\partial f}{\partial \boldsymbol{v}} = \lim_{t\to 0} \frac{f(\boldsymbol{x}_0 + t\boldsymbol{v}) - f(\boldsymbol{x}_0)}{t}$$

La derivata direzionale esprime la pendenza della funzione. Derivate direzionali di particolare rilevanza sono quelle valutate rispetto agli assi. Se il vettore $\boldsymbol{v}$ rispetto alla quale si calcola la derivata parziale é uno dei versori degli assi (ovvero, uno dei vettori della base canonica di $\mathbb{R}^n$), la derivata direzionale rispetto a questo vettore viene chiamata **derivata parziale**, e viene indicata con $\frac{\partial f}{\partial x_i}$, dove x_i indica la i-esima componente della base canonica di $\mathbb{R}_n$. Una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ si dice **derivabile** nel punto $\boldsymbol{x}_0 \in X$ se tutte le sue derivate parziali in x_0 sono definite.
La derivata direzionale (e parziale) per funzioni su $\mathbb{R}^n$ mantiene tutte le proprietá algebriche (linearitá, regola della catena, derivate notevoli, ecc...) della derivata per funzioni ad una variabile. Il calcolo della i-esima derivata parziale é molto semplice, dato che é sufficiente derivate la funzione normalmente considerando tutte le componenti al di fuori della i-esima come costanti.

$$f(x,y) = x^2\cos(y) \qquad \frac{\partial f}{\partial x}(x^2\cos(y)) = 2x\cos(y) \qquad \frac{\partial f}{\partial y}(x^2\cos(y)) = -x^2\sin(y)$$

Data una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$, viene chiamato **vettore gradiente**, indicato con il simbolo ∇f oppure come $\operatorname{grad} f$, il vettore n-dimensionale che ha per componenti le n derivate parziali di f. Rispetto al dominio della funzione nel suo complesso, il vettore gradiente é un campo vettoriale, perché il suo valore non é necessariamente lo stesso per ogni punto in cui la funzione é definita.

$$\nabla f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial x_1} \; \frac{\partial f(\boldsymbol{x})}{\partial x_2} \; \dots \; \frac{\partial f(\boldsymbol{x})}{\partial x_n}$$

Se una funzione ha un massimo o un minimo locale in un punto interno al suo dominio e la derivata direzionale della funzione in quel punto esiste, allora é nulla.

Dimostrazione. Siano date una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ ed un vettore $\boldsymbol{v} \neq 0 \in \mathbb{R}^n$. Si assuma che f abbia un massimo o un minimo locale nel punto $\boldsymbol{x}_0 \in X^o$.
Sia $g(t) = f(\boldsymbol{x}_0 + t\boldsymbol{v})$ una funzione a una variabile costruita a partire da f. Per t piccolo, la funzione g é certamente ben definita. Per funzioni ad una variabile é noto che se questa ha un massimo o un minimo locale in un punto ed é derivabile in quel punto, allora la derivata in quel punto deve essere nulla. Pertanto:

$$g'(t) = \frac{dg}{dt}(t) = \lim_{t\to 0}\frac{g(t)-g(0)}{t} = \lim_{t\to 0}\frac{f(\boldsymbol{x}_0 + t\boldsymbol{v})-f(\boldsymbol{x}_0)}{t} = D_{\boldsymbol{v}}f(\boldsymbol{x}_0) = 0$$

Che era il risultato che si stava cercando.

Se una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ ha derivata direzionale nulla nel punto $\boldsymbol{x}_0$, allora anche il gradiente in x_0 é nullo.

Dimostrazione. Questo segue direttamente dal teorema appena mostrato: se la derivata direzionale é nulla per un vettore generico, allora lo saranno anche le derivate direzionali rispetto alla base canonica di $\mathbb{R}^n$, ovvero le derivate parziali.

I punti di una funzione che hanno gradiente nullo si dicono **punti di equilibrio** oppure **punti stazionari** oppure **punti critici**. Il teorema appena mostrato fornisce un modo semplice per trovare i punti stazionari di una funzione: é sufficiente calcolarne il vettore (campo) gradiente e valutare i punti, se esistono, in cui questo si annulla.

Data la funzione $x(1-x^2-4y^2)$ ristretta al dominio $X = \{x^2 + 4y^2 < 1\}$, si é interessati a studiarne i punti stazionari. Si calcolino le derivate parziali:

$$\begin{cases}\frac{\partial f}{\partial x}(x(1-x^2-4y^2)) = x\left(\frac{\partial f}{\partial x}(1-x^2-4y^2)\right) + (1-x^2-4y^2)\left(\frac{\partial f}{\partial x}(x)\right) = 1-3x^2-4y^2\\ \frac{\partial f}{\partial y}(x(1-x^2-4y^2)) = (1-x^2-4y^2)\left(\frac{\partial f}{\partial y}(x)\right) + x\left(\frac{\partial f}{\partial y}(1-x^2-4y^2)\right) = -8xy\end{cases}$$

Queste corrispondono alle componenti del vettore (campo) gradiente. Imponendo che siano entrambe nulle, si ha:

$$\begin{cases}1-3x^2-4y^2 = 0\\ -8xy = 0\end{cases} \Rightarrow \begin{cases}1-0-4y^2 = 0\\ x = 0\end{cases} \cup \begin{cases}1-3x^2-0 = 0\\ y = 0\end{cases} \Rightarrow \begin{cases}y = \frac{1}{2}\\ x = 0\end{cases} \cup \begin{cases}x = \frac{1}{\sqrt{3}}\\ y = 0\end{cases}$$

Si ottengono le due coppie di punti $\left(0, \frac{1}{2}\right)$ e $\left(\frac{1}{\sqrt{3}}, 0\right)$. Si noti peró come:

$$(0)^2 + 4\left(\frac{1}{2}\right)^2 < 1 \;\Rightarrow\; 1 < 1 \qquad \left(\frac{1}{\sqrt{3}}\right)^2 + 4(0)^2 < 1 \;\Rightarrow\; \frac{1}{3} < 1$$

Pertanto, solamente la coppia $\left(\frac{1}{\sqrt{3}}, 0\right)$ é un punto stazionario per X.

Le derivate parziali di una funzione, essendo funzioni esse stesse, possono essere a loro volta derivate. Data la derivata parziale $\partial f \,/\, \partial i$ di una funzione f ad n componenti rispetto alla componente i, la sua derivata parziale rispetto a j viene espressa come $\partial f \partial f \,/\, \partial j \partial i$, o piú semplicemente $\partial^2 f \,/\, \partial j \partial i$. Nel caso in cui $i = j$, é possibile scrivere la derivata parziale della derivata parziale nella forma ancora piú compatta $\partial^2 f \,/\, \partial^2 i$. Nel caso piú generale possibile:

Con x, y, z, j non necessariamente distinti

$$\frac{\partial^i f}{\partial j}\left(\frac{\partial f}{\partial x \partial y \ldots \partial z}\right) = \frac{\partial^{i+1} f}{\partial j \partial x \partial y \ldots \partial z}$$

Nel caso particolare di una funzione f scalare a due variabili, le derivate parziali $\frac{\partial^2 f}{\partial x \partial y}$ e $\frac{\partial^2 f}{\partial y \partial x}$ sono dette **derivate miste**.

Teorema delle derivate miste. Se una funzione a due variabili ha definite le derivate parziali miste nell'intorno di un punto e sono entrambe continue in quel punto, allora le due derivate miste nel punto coincidono:

$$\frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) = \frac{\partial^2 f}{\partial y \partial x}(x_0, y_0)$$

Si dice **matrice hessiana** di una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ la matrice $H(f(\boldsymbol{x}))$ costruita disponendo ordinatamente le derivate parziali seconde della funzione rispetto alle n variabili:

$$H\left(f\left(\boldsymbol{x}\right)\right) = \begin{bmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial^2 x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{bmatrix}$$

Le posizioni delle derivate parziali seconde $\partial^2 f \,/\, \partial x_i \partial x_j$ e $\partial^2 f \,/\, \partial x_j \partial x_i$ siano, rispettivamente, (i, j) e (j, i) per $i \neq j$. Ricordando che se é valido il teorema delle derivate miste allora $\partial^2 f \,/\, \partial x_i \partial x_j = \partial^2 f \,/\, \partial x_j \partial x_i$, questo significa che la matrice hessiana di una funzione per la quale il teorema é valido é una matrice simmetrica.

$$f(x, y) = x^2 + 16y^2 + 6xy \qquad \nabla f\left(x, y\right) = \begin{matrix} 2x \\ 32y \end{matrix} \qquad H(f(x, y)) = \begin{bmatrix} 2 & 0 \\ 0 & 32 \end{bmatrix}$$

Data una funzione $f(x)$, siano x_p e x_q due punti appartenenti all'intervallo $[a, b] \subseteq \mathrm{Dom}(f(x))$ tali per cui $x_p < x_q$. Sia poi λ un qualsiasi valore strettamente compreso fra 0 e 1. Se vale la disuguaglianza in basso a sinistra, la funzione $f(x)$ é detta **convessa nell'intervallo [a, b]**; se vale invece quella in basso a destra, é detta **concava nell'intervallo [a, b]**:

$$f(\lambda x_q + (1-\lambda)x_p) \leq \lambda f(x_q) + (1-\lambda) f(x_p) \qquad f(\lambda x_q + (1-\lambda)x_p) \geq \lambda f(x_q) + (1-\lambda) f(x_p)$$

Le definizioni di funzione concava e funzione convessa in un intervallo non sono mutualmente esclusive. Infatti, una funzione potrebbe non essere né concava né convessa in un intervallo cosí come essere sia concava sia convessa in un intervallo.
Piú in generale, una funzione viene semplicemente detta **funzione convessa** se é convessa su tutto il suo dominio; allo stesso modo, una funzione viene semplicemente detta **funzione concava** se é concava su tutto il suo dominio.

Criterio di concavitá e convessitá per funzioni ad una variabile. Una funzione ad una variabile due volte derivabile é convessa in un intervallo se la sua derivata seconda é nulla o negativa in tutti i punti di tale intervallo. Similmente, tale funzione é concava in un intervallo se la sua derivata seconda é nulla o positiva in tutti i punti di tale intervallo.

Si noti come la non esistenza della derivata seconda di una funzione ad una variabile in un intervallo non implichi l'impossibilitá di determinare in toto se questa sia concava o convessa in tale intervallo. Infatti, sebbene il criterio sopra presentato non sia applicabile, la definizione di concavitá/convessitá potrebbe essere comunque rispettata.
Sostituendo i punti x_p e x_q nelle definizioni di concavitá e convessitá con i vettori n-dimensionali $\boldsymbol{x}_p$ e $\boldsymbol{x}_q$, si estendono tali definizioni alle funzioni a piú di una variabile:

$$f(\lambda \boldsymbol{x}_q + (1-\lambda)\boldsymbol{x}_p) \leq \lambda f(\boldsymbol{x}_q) + (1-\lambda) f(\boldsymbol{x}_p) \qquad f(\lambda \boldsymbol{x}_q + (1-\lambda)\boldsymbol{x}_p) \leq \lambda f(\boldsymbol{x}_q) + (1-\lambda) f(\boldsymbol{x}_p)$$

Criterio di concavitá e convessitá per funzioni a piú variabili.
Sia data una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ per la quale esistono e sono definite tutte le derivate parziali seconde in un intervallo. Sia poi $H(f(\boldsymbol{x}))$ la matrice Hessiana a questa associata. Si ha allora:

- Se $H(f(\boldsymbol{x}))$ é definita positiva per tutti i punti che appartengono all'intervallo, allora f é convessa in tale intervallo;
- Se $H(f(\boldsymbol{x}))$ é definita negativa per tutti i punti che appartengono all'intervallo, allora f é concava in tale intervallo;
- Se f é convessa in tutti i punti che appartengono all'intervallo, allora $H(f(\boldsymbol{x}))$ é semidefinita positiva per tutti i punti dell'intervallo;
- Se f é concava in tutti i punti che appartengono all'intervallo, allora $H(f(\boldsymbol{x}))$ é semidefinita negativa per tutti i punti dell'intervallo.

Data la funzione $f(x, y) = x^2 - 2xy + y^2$, se ne calcoli la matrice Hessiana:

$$H = \begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial y \partial x} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}(x^2 - 2xy + y^2)\right) & \frac{\partial}{\partial y}\left(\frac{\partial}{\partial x}(x^2 - 2xy + y^2)\right) \\ \frac{\partial}{\partial x}\left(\frac{\partial}{\partial y}(x^2 - 2xy + y^2)\right) & \frac{\partial}{\partial y}\left(\frac{\partial}{\partial y}(x^2 - 2xy + y^2)\right) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x}(2x - 2y) & \frac{\partial}{\partial y}(2x - 2y) \\ \frac{\partial}{\partial x}(-2x + 2y) & \frac{\partial}{\partial y}(-2x + 2y) \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix}$$

Tale matrice é semidefinita positiva. Infatti:

$$z^T H z = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x - 2y & -2x + 2y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = (2x - 2y)x + (-2x + 2y)y = 2x^2 - 4xy + 2y^2 = 2(x - y)^2$$

Che é una quantitá non negativa per qualsiasi valore di x e di y. La funzione $f(x, y)$ é allora una funzione convessa.

Se $f(x_1, x_2, ..., x_n)$ é una funzione convessa, allora $-f(x_1, x_2, ..., x_n)$ é una funzione concava, e viceversa.

Dimostrazione. Sia $f(x_1, x_2, ..., x_n)$ una funzione convessa. Allora, per qualsiasi $\boldsymbol{x}_p, \boldsymbol{x}_q \in \mathrm{Dom}(f)$ e per qualsiasi $\lambda \in (0, 1)$, deve valere:

$$f(\lambda x_q + (1 - \lambda)x_p) \leq \lambda f(x_q) + (1 - \lambda) f(x_p)$$

Moltiplicando ambo i membri per -1, si ha:

$$-(f(\lambda x_q + (1 - \lambda)x_p)) \geq -(\lambda f(x_q) + (1 - \lambda) f(x_p)) \;\Rightarrow\; -f(\lambda x_q + (1 - \lambda)x_p) \geq \lambda(-f(x_q)) + (1 - \lambda)(-f(x_p))$$

Che é la definizione di funzione concava per $-f(x_1, x_2, ..., x_n)$. La dimostrazione in senso inverso é sostanzialmente analoga.

La somma di piú funzioni convesse é una funzione convessa; la somma di piú funzioni concave é una funzione concava.

I minimi locali di una funzione convessa sono anche minimi globali; I massimi locali di una funzione concava sono anche massimi globali.

Una funzione lineare é sempre sia concava sia convessa.

Dimostrazione. La derivata seconda di una funzione lineare generica ad una variabile $ax + b$ é sempre nulla:

$$\frac{d^2 f}{dx}(ax+b) = \frac{df}{dx}\left(\frac{df}{dx}(ax+b)\right) = \frac{df}{dx}\left(\frac{df}{dx}(ax) + \frac{df}{dx}(b)\right) = \frac{df}{dx}(a+0) = 0$$

Pertanto, per definizione, la sua derivata seconda é sempre sia positiva o nulla (quindi é convessa) e sempre sia negativa o nulla (quindi é concava).

Sia data una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$, sia $\boldsymbol{x}_0$ un suo punto critico. Sia poi $H(f(\boldsymbol{x}_0))$ la matrice Hessiana di f valutata in tale punto. Si ha allora:

- Se $H(f(\boldsymbol{x}_0))$ é definita positiva, allora $\boldsymbol{x}_0$ é un punto di minimo;
- Se $H(f(\boldsymbol{x}_0))$ é definita negativa, allora $\boldsymbol{x}_0$ é un punto di massimo;
- Se gli autovalori di $H(f(\boldsymbol{x}_0))$ hanno segni discordi e nessuno di questi é nullo, allora $\boldsymbol{x}_0$ é un punto sella.

Capitolo 2
Programmazione lineare

2.1 Modello di programmazione matematica

Sia data una funzione f, chiamata **funzione obiettivo**, nella forma $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$, ovvero avente un numero n di variabili $x_1, x_2, ..., x_n$, chiamate **variabili decisionali**, e restituente un numero reale (scalare). Qualsiasi n-pla di elementi che soddisfano f é detta **soluzione** di f.
Sia poi fornito un insieme X, sottoinsieme del dominio di una funzione obiettivo f (e quindi anche di $\mathbb{R}^n$), chiamato **regione ammissibile**. Una n-pla di elementi che soddisfa f e che appartiene a X é detta **soluzione ammissibile** per f.
A partire da una funzione obiettivo f e da una regione ammissibile X é possibile definire un **problema di ottimizzazione** come la ricerca della soluzione ammissibile (o delle soluzioni ammissibili) $\boldsymbol{x}^* \in X$ per f tale che sia "ottimale". Tale soluzione (o tali soluzioni) é detta **soluzione ottimale**:

$$\boldsymbol{x}^* = \{\boldsymbol{x} \in X \text{ t.c. } \text{opt}(f(\boldsymbol{x}))\}$$

La qualitá di una soluzione ammissibile generica dipende da quanto questa é "vicina" alla soluzione ottimale $\boldsymbol{x}^*$. Se una soluzione ammissibile $\boldsymbol{x}_A$ é piú "vicina" a $\boldsymbol{x}^*$ di quanto lo sia un'altra soluzione ammissibile $\boldsymbol{x}_B$, allora si dice che $\boldsymbol{x}_A$ é una soluzione *migliore* di $\boldsymbol{x}_B$, o semplicemente che $\boldsymbol{x}_A$ é *migliore* di $\boldsymbol{x}_B$.
In genere, vi sono due possibili definizioni di "soluzione ottimale": la soluzione piú piccola assumibile dalla funzione oppure la soluzione piú grande assumibile dalla funzione. Nel primo caso si parla di **problema di minimizzazione**, mentre nel secondo caso di **problema di massimizzazione**.

$$\boldsymbol{x}^* = \{\boldsymbol{x} \in X \text{ t.c. } \min(f(\boldsymbol{x}))\} \qquad \boldsymbol{x}^* = \{\boldsymbol{x} \in X \text{ t.c. } \max(f(\boldsymbol{x}))\}$$

La regione ammissibile é definita sulla base di un certo numero di *restrizioni* definite nel problema, chiamate **vincoli**. Nel caso in cui non vi sia alcuna restrizione, ovvero nel caso in cui la ricerca dei valori ottimali della funzione obiettivo venga condotta sui valori dell'intero dominio della funzione obiettivo, si parla di **ottimizzazione non vincolata**. Se invece almeno una restrizione esiste, si parla di **ottimizzazione vincolata**.
Quando i vincoli che definiscono la regione ammissibile di un problema di ottimizzazione sono espressi come un sistema di equazioni e disequazioni, tale problema prende il nome di **problema di programmazione matematica (PM)**. Un generico vincolo g_i viene allora espresso come una funzione $g_i : X \mapsto \mathbb{R}$ nella forma $g_i(\boldsymbol{x}) = 0$, $g_i(\boldsymbol{x}) \geq 0$ oppure $g_i(\boldsymbol{x}) \leq 0$. La regione ammissibile di un problema di programmazione matematica viene quindi ad essere l'insieme X composto da tutti i valori $\boldsymbol{x}$ per i quali sono contemporaneamente verificate tutte le m equazioni/disequazioni definite nei vincoli.

$$X = \left\{ \boldsymbol{x} \in \mathbb{R}^n \text{ t.c. } g_i(\boldsymbol{x}) \begin{Bmatrix} \geq \\ = \\ \leq \end{Bmatrix} 0 \ \forall i \in \left\{ 0, 1, ..., m \right\} \right\},$$

In merito alla natura della regione ammissibile, si distinguono quattro tipi di problemi:

- Quando l'insieme X coincide con l'insieme vuoto, si parla di **problema non ammissibile**. Il problema non ha alcuna soluzione, perché nessuna soluzione é ammissibile per X, e a maggior ragione non ha alcuna soluzione ottima;
- Quando qualsiasi valore puó fare parte di X, si parla di **problema illimitato**. Nello specifico, se il problema é un problema di minimizzazione si parla di **problema illimitato inferiormente**, mentre se é un problema di massimizzazione si parla di **problema illimitato superiormente**;
- Il caso piú generico é il **problema con piú di una soluzione ottima**, dove l'insieme X contiene due o piú soluzioni ottimali, aventi tutti ugual valore della funzione obiettivo.

Si consideri il seguente problema di programmazione matematica:

$$f = x_3 \qquad x^* = \max_{x_1,x_2,x_3}\{f\} \qquad \begin{cases} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \\ 0 \leq x_3 \leq 1 \\ x_1 + x_2 + x_3 - 2 = 0 \end{cases}$$

Si consideri il problema dal punto di vista dell'algebra lineare. Le prime tre disequazioni riportate nei vincoli specificano che le soluzioni ammissibili devono necessariamente trovarsi all'interno del cubo delimitato dai vertici:

$$(0,0,0)\ (0,0,1)\ (0,1,0)\ (1,0,0)\ (1,0,1)\ (0,1,1)\ (1,1,0)\ (1,1,1)$$

L'ultima disequazione specifica che l'insieme delle soluzioni ammissibili é dato dall'intersezione fra tale cubo ed il piano di equazione $x_1 + x_2 + x_3 - 2 = 0$. Le soluzioni ottimali sono tutti i punti di $\mathbb{R}^3$ che fanno parte di tale intersezione e hanno $x_3 = 1$.

2.2 Modello di programmazione lineare

Un problema di programmazione matematica dove figurano esclusivamente funzioni lineari, sia nella funzione obiettivo che nei vincoli, prende il nome di **problema di programmazione lineare**:

$$\min / \max Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \qquad \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq/\geq/= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq/\geq/= b_2 \\ \cdots + \cdots + \cdots + \cdots \leq/\geq/= \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq/\geq/= b_m \end{cases}$$

Z é la funzione obiettivo, composta da n variabili decisionali x_j, le cui soluzioni ammissibili vanno cercate nella regione ammissibile definita dagli m vincoli. I coefficienti c_j, a_{ij} e b_i costituiscono i **parametri** del problema. Nello specifico, i primi sono chiamati **coefficienti di costo**, i secondi **termini noti sinistri** ed i terzi **termini noti destri**.
Particolare rilevanza hanno i problemi di programmazione lineare detti in **forma standard**, ovvero quei problemi dove:

1. La funzione obiettivo deve essere massimizzata (sono problemi di ottimizzazione di massimo);
2. Ogni variabile x_i ha associato un vincolo del tipo $x_i \geq 0$, detto **vincolo di non negativitá**;
3. Tutti i vincoli restanti, detti **vincoli funzionali**, sono di tipo $\leq$.

$$\max Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \qquad \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ \cdots + \cdots + \cdots + \cdots \leq \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\ x_i \geq 0 \quad \forall i \in \{1,2,\ldots,n\} \end{cases}$$

É sempre possibile convertire un problema di programmazione lineare generico in un problema di programmazione lineare equivalente in forma standard. Pertanto é possibile assumere (se non specificato diversamente) che si stia sempre parlando di problemi di programmazione lineare in forma standard senza perdita di generalitá. Per effettuare la conversione é necessario rifarsi alle seguenti manipolazioni algebriche:

- Se il problema di programmazione lineare é un problema di minimizzazione, é sufficiente cambiare il segno della funzione obiettivo per convertirlo in un problema di massimizzazione.
- Se una variabile x_i che figura nella funzione obiettivo del problema di programmazione lineare non ha un vincolo di non negativitá associato, occorre riscriverla come una differenza del tipo $x_i = x_i^+ - x_i^-$, dove x_i^+ e x_i^- sono due variabili entrambe non negative.
- Se un vincolo funzionale del problema di programmazione lineare é del tipo $\geq$, é sufficiente cambiarne il segno per ottenere un vincolo nella forma $\leq$. Se il problema presenta invece un vincolo di uguaglianza, occorre riscrivere tale vincolo come una coppia di vincoli, uno nella forma $\geq$ (che viene poi cambiato di segno) ed uno nella forma $\leq$.

I problemi di programmazione lineare in forma standard sono preferibili perché é semplice darvi una interpretazione che non sia strettamente matematica. Ad esempio, é possibile interpretarli in chiave economica:

- Z é il valore della misura di prestazione;
- x_j é il livello dell'attivitá j;
- c_j é l'incremento del valore della misura di prestazione Z corrispondente all'incremento di un'unitá del valore dell'attivitá x_j;
- b_i é la quantitá di risorsa i allocabile alle attivitá $x_j, j = 1,2,\ldots,n$;

- a_{ij} é la quantita di risorsa i consumata da ogni unitá di attivitá $x_j, j = 1, 2, ..., n$;

I problemi di programmazione lineare hanno, oltre che una interpretazione economica, anche una interpretazione geometrica come elementi dello spazio n-dimensionale.
Un generico vincolo di disuguaglianza di un problema di programmazione lineare divide lo spazio n-dimensionale in due regioni: una i cui punti soddisfano il vincolo ed una in cui lo violano. Sostituendo la disuguaglianza con un'uguaglianza si ottiene l'equazione della **frontiera** associata a tale vincolo, che nello spazio n-dimensionale corrisponde ad un **iperpiano** a n dimensioni. Nel caso particolare in cui $n = 2$, tale iperpiano corrisponde ad una retta nel piano bidimensionale (piano cartesiano), mentre se si ha $n = 3$ tale iperpiano corrisponde ad un piano nello spazio tridimensionale.
Quando un vincolo é invece espresso come uguaglianza, solo i punti che si trovano sulla frontiera soddisfano il vincolo stesso. La frontiera della regione ammissibile contiene quelle soluzioni che soddisfano una o piú equazioni che definiscono la frontiera. Dal punto di vista geometrico, qualsiasi punto sulla frontiera della regione ammissibile si trova su uno o piú degli iperpiani definiti dalle corrispondenti equazioni.
Dal punto di vista geometrico, viene detto **vertice ammissibile** il punto di intersezione delle frontiere di n vincoli oppure, equivalentemente, una soluzione ammissibile che non é presente su nessun segmento nello spazio n-dimensionale che congiunge altre due soluzioni ammissibili. Nel caso particolare in cui $n = 2$, un vertice ammissibile corrisponde al punto di intersezione di due rette, mentre se si ha $n = 3$ tale vertice corrisponde al punto di intersezione di tre piani.
Dal punto di vista algebrico, un vertice ammissibile é dato dalla soluzione di un sistema di n equazioni, dove tali equazioni definiscono la frontiera della regione ammissibile. Si noti peró come non sia necessariamente vero il contrario: in generale, i vincoli di un problema di programmazione lineare sono costituiti da n vincoli di non negativitá e da m vincoli funzionali. Essendo $m + n \leq n$, possono esistere delle soluzioni di n frontiere che non rispettano tutti i vincoli del problema; soluzioni di questo tipo, che pur corrispondendo a dei vertici dello spazio n-dimensionale non sono soluzioni ammissibili, vengono anche dette **vertici non ammissibili**.
Dal punto di vista geometrico, uno **spigolo** della regione ammissibile é un segmento dato dall'intersezione delle frontiere di $n-1$ vincoli interamente costituito da punti che sono soluzioni ammissibili. Gli estremi di uno spigolo sono vertici ammissibili; se esiste uno spigolo che congiunge due vertici ammissibili si dice che tali vertici sono **adiacenti**.
Dal punto di vista algebrico, uno spigolo é dato dalla soluzione di un sistema di $n-1$ equazioni, dove tali equazioni definiscono parte della frontiera della regione ammissibile. Se due vertici ammissibili sono soluzioni (distinte) dello stesso sistema di $n-1$ equazioni, allora sono adiacenti.

Teorema fondamentale della programmazione lineare. (**I**) Se un problema di programmazione lineare ha una sola soluzione ottima, allora tale soluzione deve essere un vertice ammissibile. (**II**) Se un problema di programmazione lineare ha piú di una sola soluzione ottima, allora almeno due soluzioni ottime sono vertici ammissibili adiacenti.

- **Dimostrazione**. Si consideri un problema di programmazione lineare con una sola soluzione ottima x^*, avente corrispondente valore della funzione obiettivo Z^*. Si supponga per assurdo che tale soluzione non sia un vertice ammissibile: allora deve esistere un segmento della regione ammissibile che contiene x^*. Si indichino gli estremi di tale segmento con x' e x'' ed i corrispondenti valori della funzione obiettivo con Z_1 e Z_2. Come per ogni altro punto sul segmento che ha x' e x'' per estremi si ha, per un certo $\alpha \in (0, 1)$:

$$x^* = \alpha x'' + (1-\alpha)x' \qquad Z^* = \alpha Z_2 + (1-\alpha)Z_1$$

  Possono esistere solamente tre possibilitá: $Z^* = Z_1 = Z_2$, $Z_1 < Z^* < Z_2$ oppure $Z_2 < Z^* < Z_1$. La prima implica che il segmento che ha x' e x'' per estremi sia degenere, ovvero che anche x' e x'' siano soluzioni ottime, ma questo va in contraddizione con l'ipotesi di partenza secondo cui l'unica soluzione ottima é x^*. La seconda e la terza possibilitá sono entrambe contraddittorie, perché implicano che x^* non sia la soluzione ottima. Dato che tutte e tre le possibilitá sono contraddittorie, occorre assumere che se esiste una sola soluzione ottima allora tale soluzione deve essere un vertice ammissibile.

  Osservazione. L'enunciato opposto, ovvero che tutti i vertici ammissibili di un problema di programmazione lineare sono soluzioni ottime, non é necessariamente vero.
- **Dimostrazione**. Se esistono due vertici ammissibili adiacenti, tali vertici devono giacere sullo stesso iperpiano n-dimensionale. Di conseguenza, tutte le soluzioni ottime possono essere ottenute come medie pesate dei vertici ottimi.

Il teorema precedente implica che, al fine di cercare la soluzione ottima di un problema di programmazione lineare, non é necessario valutare tutti i punti della regione ammissibile (che sono infiniti e non numerabili), ma solamente i vertici. Questo é possibile solamente se il numero di vertici é un numero finito.

Il numero di vertici ammissibili in un problema di programmazione lineare é finito.

Dimostrazione. Ciascun vertice (non necessariamente ammissibile) é dato dalla soluzione di un sistema di n equazioni estratte dagli $m + n$ vincoli del problema. Il numero totale di vertici (non necessariamente ammissibili) é dato dal numero di combinazioni distinte di $m + n$ equazioni considerate n alla volta:

$$\binom{m+n}{n} = \frac{(m+n)!}{m!n!}$$

Essendo m e n due valori certamente finiti, anche $(m+n)!\,/\,m!n!$ deve essere un valore finito, ed essendo il numero di vertici ammissibili almeno pari a tale valore dev'essere finito a sua volta.

Il teorema precedente conferma che, essendo numerabile e finito il numero di vertici delle regioni ammissibili, é effettivamente possibile trovare le soluzioni ottime di un problema di programmazione lineare analizzando tutti i vertici ammissibili uno per uno fino a trovare quale fra questi é la soluzione migliore. Si noti peró come questo procedimento sia inapplicabile nella pratica, perché il valore di $(m+n)!\,/\,(m!n!)$ cresce estremamente velocemente con il crescere di m e di n.

In un problema di programmazione lineare, se un vertice ammissibile non ha alcun vertice adiacente che é soluzione migliore di esso, allora anche tutti gli altri vertici ammissibili non lo sono. Di conseguenza, se un vertice ammissibile non ha alcun vertice adiacente che é soluzione migliore di esso, allora tale vertice é una soluzione ottima.

Dimostrazione. L'insieme delle soluzioni che soddisfano un generico vincolo di programmazione lineare (sia esso di uguaglianza o di disuguaglianza) é un insieme convesso. Per un qualunque problema di programmazione lineare, la regione ammissibile é un poliedro convesso formato dall'intersezione di $(m + n)$ semispazi n-dimensionali, ciascuno rappresentante uno dei vincoli. Poiché l'intersezione di insiemi convessi é a sua volta un insieme convesso, la regione ammissibile é un insieme convesso.

Il teorema precedente restringe ulteriormente il numero di vertici ammissibili da analizzare per trovare la soluzione ottima di un problema di programmazione lineare. Infatti, appena viene individuato un vertice ammissibile che non ha alcun vertice adiacente che é soluzione migliore di esso, a quel punto si ha la certezza che tale vertice é la soluzione ottima e tutti gli altri vertici della regione ammissibile non ancora considerati possono essere scartati.
Le ipotesi del modello di programmazione lineare sono, per l'appunto, che la funzione obiettivo e i vincoli siano funzioni lineari. Tuttavia, affinché si possa assumere che tali funzioni siano effettivamente lineari, occorre che i dati e le attivitá del problema modellato rispettino determinate ipotesi.

- **Ipotesi di proporzionalitá.** Il contributo di ogni attivitá al valore della funzione obiettivo e alla quantitá a sinistra in ogni vincolo funzionale é proporzionale al livello dell'attivitá stessa. In altri termini, sia nella funzione obiettivo che nei vincoli, non possono comparire variabili elevate ad un qualsiasi esponente che non sia 1. Questa ipotesi é necessaria perché, se non fosse valida, le funzioni in esame non sarebbero lineari. In un contesto reale non sempre é possibile avere tale ipotesi soddisfatta, se non a meno di una certa approssimazione. Se tale approssimazione discosta troppo il modello dalla realtá in analisi, é preferibile utilizzare modelli di programmazione distinti dalla programmazione lineare.
- **Ipotesi di additivitá.** La funzione obiettivo e la quantitá a sinistra di un vincolo funzionale devono essere il risultato della somma dei contributi individuali delle rispettive attivitá. In altri termini, sia nella funzione obiettivo che nei vincoli, non possono comparire prodotti fra variabili. Come per la precedente, in un contesto reale questa ipotesi puó essere vera solamente a meno di un certo grado di approssimazione.
- **Ipotesi di divisibilitá.** Le variabili decisionali di un modello di programmazione lineare devono poter assumere valori reali o, al piú, razionali. Se in un contesto reale questa ipotesinon é soddisfatta, la programmazione lineare é quasi certamente un modello inapplicabile.
- **Ipotesi di certezza.** I valori assegnati ad ogni parametro che compare nel modello di programmazione lineare sono costanti note. In un contesto reale questa ipotesi non potrá mai essere valida a pieno perché, essendo i modelli di programmazione lineare formulati per pianificare azioni future, i valori dei parametri sono basati su previsioni, e questo introduce inevitabilmente un certo grado di incertezza.

2.3 Metodo del simplesso

Il **metodo del simplesso** é un algoritmo iterativo, basato sui principi enunciati nei precedenti teoremi, in grado di individuare la soluzione ottima di un problema di programmazione lineare in tempo (in media) lineare. L'algoritmo puó essere descritto in maniera informale come segue: si sceglie un vertice di partenza e si osservano i vertici a questo adiacenti. Se nessuno di questi vertici é soluzione migliore del vertice in esame, allora quest'ultimo é la soluzione ottima (teorema 3), altrimenti si ripete l'algoritmo considerando peró il vertice che, fra i vertici adiacenti a quello in esame, é la soluzione migliore.

Prima di applicare il metodo del simplesso ad un problema di programmazione lineare di questo tipo occorre convertire tutti i vincoli funzionali di disuguaglianza del problema in vincoli di uguaglianza equivalenti (i vincoli di non negativitá rimangono espressi come disuguaglianza, perché vanno trattati a parte). Questo puó essere fatto aggiungendo in maniera opportuna delle variabili extra, chiamate **variabili slack** (**scarto**).

Si consideri un problema di programmazione lineare avente per funzione obiettivo $Z = 3x_1 + 5x_2$. A sinistra sono riportati i vincoli del problema espressi cosí come sono stati forniti, a destra sono riportati vincoli equivalenti riscritti come equazioni e con aggiunte le variabili di slack.

$$x_1 \leq 4$$
$$2x_2 \leq 12$$
$$3x_1 + 2x_2 \leq 18$$

Con $x_1 \geq 0$ e $x_2 \geq 0$

$$x_1 + x_3 = 4$$
$$2x_2 + x_4 = 12$$
$$3x_1 + 2x_2 + x_5 = 18$$

Con $x_i \geq 0$ per $i \in \{1, 2, 3, 4, 5\}$

La nuova forma in cui i vincoli vengono espressi viene chiamata **forma aumentata**. Una **soluzione aumentata** (*augmented solution*) é una soluzione del problema nella forma originale a cui vengono aggiunte le variabili slack. Una **soluzione di base** (*basic solution*) é un vertice del problema nella forma originale a cui vengono aggiunti i corrispondenti valori delle variabili slack. Una **soluzione di base ammissibile** (*Basic Feasible Solution*, **BFS**) é un vertice ammissibile del problema nella forma originale a cui vengono aggiunti i corrispondenti valori delle variabili slack. Se una soluzione di base non é ammissibile, é una **soluzione di base non ammissibile**.

Si consideri il problema di programmazione lineare dell'esempio precedente: a partire dalla soluzione ammissibile $(0, 6)$ é possibile costruire la soluzione di base ammissibile $(0, 6, 4, 0, 6)$.

$$\begin{cases} x_1 + x_3 = 4 \\ 2x_2 + x_4 = 12 \\ 3x_1 + 2x_2 + x_5 = 18 \end{cases} \Rightarrow \begin{cases} 0 + x_3 = 4 \\ 2 \cdot 6 + x_4 = 12 \\ 3 \cdot 0 + 2 \cdot 6 + x_5 = 18 \end{cases} \Rightarrow \begin{cases} x_3 = 4 \\ 12 + x_4 = 12 \\ 12 + x_5 = 18 \end{cases} \Rightarrow \begin{cases} x_3 = 4 \\ x_4 = 0 \\ x_5 = 6 \end{cases}$$

Ogni vertice della regione ammissibile é dato da un sistema di n equazioni (i piani n-dimensionali che compongono la frontiera), le quali *definiscono* il vertice. Ogni vincolo di un problema di programmazione lineare ha una **variabile indicatrice** (*indicating variable*) che assicura, se il suo valore é zero, che l'equazione che definisce la frontiera di quel vincolo é soddisfatta dalla soluzione corrente. Questo significa che, nella forma aumentata del problema, ogni volta che un'equazione che definisce la frontiera di un vincolo é una delle equazioni che definiscono il vertice, una specifica variabile ha valore uguale a zero. Una variabile con queste caratteristiche é denominata **variabile non di base** per la soluzione di base corrispondente.
Un vincolo di non negativitá é una disequazione nella forma $x_i \geq 0$, e la relativa equazione che definisce la frontiera di tale vincolo é $x_i = 0$. É evidente come tale equazione sia nulla nel solo ed unico caso in cui la i-esima componente della soluzione é nulla a sua volta, pertanto la variabile indicatrice di un vincolo di non negativitá del tipo $x_i \geq 0$ é x_i.
Un vincolo funzionale é una disequazione nella forma $\sum_{i=1}^{n} a_{ij}x_j \leq b_i$, dove l'equazione che ne definisce la frontiera é $\sum_{i=1}^{n} a_{ij}x_j = b_i$. Nella forma aumentata, il vincolo funzionale viene riscritto come $\sum_{i=1}^{n} a_{ij}x_j + x_{n+1} = b_i \wedge x_{n+1} \geq 0$. Affinché un vertice possa trovarsi sull'iperpiano n-dimensionale definito da un vincolo, deve valere per esso l'equazione $\sum_{i=1}^{n} a_{ij}x_j = b_i$, che non é altro che l'equazione del vincolo in forma aumentata con x_{n+1} posto a 0. Questo significa che, per un vincolo funzionale, la relativa variabile di base é x_{n+1}, la variabile che viene introdotta per costruirne la forma aumentata.

In merito al problema di programmazione lineare dell'esempio precedente, alla soluzione $(4, 3)$ del problema é associata la soluzione aumentata $(4, 3, 0, 6, 0)$. Questo significa che tale vertice é definito dal vincolo $x_1 \leq 4$ e dal vincolo $3x_1 + 2x_2 \leq 18$.

Se la variabile di una soluzione non é una variabile non di base, allora é detta **variabile di base**. Il numero delle variabili di base é uguale al numero di vincoli funzionali, pertanto il numero delle variabili di base é uguale al numero delle variabili meno il numero dei vincoli funzionali. I valori delle variabili di base, anche detti semplicemente **base**, sono ottenuti ponendo a 0 i valori delle variabili non di base e risolvendo il sistema di equazioni risultante. Se tale sistema ha soluzione e tutti i valori non sono negativi, allora la soluzione risultante é una BFS. Una variabile di base é detta **degenere** per una BFS se ha valore pari a 0.
Indicando con n il numero complessivo di variabili e con m il numero di vincoli funzionali, il numero di variabili non di base é $n-m$. Si ha quindi che il numero totale di sistemi di equazioni che é possibile costruire ponendo a 0 le variabili non di base é dato dal numero di disposizioni di n elementi presi $n-m$ alla volta, ovvero $n! / (n-m)!m!$. Si noti peró come non tutte le combinazioni costruibili possono dare luogo a delle soluzioni, perché alcuni dei sistemi di equazioni possono essere sistemi impossibili.
Se due vertici della regione ammissibile di un problema di programmazione lineare sono adiacenti, allora anche le rispettive BFS lo sono nella forma aumentata del problema. Si noti come due vertici distinti nel piano n-dimensionale sono adiacenti se é possibile sovrapporre uno sull'altro

operando una singola traslazione, e questo richiede che i due vertici differiscano per una sola componente. Ricordando che almeno $n-m$ variabili di una BFS sono nulle (le variabili di base), si ha che due BFS sono adiacenti se hanno tutte le variabili non di base uguali tranne una.

Si consideri il problema dell'esempio precedente. Il numero di variabili del problema é 5, mentre il numero di vincoli funzionali (scritti come equazioni) é 3. Il numero di variabili non di base di ciascuna soluzione é quindi $5-3=2$, ed il numero totale di combinazioni possibili é dato da $5!\ /\ (5-3)!3!=10$. É quindi possibile costruire almeno 10 vertici distinti annullando 2 delle 5 variabili.

Vertice	**Soluzione di base**	**Variabili non di base**	**É ammissibile?**	**Vertici adiacenti**
(0, 0)	(0, 0, 4, 12, 18)	x_1, x_2	sí	(0, 6), (4, 0)
non esiste	non esiste	x_1, x_3	non esiste	non esiste
(0, 6)	(0, 6, 4, 0, 6)	x_1, x_4	sí	(0, 0), (2, 6)
(0, 9)	(0, 9, 4, -6, 0)	x_1, x_5	no	non é una BFS
(4, 0)	(4, 0, 0, 12, 6)	x_2, x_3	sí	(0, 0), (4, 3)
non esiste	non esiste	x_2, x_4	non esiste	non esiste
(6, 0)	(6, 0, -2, 12, 0)	x_2, x_5	no	non é una BFS
(4, 6)	(4, 6, 0, 0, -6)	x_3, x_4	no	non é una BFS
(4, 3)	(4, 3, 0, 6, 0)	x_3, x_5	sí	(2, 6), (4, 0)
(2, 6)	(2, 6, 2, 0, 0)	x_4, x_5	sí	(0, 6), (4, 3)

Tutti i vertici del problema, sia ammissibili che non ammissibili, sono stati riportati nella tabella precedente, insieme alla rispettiva soluzione di base. Si noti come delle 10 combinazioni, solamente 8 restituiscono delle soluzioni, perché ponendo $x_1 = x_3 = 0$ oppure ponendo $x_2 = x_4 = 0$ si ottiene un sistema di equazioni senza soluzione. Delle 8 soluzioni di base esistenti, solamente le 5 soluzioni che non contengono alcun valore negativo sono delle BFS. Per queste 5 soluzioni é possibile mettere in evidenza quali sono i vertici ad esse adiacenti osservando quali sono le rispettive variabili non di base.

Quando si affrontano problemi nella forma aumentata é conveniente considerare e manipolare allo stesso tempo l'equazione che definisce la funzione obiettivo e i nuovi vincoli funzionali. Questo puó essere fatto semplicemente riscrivendo $Z = \sum_{j=1}^{n} c_j$ come $Z - \sum_{j=1}^{n} a_{0j} = 0$. L'equazione cosí riscritta figura come fosse uno dei vincoli originari, dove non é peró necessario introdurre delle variabili slack perché il vincolo era giá espresso come uguaglianza. In questo modo, il numero totale di variabili aumenta di uno, ma anche il numero di vincoli aumenta di uno, pertanto é comunque possibile ricavare tutte le variabili delle soluzioni di base.

Le soluzioni di base del precedente problema possono essere ulteriormente estese introducendo il valore della variabile Z.

$$\begin{cases} Z - 3x_1 - 5x_2 = 0 \\ x_1 + x_3 = 4 \\ 2x_2 + x_4 = 12 \\ 3x_1 + 2x_2 + x_5 = 18 \end{cases}$$

$(0, 0, 0, 4, 12, 18)$ $(30, 0, 6, 4, 0, 6)$
$(40, 0, 9, 4, -6, 0)$ $(12, 4, 0, 0, 12, 6)$
$(18, 6, 0, -2, 12, 0)$ $(42, 4, 6, 0, 0, -6)$
$(27, 4, 3, 0, 6, 0)$ $(36, 2, 6, 2, 0, 0)$

Gli elementi finora introdotti sono sufficienti a descrivere formalmente il metodo del simplesso per la risoluzione di un problema di programmazione lineare:

1. Si riscriva, se necessario, il problema in forma standard.
2. Si riscriva il problema in forma aumentata, ottenendo un sistema di m equazioni in n variabili (contando anche Z).
3. Si scelgano $n-m$ variabili delle n a disposizione come variabili non di base per ottenere un vertice di partenza, che fungerá da soluzione attuale: se possibile, é conveniente scegliere il vertice nullo scegliendo come variabili non di base le variabili che compaiono con coefficiente non nullo nella funzione obiettivo.
4. Si riscriva il sistema di equazioni nella **forma canonica**, ovvero dove ciascuna equazione ha una variabile di base con coefficiente uguale ad 1, tutte le altre variabili di base con coefficiente uguale a 0 e tutte le variabili non di base hanno coefficiente arbitrario. Se la soluzione attuale é il vertice nullo, il sistema di equazioni é giá in questa forma, altrimenti é possibile ricondurre il sistema alla forma canonica applicandovi una o piú mosse di Gauss (considerando la matrice associata al sistema di equazioni).
5. Si impongano uguali a zero le variabili non di base della soluzione attuale e si calcolino i valori delle variabili di base risolvendo il sistema di m equazioni in m incognite che ne risulta.

6. Si operi un test di ottimalitá per valutare se la soluzione corrente é la migliore. La soluzione é da considerarsi ottimale se, nella funzione obiettivo riscritta come vincolo, compaiono solo valori non negativi, perché significa che non é possibile incrementarla ulteriormente senza uscire dalla regione ammissibile. Se la soluzione attuale é la soluzione ottima, l'algoritmo termina, altrimenti si procede oltre.
7. Si determini la variabile non di base, chiamata **variabile entrante**, che va scambiata con una delle variabili della funzione obiettivo. Tale variabile viene individuata scegliendo, nell'equazione (0), la variabile con il coefficiente negativo piú grande in valore assoluto, perché é la variabile che, a paritá di incremento, influenza maggiormente il valore di Z.
8. Si determini la variabile di base della funzione obiettivo, chiamata **variabile uscente**, che va scambiata con una delle altre variabili mediante il **test del minimo rapporto**. Tale test prevede di isolare, in tutte le equazioni, la variabile entrante ed il termine noto, imporre tale espressione maggiore o uguale di zero, spostare il termine noto a destra e dividere ambo i membri per il coefficiente della variabile entrante (se in una o piú equazioni la variabile entrante non compare, tale o tali equazioni possono venire ignorate). La variabile uscente é quella che, fra queste espressioni, compare con il termine noto piú piccolo, perché é quella che prima di tutte restituisce una soluzione al di fuori della regione ammissibile.
9. Riprendere il flusso d'esecuzione dal punto 4.

Si consideri il problema di programmazione lineare dell'esempio precedente. La prima iterazione del metodo del simplesso puó essere descritta come segue:

- Essendo (0, 0) una soluzione ammissibile nella forma standard del problema, é possibile usare il vertice nullo come vertice di partenza. Questo significa che il sistema di equazioni é giá nella forma canonica.
- Si imposta a 0 il valore delle variabili x_1 e x_2, di modo da ottenere la soluzione di partenza:

$$\begin{cases} Z-3\cdot 0-5\cdot 0=0 \\ 0+x_3=4 \\ 2\cdot 0+x_4=12 \\ 3\cdot 0+2\cdot 0+x_5=18 \end{cases} \Rightarrow \begin{cases} Z=0 \\ x_3=4 \\ x_4=12 \\ x_5=18 \end{cases} \Rightarrow \Big(0,0,0,4,12,18\Big)$$

- Si opera un test di ottimalitá su $(0, 0, 0, 4, 12, 18)$. I coefficienti di x_1 e x_2 sono positivi, pertanto é possibile inferire che $(0, 0, 0, 4, 12, 18)$ non sia la soluzione ottima, ma che ve ne sia una migliore.
- Il coefficiente di x_2 é maggiore del coefficiente di x_1, pertanto x_2 va scelta come variabile entrante.
- Si determini la variabile uscente mediante test del minimo rapporto:

$$\begin{cases} 0+x_3=4\geq 0 \\ 2\cdot 0+2x_2+x_4=12\geq 0 \\ 3\cdot 0+2x_2+x_5=18\geq 0 \end{cases} \Rightarrow \begin{cases} x_3=4\geq 0 \\ x_4=12-2x_2\geq 0 \\ x_5=18-2x_2\geq 0 \end{cases} \Rightarrow \begin{cases} \forall x_2 \\ x_2\leq 6 \\ x_2\leq 9 \end{cases}$$

  Il piú piccolo termine noto é 6, pertanto la variabile uscente é x_4.
- Si procede con una nuova iterazione, sostituendo x_4 con x_2 come variabile di base.

Non sempre la scelta della variabile entrante o uscente di una iterazione del metodo del simplesso é univoca. Possono infatti presentarsi situazioni in cui esistono nessuna o piú variabili candidate ad essere la variabile entrante o la variabile uscente:

- Se esiste piú di una variabile elegibile ad essere la variabile entrante é sufficiente sceglierne una qualsiasi, perché significa che il tasso di miglioramento della soluzione é lo stesso a prescindere dalla variabile scelta. L'unica differenza che puó verificarsi nello scegliere una variabile piuttosto che un'altra é il numero di iterazioni necessarie a trovare la soluzione ottima, ma questo non é comunque conoscibile a priori.
- Se non esiste alcuna variabile elegibile ad essere la variabile uscente significa che la funzione obiettivo puó essere aumentata all'infinito senza uscire dalla regione ammissibile, e quindi il problema termina in quanto Z é illimitata.
- Se esiste piú di una variabile elegibile ad essere la variabile uscente, potrebbe presentarsi una situazione problematica. Quando la variabile entrante viene aumentata, le variabili di base che sono elegibili a variabili uscenti si annullano contemporaneamente, pertanto la soluzione che ne risulta avrá delle variabili di base nulle (degeneri). Se una variabile degenere mantiene il proprio valore nullo sino all'iterazione successiva, dove viene selezionata come variabile uscente, la corrispondente variabile entrante deve rimanere nulla a sua volta (dato che non puó essere aumentata senza far assumere un valore negativo alla variabile uscente), pertanto il valore della funzione Z resta invariato. Se il valore di Z rimane costante anziché aumentare ad ogni iterazione, il metodo del simplesso potrebbe entrare in un loop infinito ripetendo la stessa sequenza di soluzioni senza mai raggiungere la soluzione ottimale. Si noti peró che una situazione di questo tipo é piuttosto inusuale, e puó pertanto non venire presa in considerazione. Inoltre, esistono strategie apposite in grado di rompere loop infiniti di questo tipo.

Raramente il metodo del simplesso viene applicato in forma discorsiva, perché é estremamente dispendiosa. In genere, ogni iterazione del metodo del simplesso viene riportata in una forma tabellare chiamata **tableau**, dove vengono riportate solamente le informazioni essenziali:

Iterazione	**Variabili di base**	**Equazione**	**Coefficiente di:**				**Termini noti**
			Z	x_1	…	x_n	
i	Z	(0)					
	x_1	(1)					
	…						
	x_m	(m)					

Per ciascuna iterazione sono riportati i coefficienti delle variabili, i termini noti e la variabile di base di ciascuna equazione. La variabile entrante, individuata come di consueto come il coefficiente negativo maggiore in valore assoluto della funzione obiettivo, determina una colonna della tabella, chiamata **colonna pivot**. La variabile uscente é determinata a partire dai valori che si ottengono dividendo, per ciascuna equazione, il termine noto per il coefficiente della variabile entrante; la variabile uscente é la variabile di base dell'equazione che ha il piú piccolo fra tali valori. La variabile uscente determina una riga della tabella, chiamata **riga pivot**. A partire da tale riga si modificano le altre righe della tabella mediante mosse di Gauss per effettuare il passaggio di variabile.

Iterazione	**Variabili di base**	**Equazione**	**Coefficiente di:**						**Termini noti**
			Z	x_1	x_2	x_3	x_4	x_5	
0	Z	(0)	1	-3	-5	0	0	0	0
	x_3	(1)	0	1	0	1	0	0	4
	x_4	(2)	0	0	2	0	1	0	12
	x_5	(3)	0	3	2	0	0	1	18
1	Z	(0)	1	-3	0	0	$\frac{5}{2}$	0	30
	x_3	(1)	0	1	0	1	0	0	4
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_5	(3)	0	3	0	0	-1	1	6
2	Z	(0)	1	0	0	0	$\frac{3}{2}$	1	36
	x_3	(1)	0	0	0	1	$\frac{1}{3}$	$-\frac{1}{3}$	2
	x_2	(2)	0	0	1	0	$\frac{1}{2}$	0	6
	x_1	(3)	0	1	0	0	$-\frac{1}{3}$	$\frac{1}{3}$	2

Il metodo del simplesso termina quando viene trovata una BFS, ma questo non significa che non possano esisterne altre. Per il terzo teorema, se un problema di programmazione lineare ha piú di una soluzione ottima allora tali soluzioni sono vertici adiacenti. In particolare, qualsiasi **combinazione convessa** di tali vertici é un punto sul segmento che li unisce, ed é pertanto una soluzione ottima a sua volta. Una combinazione convessa é un particolare tipo di combinazione lineare dove, indicando con a_i l'i-esimo coefficiente, valgono $\sum_{i=1}^{n} a_i = 1$ e $a_i \geq 0 \forall i$.
Vi sono situazioni dove, a prescindere dai vincoli, si richiede di scegliere fra soluzioni ottime alternative. Ogni qual volta un problema di programmazione lineare ha piú di una BFS ottima, almeno una delle variabili non di base ha un coefficiente nullo nella funzione obiettivo (scritta come vincolo) nell'ultima iterazione, perché tali incognite sono incognite libere nel sistema di equazioni. Dato che cambiare il valore di queste variabili non influisce sul valore di Z, eventuali soluzioni aggiuntive possono essere trovate eseguendo ulteriori iterazioni del metodo del simplesso scegliendo come variabile entrante una variabile non di base con coefficiente nullo.
Dalla rappresentazione tabellare del metodo del simplesso deriva una terza rappresentazione, quella **matriciale**. Sia $Z = \sum_{j=1}^{n} c_j$ la funzione obiettivo. Il risultato di tale espressione puó essere visto come il prodotto riga per colonna di due matrici, una matrice colonna $\boldsymbol{x}$ che contiene le variabili che compaiono nella funzione ed una matrice riga $\boldsymbol{c}$ che ne contiene i relativi coefficienti.

$$Z = \boldsymbol{cx} = [c_1 \;\; c_2 \;\; \cdots \;\; c_n] \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix}$$

Similmente, i vincoli funzionali di un problema di programmazione lineare sono, di fatto, un sistema di m equazioni in n incognite. Pertanto, é possibile rappresentarli mediante la relativa matrice incompleta. Lo stesso puó essere fatto per i vincoli di non negativitá.

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} \;\Rightarrow\; \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \leq \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix} \qquad \boldsymbol{x} \geq 0 \;\Rightarrow\; \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ \cdots \\ 0 \end{bmatrix}$$

Per ottenere la forma aumentata del problema, la matrice colonna $\boldsymbol{x}$ deve venire unita ad un'altra matrice colonna $\boldsymbol{x}_s$, costituita dalle variabili slack. Inoltre, la matrice $\boldsymbol{A}$ deve venire estesa con la matrice identitá di dimensione $m \times n$, che rappresenta i coefficienti (iniziali) delle suddette variabili.

$$[\boldsymbol{A}, \boldsymbol{I}] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} = \boldsymbol{b} \;\Rightarrow\; \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & 1 & 1 & \dots & 1 \\ a_{21} & a_{22} & \cdots & a_{2n} & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots & 1 & 1 & \dots & 1 \\ a_{m1} & a_{m2} & \cdots & a_{mn} & 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \dots \\ x_n \\ x_{n+1} \\ \dots \\ x_{n+m} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{bmatrix} \qquad \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} \geq 0 \;\Rightarrow\; \begin{bmatrix} x_1 \\ \dots \\ x_n \\ x_{n+1} \\ \dots \\ x_{n+m} \end{bmatrix} \geq \begin{bmatrix} 0 \\ \dots \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix}$$

Per completare una iterazione, é necessario scegliere n variabili non di base fra le $n + m$ variabili a disposizione, porle uguali a 0 e risolvere il sistema di m equazioni di m incognite che ne risulta. Nel contesto della rappresentazione matriciale, questo equivale a risolvere l'equazione:

$$\boldsymbol{B}\boldsymbol{x}_B = \boldsymbol{b} \;\Rightarrow\; \begin{bmatrix} B_{11} & B_{12} & \cdots & B_{1m} \\ B_{21} & B_{22} & \cdots & B_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ B_{m1} & B_{m2} & \cdots & B_{mm} \end{bmatrix} \begin{bmatrix} x_{B1} \\ x_{B2} \\ \cdots \\ x_{Bm} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \cdots \\ b_m \end{bmatrix}$$

Dove $\boldsymbol{x}_B$ é la matrice colonna ottenuta eliminando le variabili non di base dalla matrice $\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix}$ e la matrice $\boldsymbol{B}$, detta **matrice di base**, é la matrice ottenuta eliminando le colonne che corrispondono alle variabili non di base dalla matrice $[\boldsymbol{A}, \boldsymbol{I}]$.
Spostando a destra la matrice $\boldsymbol{B}$, é possibile esprimere l'equazione rispetto a $\boldsymbol{x}_B$:

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$$

Ammesso che la matrice $\boldsymbol{B}^{-1}$ esista, ovvero che $\boldsymbol{B}$ sia una matrice singolare. Indicando con $\boldsymbol{c}_B$ la matrice riga che contiene i coefficienti della funzione obiettivo per i corrispondenti elementi di $\boldsymbol{x}_B$ e sostituendo la precedente equazione nell'equazione per Z, si ha:

$$Z = \boldsymbol{c}_B\boldsymbol{x}_B = \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b}$$

Rimane ora da mostrare come ottenere l'insieme delle equazioni che compaiono nel tableau ad ogni iterazione del metodo del simplesso. Per ottenere l'insieme di equazioni per l'iterazione iniziale é sufficiente includere Z come variabile, riscrivendo la funzione obiettivo come equazione associata ad un vincolo. Come giá visto, la variabile Z ha coefficiente 1 nella prima equazione e 0 in tutte le altre (dato che compare solamente nella funzione obiettivo, per la quale funge da variabile di base). Si ha allora:

$$\begin{bmatrix} 1 & -\boldsymbol{c} & 0 \\ 0 & \boldsymbol{A} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} Z \\ \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{b} \end{bmatrix}$$

Le mosse di Gauss eseguite sulle equazioni durante ciascuna iterazione del metodo del simplesso per effettuare il cambio di variabile si traducono nella forma matriciale pre-moltiplicando entrambi i lati dell'insieme di equazioni originali per una matrice opportuna. Questa matrice avrá gli stessi elementi della matrice identitá eccetto che ciascuna costante moltiplicativa necessaria per una operazione algebrica verrá inserita nella posizione opportuna cosí che la pre-moltiplicazione per questa matrice produca l'operazione richiesta.
Dopo ogni iterazione, $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$ e $Z = \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b}$ ed i termini noti del nuovo sistema di equazioni diventano:

$$\begin{bmatrix} Z \\ \boldsymbol{x}_B \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{c}_B\boldsymbol{B}^{-1} \\ 0 & \boldsymbol{B}^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b} \\ \boldsymbol{B}^{-1}\boldsymbol{b} \end{bmatrix}$$

Poiché viene eseguita la stessa serie di operazioni aritmetiche su entrambi i termini dell'insieme originale di equazioni, la stessa matrice, che pre-moltiplica il lato destro originale viene usata per pre-moltiplicare il lato sinistro originale. Di conseguenza, dato che:

$$\begin{bmatrix} 1 & \boldsymbol{c}_B\boldsymbol{B}^{-1} \\ 0 & \boldsymbol{B}^{-1} \end{bmatrix} \begin{bmatrix} 1 & -\boldsymbol{c} & 0 \\ 0 & \boldsymbol{A} & \boldsymbol{I} \end{bmatrix} = \begin{bmatrix} 1 & \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{A}-\boldsymbol{c} & \boldsymbol{c}_B\boldsymbol{B}^{-1} \\ 0 & \boldsymbol{B}^{-1}\boldsymbol{A} & \boldsymbol{B}^{-1} \end{bmatrix}$$

La forma della matriciale dell'insieme delle equazioni dopo ogni iterazione é

$$\begin{bmatrix} 1 & \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{A}-\boldsymbol{c} & \boldsymbol{c}_B\boldsymbol{B}^{-1} \\ 0 & \boldsymbol{B}^{-1}\boldsymbol{A} & \boldsymbol{B}^{-1} \end{bmatrix} \begin{bmatrix} Z \\ \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b} \\ \boldsymbol{B}^{-1}\boldsymbol{b} \end{bmatrix}$$

Questa non é altro che una forma piú generale per il sistema di equazioni che compare nella i-esima equazione del metodo del simplesso. Imponendo infatti $\boldsymbol{B}^{-1} = \boldsymbol{I} = \boldsymbol{B}$ e $\boldsymbol{c}_B = 0$, si riottiene l'equazione per l'iterazione iniziale:

$$\begin{bmatrix} 1 & 0\boldsymbol{I}\boldsymbol{A}-\boldsymbol{c} & 0\boldsymbol{I} \\ 0 & \boldsymbol{I}\boldsymbol{A} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} Z \\ \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} 0\boldsymbol{I}\boldsymbol{b} \\ \boldsymbol{I}\boldsymbol{b} \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & -\boldsymbol{c} & 0 \\ 0 & \boldsymbol{A} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} Z \\ \boldsymbol{x} \\ \boldsymbol{x}_s \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{b} \end{bmatrix}$$

La relazione fra notazione matriciale e notazione a tableau é riassunta nella seguente tabella:

Iterazione	**Variabili di base**	**Equazione**	**Coefficiente di:**			**Termine noto**
			Z	**Variabili originali**	**Variabili slack**	
0	Z	(0)	1	$-\boldsymbol{c}$	0	0
	$\boldsymbol{x}_B$	$(1, 2, ..., m)$	0	$\boldsymbol{A}$	$\boldsymbol{I}$	$\boldsymbol{b}$
i	Z	(0)	1	$\boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{A}-\boldsymbol{c}$	$\boldsymbol{c}_B\boldsymbol{B}^{-1}$	$\boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b}$
	$\boldsymbol{x}_B$	$(1, 2, ..., m)$	0	$\boldsymbol{B}^{-1}\boldsymbol{A}$	$\boldsymbol{B}^{-1}$	$\boldsymbol{B}^{-1}\boldsymbol{b}$

Dalle relazioni presentate si deduce come sia possibile calcolare i valori del sistema di equazioni associato alla i-esima iterazione del metodo del simplesso ricalcolando solamente $\boldsymbol{B}$ e $\boldsymbol{c}_B$ ad ogni iterazione, dato che i valori di $\boldsymbol{A}$, $-\boldsymbol{c}$ e $\boldsymbol{b}$ sono calcolati alla prima iterazione e rimangono gli stessi in tutte le successive.
La riga 0 del tableau iniziale é $\boldsymbol{t} = [-\boldsymbol{c} \;\; 0 \;\; 0]$, mentre le altre rige sono $\boldsymbol{T} = [\boldsymbol{A} \;\; \boldsymbol{I} \;\; \boldsymbol{b}]$. Dopo ogni iterazione, i coefficienti delle variabili slack nell'i-esimo tableau diventano $\boldsymbol{c}_B\boldsymbol{B}^{-1}$ per la riga 0 e $\boldsymbol{B}^{-1}$ per le altre righe. Dopo ogni iterazione, le righe del tableau divengono:

$$\text{Riga } 0: \; [-\boldsymbol{c} \;\; 0 \;\; 0] + \boldsymbol{c}_B\boldsymbol{B}^{-1}[\boldsymbol{A} \;\; \boldsymbol{I} \;\; \boldsymbol{b}] \qquad \text{Riga } (1, ..., m): \; [-\boldsymbol{c} \;\; 0 \;\; 0] + \boldsymbol{c}_B\boldsymbol{B}^{-1}[\boldsymbol{A} \;\; \boldsymbol{I} \;\; \boldsymbol{b}]$$

La matrice $\boldsymbol{B}$ e la matrice $\boldsymbol{c}_B$ cambiano ad ogni iterazione, e con la notazione finora presentata non é possibile distinguere fra l'espressione matriciale relativa all'iterazione corrente ed una iterazione generica. Si indichi allora con $\boldsymbol{B}$ la matrice di base specifica per l'ultima iterazione, quella che restituisce il valore ottimo. Siano allora:

- $\boldsymbol{S}^* = \boldsymbol{B}^{-1}$ la matrice che contiene i coefficienti delle variabili slack nelle righe $1, ..., m$ dell'ultima iterazione;
- $\boldsymbol{A}^* = \boldsymbol{B}^{-1}\boldsymbol{A}$ la matrice che contiene i coefficienti delle variabili originali nelle righe $1, ..., m$ dell'ultima iterazione;
- $\boldsymbol{y}^* = \boldsymbol{c}_B\boldsymbol{B}^{-1}$ la matrice che contiene i coefficienti delle variabili slack nella riga 0 dell'ultima iterazione;
- $\boldsymbol{z}^* = \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{A}$, quindi $\boldsymbol{z}^*-\boldsymbol{c}$ é la matrice che contiene i coefficienti delle variabili originali nella riga 0 dell'ultima iterazione;
- $Z^* = \boldsymbol{c}_B\boldsymbol{B}^{-1}\boldsymbol{b}$ il valore ottimale della funzione obiettivo;
- $\boldsymbol{b}^* = \boldsymbol{B}^{-1}\boldsymbol{b}$ la matrice che contiene i termini noti delle righe $1, ..., m$ dell'ultima iterazione;

Per ottenere i valori che compaiono nel tableau finale, é sufficiente conoscere $\boldsymbol{t}, \boldsymbol{T}, \boldsymbol{y}^*$ e $\boldsymbol{S}^*$. Si ha infatti:

$$\begin{cases} \boldsymbol{t}^* = [-\boldsymbol{c} \;\; 0 \;\; 0] + \boldsymbol{c}_B\boldsymbol{B}^{-1}[\boldsymbol{A} \;\; \boldsymbol{I} \;\; \boldsymbol{b}] = \boldsymbol{t} + \boldsymbol{y}^*\boldsymbol{T} = [\boldsymbol{y}^*\boldsymbol{A}-\boldsymbol{c} \;\; \boldsymbol{y}^* \;\; \boldsymbol{y}^*\boldsymbol{b}] \\ \boldsymbol{T}^* = \boldsymbol{B}^{-1}[\boldsymbol{A} \;\; \boldsymbol{I} \;\; \boldsymbol{b}] = \boldsymbol{S}^*\boldsymbol{T} = [\boldsymbol{S}^*\boldsymbol{A} \;\; \boldsymbol{S}^* \;\; \boldsymbol{S}^*\boldsymbol{b}] \end{cases}$$

Si consideri il problema precedente e le matrici a questo associate:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 3 & 2 \end{bmatrix} \qquad \boldsymbol{b} = \begin{bmatrix} 4 \\ 12 \\ 18 \end{bmatrix} \qquad -\boldsymbol{c} = \begin{bmatrix} -3 & -5 \end{bmatrix}$$

Le matrici relative all'ultima iterazione del metodo del simplesso sono:

$$\boldsymbol{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 2 & 3 \end{bmatrix} \qquad \boldsymbol{B}^{-1} = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \qquad \boldsymbol{c}_B = \begin{bmatrix} 0 & 5 & 3 \end{bmatrix}$$

Entrambe le uguaglianze sono soddisfatte:

$$\boldsymbol{t}^* = \boldsymbol{t} + \boldsymbol{y}^*\boldsymbol{T} \Rightarrow \begin{bmatrix} 0 & 0 & 0 & \frac{3}{2} & 1 & 36 \end{bmatrix} = \begin{bmatrix} -3 & -5 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 5 & 3 \end{bmatrix} \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 4 \\ 0 & 2 & 0 & 1 & 0 & 12 \\ 3 & 2 & 0 & 0 & 1 & 18 \end{bmatrix}$$

$$\boldsymbol{T}^* = \boldsymbol{S}^*\boldsymbol{T} \Rightarrow \begin{bmatrix} 0 & 0 & 1 & \frac{1}{3} & -\frac{1}{3} & 2 \\ 0 & 1 & 0 & \frac{1}{2} & 0 & 6 \\ 1 & 0 & 0 & -\frac{1}{3} & \frac{1}{3} & 2 \end{bmatrix} = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & 0 \\ 0 & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 4 \\ 0 & 2 & 0 & 1 & 0 & 12 \\ 3 & 2 & 0 & 0 & 1 & 18 \end{bmatrix}$$

2.4 Teoria della dualitá

Si consideri un problema di programmazione lineare espresso in forma standard (ottimizzazione di massimo, vincoli funzionali di tipo $\leq$, vincoli di non negativitá per ciascuna variabile). A ciascun problema di questo tipo, che in questo contesto prende il nome di **problema primale**, é possibile associare un problema "gemello" , chiamato **problema duale**, con il quale é fortemente correlato. Lo studio della correlazione fra problemi primali e problemi duali prende il nome di **teoria della dualitá**.
La struttura di un generico problema primale e quella del rispettivo problema duale é presentata di seguito. Si noti come quasi tutti gli elementi di un problema siano presenti nell'altro, ma con un ruolo o in una forma diversa:

$$\max Z = \sum_{j=1}^{n} c_j \cdot x_j$$

$$\sum_{j=1}^{n} a_{ij} \cdot x_j \leq b_i \qquad i = 1, 2, \ldots, m$$

$$x_j \geq 0 \qquad j = 1, 2, \ldots, n$$

$$\min W = \sum_{i=1}^{m} b_i \cdot y_i$$

$$\sum_{i=1}^{m} a_{ij} \cdot y_i \geq c_j \qquad j = 1, 2, \ldots, n$$

$$y_i \geq 0 \qquad i = 1, 2, \ldots, m$$

1. Il problema primale é un problema di ottimizzazione di massimo, mentre il problema duale é un problema di ottimizzazione di minimo;
2. I coefficienti della funzione obiettivo del problema primale figurano come termini destri dei vincoli funzionali del problema duale;
3. I termini destri dei vincoli funzionali del problema primale figurano come coefficienti della funzione obiettivo del problema duale;
4. I vincoli funzionali del problema primale sono di tipo $\leq$, mentre i vincoli funzionali del problema duale sono di tipo $\geq$;
5. Sia nel problema primale che nel problema duale é presente un vincolo di non negativitá per ciascuna variabile.

La rappresentazione tabellare ben si presta a mostrare esplicitamente la relazione fra problema primale e problema duale. Tale notazione, estesa al descrivere la relazione fra i due, prende il nome di **tabella primale-duale**.

			PROBLEMA PRIMALE					
			coefficiente di				**termine noto**	
			x_1	x_2	…	x_n		
PROBLEMA DUALE	**coefficiente di**	y_1	a_{11}	a_{12}	…	a_{1n}	b_1	**Coefficienti della funzione obiettivo**
		y_2	a_{21}	a_{22}	…	a_{2n}	b_2	
		…	…	…	…	…	…	
		y_m	a_{m1}	a_{m2}	…	a_{mn}	b_m	
	termine noto		c_1	c_2	…	c_n		
			Coefficienti della funzione obiettivo					

I problemi di programmazione lineare possono essere interpretati in termini di allocazione di risorse ad attivitá. In particolare, quando i vincoli funzionali sono nella forma $\leq$, i termini noti a destra della disuguaglianza possono essere interpretati come le quantitá delle varie risorse che sono disponibili per le attivitá in esame. In genere, queste quantitá non sono veritá assolute, ma derivano da delle scelte manageriali; questo significa che, se é vantaggioso farlo, tali quantitá possono essere aumentate o diminuite.

I termini noti che si trovano a destra nei vincoli funzionali influenzano il valore della funzione obiettivo, pertanto per poter determinare se é vantaggioso aumentarli o diminuirli é necessario avere a disposizione una misura di *quanto* una variazione nel valore di tali termini noti influenza il valore della funzione obiettivo, ovvero del **valore marginale** della risorsa relativa a tale termine noto.

Per ciascuna risorsa i, la misura del suo valore marginale é data dal cosiddetto **prezzo ombra**, indicato con y^*_i. Il valore di y^*_i puó venire ricavato facilmente a partire dai tableau del problema di programmazione lineare di riferimento, dato che corrisponde al coefficiente della i-esima variabile slack nella riga 0 del tableau finale.

Se il prezzo ombra y^*_i é un valore positivo, allora significa che incrementando la risorsa i di un valore infinitesimo si ha effettivamente un incremento pari a y^*_i del valore della funzione obiettivo. Questo accade se l'i-esimo vincolo é soddisfatto dalla soluzione ottima non solo come disuguaglianza, ma anche come uguaglianza; vincoli con queste caratteristiche vengono detti **vincoli attivi**. Una situazione di questo tipo viene interpretata come una scarsa disponibilitá della risorsa i, dato che se un suo piccolo cambiamento comporta un incremento del valore della funzione obiettivo questo significa che tale risorsa viene interamente consumata. Risorse di questo tipo vengono chiamate **scarce goods** (*risorse rare*).

Se il prezzo ombra y^*_i é nullo, allora significa che incrementando la risorsa i di un valore infinitesimo non si ha alcun incremento del valore della funzione obiettivo. Questo accade se l'i-esimo vincolo é soddisfatto dalla soluzione come disuguaglianza, ma non come uguaglianza. Una situazione di questo tipo viene interpretata come una sovrabbondanza della risorsa i, dato che se un suo piccolo cambiamento non comporta alcun incremento del valore della funzione obiettivo questo significa che tale risorsa non verrá mai interamente consumata. Risorse di questo tipo vengono chiamate **free goods** (*risorse abbonanti*).

Uno dei principali obiettivi dell'analisi della sensitivitá é quello di analizzare i **parametri sensibili**, ovvero quelli che, se modificati, comportano un cambiamento della soluzione ottima. I parametri sensibili sono i parametri che necessitano di essere stimati con piú attenzione, e quelli che devono venire monitorati con maggiore attenzione.

Per quanto riguarda i termini noti dei vincoli funzionali, per determinare se questi siano oppure non siano parametri sensibili é possibile rifarsi ai prezzi ombra. Infatti, per definizione, se il prezzo ombra associato ad una risorsa é un valore positivo, allora una variazione di tale risorsa comporta una variazione del valore della funzione obiettivo, mentre se il prezzo ombra é nullo il valore della funzione obiettivo rimane costante. Questo comporta che i termini noti dei vincoli funzionali che hanno associato un prezzo ombra positivo, specialmente se tale prezzo ombra é grande in modulo, sono da considerarsi parametri sensibili.

I coefficienti di un vincolo funzionale sono da considerarsi parametri sensibili se tale vincolo é un vincolo attivo per la soluzione ottima perché, per definizione, se un vincolo é un vincolo attivo allora la risorsa a questo associata é uno *scarce good*. Un ragionamento simile puó essere fatto per determinare se i coefficienti della funzione obiettivo siano da considerarsi parametri sensibili.

Si noti come, nelle analisi reali, si tende a concentrarsi sul determinare se i termini noti dei vincoli funzionali ed i coefficienti della funzione obiettivo siano da considerarsi parametri sensibili, e raramente si fa lo stesso con i coefficienti dei vincoli funzionali. Questo perché i vincoli funzionali dei problemi reali sono in genere moltissimi, pertanto una variazione di uno solo di questi ha difficilmente una influenza significativa sulla soluzione ottima.

Ci si chiede allora da dove provenga il problema duale e la sua struttura. Si consideri la notazione matriciale introdotta in precedenza per il metodo del simplesso: in una generica iterazione del problema primale, i coefficienti per le prime n variabili sono dati dalla matrice $\boldsymbol{z}-\boldsymbol{c}$, mentre quelli per le restanti m variabili sono dati dalla matrice $\boldsymbol{y}$ (nell'iterazione iniziale, $\boldsymbol{y}$ e $\boldsymbol{c}$ sono entrambe matrici nulle). Si ricordi inoltre che sono state validate le seguenti espressioni:

$$W = \boldsymbol{yb} \qquad \boldsymbol{z} = \boldsymbol{yA} \;\Rightarrow\; z_j = \sum_{i=1}^{m} a_{ij} y_i$$

Dove, dato che si sta considerando il problema duale, W prende il posto di Z nell'indicare il valore della funzione obiettivo. Nello specifico, la prima equazione funzione la funzione obiettivo per il problema duale, mentre la seconda equazione fornisce i termini a sinistra dei vincoli funzionali per il problema duale. Quindi sottraendo i termini noti di questi vincoli del tipo $\geq$, la quantitá $z_j - c_j$ puó essere interpretate come variabile surprlus per il j-esimo vincolo funzionale.

Occorre a questo punto approfondire a cosa tende il metodo del simplesso in termini di queste variabili introdotte. In particolare, il metodo del simplesso cerca un insieme di variabili di base e la corrispondente BFS tali che tutti i coefficienti della riga 0 siano non negativi; una volta ottenuta una soluzione di questo tipo (che é ottima), la procedura termina. In altre parole, l'obiettivo perseguito dal metodo del simplesso, la sua *condzione di ottimalitá*, puó essere espressa come:

$$z_j - c_j \geq 0 \text{ per } j = 1, 2, \ldots, n \qquad y_i \geq 0 \text{ per } i = 1, 2, \ldots, m$$

Sostituendo l'espressione per z_j nell'equazione precedente, la condizione di ottimalitá afferma che il metodo del simplesso puó essere interpretato come la ricerca di quei valori $y_1, \ldots, y_m$ tali per cui:

$$W = \sum_{i=1}^{m} b_i y_i \quad \text{soggetto ai vincoli} \quad \begin{cases} \sum_{i=1}^{m} a_{ij} y_i \geq c_j & j = 1, 2, \ldots, n \\ y_i \geq 0 & i = 1, 2, \ldots, m \end{cases}$$

Che, ad eccezione della mancata indicazione se la quantitá W debba essere minimizzata o massimizzata, é esattamente nella stessa forma del problema duale.

Il motivo per cui, in questo nuovo problema, W vada minimizzato anziché massimizzato (come si potrebbe pensare, dato che W^* e Z^* coincidono) é da cercarsi nel fatto che le uniche soluzioni ammissibili per questo problema sono quelle che soddisfano la condizione di ottimalitá per il problema primale. Quindi, solo una soluzione ottima per il problema primale costituisce una soluzione ammissibile per questo nuovo problema. Di conseguenza, il valore ottimo di Z nel problema primale é il minimo valore ammissibile di W nel nuovo problema duale, e cosí W deve essere minimizzato. Introdurre tale informazione nel problema appena presentato permette di ricostruire il problema duale completo.

Di conseguenza, il problema duale puó essere visto come una riaffermazione in chiave di programmazione lineare dell'obiettivo del metodo del simplesso, vale a dire, ottenere una soluzione per il problema primale che soddisfi le condizioni di ottimalitá. Prima che questo obiettivo sia raggiunto, il corrispondente vettore $\boldsymbol{y}$ della riga 0 (i coefficienti delle variabili slack) del tableau corrente non risulta ammissibile per il problema duale. Tuttavia, dopo che l'obiettivo viene raggiunto, il corrispondente vettore $\boldsymbol{y}$ é una soluzione ottima (e viene pertanto indicato con $\boldsymbol{y}^*$) per il problema duale, perché costituisce una soluzione ammissibile con il valore minimo possibile per W. Tale soluzione ottima $(y_1, y_2, \ldots, y_m)$ fornisce i prezzi ombra del problema primale.

Proprietá di dualitá forte: se $\boldsymbol{x}^*$ é una soluzione ottima per un problema di programmazione lineare in forma standard e $\boldsymbol{y}^*$ é una soluzione ottima per il rispettivo problema duale, allora vale $\boldsymbol{cx}^* = \boldsymbol{y}^*\boldsymbol{b}$.

Proprietá di dualitá debole: se $\boldsymbol{x}$ é una soluzione ammissibile (non necessariamente ottima) per un problema di programmazione lineare in forma standard e $\boldsymbol{y}$ é una soluzione ammissibile (non necessariamente ottima) per il rispettivo problema duale, allora vale $\boldsymbol{cx} \leq \boldsymbol{yb}$.

Dimostrazione. Come mostrato in precedenza, si ha $Z = \boldsymbol{cx}$ e $W = \boldsymbol{yb}$. Essendo la soluzione ottima per il problema duale data dal minimo valore di W ed essendo $W^* = Z^*$, allora qualsiasi valore di W non ottimale é necessariamente maggiore di qualsiasi valore di Z non ottimale.

Condizione di complementarietá. Ad ogni iterazione generica, il metodo del simplesso identifica simultaneamente un vertice bix per il problema primale ed una **soluzione complementare** $\boldsymbol{y}$ per il problema duale, dove $\boldsymbol{cx} = \boldsymbol{yb}$. Se $\boldsymbol{x}$ non é una soluzione ottima per il problema primale, allora $\boldsymbol{y}$ non é una soluzione ammissibile per il problema duale.

Condizione di complementarietá per soluzioni ottime. Nell'iterazione finale, il metodo del simplesso identifica simultaneamente una soluzione ottima bix^* per il problema primale ed una **soluzione complementare ottima** $\boldsymbol{y}^*$ per il problema duale, dove $\boldsymbol{cx}^* = \boldsymbol{y}^*\boldsymbol{b}$. Le quantitá y_i sono i prezzi ombra per il problema primale.

Simmetria. Siano dati un problema primale di programmazione lineare ed il relativo problema duale. Il problema duale del problema duale é il problema primale.

Il teorema precedente implica che non é possibile parlare di problema primale o duale in senso "assoluto"; é indifferente quale dei due problemi legati da una relazione primale-duale venga considerato come primale e quale come duale, perché a seconda del punto di vista entrambe le denominazioni sono corrette. Allo stesso modo, tutte le relazioni che il problema primale ha nei confronti del problema duale sono valide anche in senso inverso. Per convenzione, si tende a considerare come problema primale quello dei due che modella la situazione che si vuole rappresentare. Il teorema precedente implica inoltre che i problemi primali ed i rispettivi problemi duali possono essere risolti con gli stessi metodi. Il metodo del simplesso puó essere quindi applicato all'uno o all'altro problema (dopo averli eventualmente convertiti nella forma standard) e permetterá di identificare simultaneamente le soluzioni complementari per l'altro problema.

Teorema della dualitá. Siano dati un problema primale di programmazione lineare ed il relativo problema duale. Fra i due problemi deve presentarsi una ed una sola fra queste due possibili correlazioni:

- Se uno dei due problemi ha almeno una soluzione ammissibile ed una funzione obiettivo limitata (e quindi ha almeno una soluzione ottima), allora anche l'altro problema ha almeno una soluzione ammissibile ed una funzione obiettivo limitata;
- Se uno dei due problemi non ha alcuna soluzione ammissibile, allora l'altro problema non ha alcuna soluzione ammissibile oppure ha una funzione obiettivo illimitata.

Il problema duale ed i suoi elementi hanno anche una interpretazione economica, strettamente correlata all'interpretazione economica dei problemi di programmazione lineare in forma standard vista in precedenza.
Poiché $W = b_1y_1 + \ldots + b_my_m = Z$, ogni quantitá b_iy_i puó essere interpretata come il contributo al profitto corrente quando b_i unitá della risorsa i sono disponibili, mentre y_i rappresenta il prezzo ombra.
Questa definizione delle variabili duali porta naturalmente ad una interpretazione dell'intero problema duale. In particolare, poiché nel problema primale ogni unitá dell'attivitá j consuma a_{ij} unitá della risorsa i, la quantitá $\sum_{i=1}^{m} a_{ij}y_i$ puó essere interpretata come il contributo al profitto della combinazione di risorse che verrebbe consumata se venisse impiegata un'unitá dell'attivitá j.
Nel problema primale, c_j corrisponde al profitto unitario dell'attivitá j. Questo significa che, nel problema duale, se si verifica $\sum_{i=1}^{m} a_{ij}y_i > c_j$ allora significa che il contributo al profitto $\sum_{i=1}^{m} a_{ij}y_i$ é superiore a quello ottenuto utilizzando un'unitá dell'attivitá j; diversamente, queste risorse non verrebbero usate nel modo migliore. Analogamente, il vincolo di non negativitá $y_i \geq 0$ indica che il contributo dato dalla risorsa i al profitto deve essere un valore non negativo, altrimenti sarebbe deletereo farne uso.
La funzione obiettivo $W = \sum_{i=1}^{m} b_iy_i$ da minimizzare puó essere vista come la minimizzazione del valore totale implicito delle risorse consumate dalle varie attivitá.
Poiché il problema duale é un problema di programmazione lineare, anche per esso é possibile ragionare in termini di vertici e di regione ammissibile. Inoltre, cosí come il problema primale, é possibile esprimerlo in forma aumentata introducendo opportune variabili aggiuntive per esprimerne i vertici come soluzioni di base. Nello specifico, poiché i vincoli del problema duale sono di tipo $\geq$, tale forma aumentata si ottiene sottraendo una variabile aggiuntiva, detta **variabile surplus** (e non sommando una variabile slack) a sinistra di ogni vincolo. Si ha quindi:

$$z_j - c_j = \sum_{i=1}^{m} a_{ij}y_i - c_j \quad \text{per} \quad j = 1, 2, \ldots, n$$

In questo modo $z_j - c_j$ svolge il ruolo della variabile surplus per il vincolo j. Di conseguenza, da ogni vertice $(y_1, y_2, \ldots, y_m)$ si ottiene una soluzione di base $(y_1, y_2, \ldots, y_m, z_1 - c_1, z_2 - c_2, \ldots, z_n - c_n)$. Poiché la forma aumentata del problema duale ha n vincoli funzionali e $n + m$ variabili, ogni soluzione di base ha n variabili di base e m variabili non di base.

Ad ogni soluzione di base del problema primale corrisponde una soluzione di base complementare nel problema duale ed i rispettivi valori della funzione obiettivo sono uguali.

Proprietá di complementary slackness. Se una variabile del problema primale é variabile di base per una certa soluzione, allora la variabile a questa associata nel problema duale é variabile non di base per la relativa soluzione, e viceversa.

$$\max Z = 3x_1 + 5x_2$$
$$x_1 \leq 4$$
$$2x_2 \leq 12$$
$$3x_1 + 2x_2 \leq 18$$
$$x_1 \geq 0, x_2 \geq 0$$

$$\min W = 4y_1 + 12y_2 + 18y_3$$
$$y_1 + 3y_3 \geq 3$$
$$2y_2 + 2y_3 \geq 5$$
$$y_1 \geq 0, y_2 \geq 0, y_3 \geq 0$$

N.	Problema primale		Z = W	Problema duale	
	Soluzione di base	Ammissibile?		Soluzione di base	Ammissibile?
1	(0, 0, 4, 12, 18)	sí	0	(0, 0, 0, -3, -5)	no
2	(4, 0, 0, 12, 6)	sí	12	(3, 0, 0, 0, -5)	no
3	(6, 0, -2, 12, 0)	no	18	(0, 0, 1, 0, -3)	no
4	(4, 3, 0, 6, 0)	sí	27	$(-\frac{9}{2}, 0, \frac{5}{2}, 0, 0)$	no
5	(0, 6, 4, 0, 6)	sí	30	$(0, \frac{5}{2}, 0, -3, 0)$	no
6	(2, 6, 2, 0, 0)	sí	36	$(0, \frac{3}{2}, 1, 0, 0)$	sí
7	(4, 6, 0, 0, -6)	no	42	$(3, \frac{5}{2}, 0, 0, 0)$	sí
8	(0, 9, 4, -6, 0)	no	45	$(0, 0, \frac{5}{2}, \frac{9}{2}, 0)$	sí

La proprietá di complementarietá delle soluzioni ottimali viene estesa in modo molto naturale alla forma aumentata del problema.

Proprietá di complementarietá delle soluzioni di base ottime. Ad ogni soluzione di base ottima del problema primale é associata una soluzione di base complementare ottimale del problema duale, ed i valori delle rispettive funzioni obiettivo sono gli stessi.

Le soluzioni di base possono essere classificate a seconda che esse soddisfino o meno ciascuna delle seguenti condizioni. Una é la **condizione di ammissibilitá**, cioé se tutte le variabili (comprese quelle slack) nella soluzione aumentata sono non negative. L'altra é la **condizione di ottimalitá**, cioé se tutte le variabili nella soluzione di base complementare sono non negativi. Le due condizioni possono sia presentarsi contemporaneamente, sia essere entrambe assenti; in particolare, si distinguono quattro casi:

1. Se una soluzione di base soddisfa sia la condizione di ammissibilitá che la condizione di ottimalitá, si dice che tale soluzione é **ottimale**;
2. Se una soluzione di base soddisfa la condizione di ammissibilitá ma non la condizione di ottimalitá, si dice che tale soluzione é **sub-ottimale**;
3. Se una soluzione di base soddisfa la condizione di ottimalitá ma non la condizione di ammissibilitá, si dice che tale soluzione é **super-ottimale**;
4. Se una soluzione di base non soddisfa né la condizione di ammissibilitá né la condizione di ottimalitá, si dice che tale soluzione non é **né ammissibile né super-ottimale**;

In merito all'esempio precedente, la condizione di ammissibilitá é soddisfatta dalle soluzioni 1, 2, 4, 5 e 6, mentre la condizione di ottimalitá é soddisfatta dalle soluzioni 6, 7 e 8. Si ha quindi che le soluzioni 1, 2, 4 e 5 sono sub-ottime, la soluzione 6 é ottima, le soluzioni 7 e 8 sono super-ottime e la soluzione 3 non é né ammissibile né super-ottimale.

Ricordando che la relazione di dualitá fra problema primale e problema duale é simmetrica, si ha che una soluzione sub-ottimale é una soluzione super-ottimale se considera il problema primale come problema duale (e viceversa); allo stesso modo, una soluzione super-ottimale é una soluzione sub-ottimale se considera il problema primale come problema duale (e viceversa).
Mentre il metodo del simplesso visita soluzioni di base sub-ottimali, cercando di trovare la soluzione ottimale del problema primale, indirettamente e come se visitasse soluzioni complementari super-ottimali, cercando di ottenere una soluzione ammissibile per il problema duale. Per

descrivere una coppia di soluzioni di base complementari vengono anche usati i seguenti termini: **primale ammissibile** se la soluzione di base primale è ammissibile per il primale e **duale ammissibile** se la soluzione di base complementare del duale è ammissibile per il duale. Utilizzando questa terminologia, il metodo del simplesso visita soluzioni primali ammissibili, cercando di ottenere l'ammissibilità duale. Quando questo viene ottenuto, le due soluzioni di base complementari sono ottimali per i rispettivi problemi.

Dato che é sempre possibile trasformare un problema di programmazione lineare generico in un problema equivalente in forma standard, a tutti i problemi di programmazione lineare é possibile associare un problema duale, non solamente a quelli in forma standard. L'unica differenza é che tale problema duale non sará necessariamente nella forma finora considerata (problema di ottimizzazione di minimo, vincoli funzionali tutti nella forma $\geq$ e vincoli di non negativitá per ogni variabile).

Per ricavare il problema duale associato ad un problema primale in forma non standard si potrebbe convertire tale problema nella forma standard e poi ricavare il problema duale come di consueto. Questo approccio é certamente corretto, ma dispendioso. Infatti, la relazione che sussiste fra i coefficienti che compaiono nel problema primale in forma non standard ed il rispettivo problema duale é sempre la stessa, la differenza sta soltanto nella forma dei vincoli del problema e nel tipo di ottimizzazione.

Esiste un metodo piú veloce che permette di ricavare subito la forma dei vincoli del problema duale associato ad un problema primale in forma non standard, senza effettuare alcuna conversione. Tale metodo prende il nome di **metodo SOB** (**Strange-Odd-Bizarre**), e si compone di quattro passaggi. Noto un problema di programmazione lineare scelto come problema primale:

1. Se il problema primale é un problema di ottimizzazione di massimo, allora il problema duale sará un problema di ottimizzazione di minimo. Viceversa, se il problema primale é un problema di ottimizzazione di minimo, allora il problema duale sará un problema di ottimizzazione di massimo;
2. Etichettare le diverse forme di vincoli funzionali ed i vincoli sulle singole variabili decisionali del primale come Sensible, Odd o Bizarre, in accordo con la tabella in basso a sinistra. L'etichettatura dei vincoli funzionali dipende dalla natura del problema (se é di ottimizzazione di massimo o di ottimizzazione di minimo);
3. Per ogni vincolo su una variabile di decisione del problema duale, utilizzare la forma che ha medesima etichetta del vincolo funzionale per il problema primale corrispondente a questa variabile duale, in accordo con la tabella in basso a destra;
4. Per ogni vincolo funzionale del problema duale, utilizzare la forma che ha la stessa etichetta del vincolo della corrispondente variabile di decisione del problema primale, in accordo con la tabella in basso a destra.

Etichetta	**Problema primale (o duale)**	**Problema duale (o primale)**
	max Z (o W)	min W (o Z)
	Vincolo i	Variabile y_i (o x_i)
Sensible	$\leq$	$y_i \geq 0$
Odd	=	non vincolata
Bizarre	$\geq$	$y_i' \leq 0$
	Variabile x_j (o y_j)	Vincolo j
Sensible	$x_j \geq 0$	$\geq$
Odd	non vincolata	=
Bizarre	$x_j' \leq 0$	$\leq$

problema 1#	**problema 2#**
Vincolo i	Variabile i
Funzione obiettivo	Termine noto destro

Sia P un problema di programmazione lineare in forma non standard, e sia D il rispettivo problema duale. Si ha allora che D', il problema D riscritto in forma standard, é il duale di P', il problema P riscritto in forma standard.

$$\begin{aligned} \max -Z = \; & -0.4x_1 - 0.5x_2 \\ & 0.3x_1 + 0.1x_2 \leq 2.7 \\ & 0.5x_1 + 0.5x_2 = 6 \\ & 0.6x_1 + 0.4x_2 \geq 6 \\ & x_1 \geq 0, \; x_2 \geq 0 \end{aligned}$$

Si consideri il problema di programmazione lineare presentato a lato. Tale problema non si trova in forma standard, dato che non tutti i vincoli funzionali sono di tipo $\leq$. É peró possibile applicare il metodo SOB per costruirne il relativo problema duale:

1. Il problema primale é un problema di ottimizzazione di massimo, pertanto il rispettivo problema duale sará un problema di ottimizzazione di minimo;
2. Il problema primale ha tre vincoli funzionali e due variabili, pertanto il rispettivo problema duale avrá due vincoli funzionali e tre variabili;
3. Il primo vincolo funzionale del problema primale é di tipo $\leq$, pertanto la variabile y_1 del problema duale avrá associato un vincolo di non negativitá;
4. Il secondo vincolo funzionale del problema primale é di tipo =, pertanto la variabile y_2 del problema duale non avrá associato un vincolo legato al suo segno;
5. Il terzo vincolo funzionale del problema primale é di tipo $\geq$, pertanto la variabile y_3' del problema duale (costruita ad-hoc) avrá associato un vincolo di non positivitá;
6. I coefficienti dell'i-esimo vincolo del problema duale corrispondono ai coefficienti della variabile x_i nei vincoli funzionali del problema primale, mentre il termine noto destro corrisponde al coefficiente della variabile x_i nella funzione obiettivo del problema primale. Tale vincolo sará nella forma $\geq$;
7. L'i-esimo coefficiente della funzione obiettivo del problema duale é dato dal termine noto destro dell'i-esimo vincolo funzionale del problema primale.

$$\begin{aligned} \max -Z = \; & -0.4x_1 - 0.5x_2 \\ & 0.3x_1 + 0.1x_2 \leq 2.7 \\ & 0.5x_1 + 0.5x_2 = 6 \\ & 0.6x_1 + 0.4x_2 \geq 6 \\ & x_1 \geq 0, \; x_2 \geq 0 \end{aligned}$$

$$\begin{aligned} \min W = \; & 2.7y_1 + 6y_2 + 6y_3' \\ & 0.3y_1 + 0.5y_2 + 0.6y_3' \geq -0.4 \\ & 0.1y_1 + 0.5y_2 + 0.4y_3' \geq -0.5 \\ & y_1 \geq 0, \; y_3' \leq 0 \end{aligned}$$

Capitolo 3
Programmazione lineare intera

3.1 Modello di programmazione lineare intera

I problemi di programmazione lineare finora presentati rispettavano, fra le altre, l'ipotesi di divisibilitá, ovvero veniva assunto che le variabili decisionali del problema potessero assumere valori reali (o, al piú, razionali). Tuttavia, non sempre il problema da modellare rispetta questa ipotesi, e pertanto la programmazione lineare risulta inapplicabile. Ad esempio, l'ipotesi é evidentemente invalidata in problemi che riguardano: assegnazione del personale, distribuzione o stoccaggio di beni, decisioni manageriali. Ció che accomuna tutte queste situazioni é il fatto che gli elementi reali da modellare sono enti discreti.
Se un problema programmazione lineare richiede che tutte le variabili decisionali debbano assumere esclusivamente valori interi, si parla di **Programmazione Lineare Intera Pura**, o semplicemente di **Programmazione Lineare Intera** (**PLI**). Se invece soltanto una parte delle variabili decisionali devono assumere valori interi si parla di **Programmazione Lineare Intera Mista** (**PLM**).
Una particolare classe di problemi di programmazione lineare intera che merita attenzione sono i problemi che trattano di decisioni "si/no". Tali decisioni possono essere rappresentate mediante variabili che possono assumere esclusivamente due valori, che per analogia con l'algebra booleana tendono ad essere 0 e 1. In questo senso, se la i-esima variabile decisionale del problema assume valore 1 significa che la i-esima decisione é stata presa, mentre se assume valore 0 significa che l'i-esima decisione é stata rigettata. Variabili di questo tipo sono dette **variabili binarie**, e problemi di programmazione lineare intera che trattano solo variabili binarie sono indicati come **problemi di programmazione binaria**.

Una azienda manufatturiera sta considerando la possibilitá di espandersi costruendo una nuova fabbrica a Los Angeles, a San Francisco o in entrambe le cittá. Inoltre, vi é anche la possibilitá di costruire un nuovo magazzino in al piú una delle due cittá, ma solo se in questa é stata costruita anche la fabbrica. Il profitto che genererá la costruzione di una struttura, oltre al capitale necessario per l'investimento, é riportato per ciascuna di queste nella tabella di seguito (espresso in milioni di dollari). Sapendo che il capitale massimo a disposizione é 10 milioni di dollari, qual'é la combinazione di decisioni che genera il massimo profitto?

Decisione	Struttura da costruire	Luogo in cui costruire	Profitto ipotetico	Investimento necessario
1	Fabbrica	Los Angeles	9	6
2	Fabbrica	San Francisco	5	3
3	Magazzino	Los Angeles	6	5
4	Magazzino	San Francisco	4	2

Il problema puó essere modellato come un problema di programmazione lineare intera binaria.
Alla decisione di costruire ciascuna i-esima struttura viene associata una variabile binaria x_i: tale variabile assume valore 1 se si é deciso che l'i-esima struttura va costruita mentre assume valore 0 se si é deciso che l'i-esima struttura non va costruita.
Sia Z la funzione obiettivo del problema, che corrisponde al profitto totale dato dalla costruzione da zero a quattro strutture. Il problema richiede di trovare la combinazione di decisioni che restituisca il massimo profitto, pertanto tale funzione deve venire massimizzata. Si ricordi inoltre che il capitale massimo a disposizione é 10 milioni di dollari, pertanto la spesa totale non puó superare tale cifra. Si ottengono quindi l'espressione per Z ed un primo vincolo:

$$Z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \qquad 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$$

Le decisioni 3 e 4 sono mutualmente esclusive, dato che é stato imposto che venga costruito al piú un solo magazzino. In altri termini, le variabili x_3 e x_4 non possono assumere contemporaneamente il valore 1, ma possono assumere contemporaneamente il valore 0. Tale costrizione puó venire espressa aggiungendo al problema il vincolo $x_3 + x_4 \leq 1$.
La decisione 3 dipende della decisione 1, perché la costruzione di un magazzino é ammessa solamente se nella stessa cittá viene costruita anche una fabbrica. In altri termini, se la variabile x_1 vale 0 allora x_3 deve valere per forza 0, ma se x_1 vale 1 allora x_3 puó valere sia 0 che 1. Un ragionamento del tutto analogo puó essere fatto per le decisioni 4 e 2 e le rispettive variabili decisionali. Tali costrizioni possono venire espresse aggiungendo al problema i vincoli $x_3 \leq x_1$ e $x_4 \leq x_2$.
Unificando tutti i vincoli rilevati, si ottiene il seguente problema di programmazione lineare intera binaria:

$$\begin{aligned} \max Z = \; & 9x_1 + 5x_2 + 6x_3 + 4x_4 \\ & 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\ & x_3 + x_4 \leq 1 \\ & -x_1 + x_3 \leq 0 \\ & -x_2 + x_4 \leq 0 \\ x_j \in \{0,1\} \; & \text{ per } j = 1,2,3,4 \end{aligned}$$

3.2 Adattamento al modello di programmazione binaria

Vi sono diverse situazioni in cui le variabili binarie permettono di ridurre un problema complesso ad un problema di programmazione lineare intera binaria. Situazioni di questo tipo nascono quando la formulazione originaria del problema si adatta ad un modello di PLI ad eccezione di alcune condizioni che coinvolgono relazioni di tipo combinatorio del modello. Esprimento queste relazioni in termini di domande a cui bisogna dare risposta sí-no, é possibile aggiungere delle variabili binarie ausiliare al modello per rappresentare queste decisioni riducendo il problema ad un problema di PLI mista.

3.2.1 Vincoli di tipo "either-of"

Si consideri il caso in cui, dati due vincoli, solo uno di questi deve essere soddisfatto, e non é necessario che sia soddisfatto anche l'altro. In altri termini, almeno una delle due uguaglianze/disuguaglianze deve essere vera, ma non necessariamente entrambe:

$$a_1x_1 + \ldots + a_nx_n \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} \alpha \qquad\qquad b_1x_1 + \ldots + b_nx_n \begin{Bmatrix} \leq \\ \geq \\ = \end{Bmatrix} \beta$$

Una situazione di questo tipo non é esprimibile direttamente in termini di programmazione lineare, perché nel modello di programmazione lineare tutti i vincoli devono essere sempre soddisfatti.
Si assuma che la forma dei vincoli sia $\leq$ (se non lo sono, é sempre possibile convertirli in tale forma). Sia M un numero positivo molto grande; aggiungendo M a destra di una disuguaglianza $\leq$ che é noto essere verificata si ottiene una nuova disuguaglianza $\leq$ anch'essa certamente verificata. Introducendo M, la coppia di vincoli puó essere riscritta indifferentemente in uno dei due modi:

$$\begin{cases} a_1x_1 + \ldots + a_nx_n \leq \alpha + M \\ b_1x_1 + \ldots + b_nx_n \leq \beta \end{cases} \qquad\qquad \begin{cases} a_1x_1 + \ldots + a_nx_n \leq \alpha \\ b_1x_1 + \ldots + b_nx_n \leq \beta + M \end{cases}$$

L'idea é che aggiungendo M a destra di un vincolo lo si rende irrilevante, perché una soluzione che soddisfa gli altri vincoli del problema soddisfa certamente anche questo (naturalmente, si deve assumere che M sia abbastanza grande da non invalidare potenziali soluzioni ammissibili). Introducendo una variabile binaria ausiliaria y, le due forme della riscrittura possono essere riassunte in una sola:

$$\begin{cases} a_1x_1 + \ldots + a_nx_n \leq \alpha + My \\ b_1x_1 + \ldots + b_nx_n \leq \beta + M(1-y) \end{cases}$$

In questo modo, la variabile y annulla alternativamente la variabile M in uno dei due vincoli a seconda di quale valore assume, ed il vincolo in cui M permane viene reso irrilevante. In questo senso, la variabile binaria y rappresenta la decisione sí-no in merito a quale dei due vincoli é quello che si ha maggior interesse che venga soddisfatto, perché l'altro lo sará sicuramente.

3.2.2 Devono essere soddisfatti K degli N vincoli presenti

Si consideri il caso di un modello che include un insieme di N vincoli, tali che solo K di questi devono essere soddisfatti (si assuma $K < N$). Parte del processo di ottimizzazione é scegliere la combinazione dei K vincoli che permette alla funzione obiettivo di raggiungere i migliori valori possibili, mentre i rimanenti $N-K$ vincoli vengono resi irrilevanti. Si noti come questa é semplicemente una generalizzazione della situazione precedente; infatti, quest'ultima si ottiene ponendo $K = 1$ e $N = 2$.
Si assuma sempre che gli N vincoli siano nella forma $\leq$. Siano questi denotati come:

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) \leq d_1 \\ \vdots \\ f_N(x_1, x_2, \ldots, x_n) \leq d_N \end{cases}$$

Applicando la stessa logica del caso precedente, una formulazione equivalente al requisito che K di questi vincoli debbano essere soddisfatti é

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) \leq d_1 + My_1 \\ \vdots \\ f_N(x_1, x_2, \ldots, x_n) \leq d_N + My_N \\ \sum_{i=1}^{N} y_i = N-K \end{cases}$$

Dove y_i con $i = 1, 2, \ldots, N$ sono tutte variabili binarie ed M é un numero positivo molto grande. Per ciascun i-esimo vincolo, se la variabile y_i assume valore 0 il valore M viene annullato, rendendo tale vincolo equivalente a quello originale, mentre se assume valore 1 il valore M viene mantenuto, rendendo il vincolo sicuramente soddisfatto dalle soluzioni che soddisfano gli altri vincoli.

Poiché i vincoli sulle variabili y_i garantiscono che K di queste variabili siano uguali a 0 e $N-K$ siano uguali ad 1, K dei vincoli originari saranno lasciati intatti e $N-K$ verranno resi irrilevanti. La scelta di quali siano i K vincoli da dover essere mantenuti é demanata all'algoritmo risolutivo stesso.

3.2.3 Funzioni con N possibili valori

Si consideri la situazione in cui una data funzione assuma uno fra N possibili valori. Si denoti un vincolo di questo tipo con

$$f(x_1, x_2, ..., x_n) = \{d_1 \;\; d_2 \;\; ... \;\; d_N\}$$

Come in precedenza, anche una situazione di questo tipo non puó venire espressa direttamente in termini di programmazione lineare. Tale vincolo puó peró essere riscritto in una forma equivalente, compatibile con il modello di PLI, composta da una coppia di vincoli:

$$\begin{cases} f(x_1, x_2, ..., x_n) = \sum_{i=1}^{N} d_i y_i \\ \sum_{i=1}^{N} y_i = 1 \end{cases}$$

Dove y_i con $i = 1, 2, ..., N$ sono tutte variabili binarie. Il primo vincolo é il corrispettivo del vincolo originale, dove i valori d_i con $i = 1, 2, ..., N$ sono stati sostituiti da una sommatoria i cui elementi sono le variabili d_i e le variabili binarie ausiliarie y_i.
Il secondo vincolo impone che esattamente una sola variabile y_i assuma il valore 1 e tutte le altre assumano il valore 0, ed in questo modo tutti i valori d_i vengono annullati tranne uno; tale valore sará il valore che viene assunto dalla funzione. In questo caso, ci sono N domande di tipo sí-no da porsi: il valore scelto é d_i con $i = 1, 2, ..., N$?

Il vincolo a sinistra per un ipotetico problema di programmazione lineare puó essere riscritto in una forma equivalente come a destra, introducendo le tre variabili binarie y_1, y_2, y_3.

$$3x_1 + 2x_2 = \{6 \;\; 12 \;\; 18\} \qquad \begin{cases} 3x_1 + 2x_2 = 6y_1 + 12y_2 + 18y_3 \\ y_1 + y_2 + y_3 = 1 \end{cases}$$

3.2.4 Problema del "fixed charge"

Si consideri una situazione in cui le variabili decisionali x_j con $j = 1, 2, ..., n$ di un problema di programmazione lineare rappresentino delle attivitá, a cui é associato un costo. Nel caso in cui l'attivitá in questione venga avviata, il suo costo totale é dato dalla somma di una spesa fissa legata all'inizializzazione di tale attivitá e ad un costo variabile, in genere proporzionale al livello dell'attivitá stessa. Nel caso in cui l'attivitá in questione non venga avviata, il suo costo é zero.
Il costo totale di una attivitá x_j con tali caratteristiche puó rappresentato mediante la seguente funzione:

$$f_j(x_j) = \begin{cases} k_j + c_j x_j & \text{se } x_j > 0 \\ 0 & \text{se } x_j = 0 \end{cases}$$

Dove x_j denota il livello dell'attivitá j, k_j denota il costo di setup della j-esima attivitá e c_j denota il costo per ogni unitá incrementale della j-esima attivitá. La presenza del costo fisso k_j impedisce al problema di essere formulato direttamente come problema di programmazione lineare.
Si considerino allora n attivitá x_j, ciascuna con un costo di inizializzazione (con $k_j \geq 0$ in ogni caso e $k_j > 0$ per qualche $j = 1, 2, ..., n$). Se il portare a termine una attivitá implica un costo, allora il problema consiste nel minimizzare la funzione obiettivo

$$Z = f_1(x_1) + f_2(x_2) + ... + f_n(x_n) = \sum_{j=1}^{n} f_j(x_j) \qquad f_j(x_j) = \begin{cases} k_j + c_j x_j & \text{se } x_j > 0 \\ 0 & \text{se } x_j = 0 \end{cases}$$

sottostando ai vincoli del problema (quali che siano).
Questo problema puó essere trasformato in un problema di PLI equivalente introducendo n variabili binarie ausiliarie y_i con $i = 1, 2, ..., n$. Ciascuna variabile y_i rappresenta la decisione sí-no relativa all'intraprendere o al non intraprendere la i-esima attivitá. A tal fine, la funzione obiettivo Z puó venire riscritta come:

$$Z = f_1(x_1) + f_2(x_2) + ... + f_n(x_n) = \sum_{j=1}^{n} f_j(x_j) = \sum_{j=1}^{n} c_j x_j + k_j y_j \qquad y_j = \begin{cases} 1 & \text{se } x_j > 0 \\ 0 & \text{se } x_j = 0 \end{cases}$$

Se la j-esima attivitá non viene intrapresa, ovvero se sia x_j che y_j sono uguali a 0, allora il costo per la j-esima attivitá é esso stesso 0.

Questa riscrittura non é peró sufficiente a trasformare il problema originale in un problema PLI, perché la definizione di y_i sopra riportata non é esprimibile in termini di programmazione lineare. Occorre infatti compiere un ulteriore passo e tradurre la definizione di y_i come uno o piú vincoli.
In tal senso, sia M un numero intero positivo molto grande. Introducendo n vincoli

$$x_j \leq My_j$$

per ciascun $j = 1, 2, ..., n$ si ha la certezza che y_j valga 1 invece che 0 ogni volta che $x_j > 0$.
Ora occorrerebbe introdurre uno o piú vincoli che impongano che le variabili y_j assumano il valore 0 ogni volta che x_j vale 0. Tuttavia, per la natura stessa della funzione obiettivo, non é necessario esplicitare tale restrizione.
Si consideri infatti il caso in cui $x_j = 0$. Se $k_j = 0$, l'intera espressione $c_j x_j + k_j y_j$ si annulla comunque, quindi la scelta di y_j é irrilevante. Se invece $k_j > 0$, scegliere $y_j = 0$ annulla l'intera espressione, mentre scegliere $y_j = 1$ restituisce il solo k_j, che per ipotesi é un valore positivo. Pertanto, dato che il problema é un problema di ottimizzazione di minimo, la scelta migliore per il valore di y_j quando $x_j = 0$ e $k_j > 0$ é sempre e comunque 0.

Una acciaieria sta valutando come ridurre l'inquinamento atmosferico prodotto dalle sue fornaci con la minima spesa possibile. Tali fornaci sono di due tipi: fornaci di fusione e fornaci a crogiolo aperto. L'inquinamento si presenta in tre forme: polveri sottili, ossido di zolfo e gas metano. Qual'é la percentuale di utilizzo che permette di minimizzare i costi ed al contempo di ridurre le emissioni nocive della quantitá prefissata?

Sostanza inquinante	Quantitá da eliminare	Riduzione a pieno regime	
Polveri sottili	25	12	9
Ossido di zolfo	50	35	42
Gas metano	75	37	53

Costo	Fusione	Crogiolo aperto
Spesa fissa	2	3
Spesa variabile	8	10

Sia x_1 la percentuale di ottimizzazione impiegata per le fornaci a fusione, e sia mathx_2 la percentuale di ottimizzazione impiegata per le fornace a crogiolo aperto. Essendo delle percentuali, tali variabili devono assumere un valore compreso fra 0 e 1. Inoltre, la quantitá di emissioni eliminate deve essere almeno pari alla quantitá prefissata. Pertanto occorre introdurre i vincoli:

$$\begin{cases} 12x_1 + 9x_2 \geq 25 \\ 35x_1 + 42x_2 \geq 50 \\ 37x_1 + 53x_2 \geq 75 \end{cases} \qquad \begin{aligned} 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \end{aligned}$$

Il problema consiste quindi nel minimizzare la funzione:

$$Z = 8x_1 + 10x_2 + 2y_1 + 3y_2 \qquad \begin{cases} x_1 - My_1 \leq 0 \\ x_2 - My_2 \leq 0 \end{cases}$$

Dove y_1 e y_2 sono due variabili binarie, alle quali sono associati i vincoli necessari per rappresentare il costo fisso associato alla ottimizzazione delle fornaci. Il problema complessivo viene quindi definito come:

$$\begin{aligned} \min Z = \ & 8x_1 + 10x_2 + 2y_1 + 3y_2 \\ & 12x_1 + 9x_2 \geq 25 \\ & 35x_1 + 42x_2 \geq 50 \\ & 37x_1 + 53x_2 \geq 75 \\ & x_1 \geq 0 \\ & x_1 \leq 1 \\ & x_1 - My_1 \leq 0 \\ & x_2 \geq 0 \\ & x_2 \leq 1 \\ & x_2 - My_2 \leq 0 \end{aligned}$$

3.2.5 Rappresentare variabili intere come variabili binarie

Si supponga di avere un problema di PLI pura dove la maggior parte delle variabili sono variabili binarie ed una piccola parte sono variabili intere, che impediscono di risolvere il problema mediante algoritmi di programmazione lineare intera binaria (che sono molto piú veloci degli algoritmi per la programmazione lineare generica).
Se le variabili intere in questione sono poche, é possibile aggirare il problema in maniera elegante é quello di usare la rappresentazione binaria per ognuna di queste. Nello specifico, se i limiti per una variabile intera sono $0 \leq x \leq u$, dove u é un numero compreso fra la N-esima e la $N+1$-esima potenza di due, la rappresentazione binaria di x é:

$$x = \sum_{i=0}^{N} 2^i y_i$$

Dove le variabili y_i con $i = 1, 2, ..., N$ sono variabili binarie aggiuntive. Sostituendo questa rappresentazione binaria al posto di ognuna delle variabili intere, avendo cura di selezionare un insieme differente di variabili aggiuntive binarie per ciascuna di esse, l'intero problema si riduce ad un modello di programmazione binaria.

Si consideri un problema di PLI avente un certo numero di variabili binarie e due variabili intere x_1 e x_2. Si supponga che ad entrambe le variabili siano associati dei vincoli di non negativitá ed i seguenti vincoli funzionali:

$$\begin{cases} x_1 \leq 5 \\ 2x_1 + 3x_2 \leq 30 \end{cases}$$

Questi vincoli implicano che $u = 5$ per x_1 e $u = 10$ per x_2. Quindi si ha $N = 2$ per x_1 (perché $2^2 \leq 5 < 2^3$) e $N = 3$ per x_2 (perché $2^3 \leq 10 < 2^4$). Pertanto, a queste variabili é possibile associare la seguente rappresentazione binaria:

$$x_1 = \sum_{i=0}^{2} 2^i y_i = 2^0 y_0 + 2^1 y_1 + 2^2 y_2 = y_0 + 2y_1 + 4y_2$$
$$x_2 = \sum_{i=0}^{3} 2^i y_{i+3} = 2^0 y_3 + 2^1 y_4 + 2^2 y_5 + 2^3 y_6 = y_3 + 2y_4 + 4y_5 + 8y_6$$

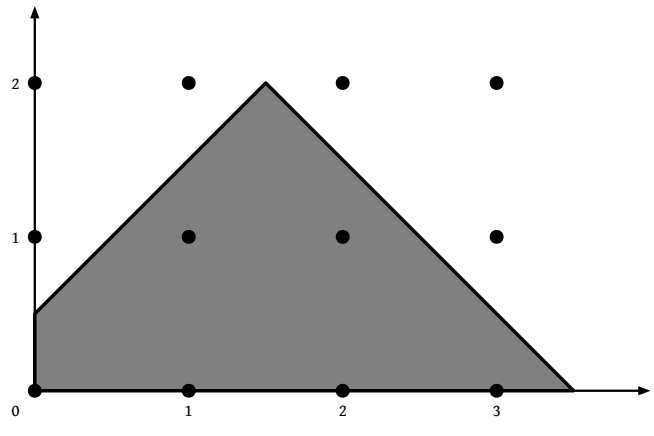
Sostituendo tale rappresentazione nei vincoli, si ottiene un problema di programmazione lineare intera in cui figurano solo ed esclusivamente variabili binarie:

$$\begin{cases} y_0 + 2y_1 + 4y_2 \leq 5 \\ 2y_0 + 4y_1 + 8y_2 + 3y_3 + 6y_4 + 12y_5 + 24y_6 \leq 30 \end{cases}$$

3.3 Metodo Branch-and-Bound per la programmazione binaria

L'unica significativa distinzione fra i problemi di programmazione lineare ed i problemi di programmazione binaria é nel dominio dal quale tali variabili provengono. Infatti, tutte le altre ipotesi che sostengono i due modelli (proporzionalitá, additivitá, certezza) sono le medesime. Dato che la differenza fra i due modelli sembrerebbe essere marginale, ci si chiede allora cosa accadrebbe se si tentasse di risolvere un problema di programmazione binaria semplicemente applicando il metodo del simplesso, cosí come é stato formulato in precedenza.
Innanzitutto, occorre puntualizzare come il metodo del simplesso sfrutti l'esistenza di una regione ammissibile di un piano n-dimensionale, avente vertici che sono a loro volta potenziali soluzioni ottimali (anzi, sono proprio tali soluzioni ad essere le uniche che il metodo del simplesso prende in considerazione). Nel caso della programmazione binaria, questo non si verifica sempre, perché gli estremi della regione ammissibile non sono necessariamente numeri interi, e quindi non sono necessariamente soluzioni ammissibili.
Si potrebbe pensare di aggirare questo sconveniente approssimando, se necessario, la soluzione ottimale restituita dal metodo del simplesso applicato al problema di programmazione binaria alla soluzione intera piú vicina. Tuttavia, questo approccio non garantisce di restituire una soluzione ammissibile, dato che la soluzione intera piú vicina a quella restituita dal metodo del simplesso potrebbe non essere parte della regione ammissibile.

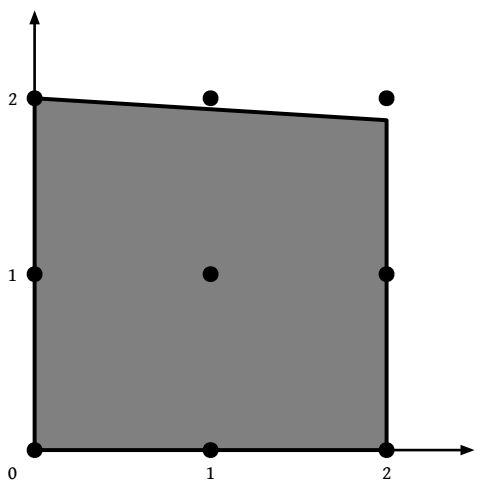
$$Z = x_2$$
$$-x_1 + x_2 \leq \frac{1}{2}$$
$$x_1 + x_2 \leq \frac{7}{2}$$
$$x_1 \geq 0,\ x_2 \geq 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Si consideri il problema di programmazione lineare intera sopra presentato. Se si tralascia il vincolo che ciascuna variabile debba essere intera e si applica il metodo del simplesso, si ottiene la soluzione $\left(\frac{3}{2}, 2\right)$. Si noti peró come approssimare tale soluzione ad una delle due soluzioni intere piú vicine, sia questa $(1, 2)$ o $(2, 2)$, restituisce comunque una soluzione non ammissibile.

Inoltre, anche ammesso che la soluzione intera piú vicina a quella restituita dal metodo del simplesso sia contenuta nella regione ammissibile del problema, non vi é garanzia che tale soluzione intera sia la soluzione ottimale.

$$Z = x_1 + 5x_2$$
$$x_1 + 10x_2 \leq 20$$
$$x_1 \leq 2$$
$$x_1 \geq 0,\ x_2 \geq 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Si consideri il problema di programmazione lineare intera sopra presentato. Se si tralascia il vincolo che ciascuna variabile debba essere intera e si applica il metodo del simplesso, si ottiene la soluzione $\left(2, \frac{9}{5}\right)$, alla quale é associato il valore della funzione obiettivo $Z = 11$. Approssimando tale soluzione alla soluzione intera piú vicina si ottiene la soluzione ammissibile $(2, 1)$. Questa é peró molto lontana dall'essere la soluzione ottima del problema, che é $(0, 2)$. Infatti, per la soluzione $(2, 1)$ si ha $Z = 7$, mentre per la soluzione $(0, 2)$ si ha $Z = 10$.

Il fatto che i problemi di programmazione binaria abbiano un numero di soluzioni finito potrebbe indurre a pensare che questi possano essere risolti semplicemente per esaustione, ovvero calcolando il valore della funzione obiettivo per ciascuna possibile soluzione ammissibile e cercando quella che ne massimizza o ne minimizza il valore. Questo é certamente vero, ma occorre ricordare che il numero di soluzioni ammissibili di un problema di programmazione lineare aumenta esponenzialmente con l'aumentare delle variabili, e anche per problemi di dimensione contenuta valutare ogni singola soluzione possibile sarebbe impraticabile.

Si consideri un problema di programmazione lineare intera avente $n = 100$ variabili binarie e nessun vincolo. Il numero totale di possibili soluzioni di tale problema é pari a $2^{100} = 1,26 \times 10^{30}$. Assumendo che il valore della funzione obiettivo restituito da una soluzione possa essere calcolato in 10^{-11} secondi (stima estremamente generosa), per controllare ogni singola soluzione di tale problema sarebbero comunque necessari circa 400 anni di computazione ininterrotta.

Tale approccio funzionerebbe se fosse possibile ridurre il numero di potenziali soluzioni ottimali ad una quantitá sufficientemente maneggevole. Il metodo **Branch-and-Bound**, presentato di seguito, opera precisamente in questo modo. Tale metodo partiziona l'intero insieme delle soluzioni ammissibili in insiemi sempre piú piccoli, determina un limite superiore per la soluzione del problema nel sottoinsieme e scarta i sottoinsiemi che non possono contenere la soluzione ottima. Queste tre fasi vengono chiamate rispettivamente **branching**, **bounding** e **fathoming**.

- Un singolo passo di branching prevede di fissare il valore di una delle variabili e considerare tutti i sottoproblemi che ne conseguono. Nel caso della programmazione binaria una variabile puó assumere soltanto due valori, pertanto un passo di branching applicato ad un problema di programmazione binaria genera due sottoproblemi ad ogni iterazione.
  La variabile che viene usata per eseguire questa suddivisione ad ogni iterazione fissandone il valore é chiamata **variabile di branching**. La scelta di una certa variabile di branching piuttosto che un'altra non inficia la correttezza del metodo branch-and-bound, ma solo la velocitá con la quale questo viene terminato. Sebbene esistano diverse procedure raffinate con le quali stimare quale sia la variabile di branching che fará terminare il metodo piú in fretta possibile, per semplicitá é possibile scegliere sempre la i-esima variabile per la i-esima iterazione.
  I sottoproblemi che vengono mano a mano generati possono essere descritti mediante una struttura ad albero, chiamata **albero delle soluzioni**. Ciascun nodo rappresenta il valore a cui una certa variabile viene fissata, ciascun i-esimo livello dell'albero rappresenta i valori assunti dalla variabile i-esima e ciascun arco rappresenta la generazione di un nuovo sottoproblema.
- Per ciascun sottoproblema costruito a partire dalla fase di branching si deve ottenere un limite (*bound*) su quanto buona possa essere la migliore soluzione ammissibile. Il metodo standard per fare questo consiste nel compiere un **rilassamento** del sottoproblema, ovvero sostituirne una ipotesi con un'altra meno stringente. Nel caso della programmazione binaria, il rilassamento piú usato prevede di imporre che le variabili possano anche assumere valori reali e non solo valori binari; in questo modo, una soluzione al sottoproblema puó essere ottenuta applicando il metodo del simplesso.
  Come giá detto in precedenza, tale soluzione non sará necessariamente intera e nemmeno necessariamente ottimale, ma ció che interessa é l'utilizzarla come limite. Infatti, se il problema originale é un problema di ottimizzazione di massimo, la sua soluzione ottimale non potrá mai restituire un valore della funzione obiettivo superiore a quello della soluzione ottenuta dal problema rilassato (usando il metodo del simplesso). Allo stesso modo, se é un problema di ottimizzazione di minimo, la sua soluzione ottimale non potrá mai restituire un valore della funzione obiettivo inferiore a quello della soluzione ottenuta dal problema rilassato.
  La migliore soluzione ammissibile finora trovata per il problema originario prende il nome di **soluzione incombente**. Di tale soluzione si ha interesse a memorizzare il relativo valore della funzione obiettivo, indicato con Z^*. Inizialmente, a tale quantitá viene assegnato $-\infty$, e nel corso delle successive iterazioni tale valore viene migliorato.
- Un sottoproblema puó essere soppresso (*fathomed*) se rispetta diversi criteri. Innanzitutto, se il metodo del simplesso applicato ad un sottoproblema rilassato restituisce una soluzione avente bound inferiore a quello della soluzione incombente, allora tale problema puó essere scartato, perché le soluzioni ottimali che tale sottoproblema e tutti i sottoproblemi che ne derivano non forniranno mai soluzioni migliori di quella incombente.
  Inoltre, se il metodo del simplesso applicato ad un sottoproblema rilassato restituisce direttamente una soluzione intera, tale soluzione sará anche la soluzione ottimale per il problema non rilassato. In questo caso, qualsiasi rilassamento ulteriore del sottoproblema non porterá mai ad una soluzione migliore di quanto questa possa essere. Inoltre, se tale soluzione intera ha un valore della funzione obiettivo maggiore di quello della soluzione incombente attuale, allora la soluzione intera in questione puó essere scelta come nuova soluzione incombente.
  Infine, molto semplicemente, se un sottoproblema non ha alcuna soluzione ammissibile puó essere scartato, perché anche tutti i sottoproblemi che ne derivano non ne avranno.

Alla luce di quanto detto finora, é possibile esporre l'algoritmo per l'applicazione del metodo branch-and-bound ai problemi di programmazione binaria:

1. Z^*, il valore della funzione obiettivo per la soluzione incombente, viene inizializzato a $-\infty$;
2. Se il problema originale é un problema di ottimizzazione di minimo, lo si converta in un problema di ottimizzazione di massimo equivalente;
3. Si convertano tutti i vincoli del problema originale nella forma $x_j \in \{0, 1\}$ in vincoli nella forma $0 \leq x_j \leq 1$.
4. *Branching*. Si scelga uno fra tutti i possibili sottoproblemi ancora aperti (se questa é la prima iterazione, il problema é uno solo e coincide con il problema originale). Si fissi il valore della i-esima variabile, generando cosí due sottoproblemi;
5. *Bounding*. Per ogni nuovo sottoproblema si ottenga un limite superiore alla soluzione del problema originale (applicando, ad esempio, il metodo del simplesso). Se la soluzione approssimata cosí ottenuta non fornisce un valore intero per la soluzione obiettivo, tale valore viene approssimato per difetto;
6. *Fathoming*. Per ogni nuovo sottoproblema si valuti se é possibile scartarlo. Questo avviene se é rispettato uno dei tre criteri:
   - Il sottoproblema rilassato non possiede alcuna soluzione ottima;
   - La soluzione fornita dal sottoproblema rilassato restituisce un valore per la funzione obiettivo inferiore a quello della soluzione incombente;
   - La soluzione fornita dal sottoproblema rilassato é una soluzione intera.
7. Se la soluzione approssimata fornita dal sottoproblema rilassato é una soluzione intera ed il valore della funzione obiettivo a questa associata é superiore al valore di Z^*, tale valore sostituisce Z^*;
8. Se vi sono ancora dei sottoproblemi da eliminare, l'algoritmo riprende dal punto 4, altrimenti si procede oltre;
9. L'algoritmo termina e Z^* é il valore della funzione obiettivo associato alla soluzione ottimale. Tale soluzione puó venire eventualmente ricostruita ripercorrendo a ritroso l'albero delle soluzioni.

Si consideri il problema di programmazione binaria presentato di seguito. Una soluzione ottimale per tale problema puó essere ottenuta applicando il metodo branch-and-bound.
Tale problema é un problema di ottimizzazione di massimo, pertanto é giá nella forma desiderata. Occorre sostituire solamente i vincoli $x_j \in \{0, 1\}$ con vincoli $0 \leq x_j \leq 1$.
Impostando la variabile x_1 prima a 0 e poi a 1 del problema originale si ottengono i due sottoproblemi seguenti:

$$\begin{aligned}
&\max Z = 9x_1 + 5x_2 + 6x_3 + 4x_4 \\
&6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\
&x_3 + x_4 \leq 1 \\
&-x_1 + x_3 \leq 0 \\
&-x_2 + x_4 \leq 0 \\
&x_j \in \{0, 1\} \text{ per } j = 1, 2, 3, 4
\end{aligned}$$

$$\begin{aligned}
&\max Z = 5x_2 + 6x_3 + 4x_4 \\
&3x_2 + 5x_3 + 2x_4 \leq 10 \\
&x_3 + x_4 \leq 1 \\
&x_3 \leq 0 \\
&-x_2 + x_4 \leq 0 \\
&0 \leq x_j \leq 1 \text{ per } j = 2, 3, 4
\end{aligned}$$

$$\begin{aligned}
&\max Z = 9 + 5x_2 + 6x_3 + 4x_4 \\
&3x_2 + 5x_3 + 2x_4 \leq 4 \\
&x_3 + x_4 \leq 1 \\
&x_3 \leq 1 \\
&-x_2 + x_4 \leq 0 \\
&0 \leq x_j \leq 1 \text{ per } j = 2, 3, 4
\end{aligned}$$

Il primo ha soluzione ottima $(1, 0, 1)$ con $Z = 9$, e puó essere eliminato perché rispetta il terzo criterio di fathoming. Il secondo ha soluzione ottima $\left(\frac{4}{5}, 0, \frac{4}{5}\right)$, con $Z = 16.2$ (arrotondato a 16), e non rispetta alcun criterio di fathoming. Inoltre, avendo il primo problema una soluzione intera il cui valore della funzione obiettivo é superiore a quello di Z^*, si impone $Z^* = 9$.
L'unico sottoproblema rimasto é quello con $x_1 = 1$. Impostando la variabile x_2 prima a 0 e poi a 1 in quest'ultimo si ottengono i due sottoproblemi seguenti:

$$\begin{aligned}
&\max Z = 9 + 6x_3 + 4x_4 \\
&5x_3 + 2x_4 \leq 4 \\
&x_3 + x_4 \leq 1 \\
&x_3 \leq 1 \\
&x_4 \leq 0 \\
&0 \leq x_j \leq 1 \text{ per } j = 3, 4
\end{aligned}$$

$$\begin{aligned}
&\max Z = 14 + 6x_3 + 4x_4 \\
&5x_3 + 2x_4 \leq 1 \\
&x_3 + x_4 \leq 1 \\
&x_3 \leq 1 \\
&x_4 \leq 1 \\
&0 \leq x_j \leq 1 \text{ per } j = 3, 4
\end{aligned}$$

Il primo ha soluzione ottima $\left(\frac{4}{5}, 0\right)$, con $Z = 13.2$ (arrotondato a 13), mentre il secondo ha soluzione ottima $\left(0, \frac{1}{2}\right)$, con $Z = 16$. Nessuno dei due rispetta alcun criterio di fathoming, pertanto vanno mantenuti entrambi. Impostando la variabile x_3 prima a 0 e poi a 1 in entrambi si ottengono i quattro sottoproblemi seguenti:

$$\begin{aligned}
&\max Z = 9 + 4x_4 \\
&2x_4 \leq 4 \\
&x_4 \leq 1 \\
&0 \leq 1 \\
&x_4 \leq 0 \\
&0 \leq x_4 \leq 1
\end{aligned}$$

$$\begin{aligned}
&\max Z = 15 + 4x_4 \\
&2x_4 \leq -1 \\
&x_4 \leq 0 \\
&1 \leq 1 \\
&x_4 \leq 0 \\
&0 \leq x_4 \leq 1
\end{aligned}$$

$$\begin{aligned}
&\max Z = 14 + 4x_4 \\
&2x_4 \leq 1 \\
&x_4 \leq 1 \\
&0 \leq 1 \\
&x_4 \leq 1 \\
&0 \leq x_4 \leq 1
\end{aligned}$$

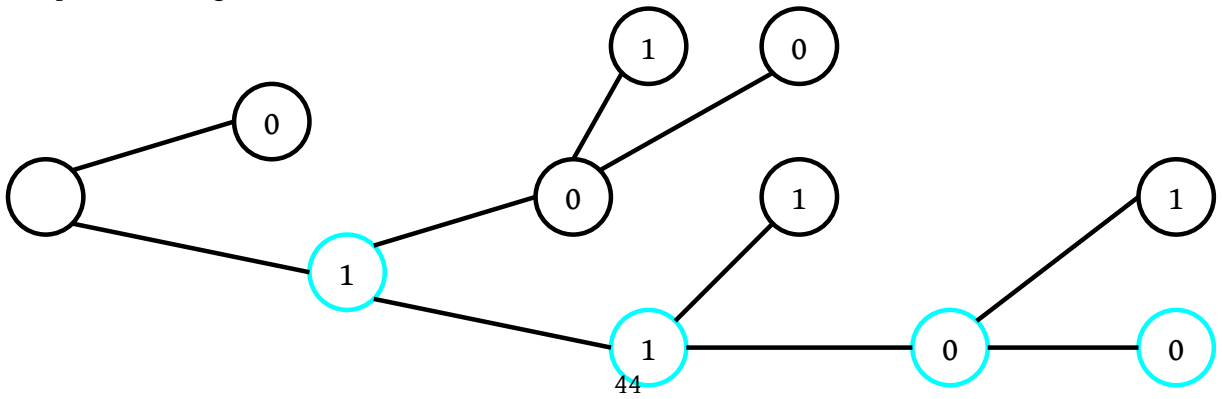
$$\begin{aligned}
&\max Z = 20 + 4x_4 \\
&2x_4 \leq -4 \\
&x_4 \leq 0 \\
&1 \leq 1 \\
&x_4 \leq 1 \\
&0 \leq x_4 \leq 1
\end{aligned}$$

Il primo, il secondo ed il quarto problema non hanno alcuna soluzione ammissibile, pertanto possono essere scartati. Il terzo ha soluzione ottima $\left(\frac{1}{2}\right)$ con $Z = 16$. Impostando la variabile x_4 prima a 0 e poi a 1 in quest'ultimo si ottengono i due sottoproblemi seguenti:

$$\begin{aligned}
&\max Z = 14 \\
&0 \leq 1, 0 \leq 1, 0 \leq 1
\end{aligned}$$

$$\begin{aligned}
&\max Z = 24 \\
&2 \leq -4, 1 \leq 0, 1 \leq 1
\end{aligned}$$

Il primo ha soluzione ottima intera (degenere) con $Z = 14$ e puó essere eliminato perché rispetta il terzo criterio di fathoming. Il secondo, invece, non ha alcuna soluzione ammissibile, pertanto puó essere scartato. Inoltre, avendo il primo problema una soluzione intera il cui valore della funzione obiettivo é superiore a quello di Z^*, si impone $Z^* = 14$.
Non esistendo piú alcun sottoproblema aperto, l'algoritmo termina. Ripercorrendo a ritroso l'albero delle soluzioni, si ottiene la soluzione ottimale per il problema originale $(1, 1, 0, 0)$.



3.4 Metodo Branch-and-Bound per la programmazione intera

L'algoritmo branch-and-bound applicato alla risoluzione di problemi di programmazione binaria puó essere rielaborato per la risoluzione di problemi di programmazione lineare intera pura. Per adattare l'algoritmo branch-and-bound a questo tipo di problemi é sufficiente operare tre modifiche.
Il primo cambiamento riguarda la scelta della variabile in esame in ciascuna iterazione. Nell'algoritmo branch-and-bound per la programmazione binaria ad ogni i-esima iterazione viene sempre presa in esame l'i-esima variabile. Nell'algoritmo branch-and-bound per la PLI viene considerata la prima variabile disponibile che nella soluzione per l'iterazione corrente non assume un valore intero. Puó infatti capitare che una stessa variabile venga scelta piú di una volta in iterazioni successive.
Il secondo cambiamento riguarda il valore che viene assegnato alla variabile in esame nell'iterazione corrente. Nell'algoritmo branch-and-bound per la programmazione binaria ad ogni i-esima iterazione vengono generati due sottoproblemi: uno in cui $x_i = 1$ ed uno in cui $x_i = 0$. Se si applicasse lo stesso approccio per la programmazione intera si dovrebbero creare tanti sottoproblemi quanti sono i possibili valori che tale variabile puó assumere; sebbene questo sia possibile, spesso risulta in un numero di sottoproblemi troppo grande per poter essere gestibile. L'approccio usato dall'algoritmo branch-and-bound per la PLI prevede di generare comunque due sottoproblemi, uno in cui viene aggiunto il vincolo supplementare $x_i \leq \lfloor x_i \rfloor$ ed uno in cui viene aggiunto il vincolo supplementare $x_i \geq \lceil x_i \rceil$.
Il terzo cambiamento riguarda il passo di bounding. Se nell'algoritmo branch-and-bound per la programmazione binaria il confronto con il valore della soluzione incombente veniva fatto con l'approssimazione (se necessario) del valore di Z per la soluzione in esame, mentre nell'algoritmo branch-and-bound per la PLI il confronto viene fatto direttamente con il valore di Z non approssimato.

1. Z^*, il valore della funzione obiettivo per la soluzione incombente, viene inizializzato a $-\infty$;
2. Se il problema originale é un problema di ottimizzazione di minimo, lo si converta in un problema di ottimizzazione di massimo equivalente;
3. *Branching*. Si scelga uno fra tutti i possibili sottoproblemi ancora aperti (se questa é la prima iterazione, il problema é uno solo e coincide con il problema originale). Si scelga la prima variabile che non assume un valore intero nella soluzione del problema corrente, sia questa x_i. Vengono generati due sottoproblemi: uno in cui é aggiunto il vincolo $x_i \leq \lfloor x_i \rfloor$ ed uno in cui é aggiunto il vincolo $x_i \leq \lceil x_i \rceil$;
4. *Bounding*. Per ogni nuovo sottoproblema si ottenga un limite superiore alla soluzione del problema originale (applicando, ad esempio, il metodo del simplesso);
5. *Fathoming*. Per ogni nuovo sottoproblema si valuti se é possibile scartarlo. Questo avviene se é rispettato uno dei tre criteri:

   - Il sottoproblema rilassato non possiede alcuna soluzione ottima;
   - La soluzione fornita dal sottoproblema rilassato restituisce un valore per la funzione obiettivo inferiore a quello della soluzione incombente;
   - La soluzione fornita dal sottoproblema rilassato é una soluzione intera.
6. Se la soluzione approssimata fornita dal sottoproblema rilassato é una soluzione intera ed il valore della funzione obiettivo a questa associata é superiore al valore di Z^*, tale valore sostituisce Z^*;
7. Se vi sono ancora dei sottoproblemi da eliminare, l'algoritmo riprende dal punto 3, altrimenti si procede oltre;
8. L'algoritmo termina e Z^* é il valore della funzione obiettivo associato alla soluzione ottimale. Tale soluzione puó venire eventualmente ricostruita ripercorrendo a ritroso l'albero delle soluzioni.

Capitolo 4
Programmazione nonlineare

4.1 Modello di programmazione nonlineare

Sebbene l'ipotesi di linearitá permetta di semplificare notevolmente il modello di programmazione lineare, in diverse classi di problemi questa assunzione rende il modello troppo semplicistico. In questi casi, occorre rinunciare all'ipotesi di linearitá e ammettere che uno o piú elementi del modello (la funzione obiettivo, i vincoli o entrambi) possano essere funzioni non lineari. Questo significa che, in modelli di questo tipo, nella funzione obiettivo e/o nei vincoli possono comparire variabili di grado superiore al primo.
Problemi in cui la funzione obiettivo e/o uno o piú vincoli sono funzioni non lineari prendono il nome di **problemi di programmazione nonlineare** (**PNL**). La forma standard di problemi di programmazione nonlineare é essenzialmente la medesima dei problemi di programmazione lineare:

$$\text{Determinare } \boldsymbol{x}^* = (x_1, x_2, ..., x_n) \text{ tale da massimizzare } f(\boldsymbol{x}) \text{ soggetto ai vincoli } \begin{cases} g_i(\boldsymbol{x}) \leq b_i \; i = 1, 2, ..., m \\ \boldsymbol{x} \geq 0 \end{cases}$$

Dove $f(\boldsymbol{x})$ e $g_i(\boldsymbol{x})$ sono funzioni note nelle n variabili decisionali.
La programmazione nonlineare presenta diverse sfide in piú rispetto alla programmazione lineare. Innanzitutto, il metodo del simplesso risulta inapplicabile perché la natura della regione ammissibile non é la medesima della programmazione lineare.
Se un problema di programmazione matematica presenta un vincolo non lineare, la regione ammissibile non é piú un politopo convesso, perché uno o piú lati di quest'ultima sono composti da coniche. Questo significa non solo che la soluzione ottimale potrebbe non trovarsi su uno dei vertici della regione ammissibile, ma che quest'ultima potrebbe non avere alcun vertice.
D'altro canto, se un problema di programmazione matematica presenta una funzione obiettivo non lineare, il valore ottimale della funzione obiettivo non é piú dato dall'intersezione di un piano n-dimensionale con la regione ammissibile ma dall'intersezione di una conica n-dimensionale. Questo significa non solo che la soluzione ottimale potrebbe non trovarsi su uno dei vertici della regione ammissibile, ma che la soluzione ottimale potrebbe non trovarsi nemmeno sulla frontiera di quest'ultima.
Piú in generale, il problema sussiste perché i vincoli funzionali di un problema di programmazione nonlineare non sono necessariamente funzioni convesse. Dato che la regione ammissibile di un problema di programmazione matematica é data dall'intersezione dei vincoli, se tali vincoli non sono funzioni convesse allora la regione ammissibile non sará un insieme convesso. Se questo accade, un massimo locale nella regione ammissibile non é necessariamente anche un massimo globale, e gli algoritmi di programmazione nonlineare non sono in grado di fare distinzione tra i due [1].
Vi sono diverse classi di problemi di programmazione non lineare, a seconda delle caratteristiche di $f(\boldsymbol{x})$ e $g_i(\boldsymbol{x})$. Per ciascuna di queste, esistono algoritmi ad hoc in grado di risolverli: questi fanno uso di una strategia di tipo **line-search**, ovvero generano una successione di punti all'interno dello spazio $\mathbb{R}^n$ in modo che il punto successivo é ottenuto a partire dal precedente muovendosi lungo una **direzione di salita** o **direzione di discesa**.
Data una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ ed un punto $\boldsymbol{x} \in \mathbb{R}^n$ in cui f é differenziabile, una direzione di discesa é definita come un generico vettore $\boldsymbol{v} \in \mathbb{R}^n$ tale per cui il prodotto scalare fra quest'ultimo ed il gradiente di f nel punto $\boldsymbol{x}$ é strettamente negativo. Allo stesso modo, una direzione di salita é definita come un generico vettore $\boldsymbol{v} \in \mathbb{R}^n$ tale per cui il prodotto scalare fra quest'ultimo ed il gradiente di f nel punto $\boldsymbol{x}$ é strettamente positivo.
Data una funzione $f : X \subseteq \mathbb{R}^n \mapsto \mathbb{R}$ ed un punto $\boldsymbol{x} \in \mathbb{R}^n$ in cui f é differenziabile, gli algoritmi che applicano la strategia line-search possono essere tutti ridotti a questo insieme di passaggi:

1. Un indice k, che indica l'iterazione corrente, viene inizializzato a 0. Il punto $\boldsymbol{x}^k$ indica l'approssimazione finora trovata per il massimo o il minimo di f;
2. Viene calcolata la direzione di discesa $\boldsymbol{d}^k$ sulla base di $\boldsymbol{x}^k$;
3. Viene determinato il punto successivo $\boldsymbol{x}^{k+1}$ lungo la direzione $\boldsymbol{x}^k$ come $\boldsymbol{x}^k \pm \alpha^k \boldsymbol{d}^k$, dove α^k é una quantitá scalare positiva chiamata **step-size**;
4. Se $\boldsymbol{x}^{k+1}$ é una approssimazione accettabile per il massimo o il minimo di f l'algoritmo termina, altrimenti si riprende dal punto 2 con $\boldsymbol{x}^{k+1}$ come nuovo punto in considerazione.

Di fatto, la differenza fra gli algoritmi di line-search sta nel modo in cui questi calcolano lo step-size.

1. La distinzione fra massimi locali e massimi globali non si presentava nel caso della programmazione lineare. Questo avviene perché, come dimostrato in precedenza, le funzioni lineari sono sempre convesse (e anche sempre concave), pertanto la regione ammissibile di un problema di programmazione lineare é sempre un insieme convesso.

4.2 Ottimizzazione non vincolata in una sola variabile

Il caso piú semplice di programmazione nonlineare é un problema di ottimizzazione in una sola variabile, senza vincoli e dove la funzione obiettivo $f(x)$ é concava. In questo caso, la soluzione ottimale é la soluzione (o le soluzioni) che rendono nulla la derivata prima della funzione obiettivo.

$$x^* = \frac{df}{dx} f(x)$$

Se la funzione $f(x)$ é sufficientemente semplice, tale valore puó essere calcolato analiticamente. Se la funzione é particolarmente ostica, é comunque possibile ottenere una approssimazione della soluzione ottima applicando specifici algoritmi.
Algoritmi di questo tipo si basano sul costruire una sequenza di punti che convergono ad una soluzione ottima. Ad ogni iterazione, a partire dal punto corrente si esegue una ricerca sistematica che culmina con l'identificazione di un punto migliore. La procedura termina quando viene ottenuta un'approssimazione della soluzione ottima sufficientemente buona.

4.2.1 Metodo di bisezione

Sia $f(x)$ una funzione continua e derivabile. Dalla definizione di derivata, si ha:

$$\frac{df(x)}{dx} > 0 \text{ se } x < x^* \qquad \frac{df(x)}{dx} = 0 \text{ se } x = x^* \qquad \frac{df(x)}{dx} < 0 \text{ se } x > x^*$$

Sia x' una certa approssimazione della soluzione ottima. Se la derivata prima valutata in x' é positiva, allora x' é inferiore alla soluzione ottima. Questo significa che qualsiasi punto nell'intervallo $[x', x^*)$ é una migliore approssimazione per x^* di quanto possa esserlo x'. Allo stesso modo, se la derivata prima valutata in x' é negativa, allora x' é superiore alla soluzione ottima. Questo significa che qualsiasi punto nell'intervallo $(x^*, x']$ é una migliore approssimazione per x^* di quanto possa esserlo x'.
Il **metodo di bisezione** si basa sul ripetere questo processo di raffinamento fino ad ottenere una approssimazione della soluzione ottima che dista da questo meno di un valore fissato ε. Indicando con x_L una approssimazione per difetto di x^* e con x_R una approssimazione per eccesso di x^*, l'algoritmo procede come segue:

1. Si fissi un certo ε;
2. Si calcolino due valori iniziali per x_L e per x_R. Tali valori devono essere scelti tali per cui la funzione é continua in $[x_L, x_R]$ e $f(x_L)$ e $f(x_R)$ hanno segno opposto;
3. Si calcoli l'approssimazione x' di x^* come $x' = x_R - x_L \ / \ 2$;
4. Si calcoli la derivata prima di $f(x)$ nel punto x'. Se tale valore é positivo, allora x_L viene sostituito con x', mentre se é negativo x_R viene sostituito con x';
5. Se $x_R - x_L < 2\varepsilon$, allora l'algoritmo termina, perché x' é una approssimazione accettabile di x^*. Altrimenti, si riprende dal punto 3.

Si voglia calcolare un valore ottimo della funzione $f(x) = 12x - 3x^4 - 2x^6$. La derivata prima di tale funzione é $f'(x) = 12(1 - x^3 - x^5)$, mentre la derivata seconda é $f''(x) = -12(3x^2 + 5x^4)$.
Si noti come non sia possibile ricavare analiticamente il valore ottimo della funzione, ovvero calcolare direttamente lo zero della derivata prima, perché tale funzione é un polinomio di quinto grado. Essendo peró $f(x)$ concava, dato che la derivata seconda é sempre negativa, é possibile applicare il metodo di bisezione.
Si fissi $\varepsilon = 0.01$. Come valori iniziali per x_L e x_R é possibile utilizzare rispettivamente i valori -1 e 2: infatti, $f(-1) = 36$ e $f(2) = -24$. Il valore x' viene allora inizializzato a $(2 + (-1)) \ / \ 2 = 0.5$. Applicando il metodo di bisezione, si ottiene il valore approssimato per x^* pari a 0.833984375

Iterazione	**Derivata in x'**	x_L	x_R	**Nuovo x'**
0		-1	2	0.5
1	+10.13	0.5	2	1.25
2	-48.06	0.5	1.25	0.875
3	-2.19	0.5	0.875	0.6875
4	+6.26	0.6875	0.875	0.78125
5	+2.79	0.78125	0.875	0.828125
6	+0.51	0.828125	0.875	0.8515625
7	-0.78	0.828125	0.8515625	0.83984375
8	-0.12	0.828125	0.83984375	0.833984375

4.2.2 Metodo di Newton

Il metodo di bisezione é molto semplice, ma converge molto lentamente ad una soluzione ottima. Un metodo tanto piú veloce quanto piú complesso, che fa uso della derivata seconda della funzione da massimizzare, é il cosiddetto **Metodo di Newton**.
L'idea alla base del metodo di Newton é quella di approssimare, nell'intorno del punto corrente, la funzione $f(x)$ con una funzione quadratica, e poi massimizzare (o minimizzare) la funzione approssimata per ottenere un nuovo punto. Questa **approssimazione quadratica** viene fatta troncando la serie di Taylor al secondo termine. In particolare, se x_i é l'approssimazione di x^* alla i-esima iterazione, allora la serie di Taylor troncata per x_{i+1} é:

$$f(x_{i+1}) \approx f(x_i) + f'(x_i)(x_{i+1}-x_i) + \frac{f''(x_i)(x_{i+1}-x_i)^2}{2}$$

Il punto x_i é fissato all'inizio della i-esima iterazione, e quindi le quantitá $f(x_i)$, $f'(x_i)$ e $f''(x_i)$ sono costanti nella funzione approssimante a destra. Quindi, essa é una funzione quadratica nella variabile x_{i+1}. Inoltre, questa funzione quadratica é una buona approssimazione di $f(x_{i+1})$ nell'intorno di x_i ed il suo valore e quello della sua derivata prima e seconda sono esattamente gli stessi per $x_{i+1} = x_i$.
Tale funzione quadratica puó essere massimizzata nella maniera consueta ponendo uguale a zero la sua derivata prima e risolvendo per x_{i+1}. Questa derivata prima allora:

$$\frac{d}{dx}(f(x_{i+1})) \approx \frac{d}{dx}\left(f(x_i) + f'(x_i)(x_{i+1}-x_i) + \frac{f''(x_i)(x_{i+1}-x_i)^2}{2}\right) \approx f'(x_{i+1}) \approx f'(x_i) + f''(x_i)(x_{i+1}-x_i)$$

Imponendo che la derivata prima della funzione a destra sia zero e risolvendo per x_{i+1}, si ottiene:

$$f'(x_{i+1}) = 0 \;\Rightarrow\; f'(x_i) + f''(x_i)(x_{i+1}-x_i) = 0 \;\Rightarrow\; f'(x_i) + f''(x_i)x_{i+1} - f''(x_i)x_i = 0 \;\Rightarrow\; f''(x_i)x_{i+1} = f''(x_i)x_i - f'(x_i) \;\Rightarrow\; x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$$

Il metodo termina quando la distanza fra x_{i+1} e x_i é inferiore ad un valore fissato ε. L'algoritmo procede come segue:

1. Si fissi un certo ε;
2. Si ponga $i = 1$;
3. Si scelga un punto iniziale x_1;
4. Si calcolino $f'(x_i)$ e $f''(x_i)$;
5. Si calcoli x_{i+1} come $x_i - (f'(x_i) \,/\, f''(x_i))$;
6. Se $| x_{i+1} - x_i | \leq \varepsilon$, allora l'algoritmo termina, perché $x + 1$ é una approssimazione accettabile di x^*. Altrimenti, si aumenta i di 1 e si riprende dal punto 4.

Il problema precedente puó essere affrontanto applicando il metodo di Newtown, ottenendo un'approssimazione per x^* molto piú rapidamente di quanto possa fare il metodo di bisezione. Fissato $\varepsilon = 0.00001$, si ha:

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} = x_i - \frac{12\left(1-x_i^3-x_i^5\right)}{-12\left(3x_i^2+5x_i^4\right)} = x_i + \frac{1-x_i^3-x_i^5}{3x_i^2+5x_i^4}$$

Iterazione	x_i	$f'(x_i)$	$f''(x_i)$	x_{i+1}
1	1	-12	-96	0.875
2	0.875	-2.1940	-62.733	0.84003
3	0.84003	-0.1325	-55.279	0.83763
4	0.83763	-0.0006	-54.795	0.83762

4.3 Ottimizzazione non vincolata in piú variabili

Si consideri ora il problema di massimizzare una funzione $f(\boldsymbol{x})$ a piú variabili, con $\boldsymbol{x} = (x_1, x_2, ..., x_n)$, in cui non esistono vincoli sui valori ammissibili. In questo caso, la soluzione ottimale é la soluzione (o le soluzioni) che rendono nulle tutte le derivate parziali della funzione obiettivo. Sia $f(\boldsymbol{x})$ una funzione in n variabili. Il vettore avente per componenti le derivate n parziali prime di $f(\boldsymbol{x})$ prende il nome di **vettore gradiente**, o semplicemente **gradiente**:

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n}\right)$$

Se $f(\boldsymbol{x})$ é differenziabile nel punto $\boldsymbol{x}_0$, allora esiste $\nabla f(\boldsymbol{x}_0)$, altrimenti il gradiente della funzione in tale punto non é definito.

Se la funzione $f(\boldsymbol{x})$ é sufficientemente semplice ed é differenziabile nell'intorno di $\boldsymbol{x}^*$, il suo massimo puó essere calcolato semplicemente ponendo a zero tutte le componenti di $\nabla f(\boldsymbol{x})$ e risolvendo per ciascuna x_i. Se questo non é possibile, una soluzione approssimata puó essere comunque ricavata applicando specifici algoritmi.

4.3.1 Metodo del gradiente

Si ricordi che per definizione di gradiente la variazione infinitesimale di $\boldsymbol{x}$ che massimizza il tasso a cui la funzione $f(\boldsymbol{x})$ cresce é una variazione proporzionale a $\nabla f(\boldsymbol{x})$. Una efficiente procedura di ricerca consisterebbe quindi nel "muoversi" nella direzione del gradiente identificando punti di $f(\boldsymbol{x})$ in cui il gradiente é sempre piú vicino allo zero per tutte le sue componenti.
Di norma non sarebbe pratico modificare $\boldsymbol{x}$ continuamente nella direzione di $\nabla f(\boldsymbol{x})$, perché questo richiederebbe di ricalcolare tutte le componenti del vettore gradiente ad ogni iterazione. Un approccio migliore consiste nel fissare una direzione, modificare soltanto la componente associata a tale direzione per poi passare ad un'altra. Tale approccio é utilizzato dal cosiddetto **Metodo del gradiente**.
Sia $\boldsymbol{x}'$ una certa approssimazione della soluzione ottima. Con questo approccio, ad ogni iterazione tale approssimazione viene sostituita con $\boldsymbol{x}' + t^*\nabla f(\boldsymbol{x}')$, dove t^* é il valore (positivo) che massimizza la quantitá $f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))$:

$$f(\boldsymbol{x}' + t^*\nabla f(\boldsymbol{x}')) = \max\{f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))\}$$

Si noti che $f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))$ é semplicemente $f(\boldsymbol{x})$ dove:

$$x_j = x'_j + t\left(\frac{\partial f}{\partial x_j}\right)_{x=x'} \text{ per } j = 1, 2, \ldots, n$$

e che queste espressioni per x_j dipendono solo da quantitá costanti e da t, quindi la funzione $f(\boldsymbol{x})$ é una funzione nella sola variabile t. Tale procedura viene ripetuta finché l'approssimazione ottenuta per tutte le componenti del gradiente restituisce un valore inferiore ad una soglia ε:

$$\left|\frac{\partial f}{\partial x_j}\right| \leq \varepsilon \text{ per } j = 1, 2, \ldots, n$$

Poiché $\boldsymbol{x}$ e $\nabla f(\boldsymbol{x})$ sono fissati e poiché $f(\boldsymbol{x})$ é concava, determinare t^* equivale a massimizzare una funzione concava nella singola variabile t. Questo puó essere fatto analiticamente (se possibile) oppure applicando le apposite procedure giá trattate, come il metodo di bisezione o il metodo di Newton.
L'algoritmo procede come segue:

1. Si fissi un certo ε;
2. Si scelga un punto iniziale $\boldsymbol{x}$;
3. Si esprima $f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))$ come funzione di t ponendo

$$x_j = x'_j + t\left(\frac{\partial f}{\partial x_j}\right)_{x=x'} \text{ per } j = 1, 2, \ldots, n$$

4. Si determini t^*, il valore di t che massimizza $f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))$, analiticamente o in mediante approssimazione;
5. Si sostituisca $\boldsymbol{x}'$ con $\boldsymbol{x}' + t^*\nabla f(\boldsymbol{x}')$;
6. Si calcoli $\nabla f(\boldsymbol{x}')$. Se ciascuna componente di tale vettore é sufficientemente vicina ad ε l'algoritmo termina, perché $\boldsymbol{x}'$ é una approssimazione accettabile di $\boldsymbol{x}^*$. Altrimenti si riprende dal punto 3.

Se la funzione $f(\boldsymbol{x})$ fosse invece convessa e si volesse minimizzarla, occorrerebbe modificare la procedura richiedendo di muoversi, ad ogni iterazione, nella direzione opposta a quella del gradiente.
In questo caso, l'approssimazione successiva per $\boldsymbol{x}^*$ non si ottiene sostituendo $\boldsymbol{x}'$ con $\boldsymbol{x}' + t^*\nabla f(\boldsymbol{x}')$, bensí con $\boldsymbol{x}' - t^*\nabla f(\boldsymbol{x}')$. Inoltre, t^* non sarebbe piú il valore che massimizza $f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))$, bensí quello che minimizza $f(\boldsymbol{x}' - t\nabla f(\boldsymbol{x}'))$.

Si consideri la funzione $f(\boldsymbol{x}) = 2xy + 2y - x^2 - 2y^2$. Tale funzione é concava, in quanto la sua matrice Hessiana é semidefinita negativa:

$$z^T Hz = [x \;\; y]\begin{bmatrix} \frac{\partial^2 f(x,y)}{\partial x^2} & \frac{\partial^2 f(x,y)}{\partial x \partial y} \\ \frac{\partial^2 f(x,y)}{\partial y \partial x} & \frac{\partial^2 f(x,y)}{\partial y^2} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = [x \;\; y]\begin{bmatrix} -2 & 2 \\ 2 & -4 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = -2x^2 + 4xy - 4y^2 = -2\left[(x-y)^2 + y^2\right]$$

Il punto che massimizza tale funzione puó essere trovato per via analitica, perché le componenti del relativo vettore gradiente sono semplici funzioni polinomiali. Infatti, si ha:

$$\nabla f(\boldsymbol{x}) = 0 \;\Rightarrow\; \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}\right) = 0 \;\Rightarrow\; (-2x + 2y, 2x + 2 - 4y) = 0 \;\Rightarrow\; \begin{cases} -2x + 2y = 0 \\ 2x + 2 - 4y = 0 \end{cases} \Rightarrow \begin{cases} x = 1 \\ y = 1 \end{cases}$$

Applicando il metodo del gradiente, si ottiene una approssimazione che si avvicina molto a tale valore.

Iterazione	$\boldsymbol{x}'$	$\nabla f(\boldsymbol{x}')$	$\boldsymbol{x}' + t\nabla f(\boldsymbol{x}')$	$f(\boldsymbol{x}' + t\nabla f(\boldsymbol{x}'))$	t^*	$\boldsymbol{x}' + t^*\nabla f(\boldsymbol{x}')$
1	$(0, 0)$	$(0, 2)$	$(0, 2t)$	$4t(1-2t)$	$\frac{1}{4}$	$\left(0, \frac{1}{2}\right)$
2	$\left(0, \frac{1}{2}\right)$	$(1, 0)$	$\left(t, \frac{1}{2}\right)$	$-t^2 + t + \frac{1}{2}$	$\frac{1}{2}$	$\left(\frac{1}{2}, \frac{1}{2}\right)$
3	$\left(\frac{1}{2}, \frac{1}{2}\right)$	$(0, 1)$	$\left(\frac{1}{2}, t + \frac{1}{2}\right)$	$-2t^2 + t + \frac{3}{4}$	$\frac{1}{4}$	$\left(\frac{1}{2}, \frac{3}{4}\right)$
4	$\left(\frac{1}{2}, \frac{3}{4}\right)$	$\left(\frac{1}{2}, 0\right)$	$\left(\frac{1}{2}t + \frac{1}{2}, \frac{3}{4}\right)$	$-\frac{1}{4}t^2 + \frac{1}{4}t + \frac{7}{8}$	$\frac{1}{2}$	$\left(\frac{3}{4}, \frac{3}{4}\right)$
5	$\left(\frac{3}{4}, \frac{3}{4}\right)$	$\left(0, \frac{1}{2}\right)$	$\left(\frac{3}{4}, \frac{1}{2}t + \frac{3}{4}\right)$	$-\frac{1}{2}t^2 + \frac{1}{4}t + \frac{15}{16}$	$\frac{1}{4}$	$\left(\frac{3}{4}, \frac{7}{8}\right)$
6	$\left(\frac{3}{4}, \frac{7}{8}\right)$	$\left(\frac{1}{4}, 0\right)$	$\left(\frac{1}{4}t + \frac{3}{4}, \frac{7}{8}\right)$	$-\frac{1}{16}t^2 + \frac{1}{16}t + \frac{31}{32}$	$\frac{1}{2}$	$\left(\frac{7}{8}, \frac{7}{8}\right)$
7	$\left(\frac{7}{8}, \frac{7}{8}\right)$	$\left(0, \frac{1}{4}\right)$	$\left(\frac{7}{8}, \frac{1}{4}t + \frac{7}{8}\right)$	$-\frac{1}{8}t^2 + \frac{1}{16}t + \frac{63}{64}$	$\frac{1}{4}$	$\left(\frac{7}{8}, \frac{15}{16}\right)$
8	$\left(\frac{7}{8}, \frac{15}{16}\right)$	$\left(\frac{1}{8}, 0\right)$	$\left(\frac{1}{8}t + \frac{7}{8}, \frac{15}{16}\right)$	$-\frac{1}{64}t^2 + \frac{1}{64}t + \frac{127}{128}$	$\frac{1}{2}$	$\left(\frac{15}{16}, \frac{15}{16}\right)$
9	$\left(\frac{15}{16}, \frac{15}{16}\right)$	$\left(0, \frac{1}{8}\right)$	$\left(\frac{15}{16}, \frac{1}{8}t + \frac{15}{16}\right)$	$-\frac{1}{32}t^2 + \frac{1}{64}t + \frac{255}{256}$	$\frac{1}{4}$	$\left(\frac{15}{16}, \frac{31}{32}\right)$
10	$\left(\frac{15}{16}, \frac{31}{32}\right)$	$\left(\frac{1}{16}, 0\right)$	$\left(\frac{1}{16}t + \frac{15}{16}, \frac{31}{32}\right)$	$-\frac{1}{256}t^2 + \frac{1}{256}t + \frac{511}{512}$	$\frac{1}{2}$	$\left(\frac{31}{32}, \frac{31}{32}\right)$

4.3.2 Metodo di Newton

Il metodo di Newton, presentato per la ricerca del massimo di una funzione in una sola variabile, puó essere esteso in maniera molto naturale a funzioni in piú variabili. Il metodo prevede sempre di calcolare un'approssimazione quadratica della funzione obiettivo $f(\boldsymbol{x})$, dove peró $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$. Questa funzione quadratica é ottenuta troncando al secondo termine la serie di Taylor nel punto corrente. Questa funzione approssimante viene quindi massimizzata/minimizzata per ottenere il nuovo punto da cui partire alla successiva iterazione.
Quando la funzione obiettivo é concava, l'approssimazione $\boldsymbol{x}'$ ha la forma:

$$\boldsymbol{x}' = \boldsymbol{x} - \left(\nabla^2 f(\boldsymbol{x})\right)^{-1}\nabla f(\boldsymbol{x}) = \boldsymbol{x} - H^{-1}\nabla f(\boldsymbol{x})$$

Dove H^{-1} é l'inversa della matrice Hessiana di $f(\boldsymbol{x})$.
Sia $\boldsymbol{x}_i$ l'approssimazione per $\boldsymbol{x}'$ ottenuta alla i-esima iterazione. Il metodo termina quando la distanza fra $\boldsymbol{x}_{i+1}$ e $\boldsymbol{x}_i$ (che sono ora vettori e non piú scalari) é inferiore ad un valore fissato ε. L'algoritmo procede come segue:

1. Si fissi un certo ε;
2. Si ponga $i = 1$;
3. Si scelga un punto iniziale $\boldsymbol{x}_1$;
4. Si calcolino il gradiente e l'Hessiana di $f(\boldsymbol{x})$ nel punto $\boldsymbol{x}_i$;
5. Si calcoli $\boldsymbol{x}_{i+1}$ come $\boldsymbol{x}_i - (\nabla^2 f(\boldsymbol{x}_i))^{-1}\nabla f(\boldsymbol{x}_i)$
6. Se $|\, \boldsymbol{x}_{i+1} - \boldsymbol{x}_i \,| \leq \varepsilon$, allora l'algoritmo termina, perché $\boldsymbol{x} + 1$ é una approssimazione accettabile di $\boldsymbol{x}^*$. Altrimenti, si aumenta i di 1 e si riprende dal punto 4.

4.4 Ottimizzazione con vincoli

Un problema di programmazione non lineare in cui sono presenti dei vincoli non ha un metodo generale in grado di essere risolti. É peró possibile impiegare diversi metodi che permettono di approssimarne le soluzioni.

4.4.1 Riduzione del numero di variabili libere

Si supponga che il problema di programmazione non lineare in esame abbia n variabili e presenti l vincoli di uguaglianza, con $n > l$. Nel caso in cui sia possibile esplicitare l variabili in funzione delle restanti $n-l$ utilizzando tali vincoli di uguaglianza é possibile semplificare il problema trasformandolo in un problema di programmazione non lineare in $n-l$ variabili.

Si consideri un problema di programmazione non lineare avente per funzione obiettivo min $(x_1-2)^2 + 2(x_2-1)^2$ e come unico vincolo il vincolo di uguaglianza $x_1 + 4x_2 = 3$. É possibile semplificare nettamente tale problema esplicitando x_1 nel vincolo di uguaglianza e sostituendolo nella funzione obiettivo, ottenendo un problema di programmazione non lineare nella sola variabile x_2 avente per funzione obiettivo min $18x_2^2-12x_2 + 3$.

Naturalmente questo approccio non é sempre applicabile, dato che non tutti i problemi di programmazione non lineare presentano vincoli di uguaglianza e, se ne hanno, non necessariamente questi permettono di rendere esplicite delle variabili.

4.4.2 metodo dei moltiplicatori di Lagrange

Si supponga che il problema di programmazione non lineare in esame abbia una certa funzione obiettivo min / max $f(x_1, ..., x_n)$ e sia soggetto ad m vincoli di uguaglianza $g_i(x_1, ..., x_n) = 0$ con $i = 1, ..., m$. Prende il nome di **funzione Lagrangiana**, o semplicemente **Lagrangiana**, la funzione in $n + m$ variabili:

$$L(x_1, ..., x_n, \lambda_1, ..., \lambda_m) = f(x_1, ..., x_n) + \sum_{i=0}^{m} \lambda_i \cdot g_i(x_1, ..., x_n)$$

Dove le n variabili $x_1, ..., x_n$ sono le medesime n variabili presenti nella funzione obiettivo e nei vincoli mentre le m variabili $\lambda_1, ..., \lambda_m$, ciascuna associata ad un vincolo, prendono il nome di **moltiplicatori di Lagrange**.

La Lagrangiana associata al problema di programmazione non lineare dell'esempio precedente é:

$$L(x_1, x_2, \lambda_1) = (x_1-2)^2 + 2(x_2-1)^2 + \lambda_1(x_1 + 4x_2-3)$$

I punti stazionari della Lagrangiana sono fortemente legati ai punti di massimo/minimo della funzione obiettivo del problema a cui é associata.

Sia P un problema di programmazione non lineare avente per funzione obiettivo min / max $f(x_1, ..., x_n)$ ed avente m vincoli $g_i(x_1, ..., x_n) = 0$ con $i = 1, ..., m$. Sia $L(\boldsymbol{x}, \lambda)$ la Lagrangiana associata a P; se $\boldsymbol{x}^* = (x_1^*, ..., x_n^*)$ é un punto stazionario per f, allora esistono m moltiplicatori di Lagrange $\lambda^* = (\lambda_1^*, ..., \lambda_m^*)$ tali per cui $(\boldsymbol{x}^*, \lambda^*)$ é un punto stazionario (non necessariamente dello stesso tipo) per $L(\boldsymbol{x}, \lambda)$.

Il teorema fornisce un possibile approccio per la risoluzione di un problema di programmazione non lineare con vincoli di uguaglianza: essendo la Lagrangiana una funzione sola, é possibile trasformare il problema in un problema di programmazione non lineare non vincolato, dove la soluzione é data dai punti che annullano il vettore gradiente della Lagrangiana.
Il punto di massimo della Lagrangiana é, come di consueto, il punto che annulla la derivata prima parziale rispetto a tutte le sue componenti, sia quelle relative alle variabili x che quelle relative ai moltiplicatori λ:

$$\frac{\partial L(\boldsymbol{x}^*, \lambda^*)}{\partial \lambda_i} = 0 \; j = 1, ..., m \qquad \frac{\partial L(\boldsymbol{x}^*, \lambda^*)}{\partial x_i} = 0 \; i = 1, ..., n$$

Si osservi come le condizioni a sinistra non siano altro che i vincoli stessi. Infatti:

$$\frac{\partial L(\boldsymbol{x}^*, \lambda^*)}{\partial \lambda_i} = 0 \Rightarrow \frac{\partial}{\partial \lambda_i}\left(f(x_1, ..., x_n) + \sum_{i=0}^{m} \lambda_i \cdot g_i(x_1, ..., x_n)\right) = 0 \Rightarrow \frac{\partial}{\partial \lambda_i}(f(x_1, ..., x_n)) + \frac{\partial}{\partial \lambda_i}\left(\sum_{i=0}^{m} \lambda_i \cdot g_i(x_1, ..., x_n)\right) = 0 \Rightarrow$$
$$\Rightarrow 0 + \frac{\partial}{\partial \lambda_i}(\lambda_1 g_1(x_1, ..., x_n)) + ... + \frac{\partial}{\partial \lambda_i}(\lambda_i g_i(x_1, ..., x_n)) + ... + \frac{\partial}{\partial \lambda_i}(\lambda_n g_n(x_1, ..., x_n)) = 0 \Rightarrow$$
$$\Rightarrow 0 + 0 + ... + \frac{\partial}{\partial \lambda_i}(\lambda_i g_i(x_1, ..., x_n)) + ... + 0 = 0 \Rightarrow g_i(x_1, ..., x_n) = 0$$

D'altro canto, le condizioni a destra possono essere riscritte come:

$$\frac{\partial L(\boldsymbol{x}^*, \lambda^*)}{\partial \lambda_i} = \frac{\partial f(\boldsymbol{x}^*)}{\partial x_i} + \sum_{i=1}^{m} {\lambda_i}^* \cdot \frac{\partial g_i(\boldsymbol{x}^*)}{\partial x_i} = 0$$

Da cui si ricava una relazione fra il gradiente di f valutato in $\boldsymbol{x}^*$ ed il gradiente dei vincoli:

$$\nabla L(\boldsymbol{x}^*, \lambda^*) = \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} {\lambda_i}^* \cdot \nabla g_i(\boldsymbol{x}^*) = 0 \Rightarrow \nabla f(\boldsymbol{x}^*) = -\sum_{i=1}^{m} {\lambda_i}^* \cdot \nabla g_i(\boldsymbol{x}^*)$$

Questo significa che il gradiente della funzione f valutato nel punto di massimo $\boldsymbol{x}^*$ puó essere riscritto come combinazione lineare dei gradienti dei vincoli valutati in $\boldsymbol{x}^*$.
Oltre ad una condizione necessaria per garantire che un punto di massimo di una Lagrangiana sia punto di massimo anche della funzione a cui si riferisce, occorre disporre anche di una condizione sufficiente. A tal proposito, sia J una matrice, detta **matrice Jacobiana**, costruita disponendo in ciascuna cella i, j la derivata parziale dell'i-esimo vincolo rispetto alla j-esima componente:

$$J = \begin{matrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \cdots & \frac{\partial g_m}{\partial x_n} \end{matrix}$$

Si indichi con $J(\boldsymbol{x})$ la matrice Jacobiana valutata nel punto x. Si consideri l'insieme dei vettori $\boldsymbol{y} \in \mathbb{R}^n$ tali per cui il prodotto scalare fra questi e la Jacobiana valutata in $\boldsymbol{x}^*$ sia nullo:

$$J(\boldsymbol{x}^*) \cdot \boldsymbol{y} = 0 \Rightarrow \begin{cases} \nabla g_1(\boldsymbol{x}^*) y_1 = 0 \\ \cdots \\ \nabla g_m(\boldsymbol{x}^*) y_m = 0 \end{cases}$$

Tale sottospazio rappresenta l'insieme dei vettori perpendicolari ai gradienti dei vari vincoli. Sia allora $H_L{}'(\boldsymbol{x}^*, \lambda^*)$ la matrice Hessiana della Lagrangiana ristretta alle sole variabili iniziali $x_1, ..., x_n$:

$$H_L{}'(\boldsymbol{x}^*, \lambda^*) = \begin{matrix} \frac{\partial^2 L}{\partial x_1^2} & \cdots & \frac{\partial^2 L}{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 L}{\partial x_1 x_n} & \cdots & \frac{\partial^2 L}{\partial x_n^2} \end{matrix}$$

La condizione sufficiente affinché $\boldsymbol{x}^*$ sia un punto di minimo é che il prodotto matriciale fra il vettore $\boldsymbol{y}$ trasposto, la matrice $H_L{}'(\boldsymbol{x}^*, \lambda^*)$ ed il vettore $\boldsymbol{y}$ originale sia strettamente positivo, mentre la condizione sufficiente affinché $\boldsymbol{x}^*$ sia un punto di massimo é che tale prodotto sia strettamente negativo.

Si consideri un problema di programmazione non lineare avente per funzione obiettivo $\min x_1 + x_2$ e come unico vincolo il vincolo di uguaglianza $x_1^2 + x_2^2 = 1$. Se ne calcoli la Lagrangiana ed i relativi punti stazionari:

$$L(x_1, x_2, \lambda_1) = x_1 + x_2 + \lambda_1(x_1^2 + x_2^2 - 1)$$

$$\nabla L = \begin{cases} \dfrac{\partial L}{\partial x_1} = 1 + 2\lambda_1 x_1 = 0 \\ \dfrac{\partial L}{\partial x_2} = 1 + 2\lambda_1 x_2 = 0 \\ \dfrac{\partial L}{\partial \lambda_1} = \left(x_1^2 + x_2^2 - 1\right) = 0 \end{cases} \Rightarrow \begin{cases} x_1 = -\dfrac{1}{2\lambda_1} \\ x_2 = -\dfrac{1}{2\lambda_1} \\ \left(-\dfrac{1}{2\lambda_1}\right)^2 + \left(-\dfrac{1}{2\lambda_1}\right)^2 - 1 = 0 \end{cases} \Rightarrow \begin{cases} x_1 = \pm\dfrac{1}{\sqrt{2}} \\ x_2 = \pm\dfrac{1}{\sqrt{2}} \\ \lambda_1 = \mp\dfrac{\sqrt{2}}{2} \end{cases}$$

Da cui si ottengono i punti stazionari $A = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, -\frac{\sqrt{2}}{2}\right)$ e $B = \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, \frac{\sqrt{2}}{2}\right)$. Il vettore $\boldsymbol{y}$ é dato da:

$$J(\boldsymbol{x}^*) \cdot \boldsymbol{y} = 0 \Rightarrow \left(\frac{\partial g_1}{\partial x_1}, \frac{\partial g_2}{\partial x_2}\right) \cdot \boldsymbol{y} = 0 \Rightarrow [2x_1^*, 2x_2^*] \cdot \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 0 \Rightarrow \boldsymbol{y} = \begin{bmatrix} \pm 1 \\ \mp 1 \end{bmatrix}$$

4.4.3 Condizioni di Karush-Kuhn-Tucker

Si supponga che il problema di programmazione non lineare in esame abbia una certa funzione obiettivo min / max $f(x_1, ..., x_n)$ e sia soggetto ad $m + p$ vincoli funzionali, dove m di questi sono vincoli di uguaglianza $g_i(x_1, ..., x_n) = 0$ con $i = 1, ..., m$ e p vincoli di disuguaglianza $h_j(x_1, ..., x_n) \leq 0$ con $j = 1, ..., p$.
É comunque possibile costruire una Lagrangiana associata ad un problema di programmazione non lineare di questo tipo introducendo, oltre alle m variabili λ_i associate ai vincoli di uguaglianza, p variabili $\mu_1, ..., \mu_p$ associate ai vincoli di disuguaglianza:

$$L(x_1, ..., x_n, \lambda_1, ..., \lambda_m, \mu_1, ..., \mu_p) = f(x_1, ..., x_n) + \sum_{i=1}^{m} \lambda_i \cdot g_i(x_1, ..., x_n) + \sum_{j=1}^{p} \mu_j \cdot g_j(x_1, ..., x_n)$$

Sia $\boldsymbol{x}^*$ un generico punto di minimo o di massimo per f. Affinché tale punto possa essere una soluzione ottimale per il problema di programmazione non lineare vincolata, i vincoli in questo valutati devono essere rispettati. In tal senso, deve valere $g_i(\boldsymbol{x}^*) = 0$ per qualsiasi $i = 1, ..., m$. Inoltre, per alcuni $j = 1, ..., p$ varrá $h_j(\boldsymbol{x}^*) = 0$, mentre per gli altri j varrá $h_j(\boldsymbol{x}^*) < 0$.
Se rispetto al j-esimo vincolo é valida la prima situazione, ovvero $h_j(\boldsymbol{x}^*) = 0$, allora il vincolo é un vincolo attivo, mentre se si verifica $h_j(\boldsymbol{x}^*) < 0$ allora tale vincolo é un vincolo inattivo. Se un vincolo é inattivo, questo significa che un piccolo cambiamento nel valore di $\boldsymbol{x}^*$ non lo invalida. Sia $I(\boldsymbol{x}^*)$ l'insieme dei vincoli h_j che sono attivi per $\boldsymbol{x}^*$.

Condizioni di Karush-Kuhn-Tucker. Sia dato un problema di programmazione lineare avente una funzione obiettivo $f(\boldsymbol{x})$, m vincoli di uguaglianza $g_i(x_1, ..., x_n) = 0$ e p vincoli di disuguaglianza $h_j(x_1, ..., x_n) \leq 0$.
Sia $\boldsymbol{x}^* = (x_1^*, ..., x_n^*)$ un punto di massimo o di minimo per f. Allora esistono m moltiplicatori $\lambda_1^*, ..., \lambda_m^*$ e p moltiplicatori $\mu_1^*, ..., \mu_p^*$ tali per cui valga la **condizione di stazionarietá** riportata di seguito; quella a sinistra é relativa ai problemi di minimo mentre quella destra ai problemi di massimo:

$$\nabla f(\boldsymbol{x}^*, \lambda^*, \mu^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \mu_j^* \nabla h_j(\boldsymbol{x}^*) = 0 \qquad \nabla f(\boldsymbol{x}^*, \lambda^*, \mu^*) - \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) - \sum_{j=1}^{p} \mu_j^* \nabla h_j(\boldsymbol{x}^*) = 0$$

Oltre a queste devono valere altre tre condizioni, rispettivamente indicate come **ammissibilitá primale**, **ammissibilitá duale** e **condizioni di complementarietá**:

$$\begin{aligned} &g_i(\boldsymbol{x}^*) = 0 \quad \forall i = 1, ..., m \\ &h_j(\boldsymbol{x}^*) \leq 0 \quad \forall j = 1, ..., p \end{aligned} \qquad \mu_j^* \geq 0 \ \forall j = 1, ..., p \qquad \mu_j^* \cdot h_j(\boldsymbol{x}^*) = 0 \ \forall j = 1, ..., p$$