

ALGORITMI DI
OTTIMIZZAZIONE
MONO E MULTI
OBIETTIVO

In questi ultimi anni si è avuto un crescendo nell'utilizzo di algoritmi di ottimizzazione, in quanto durante il processo di progettazione, sta avendo sempre più importanza la fase di ottimizzazione.

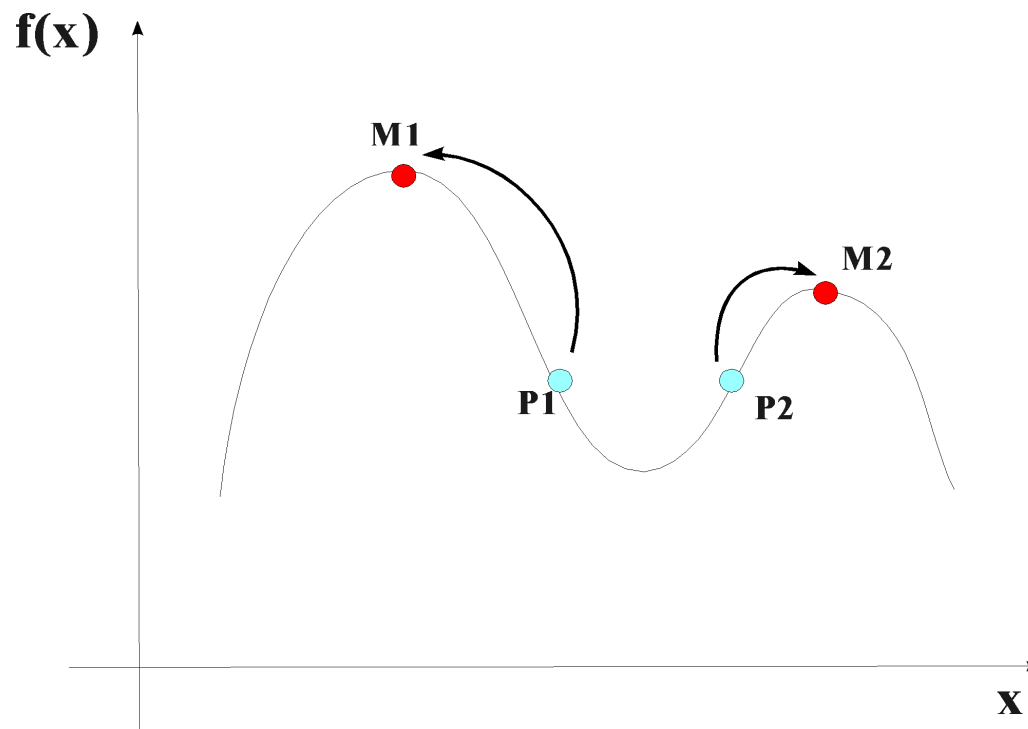
Per questo motivo è stata creata una grande varietà di algoritmi, ognuno con le proprie caratteristiche, e quindi adatti a tipologie di ottimizzazione diverse tra loro.

In questa sezione verranno mostrati alcuni algoritmi, facendo riferimento soprattutto alle differenze tra loro e tra i loro campi di applicazione più caratteristici.

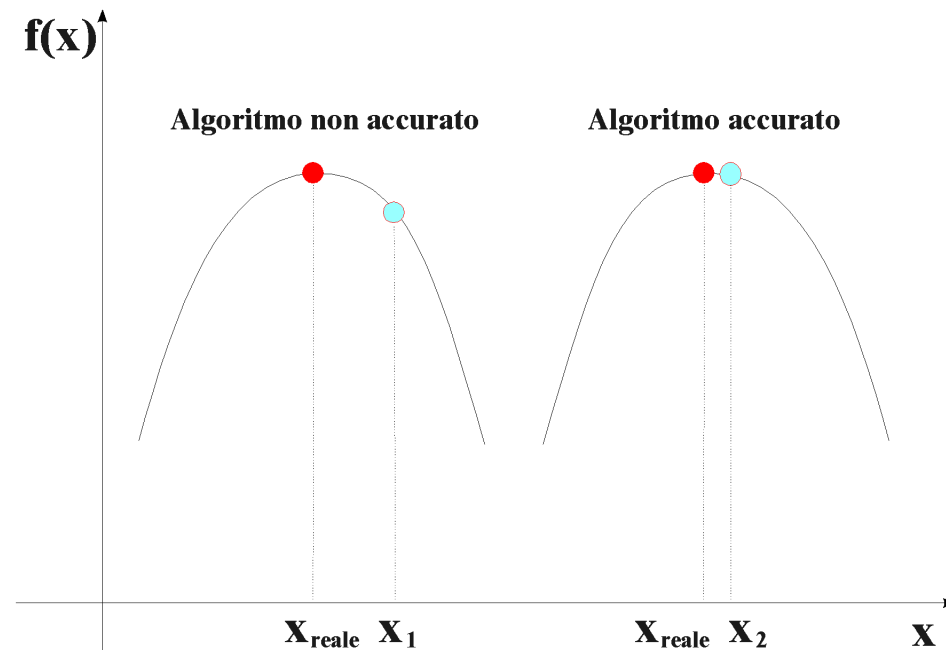
Per riuscire a comprendere meglio le differenze tra i vari algoritmi è utile dare due definizioni che vengono usualmente usate per classificare gli algoritmi:

- robustezza;**
- accuratezza.**

Per **robustezza** di un algoritmo di ottimizzazione si intende la capacità di trovare il massimo assoluto della funzione obiettivo.



Per **accuratezza** di un algoritmo di ottimizzazione si intende la capacità di avvicinarsi al vero valore del massimo assoluto della funzione obiettivo



In base alle definizioni date precedentemente (robustezza, accuratezza, velocità di convergenza) e all'utilizzo o meno del gradiente della funzione obiettivo $\vec{\nabla}f(x_0) = \left(\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}\right)$ è possibile definire la seguente tabella:

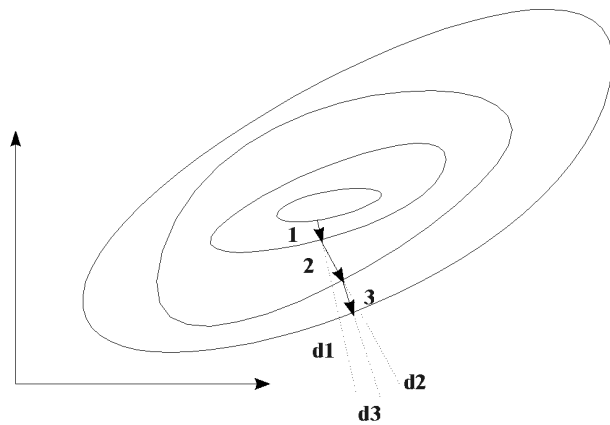
	Algoritmi che usano il gradiente	Algoritmi che NON usano il gradiente
Robustezza	Bassa	Alta
Accuratezza	Alta	Bassa
Velocità di convergenza	Alta	Bassa

$$\vec{\nabla} f(x_0) = \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{Bmatrix}$$

Valore **locale** (accuratezza \uparrow , robustezza \downarrow)

Dà la direzione di **massimo incremento** della funzione \Rightarrow **velocità in convergenza**

Necessita una notevole quantità di calcoli aggiuntivi (derivate parziali)



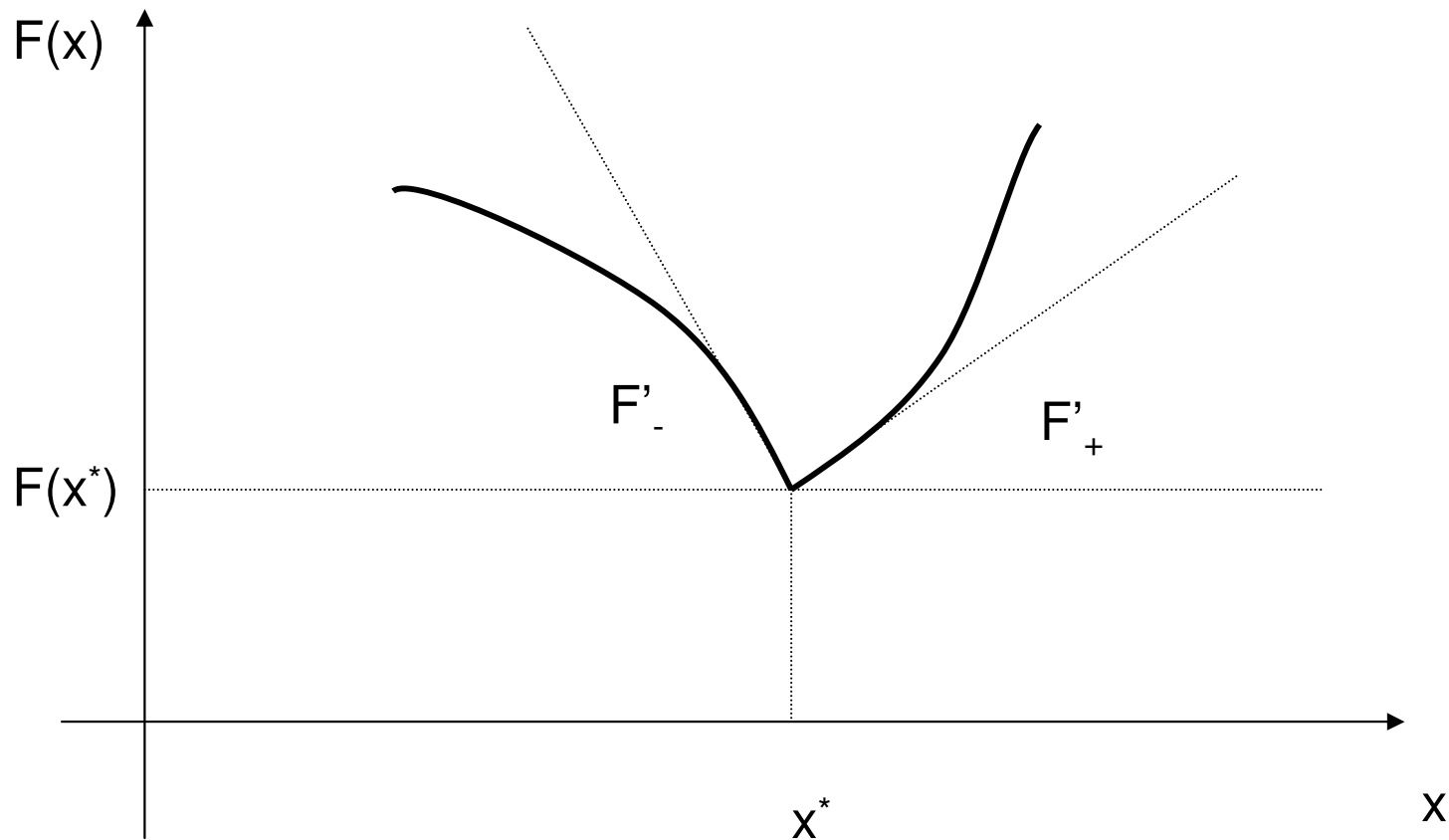
Differenze finite in avanti:

$$\left. \frac{\partial f}{\partial x_i} \right|_{X_m} \cong \frac{f(X_m + \Delta x_i u_i) - f(X_m)}{\Delta x_i}$$

Differenze finite centrate:

$$\left. \frac{\partial f}{\partial x_i} \right|_{X_m} \cong \frac{f(X_m + \Delta x_i u_i) - f(X_m - \Delta x_i u_i)}{2\Delta x_i}$$

Attenzione: non sempre è possibile definire il gradiente
Variabili discrete o di categoria (parametrizzazione)



Derivata sinistra diversa da derivata destra

- **Esiste una grande varietà di algoritmi di ottimizzazione che sfruttano il gradiente della funzione obiettivo (Cauchy, Powell, Newton, Quasi-Newton, BFGS, SQP, ecc.).**
- **Le caratteristiche più interessante che accomuna questi metodi è l'accuratezza della soluzione trovata.**
- **Noi studieremo:**
 - **Cauchy**
 - **Newton**
 - **Quasi-Newton (Broyden-Fletcher-Goldfarb-Shanno)**
 - **Sequential Quadratic Programming**

Condizione necessaria: $f : \mathcal{R}^n \Rightarrow \mathcal{R}$

$$\bar{\nabla} f(\bar{x}^*) = \bar{0}$$

$$\frac{\partial f}{\partial x_1}(\bar{x}^*) = \frac{\partial f}{\partial x_2}(\bar{x}^*) = \dots = \frac{\partial f}{\partial x_n}(\bar{x}^*) = 0$$

Condizione sufficiente:

$$\text{Matrice hessiana } [\mathbf{H}] = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

$$\text{Simmetrica } \frac{\partial^2 f}{\partial x_i \partial x_j} = \frac{\partial^2 f}{\partial x_j \partial x_i} \quad [\mathbf{H}](\bar{x}^*) = \begin{cases} \text{definita positiva} \Rightarrow \text{Minimo relativo} \\ \text{definita negativa} \Rightarrow \text{Massimo relativo} \\ \text{indefinita} \Rightarrow \text{sella} \end{cases}$$

Una matrice \mathbf{A} è definita positiva se

$$|\mathbf{A} - \lambda \mathbf{I}| = 0$$

tutti gli autovalori sono positivi

Vincoli di uguaglianza

$$\text{Min } f : \mathfrak{R}^n \Rightarrow \mathfrak{R}$$

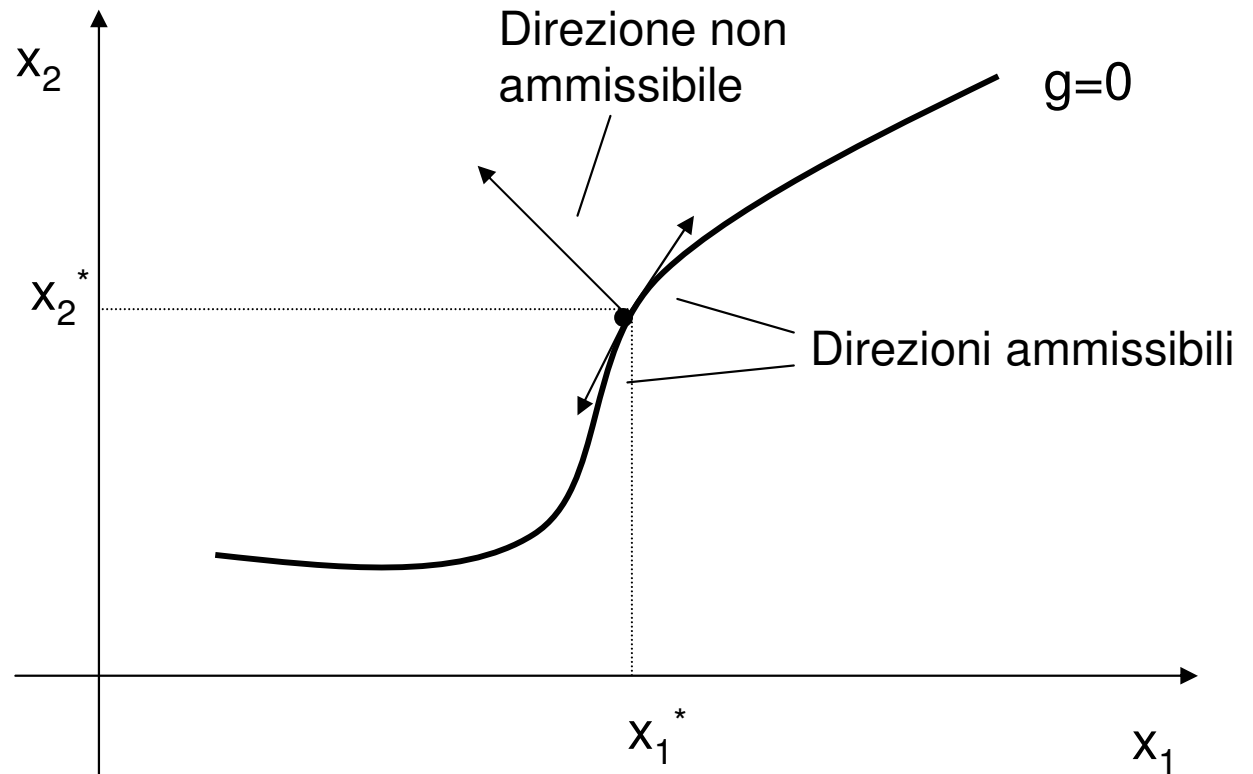
$$g_j(\bar{x}) = 0 \quad j = 1, \dots, m$$

utilizzo dei **MOLTIPLICATORI DI LAGRANGE**

Esempio n=2 m=1

MIN $f(x_1, x_2)$

$g(x_1, x_2) = 0$



$$df = \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 = 0$$

esiste direzione ammissibile tale che :

$$g(x_1^* + dx_1, x_2^* + dx_2) = 0$$

$$g(x_1^*, x_2^*) + \frac{\partial g}{\partial x_1}(x_1^*, x_2^*) dx_1 + \frac{\partial g}{\partial x_2}(x_1^*, x_2^*) dx_2 = 0$$

$$\Rightarrow dg = \frac{\partial g}{\partial x_1}(x_1^*, x_2^*) dx_1 + \frac{\partial g}{\partial x_2}(x_1^*, x_2^*) dx_2 = 0 \quad \text{in } (x_1^*, x_2^*)$$

$$\text{se } \frac{\partial g}{\partial x_2}(x_1^*, x_2^*) \neq 0$$

$$dx_2 = -\frac{\frac{\partial g}{\partial x_1}(x_1^*, x_2^*)}{\frac{\partial g}{\partial x_2}(x_1^*, x_2^*)} dx_1 \quad \text{cioè scelto } dx_1 \text{ } dx_2 \text{ è univoco}$$

$$df = \left(\frac{\partial f}{\partial x_1} - \frac{\partial g / \partial x_1}{\partial g / \partial x_2} \frac{\partial f}{\partial x_2} \right) \Bigg|_{x^*} dx_1 = 0$$

$$\left(\frac{\partial f}{\partial x_1} \frac{\partial g}{\partial x_2} - \frac{\partial f}{\partial x_2} \frac{\partial g}{\partial x_1} \right) \Bigg|_{x^*} = 0 \quad \text{condizione necessaria}$$

$$\text{MIN } f(x_1, x_2) \quad g(x_1, x_2) = 0$$

$$\left(\frac{\partial f}{\partial x_1} - \frac{\partial f / \partial x_2}{\partial g / \partial x_2} \frac{\partial g}{\partial x_1} \right) \Big|_{x^*} = 0$$

$$\lambda = - \left(\frac{\partial f / \partial x_2}{\partial g / \partial x_2} \right) \Big|_{x^*} \quad \text{moltiplicatore di Lagrange}$$

$$\begin{cases} \frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1} = 0 \\ \frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2} = 0 \\ g(x_1, x_2) = 0 \end{cases}$$

FUNZIONE DI LAGRANGE

$$L(x_1, x_2, \lambda) = f(x_1, x_2) + \lambda$$

$$\begin{cases} \frac{\partial L}{\partial x_1} = \frac{\partial f}{\partial x_1} + \lambda \frac{\partial g}{\partial x_1} = 0 \\ \frac{\partial L}{\partial x_2} = \frac{\partial f}{\partial x_2} + \lambda \frac{\partial g}{\partial x_2} = 0 \\ \frac{\partial L}{\partial \lambda} = g = 0 \end{cases}$$

$$\text{MIN } f : \mathfrak{R}^n \Rightarrow \mathfrak{R}$$

$$g_j(\bar{x}) = 0 \quad j = 1, \dots, m$$

$$L(x_1, \dots, x_n, \lambda_1, \dots, \lambda_m) = f(\bar{x}) + \sum_{j=1}^m \lambda_j g_j(\bar{x})$$

$$\text{condizione necessaria} \left\{ \begin{array}{l} \frac{\partial L}{\partial x_1} = \frac{\partial f}{\partial x_1} + \sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial x_1} = 0 \\ \vdots \\ \frac{\partial L}{\partial \lambda_1} = g_1 = 0 \\ \vdots \end{array} \right.$$

condizione sufficiente

$$Q = \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 L}{\partial x_i \partial x_j} dx_i dx_j \Rightarrow \begin{cases} \text{definita positiva} \Rightarrow \text{minimo} \\ \text{definita negativa} \Rightarrow \text{massimo} \end{cases}$$

Interpretazione moltiplicatori di Lagrange

$$\text{MIN } f(\bar{x}) : \mathfrak{R}^n \Rightarrow \mathfrak{R} \quad \tilde{g}(\bar{x}) = b$$

$$g(\bar{x}) = b - \tilde{g}(\bar{x}) = 0$$

$$\begin{cases} \frac{\partial f}{\partial x_i} + \lambda \frac{\partial g}{\partial x_i} = 0 & i = 1, \dots, n \\ g = 0 \end{cases}$$

$$db = d\tilde{g}$$

$$db = d\tilde{g} = \sum_{i=1}^n \frac{\partial \tilde{g}}{\partial x_i} dx_i$$

$$\frac{\partial f}{\partial x_i} + \lambda \frac{\partial g}{\partial x_i} = \frac{\partial f}{\partial x_i} - \lambda \frac{\partial \tilde{g}}{\partial x_i} = 0$$

$$\frac{\partial \tilde{g}}{\partial x_i} = \frac{\partial f / \partial x_i}{\lambda}$$

$$db = \sum_{i=1}^n \frac{1}{\lambda} \frac{\partial f}{\partial x_i} dx_i = \frac{df}{\lambda}$$

e quindi nel punto di ottimo

$$df_* = \lambda_* db$$

Ciò significa che il moltiplicatore di Lagrange è un indicatore della sensibilità del punto di ottimo dei confronti del vincolo di uguaglianza

Vincoli di disuguaglianza

Anch'essi possono essere risolti con l'utilizzo dei moltiplicatori di Lagrange.

$$\text{MIN } f(\bar{x}) : \mathfrak{R}^n \Rightarrow \mathfrak{R}$$

$$g_j(\bar{x}) \leq 0 \quad j = 1, \dots, m$$

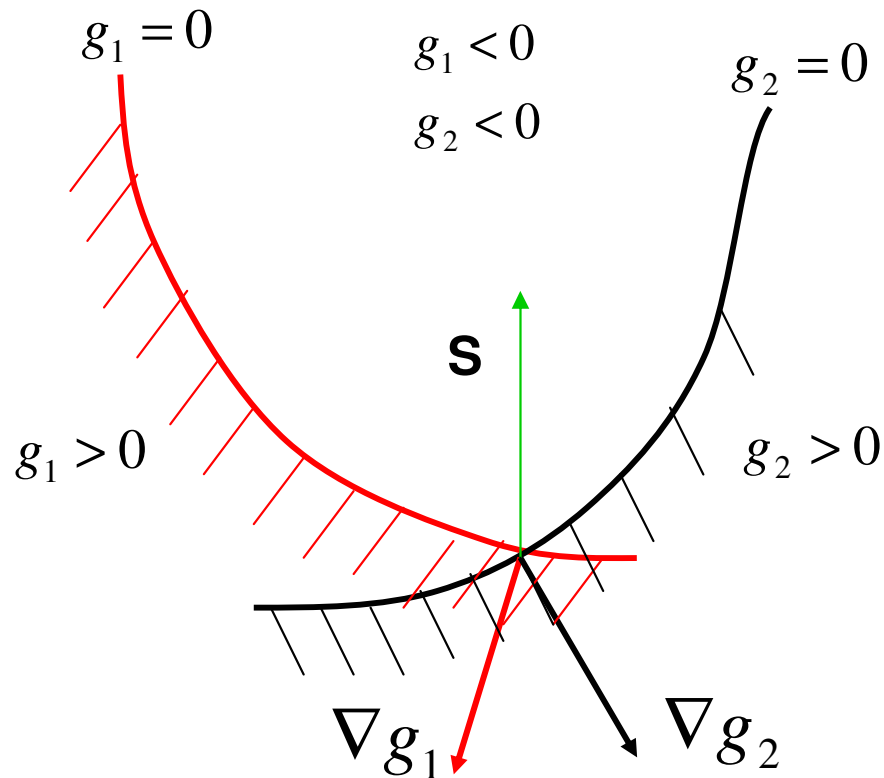
il vincolo si può trasformare in

$$G_j(\bar{x}, \bar{y}) = g_j(\bar{x}) + y_j^2 = 0$$

$$L(\bar{x}, \bar{y}, \bar{\lambda}) = f(\bar{x}) + \sum_{j=1}^m \lambda_j G_j(\bar{x}, \bar{y})$$

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial x_i}(\bar{x}, \bar{y}, \bar{\lambda}) = \frac{\partial f}{\partial x_i}(\bar{x}) + \sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial x_i}(\bar{x}) \quad i = 1, \dots, n \\ \frac{\partial L}{\partial \lambda_j}(\bar{x}, \bar{y}, \bar{\lambda}) = G_j(\bar{x}, \bar{y}) = g_j(\bar{x}) + y_j^2 = 0 \quad j = 1, \dots, m \\ \frac{\partial L}{\partial y_j}(\bar{x}, \bar{y}, \bar{\lambda}) = 2\lambda_j y_j = 0 \quad j = 1, \dots, m \end{array} \right.$$

Interpretazione dei moltiplicatori di Lagrange (caso vincoli disuguaglianza)



$$\nabla f + \bar{\lambda}^T \nabla g = 0$$

2 vincoli

$$-\nabla f + \lambda_1 \nabla g_1 + \lambda_2 \nabla g_2$$

$$-\bar{S}^T \nabla f = \lambda_1 \bar{S}^T \nabla g_1 + \lambda_2 \bar{S}^T \nabla g_2$$

$$\begin{cases} \bar{S}^T \nabla g_1 < 0 \\ \bar{S}^T \nabla g_2 < 0 \end{cases} \Rightarrow \text{direzione consentita}$$

(mi dà la diminuzione dei vincoli)

Ciò significa che se i moltiplicatori sono maggiori di 0 non c'è nessuna direzione consentita per far diminuire la funzione all'interno dei vincoli (minori di zero)

Condizioni di Kuhn-Tucker (condizione necessaria)

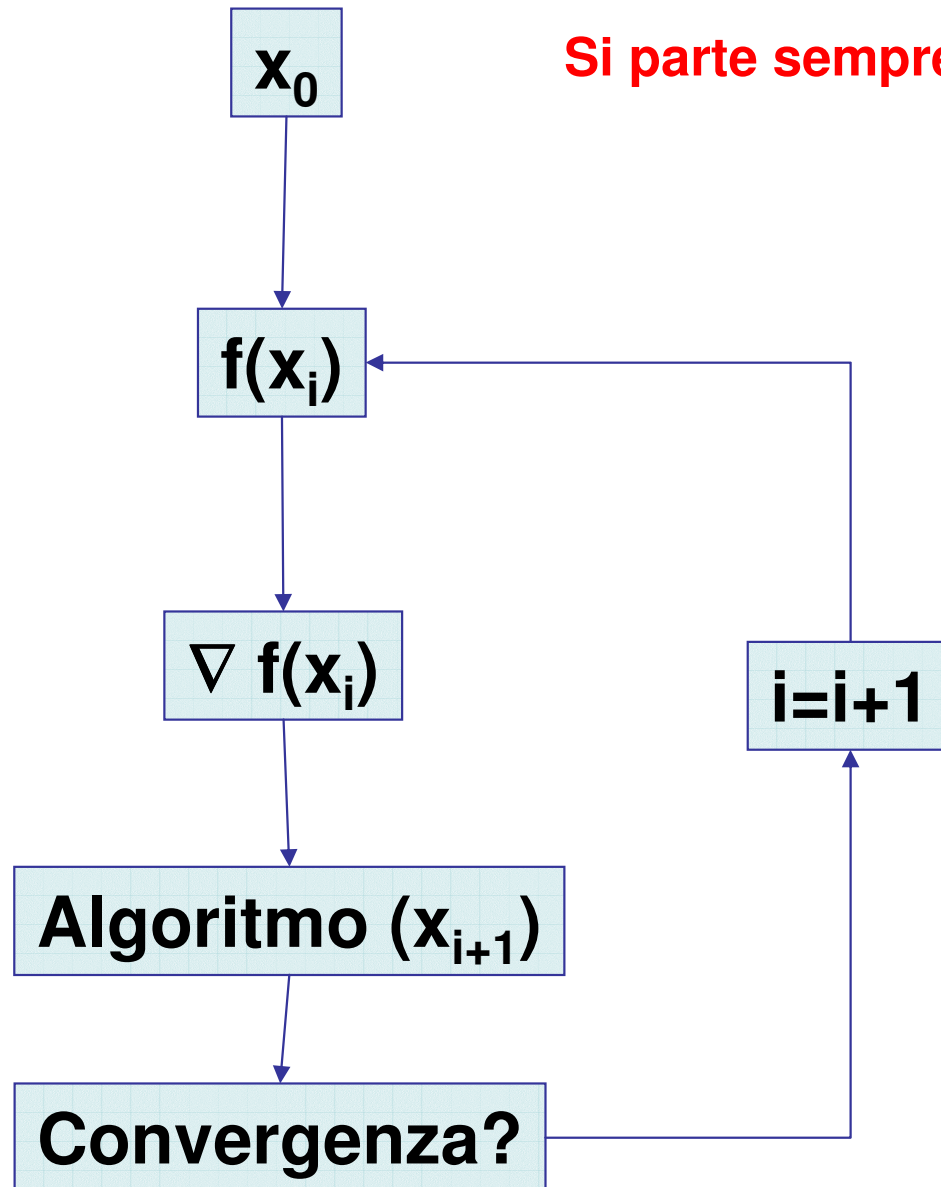
$$\text{MIN } f(\bar{x}) : \mathfrak{R}^n \Rightarrow \mathfrak{R}$$

$$g_j(\bar{x}) \leq 0 \quad j = 1, \dots, m$$

$$h_k(\bar{x}) = 0 \quad k = 1, \dots, p$$

$$\left\{ \begin{array}{l} \nabla f + \bar{\lambda}^T \nabla g - \bar{\beta}^T \nabla h = 0 \\ \lambda_j g_j = 0 \\ g_j \leq 0 \\ h_k = 0 \\ \lambda_j \geq 0 \end{array} \right.$$

Si parte sempre da una soluzione conosciuta



Convergenza

1) $i \geq ntot$

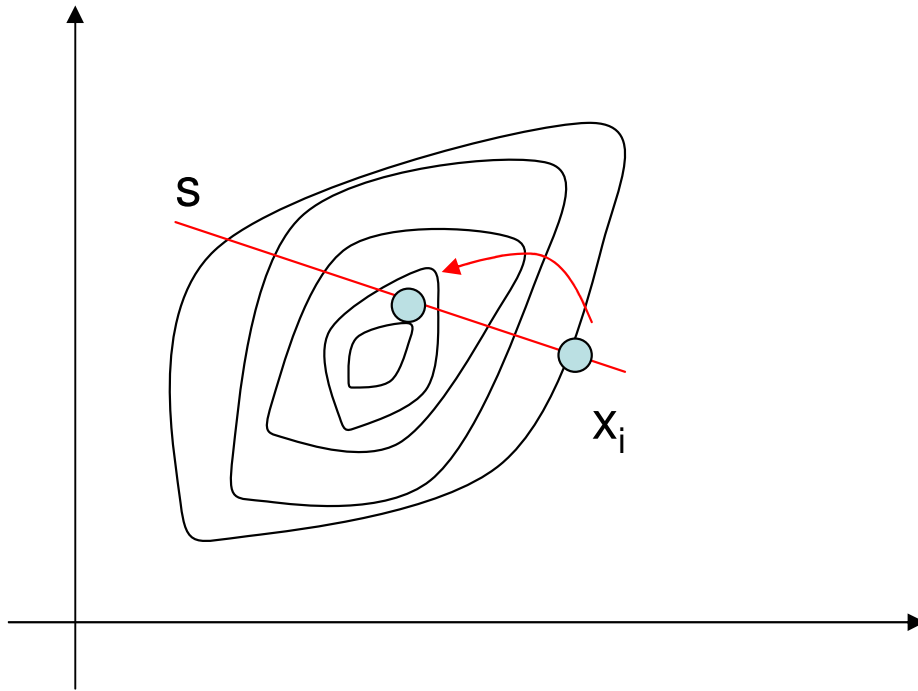
2) $\left| \frac{f(x_{i+1}) - f(x_i)}{f(x_i)} \right| \leq \varepsilon_1$

3) $|f(x_{i+1}) - f(x_i)| \leq \varepsilon_2$

4) $|x_{i+1} - x_i| \leq \varepsilon_3$

5) $\max \left| \frac{\partial f}{\partial x_i} \right| \leq \varepsilon_4$

Step ottimale lungo una direzione



$$f : \mathcal{R}^n \Rightarrow \mathcal{R}$$

$$\bar{x} = \bar{x}_i + \lambda \bar{s}$$

$$\frac{\partial f}{\partial \lambda} = 0 \Rightarrow \sum_{j=1}^n \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial \lambda} = 0$$

$$\frac{\partial x_j}{\partial \lambda} = \frac{\partial}{\partial \lambda} (x_{ij} + \lambda s_{ij}) = s_{ij}$$

$$\frac{\partial f}{\partial \lambda} = \sum_{j=1}^n \frac{\partial f}{\partial x_j} s_{ij} = \nabla f|_{\lambda^*}^T \bar{s}_i = 0$$

Algoritmo di Cauchy

Il metodo di Cauchy è uno degli algoritmi base che utilizzano il gradiente della funzione obiettivo.

Punto 1) scelta del punto di partenza \vec{x}_1

Punto 2) calcolo della direzione $\vec{S}_i = \nabla f(\vec{x}_i)$

Punto 3) calcolo del nuovo punto :

$$\vec{x}_{i+1} = \vec{x}_i + \lambda_i^* \vec{S}_i$$

con λ_i^* passo ottimale lungo la direzione \vec{S}_i

ESEMPIO

$$\text{MIN } f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

$$\text{partendo dal punto } \bar{x}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

$$\nabla f = \begin{Bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \end{Bmatrix} = \begin{Bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{Bmatrix}$$

Iterazione 1

$$\nabla f(\bar{x}_1) = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$

$$\bar{S}_1 = -\nabla f_1 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$f(\bar{x}_1 + \lambda_1 \bar{S}_1) = f(-\lambda_1, \lambda_1) = \lambda_1^2 - 2\lambda_1 \Rightarrow \lambda_1^* = 1$$

$$\bar{x}_2 = \bar{x}_1 + \lambda_1^* \bar{S}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + 1 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$\nabla f_2 \neq 0$$

Iterazione 2

$$\bar{S}_2 = -\nabla f_2 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

$$f(\bar{x}_2 + \lambda_2 \bar{S}_2) = f(-1 + \lambda_2, 1 + \lambda_2) = 5\lambda_2^2 - 2\lambda_2 - 1 \Rightarrow \lambda_2^* = \frac{1}{5}$$

$$\bar{x}_3 = \bar{x}_2 + \lambda_2^* \bar{S}_2 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} + \frac{1}{5} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -0.8 \\ 1.2 \end{Bmatrix}$$

$$\nabla f_3 \neq 0$$

Iterazione 3

$$\bar{S}_3 = -\nabla f_3 = \begin{Bmatrix} -0.2 \\ 0.2 \end{Bmatrix}$$

$$f(\bar{x}_3 + \lambda_3 \bar{S}_3) = f(-0.8 + 0.2\lambda_3, 1.2 + 0.2\lambda_3) = 0.04\lambda_3^2 - 0.08\lambda_3 - 1.2 \Rightarrow \lambda_3^*$$

$$\bar{x}_4 = \bar{x}_3 + \lambda_3^* \bar{S}_3 = \begin{Bmatrix} -0.8 \\ 1.2 \end{Bmatrix} + 1 \begin{Bmatrix} -0.2 \\ 0.2 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1.4 \end{Bmatrix}$$

$$\nabla f_4 = \begin{Bmatrix} -0.2 \\ -0.2 \end{Bmatrix} \neq 0$$

Algoritmo del gradiente coniugato (Fletcher-Reeves)

Ottimizza una forma quadratiche di ordine n in n steps (meglio che Cauchy)

$$f(\bar{x}) = \frac{1}{2} \bar{x}^T \mathbf{A} \bar{x} + \bar{B}^T \bar{x} + c \quad \mathbb{R}^n \Rightarrow \mathbb{R}$$

$$\bar{S}_1 = -\nabla f_1 = -\mathbf{A} \bar{x}_1 - \bar{B}$$

$$\bar{x}_2 = \bar{x}_1 + \lambda_1^* \bar{S}_1$$

$$\bar{S}_1 = \frac{\bar{x}_2 - \bar{x}_1}{\lambda_1^*}$$

quando vogliamo lo step ideale deve valere:

$$\bar{S}_1^T \nabla f|_{x_2} = 0 \quad \text{quindi}$$

$$\bar{S}_1^T [\mathbf{A}(\bar{x}_1 + \lambda_1^* \bar{S}_1) + \bar{B}] = 0$$

esplicitando λ_1^*

$$\lambda_1^* = \frac{-\bar{S}_1^T (\mathbf{A} \bar{x}_1 + \bar{B})}{\bar{S}_1^T \mathbf{A} \bar{S}_1} = -\frac{\bar{S}_1^T \nabla f_1}{\bar{S}_1^T \mathbf{A} \bar{S}_1}$$

$$\bar{S}_2 = -\nabla f_2 + \beta_2 \bar{S}_1$$

scelgo β_2 affinché \bar{S}_1, \bar{S}_2 siano coniugate

$$\bar{S}_1^T \mathbf{A} \bar{S}_2 = 0$$

$$\bar{S}_1^T \mathbf{A} (-\nabla f_2 + \beta_2 \bar{S}_1) = 0$$

$$-\left[\frac{(\bar{x}_2 - \bar{x}_1)^T}{\lambda_1^*} \right] \mathbf{A} (\nabla f_2 - \beta_2 \bar{S}_1) = 0 \quad (1)$$

La differenza tra i gradienti può essere calcolata come :

$$\nabla f_2 - \nabla f_1 = (\mathbf{A} \bar{x}_2 + \bar{B}) - (\mathbf{A} \bar{x}_1 + \bar{B}) = \mathbf{A} (\bar{x}_2 - \bar{x}_1)$$

considerando Eq.1 (A simmetrica)

$$(\nabla f_2 - \nabla f_1)^T (\nabla f_2 - \beta_2 \bar{S}_1) = 0$$

$$\nabla f_2^T \nabla f_2 - \nabla f_1^T \nabla f_2 - \beta_2 \nabla f_2^T \bar{S}_1 + \beta_2 \nabla f_1^T \bar{S}_1 = 0$$

$$\text{poichè } \nabla f_1^T \nabla f_2 = -\bar{S}_1^T \nabla f_2 = 0$$

$$\beta_2 = -\frac{\nabla f_2^T \nabla f_2}{\nabla f_1^T \bar{S}_1} = \frac{\nabla f_2^T \nabla f_2}{\nabla f_1^T \nabla f_1}$$

$$\bar{S}_3 = -\nabla f_3 + \beta_3 \bar{S}_2 + \sigma_3 \bar{S}_1$$

β_3, σ_3 vengono determinati imponendo (conjugate):

$$\bar{S}_1^T \mathbf{A} \bar{S}_3 = 0$$

$$\bar{S}_2^T \mathbf{A} \bar{S}_3 = 0$$

si vede come:

$$\sigma_3 = 0$$

$$\beta_3 = \frac{\nabla f_3^T \nabla f_3}{\nabla f_2^T \nabla f_2}$$

di conseguenza si può trovare la formula ricorsiva

$$\bar{S}_i = -\nabla f_i + \beta_i \bar{S}_{i-1}$$

$$\beta_i = \frac{\nabla f_i^T \nabla f_i}{\nabla f_{i-1}^T \nabla f_{i-1}}$$

1. Scelta del punto iniziale \mathbf{X}_1

2. Calcolo della prima direzione \mathbf{S}_1

$$\bar{S}_1 = -\nabla f(\bar{x}_1) = -\nabla f_1$$

3. Calcolo del punto \mathbf{X}_2 :

$$\bar{x}_2 = \bar{x}_1 + \lambda_1^* \bar{S}_1$$

4. Calcolo della direzione \mathbf{S}_i

$$\bar{S}_i = -\nabla f_i + \frac{|\nabla f_i|^2}{|\nabla f_{i-1}|^2} \bar{S}_{i-1}$$

5. Calcolo del nuovo punti \mathbf{X}_{i+1}

$$\bar{x}_{i+1} = \bar{x}_i + \lambda_i^* \bar{S}_i$$

6. Convergenza?

Considerazioni:

- 1) La metodologia gradiente coniugata ha una velocità di convergenza maggiore rispetto la metodologia Cauchy (forma quadratica di grado n viene ottimizzata in n steps);
- 2) Poiché la ricerca dello step ottimale avviene per via numerica bisogna fare un restart del calcolo della direzione ogni determinato numero di iterazioni ($S = -\nabla f$)

$$\text{MIN } f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2$$

$$\text{partendo dal punto } \bar{x}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

$$\nabla f = \begin{Bmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \end{Bmatrix} = \begin{Bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{Bmatrix}$$

Iterazione 1

$$\nabla f(\bar{x}_1) = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$

$$\bar{S}_1 = -\nabla f_1 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$f(\bar{x}_1 + \lambda_1 \bar{S}_1) = f(-\lambda_1, \lambda_1) = \lambda_1^2 - 2\lambda_1 \Rightarrow \lambda_1^* = 1$$

$$\bar{x}_2 = \bar{x}_1 + \lambda_1^* \bar{S}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + 1 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$\nabla f_2 \neq 0$$

Iterazione 2

$$\nabla f_2 = \begin{Bmatrix} 1 \\ 1 \end{Bmatrix}$$

$$\bar{S}_2 = -\nabla f_2 + \frac{|\nabla f_2|^2}{|\nabla f_1|^2} \bar{S}_1$$

$$|\nabla f_2|^2 = |\nabla f_1|^2 = 2$$

$$\bar{S}_2 = -\begin{Bmatrix} -1 \\ -1 \end{Bmatrix} + \left(\frac{2}{2}\right) \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 2 \end{Bmatrix}$$

per determinare λ_2^*

$$\begin{aligned} \min f(\bar{x}_2 + \lambda_2 \bar{S}_2) &= f(-1, 1 + 2\lambda_2) = \\ &= 4\lambda_2^2 - 2\lambda_2 - 1 \end{aligned}$$

ESEMPIO

$$\lambda_2^* = \frac{1}{4}$$

$$\bar{x}_3 = \bar{x}_2 + \lambda_2^* \bar{S}_2 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} + \frac{1}{4} \begin{Bmatrix} 0 \\ 2 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1.5 \end{Bmatrix}$$

Iterazione 3

$$\nabla f_3 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \Rightarrow \text{punto di ottimo}$$

Trattazione Vincoli

$$\max f(x) \quad g(x) < 0$$

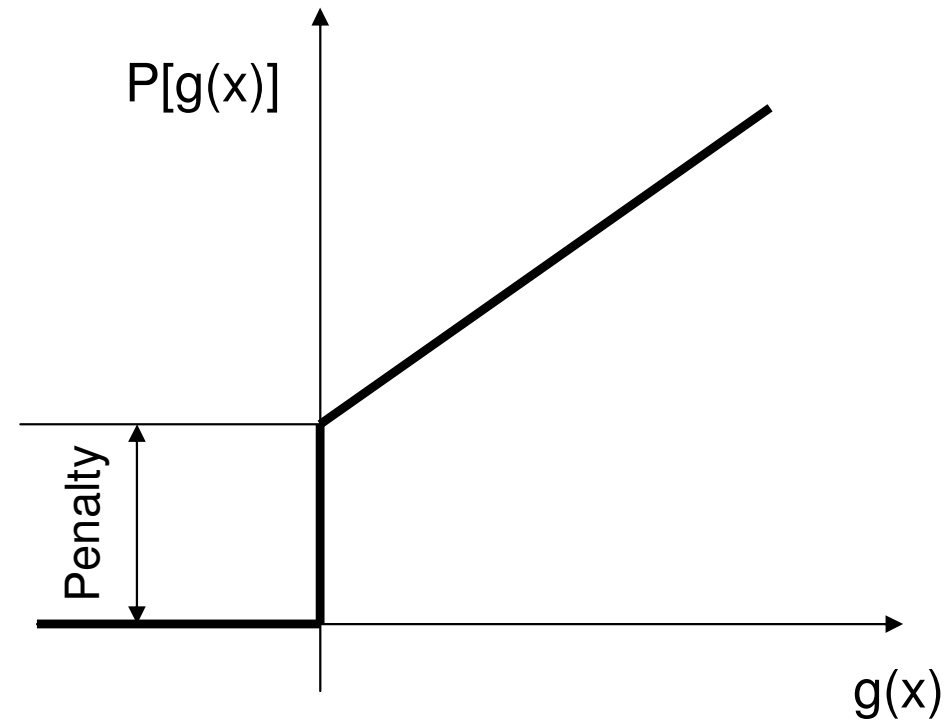
Utilizzo di Penalty function

$$\max f(\bar{x}) : \mathfrak{R}^n \Rightarrow \mathfrak{R}$$

$$g_j(\bar{x}) < 0 \quad j = 1, \dots, m$$

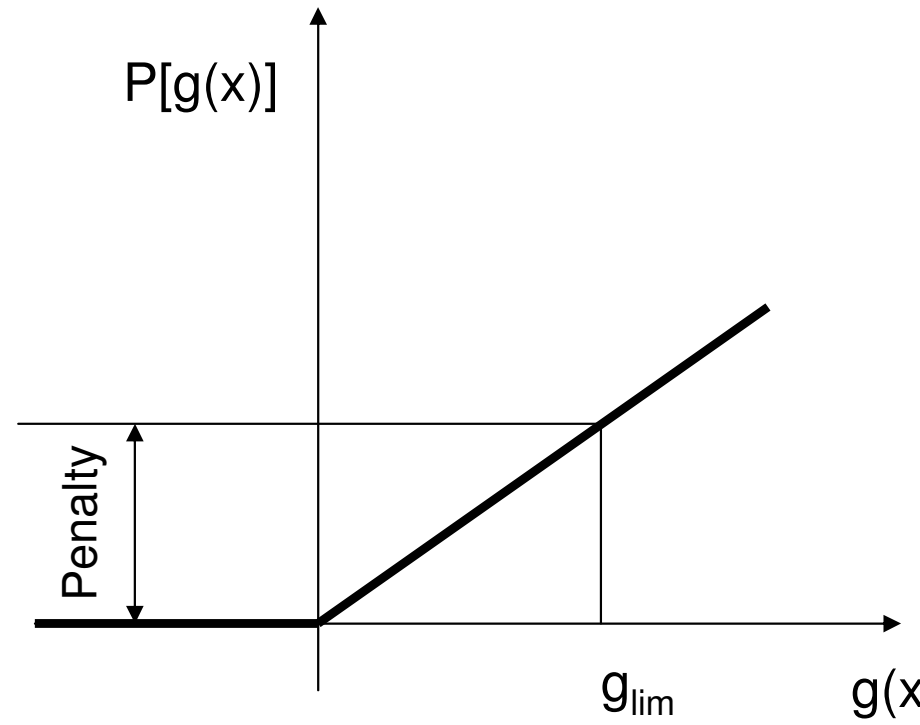
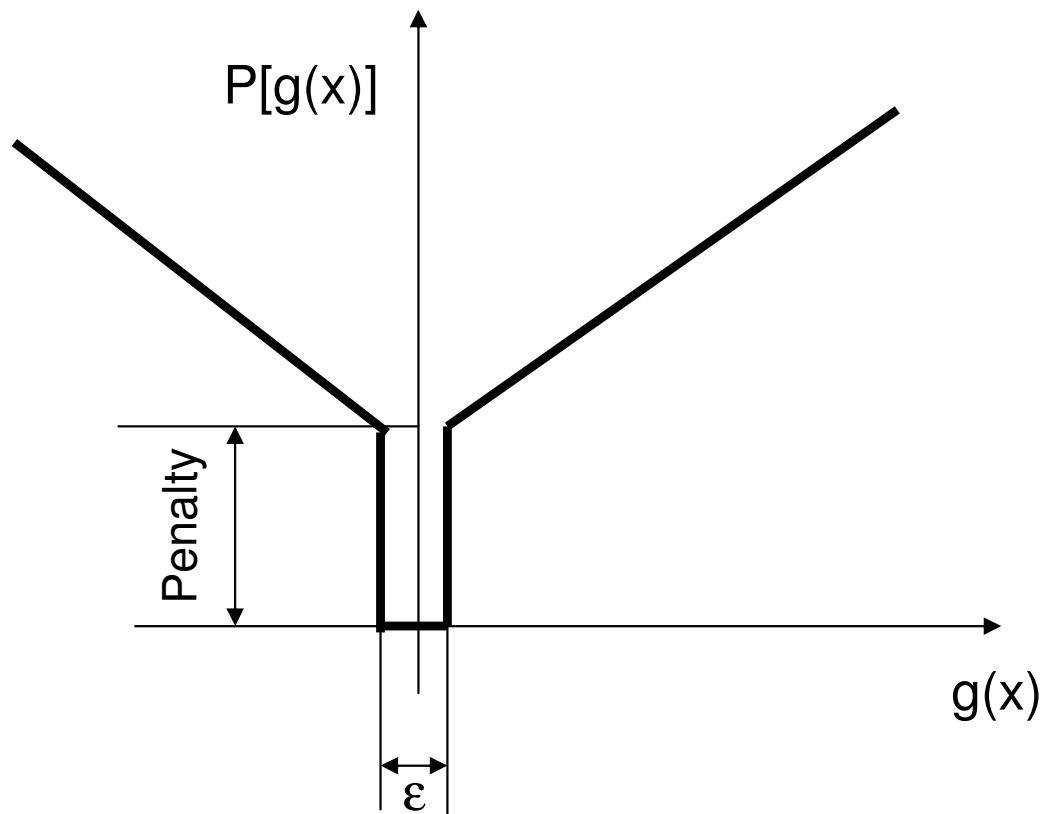
$$\max Fit(\bar{x}) = f(\bar{x}) + \sum_{j=1}^m P[g_j(\bar{x})]$$

$$P[g_j(\bar{x})] = \begin{cases} Penalty + g_j(\bar{x}) & \text{se } g_j(\bar{x}) > 0 \\ 0 & \text{se } g_j(\bar{x}) < 0 \end{cases}$$



Importante: l'ordine di grandezza di Penalty deve essere maggiore della $f(x)$ in modo da assicurare che i vincoli vengano soddisfatti

Caso con presenza di tolleranza sul vincolo



Trattamento di vincoli di uguaglianza (con tolleranza ϵ)

Metodo di Newton

Necessità di un'unica iterazione per ottimizzare una forma quadratica di ordine n

$$f(\bar{x}) = f(\bar{x}_i) + \nabla f_i^T (\bar{x} - \bar{x}_i) + \frac{1}{2} (\bar{x} - \bar{x}_i)^T [\mathbf{J}] (\bar{x} - \bar{x}_i)$$

$$\nabla f = 0 \Rightarrow \frac{\partial f}{\partial x_j} = 0 \quad j = 1, \dots, n$$

$$\nabla f = \nabla f_i + [\mathbf{J}] (\bar{x} - \bar{x}_i) = \bar{0}$$

$$\Rightarrow \bar{x}_{i+1} = \bar{x}_i - [\mathbf{J}]^{-1} \nabla f$$

oppure

$$\bar{x}_{i+1} = \bar{x}_i + \lambda_i^* \bar{S}_i = \bar{x}_{i+1} = \bar{x}_i - \lambda_i^* [\mathbf{J}_i]^{-1} \nabla f_i$$

- Il calcolo della matrice hessiana può essere troppo complesso in termini di tempo macchina (calcolo delle derivate seconde miste).
- Idea base : approssimare iterativamente la matrice l'inversa della matrice hessiana:

$$\lim_{i \rightarrow \infty} [\mathbf{H}_i] = [\mathbf{J}]^{-1}$$

- **Quasi-Newton**
-

$$\nabla f(\bar{x}) \cong \nabla f(\bar{x}_0) + [\mathbf{J}_0](\bar{x} - \bar{x}_0)$$

$$\nabla f_{i+1} = \nabla f(\bar{x}_0) + [\mathbf{A}_i](\bar{x}_{i+1} - \bar{x}_0)$$

$$\nabla f_i = \nabla f(\bar{x}_0) + [\mathbf{A}_i](\bar{x}_i - \bar{x}_0)$$

$$[\mathbf{A}_i]\bar{d}_i = \bar{g}_i$$

$$\bar{d}_i = \bar{x}_{i+1} - \bar{x}_i$$

$$\bar{g}_i = \nabla f_{i+1} - \nabla f_i$$

$$\bar{d}_i = [\mathbf{B}_i]\bar{g}_i$$

$$[\mathbf{B}_i] = [\mathbf{A}_i]^{-1} = [\mathbf{J}_0]$$

$$\bar{x}_{i+1} = \bar{x}_i - \lambda_i^* [\mathbf{B}_i] \nabla f(\bar{x}_i)$$

Calcolo della matrice B (rango 1)

$$[\mathbf{B}_{i+1}] = [\mathbf{B}_i] + [\Delta \mathbf{B}_i]$$

$$\Delta \mathbf{B}_i = c \bar{\mathbf{z}} \bar{\mathbf{z}}^T \quad \text{rango } 1$$

$$[\mathbf{B}_{i+1}] = [\mathbf{B}_i] + c \bar{\mathbf{z}} \bar{\mathbf{z}}^T$$

forzando

$$\bar{d}_i = [\mathbf{B}_{i+1}] \bar{g}_i$$

$$\bar{d}_i = ([\mathbf{B}_i] + c \bar{\mathbf{z}} \bar{\mathbf{z}}^T) \bar{g}_i = [\mathbf{B}_i] \bar{g}_i + c \bar{\mathbf{z}} (\bar{\mathbf{z}}^T \bar{g}_i)$$

poichè il termine in parentesi è uno scalare

$$c \bar{\mathbf{z}} = \frac{\bar{d}_i - [\mathbf{B}_i] \bar{g}_i}{\bar{\mathbf{z}}^T \bar{g}_i}$$

\Rightarrow

$$\bar{\mathbf{z}} = \bar{d}_i - [\mathbf{B}_i] \bar{g}_i$$

$$c = \frac{1}{\bar{\mathbf{z}}^T \bar{g}_i}$$

$$[\mathbf{B}_{i+1}] = [\mathbf{B}_i] + \frac{(\bar{d}_i - [\mathbf{B}_i] \bar{g}_i)(\bar{d}_i - [\mathbf{B}_i] \bar{g}_i)^T}{(\bar{d}_i - [\mathbf{B}_i] \bar{g}_i)^T \bar{g}_i}$$

Se la matrice \mathbf{B}_i è definita positiva altrettanto **NON si può dire per la matrice \mathbf{B}_{i+1}**

Più utilizzata è la metodologia con rango 2

$$[\Delta \mathbf{B}_i] = c_1 \bar{z}_1 \bar{z}_1^T + c_2 \bar{z}_2 \bar{z}_2^T$$

$$\bar{d}_i = [\mathbf{B}_i] \bar{g}_i + c_1 \bar{z}_1 (z_1^T g_i) + c_2 \bar{z}_2 (z_2^T g_i)$$

poichè i termini in parentesi sono scalari

$$\bar{z}_1 = \bar{d}_i$$

$$\bar{z}_2 = [\mathbf{B}_i] \bar{g}_i$$

$$c_1 = \frac{1}{z_1^T g_i}$$

$$c_2 = -\frac{1}{z_2^T g_i}$$

$$[\mathbf{B}_{i+1}] = [\mathbf{B}_i] + [\Delta \mathbf{B}_i] \equiv [\mathbf{B}_i] + \frac{\bar{d}_i \bar{d}_i^T}{\bar{d}_i^T \bar{g}_i} - \frac{([\mathbf{B}_i] \bar{g}_i)([\mathbf{B}_i] \bar{g}_i)^T}{([\mathbf{B}_i] \bar{g}_i)^T \bar{g}_i}$$

Broyden-Fletcher-Goldfarb-Shanno (BFGS)

1) Inizializzare \bar{x}_1 (soluzione di partenza)

Inizializzare $[\mathbf{B}_1] \equiv [\mathbf{I}]$

Calcolare $\nabla f_1 = \nabla f(\bar{x}_1)$

$$2) \quad \bar{S}_i = -[\mathbf{B}_i] \nabla f_i$$

$$3) \quad \bar{x}_{i+1} = \bar{x}_i + \lambda_i^* \bar{S}_i$$

$$4) \quad \text{se } \|\nabla f_{i+1}\| \leq \varepsilon \quad \bar{x}^* = x_{i+1}$$

$$5) \quad [\mathbf{B}_{i+1}] = [\mathbf{B}_i] + \left(1 + \frac{\bar{g}_i^T [\mathbf{B}_i] \bar{g}_i}{\bar{d}_i^T \bar{g}_i} \right) \frac{\bar{d}_i \bar{d}_i^T}{\bar{d}_i^T \bar{g}_i} - \frac{\bar{d}_i \bar{g}_i^T [\mathbf{B}_i]}{\bar{d}_i^T \bar{g}_i} - \frac{[\mathbf{B}_i] \bar{g}_i \bar{d}_i^T}{\bar{d}_i^T \bar{g}_i}$$

dove

$$\bar{d}_i = \bar{x}_{i+1} - \bar{x}_i = \lambda_i^* \bar{S}_i$$

$$\bar{g}_i = \nabla f_{i+1} - \nabla f_i$$

- 1. E' il metodo Quasi-Newton più efficiente ed utilizzato;**
- 2. Approssima l'inverso della matrice hessiana;**
- 3. Se lo step ideale λ viene calcolato analiticamente la matrice rimane definita positiva; poiché questo in algoritmo non avviene è necessaria la ri-inizializzazione della matrice B con la matrice identica**

ESEMPIO

$$\min f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \quad \bar{x}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \quad [\mathbf{B}_1] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Iterazione 1 (i = 1)

$$\nabla f_1 = \nabla f(x_1) = \left. \begin{Bmatrix} 1 + 4x_1 + 2x_2 \\ -1 + 2x_1 + 2x_2 \end{Bmatrix} \right|_{(0,0)} = \begin{Bmatrix} 1 \\ -1 \end{Bmatrix}$$

$$\bar{S}_1 = -[\mathbf{B}_1] \nabla f_1 = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$f(\bar{x}_1 + \lambda_1 \bar{S}_1) = f\left(\begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + \lambda_1 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}\right) = f(-\lambda_1, \lambda_1) = \lambda_1^2 - 2\lambda_1$$

$$\lambda_1^* = 1$$

$$\bar{x}_2 = \bar{x}_1 + \lambda_1^* \bar{S}_1 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} + 1 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$\text{poichè } \nabla f_2 = \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} \quad \|\nabla f_2\| = 1.4142 > \varepsilon$$



$$\bar{g}_1 = \nabla f_2 - \nabla f_1 = \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} - \begin{Bmatrix} 1 \\ -1 \end{Bmatrix} = \begin{Bmatrix} -2 \\ 0 \end{Bmatrix}$$

$$\bar{d}_1 = \lambda_1^* \bar{S}_1 = 1 \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix}$$

$$\bar{d}_1 \bar{d}_1^T = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \begin{bmatrix} -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\bar{d}_1^T \bar{g}_1 = \begin{bmatrix} -1 & 1 \end{bmatrix} \begin{Bmatrix} -2 \\ 0 \end{Bmatrix} = 2$$

$$\bar{d}_1 \bar{g}_1^T = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} \begin{bmatrix} -2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix}$$

$$\bar{g}_1 \bar{d}_1^T = \begin{Bmatrix} -2 \\ 0 \end{Bmatrix} \begin{bmatrix} -1 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix}$$

$$\bar{g}_1^T [\mathbf{B}_1] \bar{g}_1 = \begin{bmatrix} -2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} -2 \\ 0 \end{Bmatrix} = 4$$

$$\bar{d}_1 \bar{g}_1^T [\mathbf{B}_1] = \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix}$$

$$[\mathbf{B}_1] \bar{g}_1 \bar{d}_1^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix}$$

$$\begin{aligned}
[\mathbf{B}_2] &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \left(1 + \frac{4}{2}\right) \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 2 & 0 \\ -2 & 0 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 2 & -2 \\ 0 & 0 \end{bmatrix} = \\
&= - \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{5}{2} \end{bmatrix}
\end{aligned}$$

$$\bar{S}_2 = -[\mathbf{B}_2] \nabla f_2 = - \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{5}{2} \end{bmatrix} \begin{Bmatrix} -1 \\ -1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 2 \end{Bmatrix}$$

$$f(\bar{x}_2 + \lambda_2 \bar{S}_2) = 4\lambda_2^2 - 2\lambda_2 - 1$$

$$\lambda_2^* = \frac{1}{4}$$

$$\bar{x}_3 = \bar{x}_2 + \lambda_2^* \bar{S}_2 = \begin{Bmatrix} -1 \\ 1 \end{Bmatrix} + \frac{1}{4} \begin{Bmatrix} 0 \\ 2 \end{Bmatrix} = \begin{Bmatrix} -1 \\ \frac{3}{2} \end{Bmatrix}$$

$$\nabla f_3 = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

Sequential Quadratic Programming (SQP)

E' il migliore algoritmo che utilizza il gradiente delle funzione obiettivo (velocità di convergenza, robustezza).

Utilizza i moltiplicatori di Lagrange \Rightarrow **non** ha bisogno della penalty function per i vincoli

$$\min f(\bar{x}) : \mathcal{R}^n \Rightarrow \mathcal{R}$$

con

$$h_k(\bar{x}) = 0 \quad k = 1, \dots, p$$

$$L(\bar{x}, \bar{\lambda}) = f(\bar{x}) + \sum_{k=1}^p \lambda_k h_k(\bar{x})$$

utilizzando le condizioni di Kuhn - Tucker

$$\begin{cases} \nabla L = 0 \Rightarrow \nabla f + \sum_{k=1}^p \lambda_k \nabla h_k(\bar{x}) = 0 \Rightarrow \nabla f + [\mathbf{H}]^T \bar{\lambda} = 0 \\ h_k(\bar{x}) = 0 \end{cases}$$

N+P equazioni

$$\mathbf{F}(\mathbf{Y}) = \mathbf{0}$$

dove

$$\mathbf{F} = \begin{Bmatrix} \nabla L \\ \bar{h} \end{Bmatrix}_{(n+p) \times 1} \quad \mathbf{Y} = \begin{Bmatrix} \bar{x} \\ \bar{\lambda} \end{Bmatrix}_{(n+p) \times 1} \quad \mathbf{0} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}_{(n+p) \times 1}$$

utilizzando Newton

$$\mathbf{Y}_{j+1} = \mathbf{Y}_j + \Delta \mathbf{Y}_j$$

$$[\nabla \mathbf{F}]_j^T \Delta \mathbf{Y}_j = -\mathbf{F}(\mathbf{Y}_j)$$

$$\begin{bmatrix} [\nabla^2 L] & [\mathbf{H}] \\ [\mathbf{H}]^T & [\mathbf{0}] \end{bmatrix}_j \begin{Bmatrix} \Delta \bar{x} \\ \Delta \bar{\lambda} \end{Bmatrix}_j = - \begin{Bmatrix} \nabla L \\ \bar{h} \end{Bmatrix}_j \quad (1)$$

$$\Delta \bar{x}_j = \bar{x}_{j+1} - \bar{x}_j$$

$$\Delta \bar{\lambda}_j = \bar{\lambda}_{j+1} - \bar{\lambda}_j$$

prendendo il primo set di equazioni da (1)

$$[\nabla^2 L]_j \Delta \bar{x}_j + [\mathbf{H}]_j \Delta \bar{\lambda}_j = -\nabla L_j$$

$$[\nabla^2 L]_j \Delta \bar{x}_j + [\mathbf{H}]_j (\bar{\lambda}_{j+1} - \bar{\lambda}_j) = -\nabla f_j - [\mathbf{H}]_j^T \bar{\lambda}_j$$

$$[\nabla^2 L]_j \Delta \bar{x}_j + [\mathbf{H}]_j \bar{\lambda}_{j+1} = -\nabla f_j$$

$$\begin{bmatrix} [\nabla^2 L] & [\mathbf{H}] \\ [\mathbf{H}]^T & [\mathbf{0}] \end{bmatrix}_j \begin{Bmatrix} \Delta \bar{x} \\ \bar{\lambda}_{j+1} \end{Bmatrix} = - \begin{Bmatrix} \nabla f_j \\ \bar{h}_j \end{Bmatrix}$$

se prendiamo la forma quadratica (da minimizzare)

$$Q = \nabla f^T \Delta \bar{x} + \frac{1}{2} \Delta \bar{x}^T [\nabla^2 L] \Delta \bar{x}$$

con i vincoli

$$h_k + \nabla h_k^T \Delta \bar{x} = 0 \quad k = 1, \dots, p$$

la forma Lagrangiana diventa

$$\tilde{L} = \nabla f^T \Delta \bar{x} + \frac{1}{2} \Delta \bar{x}^T [\nabla^2 L] \Delta \bar{x} + \sum_{k=1}^p \lambda_k (h_k + \nabla h_k^T \Delta \bar{x})$$

con le condizioni di Kuhn - Tucker

$$\begin{cases} \nabla f + [\nabla^2 L] \Delta \bar{x} + [\mathbf{H}] \bar{\lambda} = 0 \\ h_k + \nabla h_k^T \Delta \bar{x} = 0 \quad k = 1, \dots, p \end{cases}$$

Come si nota la minimizzazione della forma quadratica porta alle stesse equazioni della forma originale

Procedura di soluzione

Trovare \bar{S} che minimizzi

$$Q(\bar{S}) = \nabla f^T \bar{S} + \frac{1}{2} \bar{S}^T [\mathbf{H}] \bar{S}$$

con i vincoli

$$\beta_j g_j + \nabla g_j^T \bar{S} \leq 0 \quad j = 1, \dots, m$$

$$\bar{\beta} h_k + \nabla h_k^T \bar{S} = 0 \quad k = 1, \dots, p$$

con

$$\bar{\beta} \approx 0.9 \quad \beta_j = \begin{cases} 1 & \text{se } g_j \leq 0 \\ \bar{\beta} & \text{se } g_j \geq 0 \end{cases}$$

si può utilizzare una metodologia quasi - Newton

$$\bar{x}_{j+1} = \bar{x}_j + \alpha^* \bar{S}$$

dove α è lo step ottimale lungo la direzione s che ottimizza

$$\theta = f(\bar{x}) + \sum_{j=1}^m \lambda_j (\max[0, g_j(\bar{x})]) + \sum_{k=1}^p \lambda_{m+k} |h_k(\bar{x})|$$

con

$$\lambda_j = \begin{cases} |\lambda_j| & j = 1, \dots, m + p \quad \text{prima iterazione} \\ \max \left\{ |\lambda_j|, \frac{1}{2} (\tilde{\lambda}_j, |\lambda_j|) \right\} & \text{iterazioni successive} \end{cases}$$

Utilizzando quasi - Newton

$$[\mathbf{H}_{i+1}] = [\mathbf{H}_i] - \frac{[\mathbf{H}_i] \bar{P}_i \bar{P}_i^T [\mathbf{H}_i]}{\bar{P}_i^T [\mathbf{H}_i] \bar{P}_i} + \frac{\bar{\gamma} \bar{\gamma}^T}{\bar{P}_i^T \bar{P}_i}$$

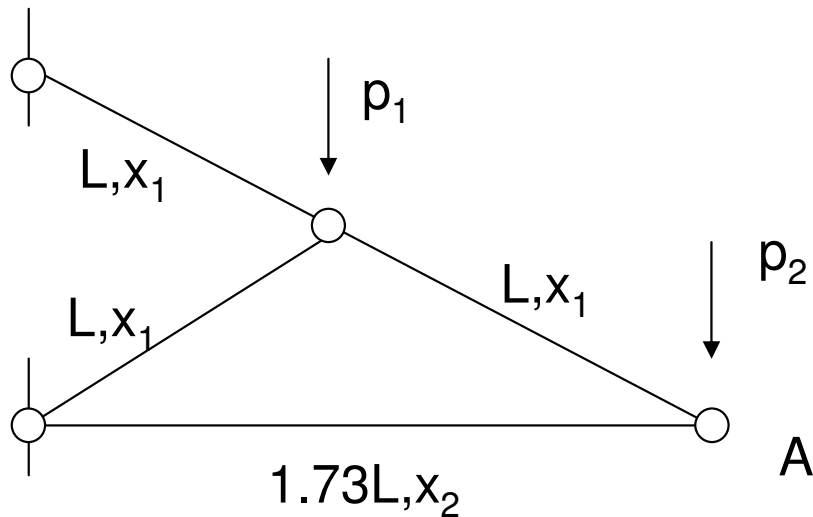
$$\bar{P}_i = \bar{x}_{i+1} - \bar{x}_i$$

$$\bar{\gamma} = \theta \bar{Q}_i + (1 - \theta) [\mathbf{H}_i] \bar{P}_i$$

$$\bar{Q}_i = \nabla_x \tilde{L}(\bar{x}_{i+1}, \bar{\lambda}_{i+1}) - \nabla_x \tilde{L}(\bar{x}_i, \bar{\lambda}_i)$$

$$\theta = \begin{cases} 1.0 & \text{se } \bar{P}_i^T \bar{Q}_i \geq 0.2 \bar{P}_i^T [\mathbf{H}_i] \bar{P}_i \\ \frac{0.8 \bar{P}_i^T [\mathbf{H}_i] \bar{P}_i}{\bar{P}_i^T [\mathbf{H}_i] \bar{P}_i - \bar{P}_i^T \bar{Q}_i} & \text{se } \bar{P}_i^T \bar{Q}_i < 0.2 \bar{P}_i^T [\mathbf{H}_i] \bar{P}_i \end{cases}$$

Esempio



Modulo di Young: 30E6 psi

Densità: 0.03333 lb/in³

P1= 10000 lb

P2= 20000 lb

Il peso della struttura è:

$$f=0.1x_1+0.05773x_2$$

La deflessione del punto A (verticale):

$$\delta_A = \frac{0.6}{x_1} + \frac{0.3464}{x_2}$$

$$\sigma_1 = \frac{50000}{x_1} \quad \sigma_4 = -\frac{34640}{x_2} \quad \text{stress}$$

$$\sigma_{\max} (\text{tensione}) = 8333.33 \text{ psi}$$

$$\sigma_{\min} (\text{compressione}) = -4948.57 \text{ psi}$$

$$\min f(x_1, x_2) = 0.1x_1 + 0.05773x_2$$

$$g_1(x_1, x_2) = \frac{0.6}{x_1} + \frac{0.3464}{x_2} - 0.1 \leq 0$$

$$g_2(x_1, x_2) = 6 - x_1 \leq 0$$

$$g_3(x_1, x_2) = 7 - x_2 \leq 0$$

$$\bar{x}_1 = (11.8765, 7.0)^T \quad g_1(\bar{x}_1) = g_3(\bar{x}_1) = 0 \quad g_2(\bar{x}_1) = -5.8765$$

$$f(\bar{x}_1) = 1.5917$$

$$\nabla f(\bar{x}_1) = \begin{Bmatrix} 0.1 \\ 0.05773 \end{Bmatrix} \quad \nabla g_1(\bar{x}_1) = \begin{Bmatrix} \frac{-0.6}{x_1^2} \\ \frac{-0.3464}{x_2^2} \end{Bmatrix} = \begin{Bmatrix} -0.004254 \\ -0.007069 \end{Bmatrix}$$

$$\nabla g_2(\bar{x}_1) = \begin{Bmatrix} -1 \\ 0 \end{Bmatrix} \quad \nabla g_3(\bar{x}_1) = \begin{Bmatrix} 0 \\ -1 \end{Bmatrix}$$

$$[\mathbf{H}_1] = [\mathbf{I}]$$

conseguentemente l'equazione della quadratica associata diventa

$$Q(\bar{s}) = 0.1s_1 + 0.05773s_2 + 0.5s_1^2 + 0.5s_2^2$$

$$\text{poichè } g_1 = g_3 = 0 \Rightarrow \beta_1 = \beta_3 = 0$$

$$\text{poichè } g_2 < 0 \Rightarrow \beta_2 = 1$$

i vincoli diventano :

$$\tilde{g}_1 = -0.004254s_1 - 0.007069s_2 \leq 0$$

$$\tilde{g}_2 = -5.8765 - s_1 \leq 0$$

$$\tilde{g}_3 = -s_2 \leq 0$$

Per risolvere il problema si utilizzano le condizioni di Kuhn - Tucker

$$\frac{\partial Q}{\partial s_1} + \sum_{j=1}^3 \lambda_j \frac{\partial \tilde{g}_j}{\partial s_1} = 0$$

$$\frac{\partial Q}{\partial s_2} + \sum_{j=1}^3 \lambda_j \frac{\partial \tilde{g}_j}{\partial s_2} = 0$$

$$\lambda_j \tilde{g}_j = 0 \quad j = 1, 2, 3$$

$$\tilde{g}_j \leq 0 \quad j = 1, 2, 3$$

$$\lambda_j \geq 0 \quad j = 1, 2, 3$$

Le prime due equazioni portano a

$$0.1 + s_1 - 0.004254\lambda_1 - \lambda_2 = 0$$

$$0.05773 + s_2 - 0.007069\lambda_1 - \lambda_3 = 0$$

la soluzione

$$s_1^* = -0.04791, s_2^* = 0.02883, \lambda_1^* = 12.2450, \lambda_2^* = 0, \lambda_3^* = 0$$

di conseguenza :

$$\bar{x} = \bar{x}_1 + \alpha \bar{s} = \begin{Bmatrix} 11.8765 - 0.049791\alpha \\ 7.0 + 0.02883\alpha \end{Bmatrix}$$

dove α minimizza

$$\begin{aligned} \phi = & 0.1(11.8765 - 0.049791\alpha) + 0.05773(7.0 + 0.02883\alpha) \\ & + 12.2450 \left(\frac{0.6}{11.8765 - 0.049791\alpha} + \frac{0.3464}{7.0 + 0.02883\alpha} - 0.1 \right) \end{aligned}$$

$$\alpha^* = 64.93$$

$$\bar{x}_2 = \begin{Bmatrix} 8.7657 \\ 8.8719 \end{Bmatrix} \Rightarrow f(\bar{x}_2) = 1.38874 \quad g_1(\bar{x}_2) = 0.0074932$$

Il prossimo passo è l'update della matrice $[\mathbf{H}]$

$$\tilde{L} = 0.1x_1 + 0.05773x_2 + 12.2450 \left(\frac{0.6}{x_1} + \frac{0.3464}{x_2} - 0.1 \right)$$

$$\nabla_x \tilde{L} = \begin{Bmatrix} \frac{\partial \tilde{L}}{\partial x_1} \\ \frac{\partial \tilde{L}}{\partial x_2} \end{Bmatrix} = \begin{Bmatrix} 0.1 - \frac{7.3470}{x_1^2} \\ 0.05773 - \frac{4.2417}{x_2^2} \end{Bmatrix}$$

$$\bar{P}_1 = \bar{x}_2 - \bar{x}_1 = \begin{Bmatrix} -3.1108 \\ 1.8719 \end{Bmatrix}$$

$$\bar{Q}_1 = \nabla_x \tilde{L}(\bar{x}_2) - \nabla_x \tilde{L}(\bar{x}_1) = \begin{Bmatrix} -0.04353 \\ 0.03267 \end{Bmatrix}$$

poichè $\bar{P}_1^T \bar{Q}_1 < 0.2 \bar{P}_1^T [\mathbf{H}_1] \bar{P}_1$

$$\theta = \frac{(0.8)(13.18)}{13.18 - 0.19656} = 0.81211$$

$$\bar{y} = \theta \bar{Q}_1 + (1 - \theta) [\mathbf{H}_1] \bar{P}_1 = \begin{Bmatrix} 0.54914 \\ -0.32518 \end{Bmatrix}$$

$$\Rightarrow [\mathbf{H}_2] = \begin{bmatrix} 0.2887 & 0.4283 \\ 0.4283 & 0.7422 \end{bmatrix}$$

Gli algoritmi basati sul calcolo del gradiente della funzione obiettivo sono ancora massicciamente usati per il design di sistemi ingegneristici mediante ottimizzazione.

I **vantaggi** di questi metodi sono ben conosciuti:

- elevata accuratezza della soluzione;
- necessitano di poche iterazioni per la convergenza.

Si presentano però alcune **problematiche**:

- bassa robustezza;
- calcolo delle derivate parziali (tempo di calcolo);
- impossibilità di lavorare con variabili intere (numero di palette, ecc.)

Per ovviare ad alcune problematiche degli algoritmi basati sull'utilizzo del gradiente sono stati sviluppati algoritmi di ottimizzazione stocastici.

Non necessitano il calcolo del gradiente, ma valutano semplicemente il valore della funzione obiettivo.

Maggiore robustezza, **minore** accuratezza.

Possibilità di effettuare ottimizzazioni multi obiettivo.

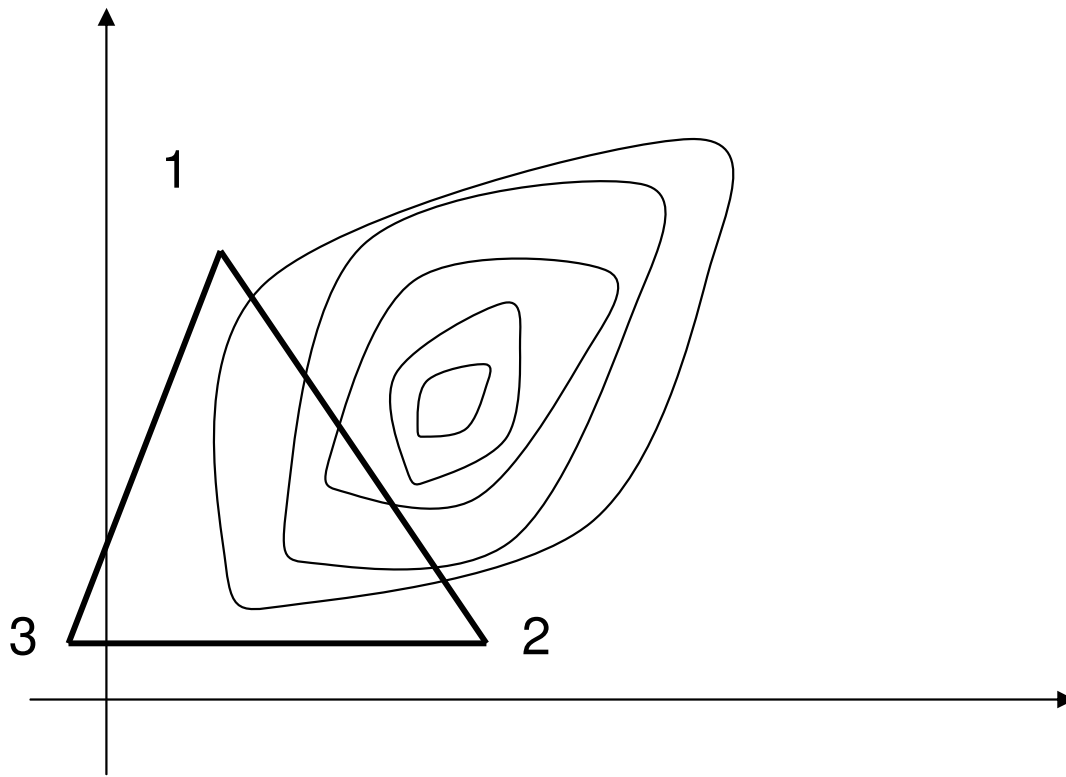
Algoritmi: Simplex, Simulated Annealing, Algoritmi genetici, Evolution Strategies.

SIMPLEX

In uno spazio ad n dimensioni qualunque figura formata da $n+1$ vertici viene chiamata **simpleso**

$N=2$ simpleso \Rightarrow triangolo

$N=3$ simpleso \Rightarrow tetraedro



Muovendo nello spazio di definizione delle variabili il simpleso si troverà il massimo della funzione

- **Riflessione**
- **Espansione**
- **Contrazione**

Riflessione

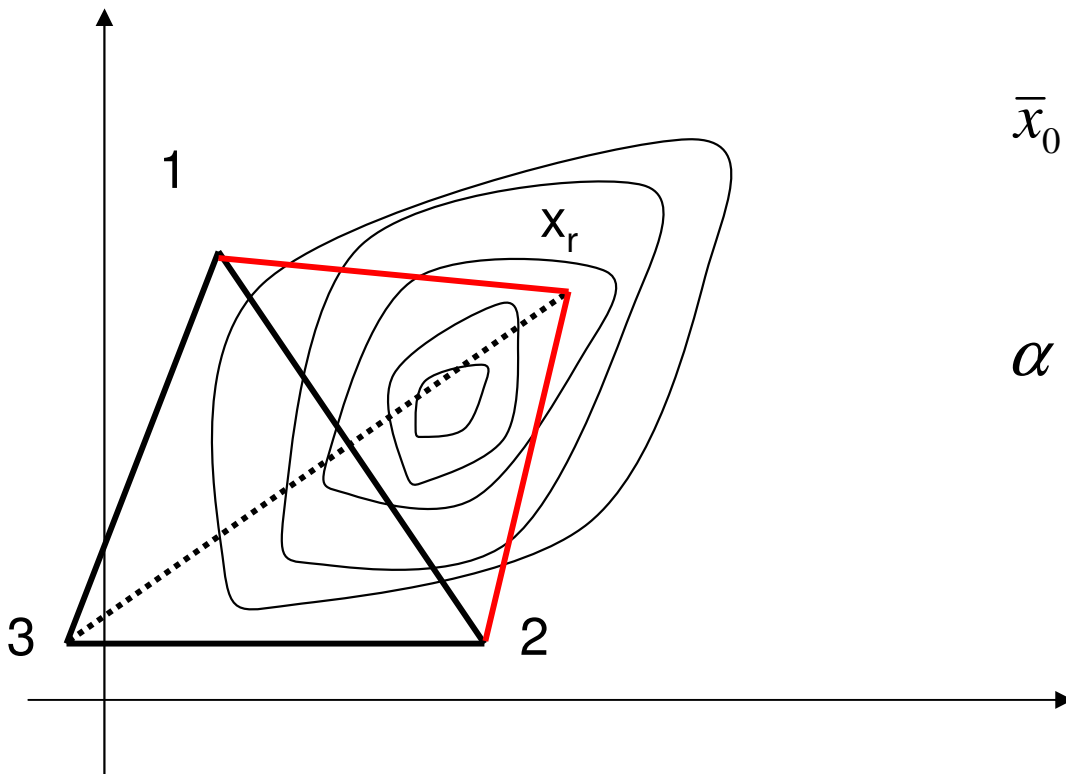
$$\min f : \mathcal{K}^n \Rightarrow \mathcal{K}$$

$$\bar{x}_r = (1 + \alpha)\bar{x}_0 - \alpha\bar{x}_h$$

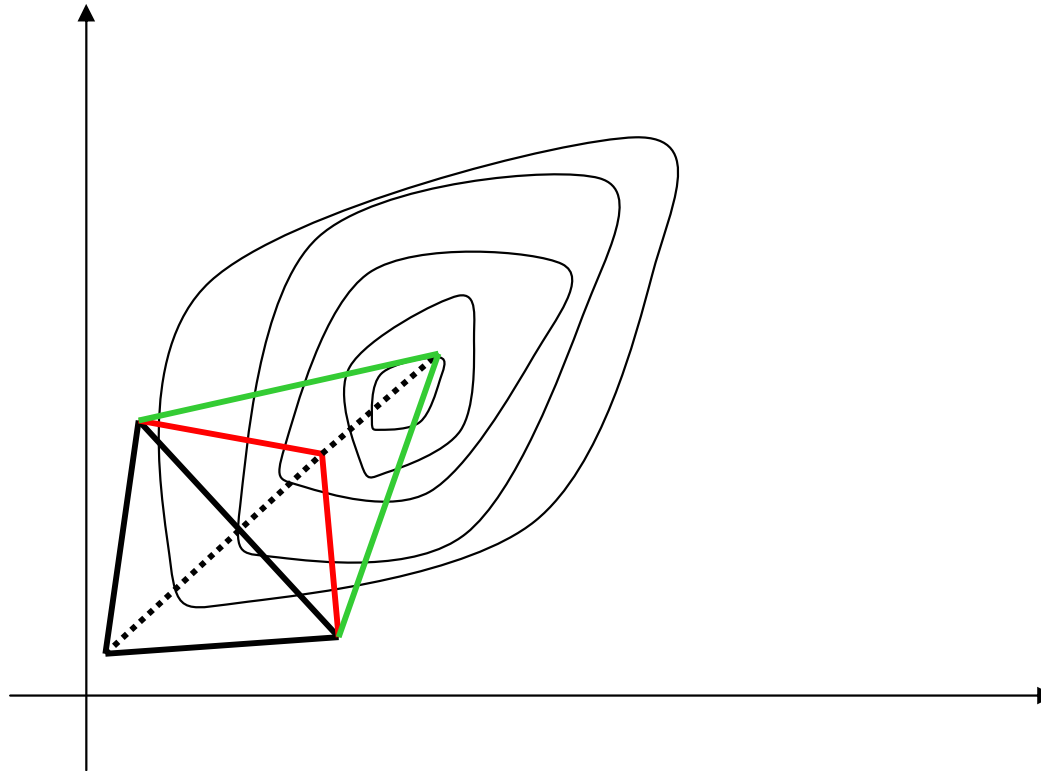
$$f(\bar{x}_h) = \max_{i=1 \text{ to } n+1} f(\bar{x}_i) \quad \text{vertice peggiore}$$

$$\bar{x}_0 = \frac{1}{n} \sum_{\substack{i=1 \\ i \neq h}}^{n+1} \bar{x}_i$$

$$\alpha = \frac{\|\bar{x}_r, \bar{x}_0\|}{\|\bar{x}_h, \bar{x}_0\|} \cong 1 \quad (> 0)$$



Espansione

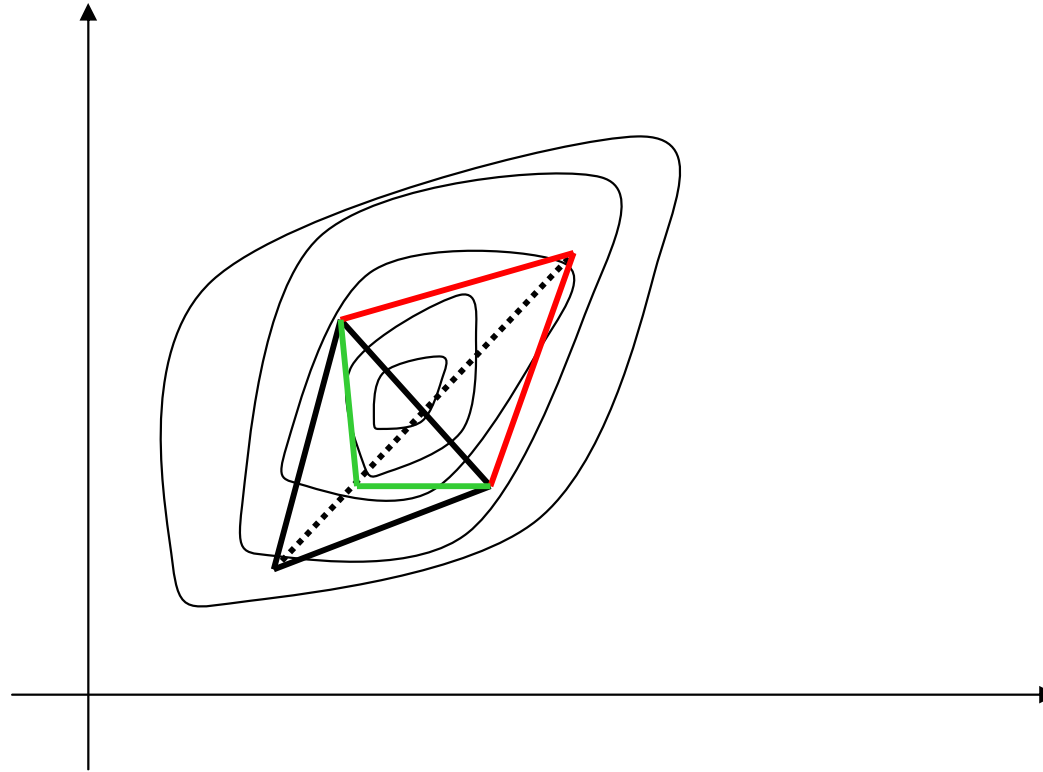


Se la riflessione porta ad un punto \bar{x}_r per il quale $f(\bar{x}_r) < f(\bar{x}_l)$

$$\bar{x}_e = \gamma \bar{x}_r + (1 - \gamma) \bar{x}_0$$

$$\gamma = \frac{\|\bar{x}_e, \bar{x}_0\|}{\|\bar{x}_r, \bar{x}_0\|} \cong 2 (> 1)$$

Contraazione



Se la riflessione porta ad un punto \bar{x}_r per il quale $f(\bar{x}_r) > f(\bar{x}_i)$ per tutti gli $i \neq h$ allora \bar{x}_h viene rimpiazzato da \bar{x}_r

$$\bar{x}_c = \beta \bar{x}_h + (1 - \beta) \bar{x}_0$$

$$\beta = \frac{\|\bar{x}_e, \bar{x}_0\|}{\|\bar{x}_h, \bar{x}_0\|}$$

$$Q = \left\{ \frac{\sum_{i=1}^{n+1} [f(\bar{x}_i) - f(\bar{x}_0)]^2}{n+1} \right\}^{1/2} \leq \varepsilon$$

CONVERGENZA

Creazione del Simpleso iniziale:

si può utilizzare qualunque metodologia DOE; oppure se si volesse un simpleso regolare:

$$\bar{x}_i = \bar{x}_0 + p\bar{u}_i + \sum_{j=1, j \neq i}^n q\bar{u}_j \quad i = 1, \dots, n$$

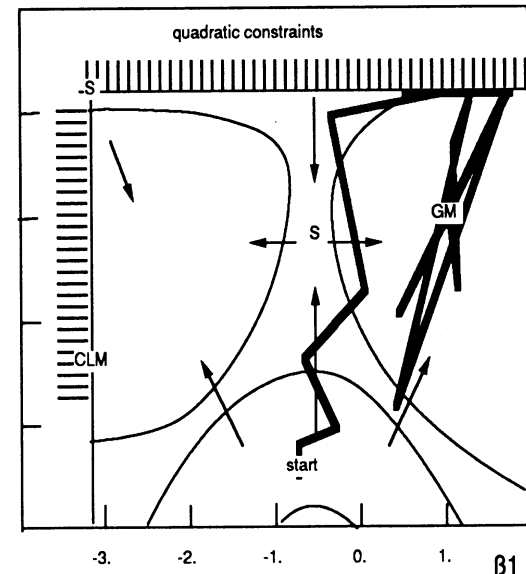
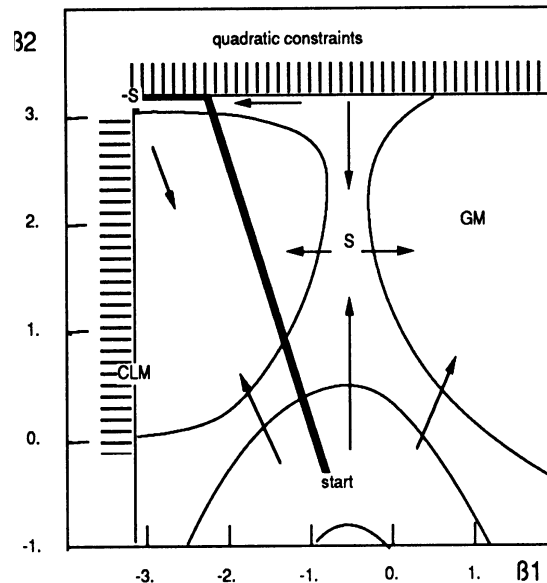
$$p = \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} + n - 1 \right) \quad q = \frac{a}{n\sqrt{2}} \left(\sqrt{n+1} - 1 \right)$$

- **Metodo robusto**
- **Normalmente computazionalmente costoso**
- **NON si possono usare variabili di categoria**
- **possono essere utilizzate variabili discrete (con base elevata)**

- Funzione da ottimizzare : TEST 1

BFGS

Simplex

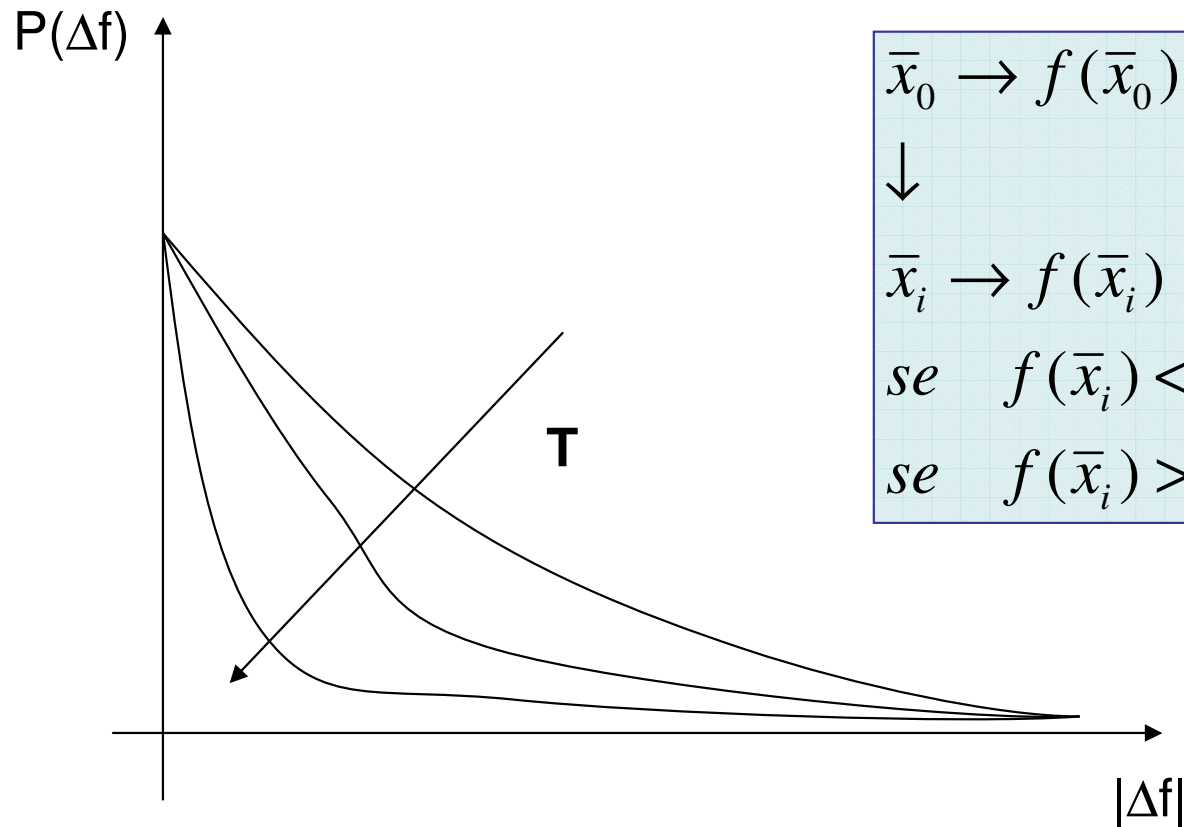


- Si nota come BFGS, al contrario di Simplex, non trova il massimo assoluto della funzione.

- L'algoritmo Simulated Annealing è stato uno dei primi algoritmi stocastici ad essere sviluppato, ed è ancora notevolmente usato per la sua robustezza (**problema del commesso viaggiatore**)
- L'algoritmo prende origine dalla solidificazione di metallo liquido: se la temperatura diminuisce lentamente si forma una struttura cristallina che rappresenta la **minima** energia interna del sistema.

- Viene usato il **Metropolis criterion** :

$$P(\Delta f) = e^{-\Delta f / kT} \quad \Delta f = f(X_{i+1}) - f(X_i)$$



$$\bar{x}_0 \rightarrow f(\bar{x}_0)$$

↓

$$\bar{x}_i \rightarrow f(\bar{x}_i)$$

se $f(\bar{x}_i) < f(\bar{x}_{i-1}) \Rightarrow$ sostituisco \bar{x}_{i-1} con \bar{x}_i

se $f(\bar{x}_i) > f(\bar{x}_{i-1}) \Rightarrow$ applico Metropolis

- **La metodologia non dipende dalla scelta iniziale di x_0 ;**
- **Non interessa la continuità o differenziabilità della funzione;**
- **Algoritmo estremamente robusto;**
- **Estremamente costoso da un punto di vista computazionale;**
- **Possibilità di utilizzare variabili discrete e di categoria;**
- **Vincoli implementati con la metodologia Penalty Function**

Algoritmo Genetico

L'algoritmo genetico si sta sempre maggiormente sviluppando grazie alle sue principali caratteristiche: robustezza, possibilità di effettuare ottimizzazioni multi obiettivo.

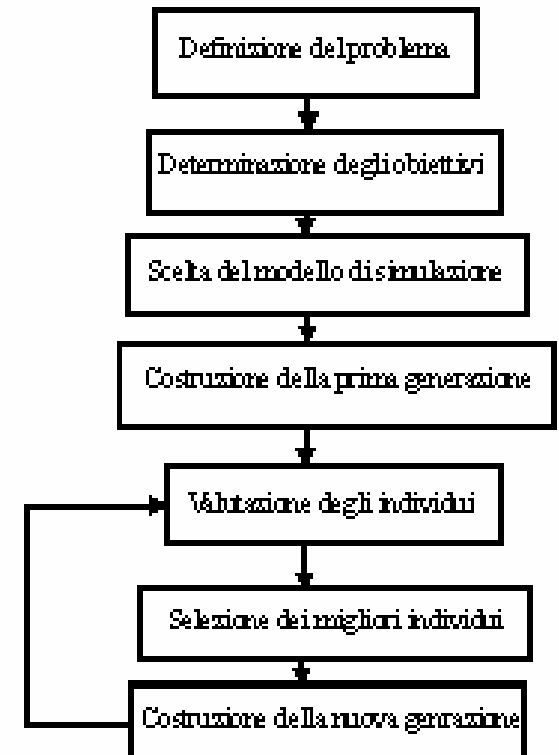
Concetto di **individuo** : punto dello spazio di definizione della funzione al quale e' associato il valore della funzione stessa.

$$f : \mathcal{R}^n \Rightarrow \mathcal{R}^m$$

$$i\text{-esimo individuo} \quad (x_1^i, \dots, x_n^i) \Rightarrow (f_1^i, \dots, f_m^i)$$

Operatori principali

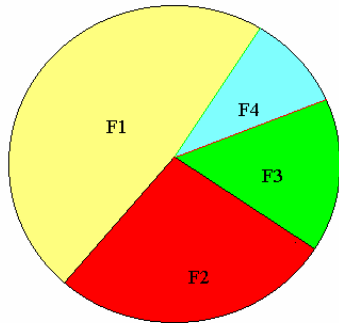
- 1) Selezione
- 2) Cross-over
- 3) Mutazione



Operatore: **selezione**

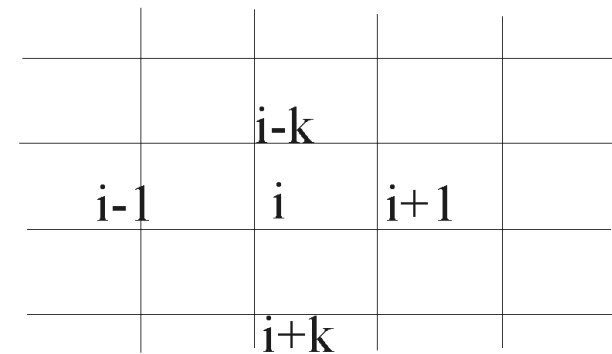
- **Scopo :**
trasmettere alle generazioni successive le caratteristiche migliori

Roulette



$F1 > F2 > F3 > F4$

Selezione locale

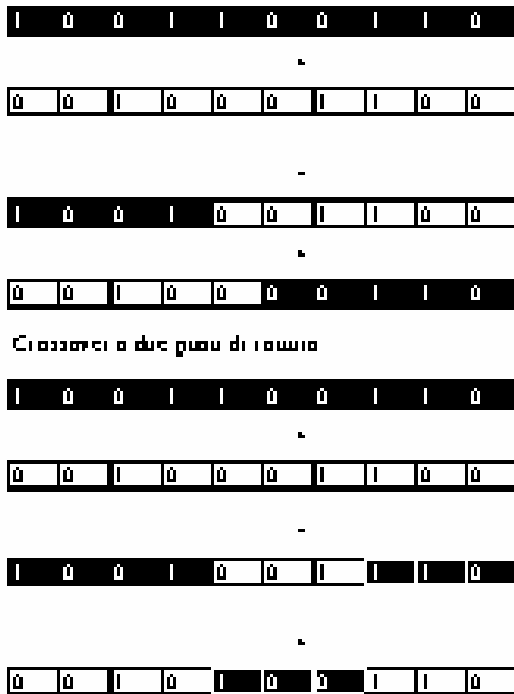


$$k = \text{sqrt}(\text{nind})$$

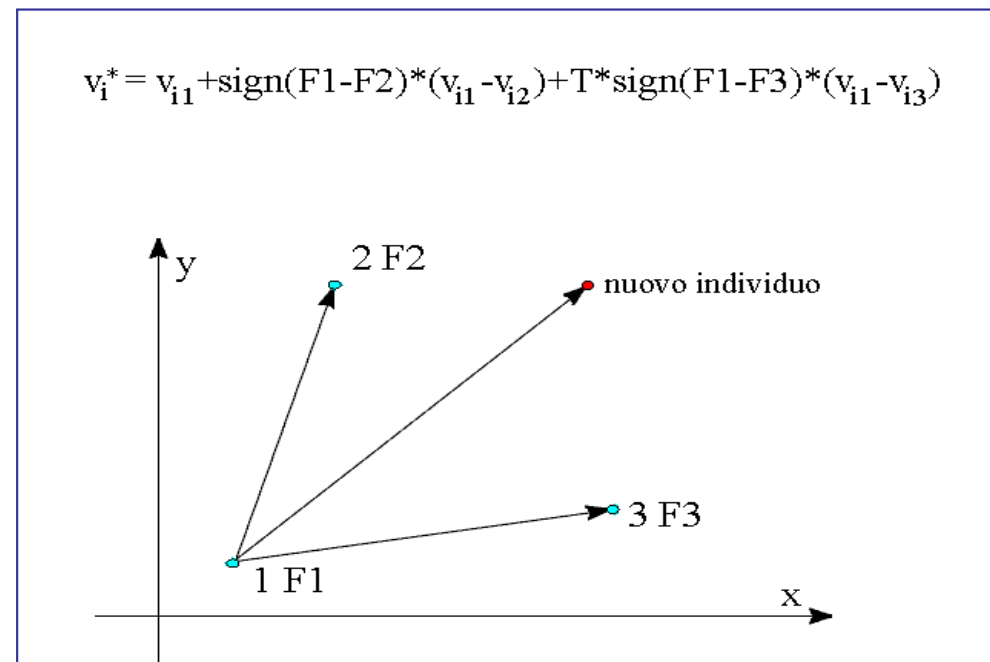
Scopo: creare le nuove configurazioni (individui)

Caso mono obiettivo

Cross-over classico



Cross-over direzionale

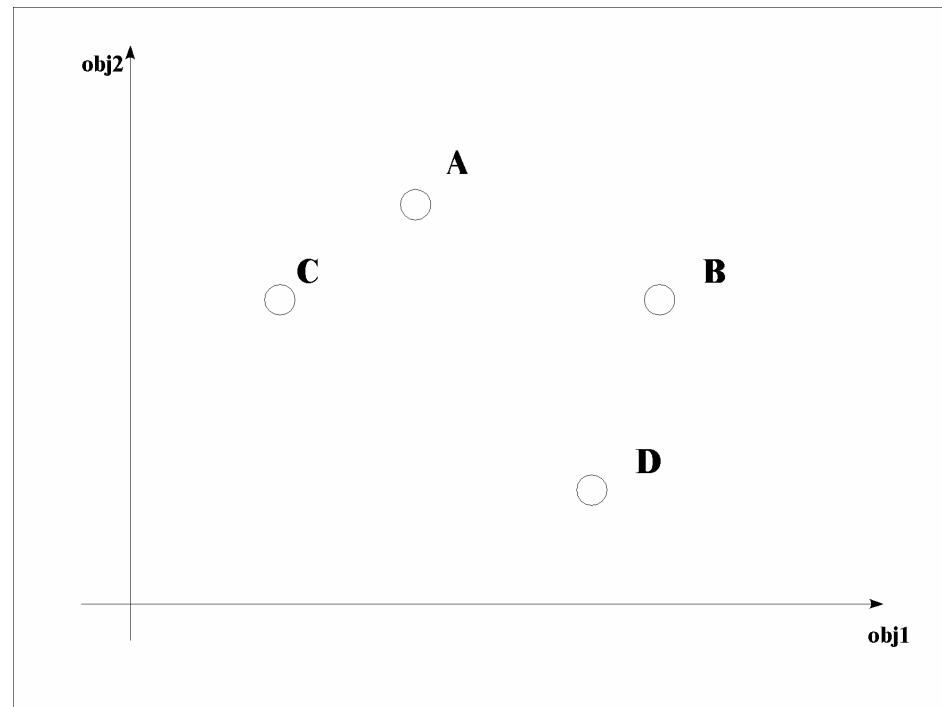


Mutazione GA

- Operatore necessario per evitare la **convergenza prematura** (uniformita' degli individui verso una soluzione di massimo relativo)



Caso Multi obiettivo



Dominanza secondo Pareto:

$$A >_P B \Leftrightarrow (\forall_i F_i(A) \geq F_i(B)) \cap (\exists j F_j(A) > F_j(B))$$

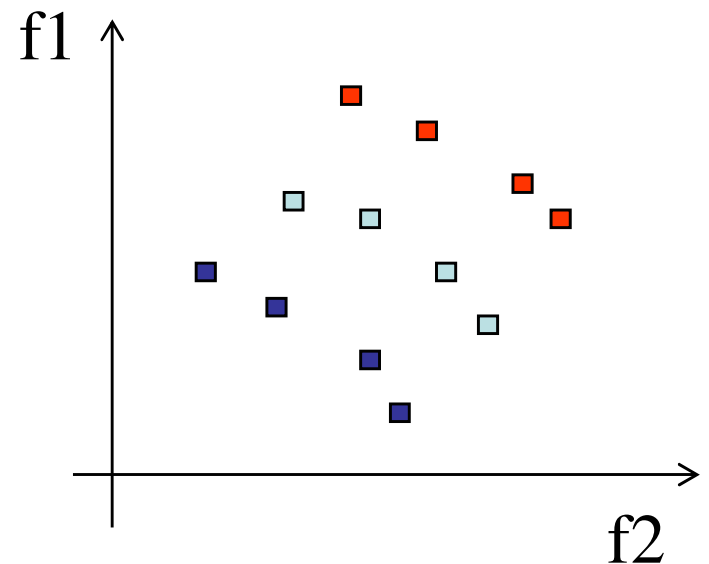
Caso Multi obiettivo

Si vede come per effettuare una reale ottimizzazione multi obiettivo, bisogna ottimizzare **contemporaneamente** ogni obiettivo **indipendentemente** uno dall'altro.

Critica alla formulazione $f = k_1 * f_1 + k_2 * f_2$
 $k_1 + k_2 = 1$

Soluzioni finali:

Fronte di Pareto = soluzioni non dominate



$$F_1(x, y) = -[1 + (A_1 + B_1)^2 + (A_2 + B_2)^2]$$

$$A_i = \sum_{j=1}^2 (a_{i,j} \cdot \sin(\alpha_j) + b_{i,j} \cdot \cos(\alpha_j))$$

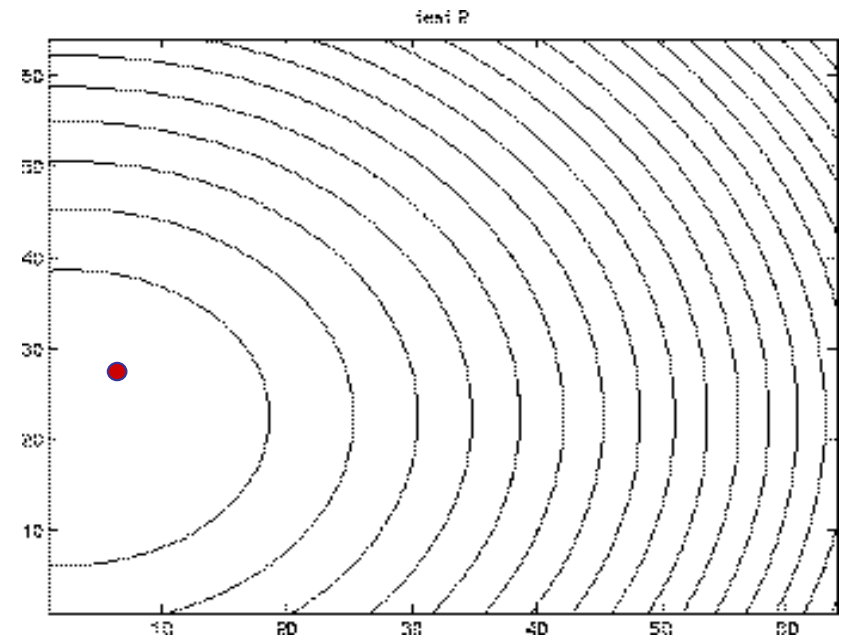
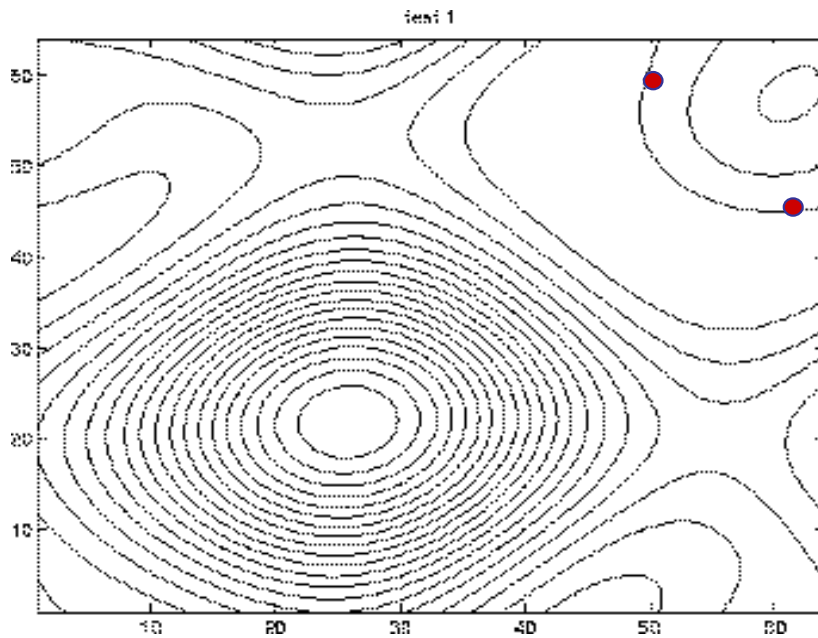
$$B_i = \sum_{j=1}^2 (a_{i,j} \cdot \sin(\beta_j) + b_{i,j} \cdot \cos(\beta_j))$$

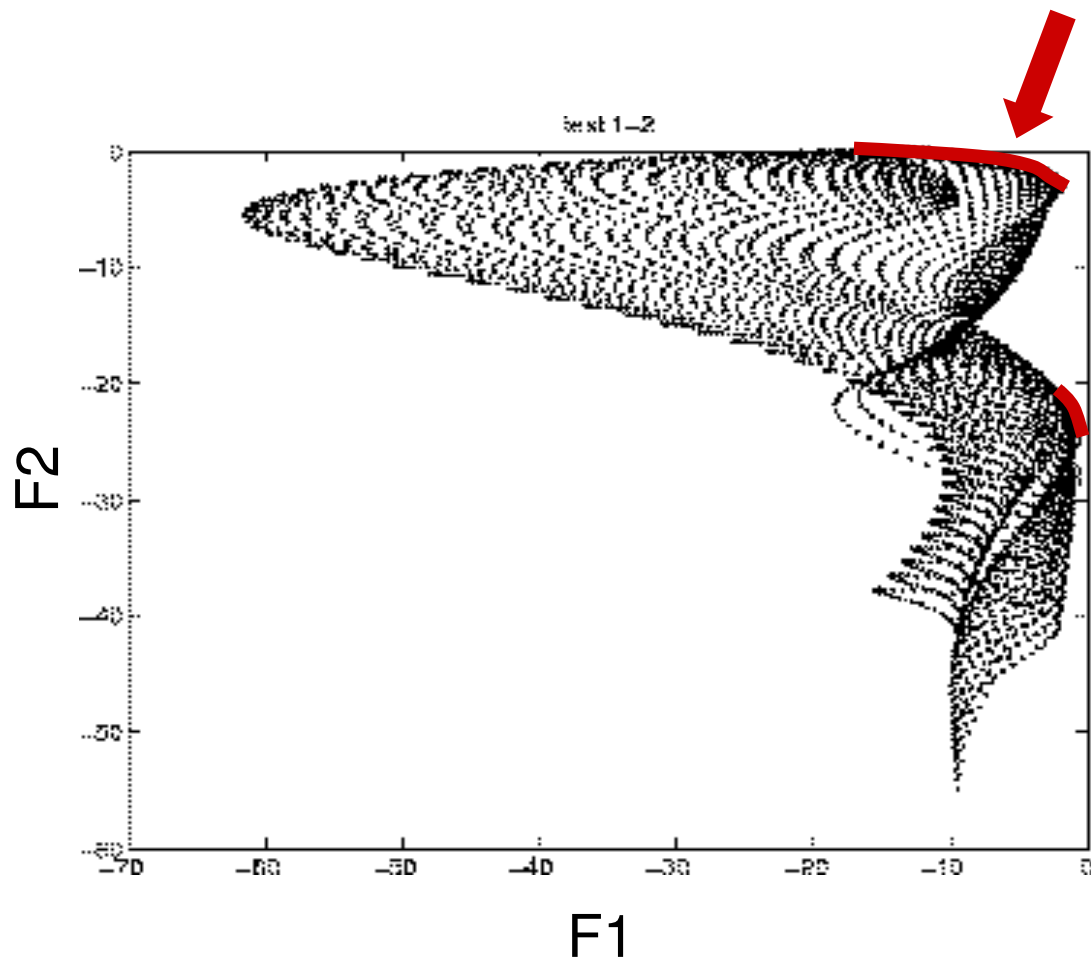
$$a = \begin{bmatrix} 0.5 & 1.0 \\ 1.5 & 2.0 \end{bmatrix} \quad b = \begin{bmatrix} -2.0 & -1.5 \\ -1.0 & -0.5 \end{bmatrix} \quad \alpha = [1.0 \quad 2.0]$$

Find the MAX for F1 and F2

$$x, y \in [-\pi, \pi]$$

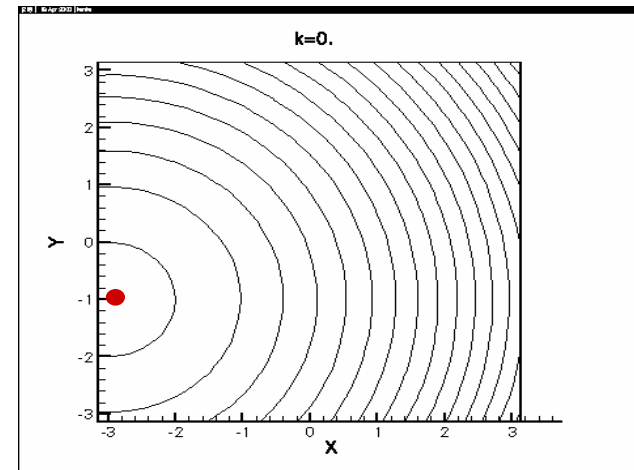
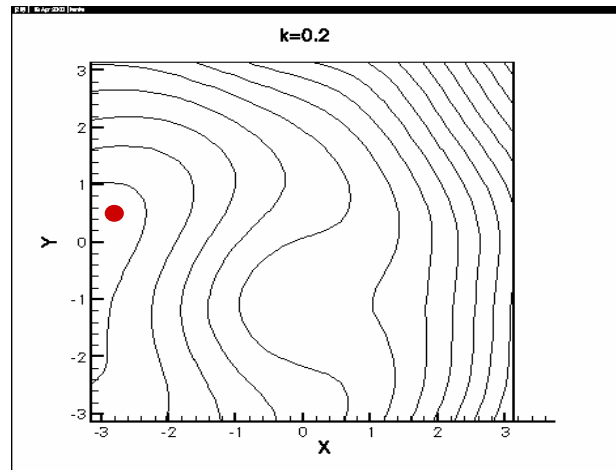
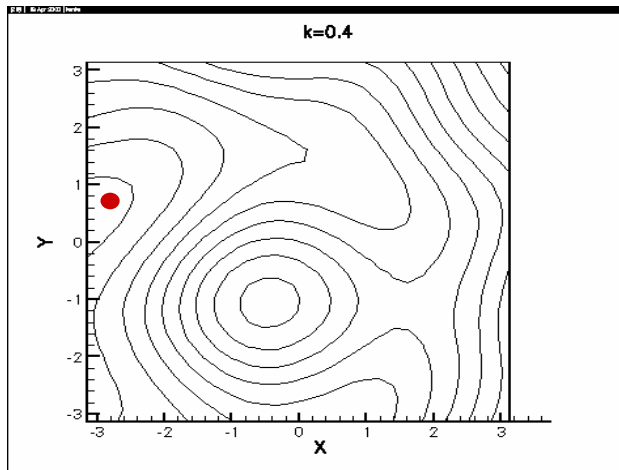
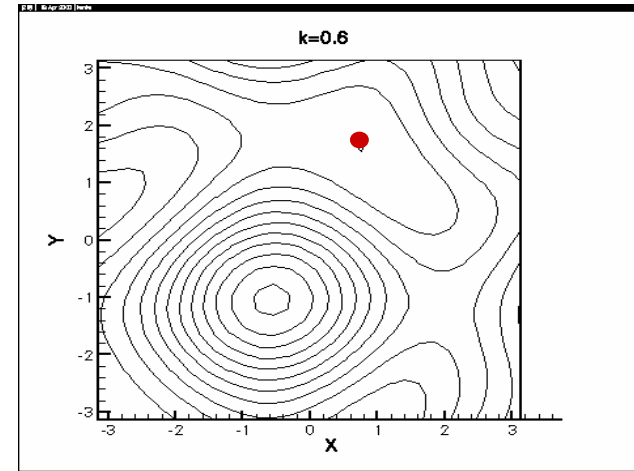
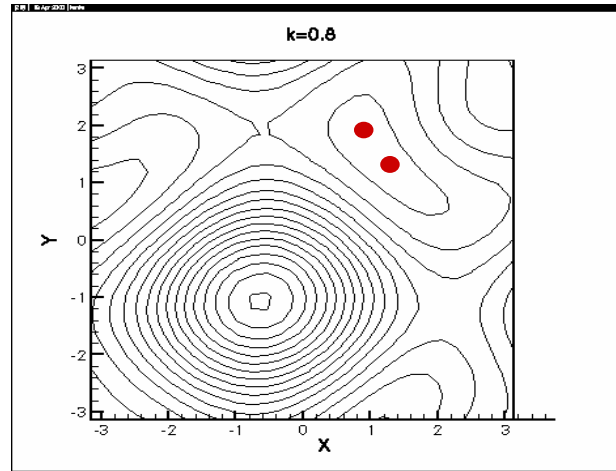
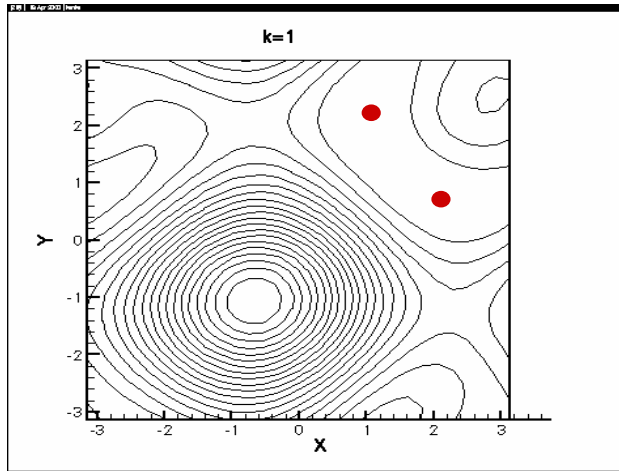
$$F_2(x, y) = -[(x + 3)^2 + (y + 1)^2]$$





**Si può notare
come le 2 zone
efficienti
(Fronte di
Pareto) sono 2
zone distinte**

Somma pesata: $k \cdot F1 + (1-k) \cdot F2$

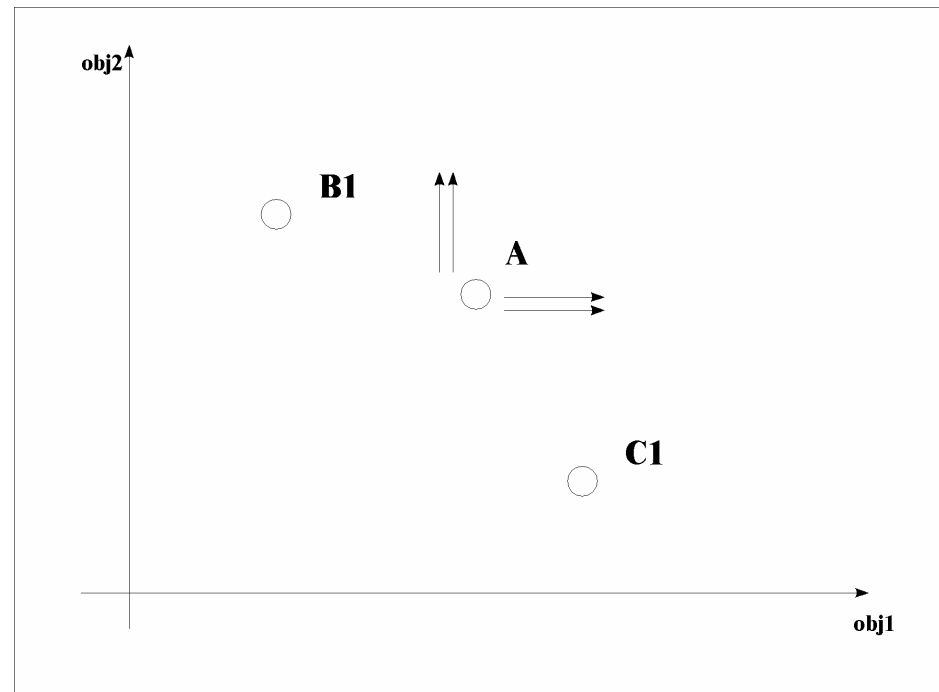


SELEZIONE: implementazione della dominanza secondo Pareto

CROSS OVER:

$$x_i^* = x_{1i} + \sum_{k=1}^m \left[\frac{s_k * sig(f_{1k} - f_{2k}) * (v_{1i} - v_{2i}) + t_k * sig(f_{1k} - f_{3k}) * (v_{1i} - v_{3i})}{m} \right]$$

s_k, t_k random



- **Normalmente non e' possibile definire a priori una strategia di approccio per un problema di ottimizzazione.**
- **Affinamento di una soluzione esistente : generalmente e' conveniente usare un metodo classico (SQP, BFGS).**
- **Nel caso contrario l' utilizzo di GA e' da privilegiare (multi obiettivo)**
- **Ibridizzazione** : ricerca delle soluzioni ottimali con GA, affinamento delle soluzioni ottenute con alcune iterazioni di SQP, BFGS.

- **Nel caso di ottimizzazioni multi obiettivo, il risultato non è una sola soluzione ma un insieme di più soluzioni **NON DOMINATE** (Fronte di Pareto).**
- **A volte la scelta della configurazione più interessante può non essere facile.**
- **Sviluppo di metodologie numeriche di aiuto alle decisioni:**
- **MCDM (Multi Criteria Decision Method)**