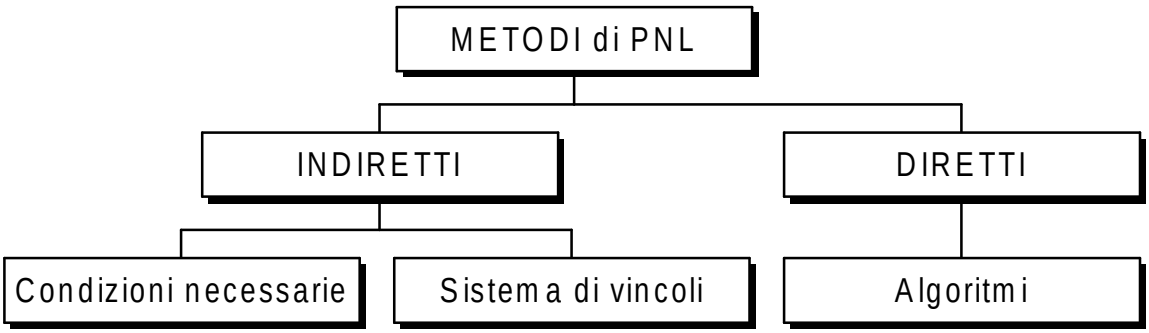
### *Programmazione non lineare: metodiche risolutive*




I *metodi indiretti* forniscono soltanto condizioni necessarie per l'ottimo. Tali condizioni, nella forma di sistema di equazioni e disequazioni, vengono utilizzate al fine di determinare la soluzione ottima.
I *metodi diretti* utilizzano di solito procedure numeriche iterative o *algoritmi* che operano sulla funzione obiettivo direttamente, per ottenere la soluzione numerica. In questo caso non c'è una fase intermedia di specifica delle condizioni per l'ottimo o meglio l'algoritmo ne tiene già conto strutturalmente. Fra i metodi diretti rientrano quelli che si basano sul gradiente. Tali algoritmi, pur presentando alcuni problemi di convergenza e di accuratezza della soluzione, vengono applicati ampiamente per la loro semplicità.

### *Algoritmi del gradiente per problemi di PNL non vincolata*

#### Il gradiente direzione

Il gradiente di una funzione vettoriale *n*-dimensionale differenziabile è il vettore di dimensione *n*:

$$\nabla f(\underline{x}) = \left[ \frac{\partial f}{\partial x_1}(\underline{x}), \frac{\partial f}{\partial x_2}(\underline{x}), \cdots, \frac{\partial f}{\partial x_n}(\underline{x}) \right]^T$$

Tale vettore definisce una direzione chiamata *gradiente direzione* ed è la base di tutti gli algoritmi che utilizzano il gradiente. Tale direzione sarà definita *direzione di discesa più veloce* (*steepest descent*) per i problemi di minimizzazione e *direzione di ascesa più veloce* (*steepest ascent*) per quelli di massimizzazione.

Individuiamo lo steepest descent per il problema non vincolato

$$\min_{\underline{x}} f(\underline{x})$$

con $\underline{x}$ vettore di dimensione $n$ ed $f(\underline{x})$ funzione differenziabile.

Scegliendo un punto di partenza arbitrario $\underline{x}_0$ si deve determinare un punto vicino $\underline{x}_0 + d\underline{x}$ tale che $f(\underline{x}_0 + d\underline{x}) < f(\underline{x}_0)$. Si vuole scegliere un $d\underline{x}$ tale che la differenza $f(\underline{x}_0) - f(\underline{x}_0 + d\underline{x})$ è massimizzata.

Il vettore $d\underline{x} = [dx_1, dx_2, \ldots, dx_n]^T$ descrive uno spostamento incrementale $ds$ lungo il percorso steepest descent. Tale spostamento è ovviamente $(ds)^2 = \sum_{i=1}^{n} (dx_i)^2$, ossia:

$$1 - \sum_{i=1}^{n} \left( \frac{dx_i}{ds} \right)^2 = 0$$

ove $\frac{dx_i}{ds}$ ($i$ = 1,2,…,$n$) sono i coseni direttori dello spostamento incrementale nello spazio $n$-dimensionale.

Si deve scegliere i coseni direttori in modo tale che il tasso di spostamento $\frac{df}{ds}$ sia massimizzato.

Dato che $\frac{df}{ds} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \cdot \frac{dx_i}{ds}$, il percorso steepest descent è ottenuto risolvendo il problema

$$\max_{dx_i/ds} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \cdot \frac{dx_i}{ds}$$

$$s.a.$$

$$1 - \sum_{i=1}^{n} \left( \frac{dx_i}{ds} \right)^2 = 0$$

A tal fine si possono utilizzare i moltiplicatori di Lagrange:

$$L\left( \frac{dx_i}{ds}, \lambda \right) = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \cdot \frac{dx_i}{ds} + \lambda \left[ 1 - \sum_{i=1}^{n} \left( \frac{dx_i}{ds} \right)^2 \right]$$

Le condizioni necessarie sono:

(1) $$\nabla_{dx_i/ds} L = \frac{\partial f}{\partial x_i} - 2\lambda \frac{dx_i}{ds} = 0 \qquad i = 1,2,\ldots,n$$

(2) $$\nabla_{\lambda} L = 1 - \sum_{i=1}^{n} \left( \frac{dx_i}{ds} \right)^2 = 0$$

Dalle (1) si ha $\frac{dx_i}{ds} = \frac{1}{2\lambda} \cdot \frac{\partial f}{\partial x_i}$, $i$ = 1,2,…,$n$, che sostituiti nella (2) danno $\lambda = \pm \frac{1}{2}\left[\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}\right)^2\right]^{1/2}$ e quindi la:

(3) $$\frac{dx_i}{ds}(\underline{x}) = \pm \frac{\frac{\partial f}{\partial x_i}(\underline{x})}{\left[\sum_{i=1}^{n}\left(\frac{\partial f}{\partial x_i}(\underline{x})\right)^2\right]^{1/2}} \qquad i = 1,2,\ldots,n$$

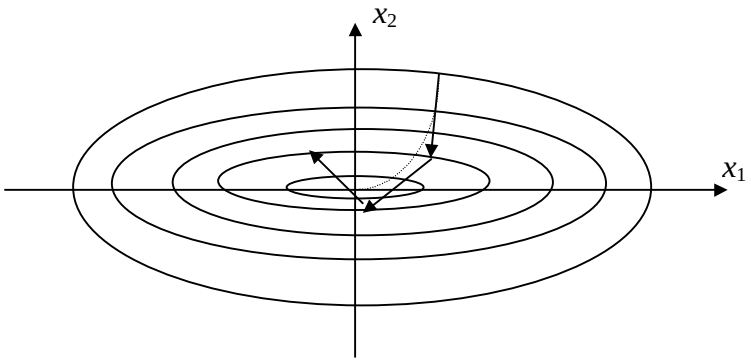
Dato che il denominatore è un fattore normalizzante positivo, si vede che lo steepest descent è definito dal vettore gradiente (numeratore). Per esso si assume segno positivo nella massimizzazione e negativo nella minimizzazione.

**Algoritmo del gradiente**

La (3) risolve il problema di ottimizzazione non vincolata ma ha il problema di essere una equazione *continua* rispetto $\underline{x}$. Utilizzando sistemi di calcolo digitale, quali sono gli attuali calcolatori, si adottano forme iterative. La forma iterativa per la (3) è chiamata *algoritmo del gradiente*. Tale algoritmo alla iterazione ($p$ + 1), partendo da un punto $\underline{x}_p \in S \subset E^n$ individua il punto $\underline{x}_{p+1} \in S \subset E^n$, nella direzione di ottimizzazione. Se riscriviamo la (3) nella forma $dx_i = \pm ds \frac{\partial f}{\partial x_i}^n(\underline{x})$, $i$ = 1,2,…,$n$, con $\frac{\partial f}{\partial x_i}^n$ la $i$-esima componente del gradiente normalizzato ed interpretando $ds$ come uno spostamento arbitrario $k$ da effettuare, l'algoritmo del gradiente si può scrivere come:

(4) $$\underline{x}^{p+1} = \underline{x}^p + k\nabla f(\underline{x}^p) \quad \begin{cases} k > 0 & \text{massimizzazione} \\ k < 0 & \text{minimizzazione} \end{cases}$$

ove ($p$ + 1) è il passo corrente e $\nabla f(\underline{x}^p)$ è il gradiente normalizzato calcolato in $\underline{x}^p$. $K$ non può essere molto grande in quanto, se lo fosse, si correrebbe il rischio di superare il valore ottimo e cominciare un processo di oscillazione intorno ad esso senza mai raggiungerlo (vedi figura).




Per questo motivo bisogna individuare alcuni meccanismi per ridurre il valore *k* in prossimità del valore ottimo. Si intuisce che comunque *k* dovrebbe essere grande per punti lontani dal valore ottimo, in modo da ridurre il numero di iterazioni e quindi il tempo di calcolo.

Vi sono molti metodi che sono stati proposti, a tal fine, come la riduzione di *k* ogni *m* iterazioni o quando la funzione cresce anziché decrescere, ma sicuramente il modo più efficiente, dal punto di vista del numero di iterazioni, è l'*algoritmo del gradiente con decremento ottimo*.

### *Algoritmo del gradiente con decremento ottimo*

In tale algoritmo si segue il procedimento del metodo del gradiente fino al punto $\underline{x}^p$ per il quale si raggiunge un minimo locale. Dato che nell'iterazione successiva la funzione obiettivo crescerà in corrispondenza del valore di $\underline{x}^{p+1}$ calcolato senza variare *k*, viene effettuata la determinazione numerica di un nuovo valore di *k*. In pratica il problema da risolvere è quello di determinare il valore ottimo di *k* che minimizza la $f(\underline{x}^{p+1})$ lungo la linea $\underline{x}^{p+1} - \underline{x}^p = \nabla f(\underline{x}^p)$. Esso si può riscrivere come:

$$\min_k f\left(\underline{x}^p + k\nabla f(\underline{x}^p)\right)$$

funzione di *k* solamente.

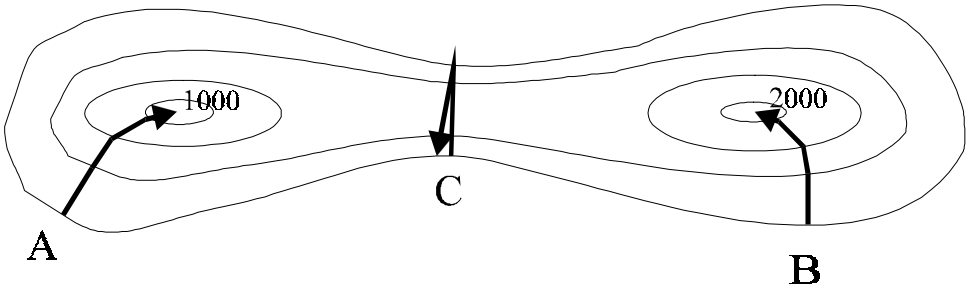
Le condizioni necessarie analitiche per tale problema di minimo sono $\frac{\partial f}{\partial k}\left(\underline{x}^p + k\nabla f(\underline{x}^p)\right) = 0$.

Poiché di solito non è nota la dipendenza analitica della funzione da *k* ma solo il valore della funzione per $\underline{x}^p$ fissato, al variare di *k*, si adottano i metodi numerici per funzioni unidimensionali, quali la *Fibonacci Search*.

Questo metodo tende a ridurre il numero di iterazioni anche se aumenta il tempo di calcolo a causa della determinazione ottima del valore di *k*.

**Considerazioni sulla convergenza e sulla bontà della soluzione**

Gli algoritmi fondati sul metodo del gradiente utilizzano solo informazioni locali per convergere ad un minimo locale. Il *punto di partenza* quindi riveste un ruolo fondamentale. Un buon punto di partenza è quello che si trova vicino all'ottimo, non solo perché ne permette l'individuazione, ma anche perché minimizza il numero di iterazioni e quindi il tempo di calcolo. Per ottenere un punto di partenza, nei casi più complessi, si estrae il punto più adatto da una griglia sovrapposta alla funzione. Questa metodica viene utilizzata soprattutto nel caso di superfici non unimodali. Nella figura seguente è rappresentato un tipico caso in cui, dipendentemente dal punto iniziale si può ottenere l'ottimo o un minimo locale o addirittura non convergere.




Allo stesso modo ha molta importanza anche il *test di interruzione* dell'algoritmo. Poiché non è nota la misura $f(\underline{x}^{p}) - f(\underline{x}^{*})$, ove $\underline{x}^{*}$ è il valore della variabile cui corrisponde l'ottimo, è necessario stabilire un test di interruzione dell'algoritmo che altrimenti potrebbe continuare indefinitamente nelle iterazioni. Di solito come criterio di interruzione si stabilisce un valore ε (*tolleranza* accettata):
nel caso di minimizzazione si interrompe se $f(\underline{x}^{p}) - f(\underline{x}^{p+1}) \le \varepsilon$, mentre nel caso di massimizzazione se $f(\underline{x}^{p+1}) - f(\underline{x}^{p}) \le \varepsilon$. Naturalmente l'ε più piccolo accettabile dipende dall'approssimazione di macchina. Inoltre ε grandi implicano soluzioni finali con cattiva approssimazione dell'ottimo e tempo di calcolo basso. Viceversa valori piccoli di ε conducono ad buona approssimazione della soluzione ottima ma alto tempo di calcolo.
Un'ultima considerazione va fatta relativamente al *tasso di convergenza* di tali algoritmi. Tale tasso può essere calcolato come $\lim_{p\to\infty} \frac{f(\underline{x}^{p+1})}{f(\underline{x}^{p})} = a$, con $0 \le a \le 1$. Per accelerare il tasso di convergenza sono stati adottati svariati metodi, tra cui:

$$\underline{x}^{p+1} = \underline{x}^{p} + k_1 \nabla f(\underline{x}^{p}) + k_2 \nabla f(\underline{x}^{p-1})$$

ossia utilizzando i gradienti direzione delle ultime due iterazioni. Se $k_1$ e $k_2$ sono decrementi ottimi, l'algoritmo prende il nome di *metodo delle tangenti parallele* anche noto col nome di *PARTAN*.

### *Algoritmi del gradiente per problemi di PNL vincolata*

Vi è un numero molto elevato di algoritmi che utilizzano la tecnica del gradiante per risolvere problemi di PNL vincolata. Essi, in generale, possono essere suddivisi in due grossi gruppi:

1. metodi *boundary-following*
2. metodi alle *funzioni di penalità*.

In questa sede ci occuperemo dei primi. I metodi boundary-following vengono applicati a particolari classi di problemi. Tale classe è quella dei problemi di PNL vincolata in cui o la funzione obiettivo o i vincoli sono lineari. In questo caso si possono adottare tecniche di linearizzazione delle equazioni o disequazioni non lineari e poi adottare algoritmi di affinamento della soluzione. Questi metodi possono essere applicati con particolari accorgimenti anche a problemi fortemente non lineari. Il *metodo delle direzioni ammissibili* e quello *della proiezione del gradiente* appartengono a questa classe di algoritmi.

#### Metodi boundary-following con vincoli rappresentati da diseguaglianze

Nella ottimizzazione *n*-dimensionale non vincolata si assume che lo spazio delle soluzioni ammissibili sia interamente $E^n$. Quando vengono aggiunti vincoli diseguaglianza la regione ammissibile viene ristretta alla regione interna e/o al bordo di questa regione. Scegliendo un punto di partenza interno alla regione di ammissibilità è possibile adottare un metodo del gradiente non vincolato ma solo fino a quando si raggiunge il limite della regione. A questo punto sarà necessario utilizzare un algoritmo che migliori la soluzione senza uscire al di fuori della regione di ammissibilità. Si intuisce allora che tali algoritmi dovranno, per i punti al limite della regione di ammissibilità, effettuare un spostamento lungo il limite stesso e non più seguendo la direzione gradiente: da ciò deriva il nome di metodi boundary-following. L'approccio più logico è quello di sostituire la direzione gradiente della funzione obiettivo con la direzione gradiente del vincolo leso. Tale metodo è noto col nome di *hemstitching* (ossia cucitrice), a causa del percorso a zigzag lungo il vincolo. Tale metodo pur essendo di facile implementazione purtroppo non è molto efficiente dato che sono necessarie molte iterazioni per spostarsi di poco lungo il vincolo. La cosa interessante è che fornisce un test di interruzione automatico dato che l'algoritmo si interrompe quando i vettori gradiente della funzione obiettivo e del vincolo leso sono paralleli. Dato che anche questo algoritmo utilizza informazioni locali, c'è sempre la possibilità di individuare un ottimo locale che non sia ottimo globale. Questo

può accadere soprattutto se la funzione obiettivo o lo spazio delle soluzioni ammissibili sono non convessi. A tal fine bisognerà estrarre più punti di partenza ed eseguire più volte l'algoritmo. Una alternativa è utilizzare un metodo di risoluzione per PNL non vincolata: se l'ottimo trovato è all'interno della regione ammissibile questo è anche l'ottimo per il problema vincolato. In caso contrario bisogna portarsi sul limite della regione ammissibile. Per questo si può utilizzare il gradiente direzione del vincolo leso o una combinazione lineare dei gradienti direzione dei vincoli lesi. Tali metodi vengono detti *esterni* dato che partono da punti non ammissibili.

**Metodo alle direzioni ammissibili**

Tale metodo incorpora un problema di PL che ad ogni iterazione determina la direzione ammissibile più vicina al gradiente della funzione obiettivo seguendo i vincoli ma senza uscire dalla regione di ammissibilità. In tal senso rientra in quei metodi detti *interni*. Ad ogni iterazione il nuovo punto viene individuato tramite la:

(5) $$\underline{x}^{p+1} = \underline{x}^{p} + k_p \underline{d}^{p}$$

dove $\underline{d}^p$ è il gradiente della funzione obiettivo se $\underline{x}^p$ è interno alla regione ammissibile, altrimenti:

(6) $$\underline{d}^{p} = \frac{\nabla f(\underline{x}^{p})}{\left\|\nabla f(\underline{x}^{p})\right\|} + \sum_{i=1}^{m} \frac{\nabla g_i(\underline{x}^{p})}{\left\|\nabla g_i(\underline{x}^{p})\right\|}.$$

Se $\underline{x}^p$ è all'interno della regione ammissibile esiste qualche $k_p > 0$ tale che la (5) è ammissibile per qualunque $\underline{d}^p$. Se invece $\underline{x}^p$ è su un vincolo si potrà individuare tramite un algoritmo di ottimizzazione lineare la direzione, più vicina al gradiente della funzione obiettivo, che non lede il vincolo, massimizzando lo spostamento nella direzione dell'ottimo.

Il metodo può essere realizzato iterativemente con una particolare procedura che utilizza un programma lineare ausiliario. Vediamo come funziona:

Supponiamo di dover risolvere il problema

$$\begin{aligned} & \max z = f(x) \\ \text{s.a} \quad & g_1(x) \le 0 \\ & g_2(x) \le 0 \\ & \cdots\cdots \\ & g_m(x) \le 0 \\ & x \ge 0 \end{aligned}$$

In cui le $g_i(x)$ sono sia funzioni lineari che non. Si può applicare la seguente procedura iterativa:

PASSO 1 - Si determina una soluzione iniziale ammissibile, sia $B$

PASSO 2 - Si risolva il programma lineare con le variabili $d_1, d_2, \ldots, d_{n+1}$:

$$\max z = d_{n+1}$$

$$\text{s.a} \quad -\left.\frac{\partial f}{\partial x_1}\right|_B d_1 - \left.\frac{\partial f}{\partial x_2}\right|_B d_2 - \cdots - \left.\frac{\partial f}{\partial x_n}\right|_B d_n + d_{n+1} \le 0$$

$$\left.\frac{\partial g_1}{\partial x_1}\right|_B d_1 + \left.\frac{\partial g_1}{\partial x_2}\right|_B d_2 + \cdots + \left.\frac{\partial g_1}{\partial x_n}\right|_B d_n + \nu_1 d_{n+1} \le -g_1(B)$$

$$\left.\frac{\partial g_2}{\partial x_1}\right|_B d_1 + \left.\frac{\partial g_2}{\partial x_2}\right|_B d_2 + \cdots + \left.\frac{\partial g_2}{\partial x_n}\right|_B d_n + \nu_2 d_{n+1} \le -g_2(B)$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\left.\frac{\partial g_m}{\partial x_1}\right|_B d_1 + \left.\frac{\partial g_m}{\partial x_2}\right|_B d_2 + \cdots + \left.\frac{\partial g_m}{\partial x_n}\right|_B d_n + \nu_m d_{n+1} \le -g_m(B)$$

$$d_j \le 1, \qquad j = 1, 2, \ldots, n+1$$

essendo $\nu_i$ ( $i = 1, 2, \ldots, m$) = 0 se $g_i(x)$ è lineare, 1 altrimenti

PASSO 3 - Se $d_{n+1} = 0 \Rightarrow x^* = B$, altrimenti vai al PASSO 4

PASSO 4 - $D = [d_1\ d_2\ \ldots\ d_n]^T$. Si determini il valore $k^*$ (> 0) massimo che renda ammissibili i vincoli del problema originale in cui $x = B + k\,D$. Se esiste $k^*$ vai al PASSO 5

PASSO 5 - $B = B + k^*\,D$. Tornare al PASSO 2

**Metodi boundary-following con vincoli rappresentati da uguaglianze**

L'applicazione delle tecniche del gradiente per problemi con vincoli uguaglianza è simile a quella del caso di vincoli disuguaglianza con l'eccezione che il percorso iterativo verso l'ottimo deve passare su punti che stanno sui vincoli. Chiaramente anche il punto di partenza apparterrà ai vincoli. In questo caso il problema potrebbe essere molto semplificato se alcuni dei vincoli potessero essere utilizzati per eliminare qualche variabile. Ciò si può fare per molte funzioni non lineari e non bisogna mai trascurare questa possibilità che permette di semplificare il problema e quindi di diminuire il tempo di calcolo.

**Metodo della proiezione del gradiente**

Tale algoritmo è applicato nel caso di vincoli lineari e funzione obiettivo non lineare, ossia il problema del tipo:

$$\min_{\underline{x}} f(\underline{x})$$

*s.a.*

$$\underline{\underline{A}}\,\underline{x} - \underline{b} = \underline{0}$$

Per tale algoritmo ad ogni iterazione viene individuato un nuovo punto tramite la:

(7) $$\underline{x}^{p+1} = \underline{x}^{p} + k_p \underline{\underline{P}} \nabla f(\underline{x}^{p}),$$

ove $\underline{\underline{P}} = [\underline{\underline{I}} - \underline{\underline{A}}^T (\underline{\underline{A}}\underline{\underline{A}}^T)^{-1}\underline{\underline{A}}]$ è chiamata *matrice di proiezione normalizzata*. Tale matrice proietta il gradiente sul sottospazio lineare definito dall'insieme di vincoli.

Tale algoritmo può essere esteso al caso di vincoli di diseguaglianza.