





**SAPIENZA**  
UNIVERSITÀ DI ROMA

UNIVERSITÀ di ROMA “LA SAPIENZA”

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA  
GESTIONALE

Anno Accademico 2022 –2023

**Appunti dalle lezioni di**  
**OTTIMIZZAZIONE CONTINUA**

**STEFANO LUCIDI**

Dipartimento di Ingegneria Informatica Automatica e Gestionale “Antonio  
Ruberti”

Università di Roma “La Sapienza”

versione preliminare



# Indice

<b>1</b>	<b>Problemi di Ottimizzazione Nonlineare</b>	<b>4</b>
1.1	Introduzione . . . . .	4
1.2	Condizioni di esistenza . . . . .	6
1.3	Condizioni di ottimalità locale . . . . .	8
1.3.1	Condizioni di ottimalità per problemi nonvincolati . . . . .	9
1.3.2	Condizioni di ottimalità per problemi vincolati . . . . .	9
1.4	Problemi particolari di Ottimizzazione . . . . .	11
1.4.1	Problemi di programmazione convessa . . . . .	11
1.4.2	Problemi di programmazione concava . . . . .	14
1.5	Condizioni di ottimalità globale . . . . .	15
1.5.1	Condizione di ottimalità necessaria e sufficiente . . . . .	15
1.5.2	Involucro convesso di una funzione . . . . .	19
1.5.3	Caratterizzazione dei minimi globali unici . . . . .	21
1.6	Proprietà generali dei metodi di ottimizzazione globale . . . . .	23
1.6.1	Definizione e funzione di perturbazione . . . . .	24
1.6.2	Proprietà di convergenza degli algoritmi di ottimizzazione globale deterministici . . . . .	26
1.6.3	Proprietà di convergenza degli algoritmi di ottimizzazione globale probabilistici . . . . .	28
<b>2</b>	<b>Metodi di ottimizzazione locale</b>	<b>29</b>
2.1	Introduzione . . . . .	29
2.2	Convergenza globale . . . . .	31
2.3	Rapidità di convergenza . . . . .	32
2.4	Metodi di ottimizzazione non vincolata . . . . .	33
2.4.1	Introduzione . . . . .	33
2.4.2	Algoritmi di ricerca unidimensionale . . . . .	38
2.4.3	Metodo del gradiente . . . . .	45
2.4.4	Metodo di Newton . . . . .	53
2.4.5	Metodi Quasi-Newton . . . . .	62
2.4.6	Metodo del gradiente coniugato . . . . .	66
2.4.7	Metodi di Newton troncato . . . . .	70
2.4.8	Metodi che non usano le derivate . . . . .	72
2.5	Metodi di ottimizzazione vincolata . . . . .	75

2.5.1	Introduzione . . . . .	75
2.5.2	Metodi sequenziali . . . . .	77
2.5.3	Funzioni di penalità . . . . .	81
<b>3</b>	<b>Metodi di ottimizzazione globale</b>	<b>92</b>
3.1	Metodi Probabilistici . . . . .	93
3.1.1	Algoritmi che utilizzano punti scelti a caso su $\mathcal{D}$ . . . . .	93
3.1.2	Metodi di tipo “simulated annealing” . . . . .	97
3.1.3	Metodi che usano “popolazioni” di punti (questa sezione non fa parte del programma di esame). . . . .	102
3.2	Metodi che utilizzano partizioni dell’insieme ammissibile . . . . .	106
3.2.1	Schema generale di un metodo che usa partizioni dell’insieme ammissibile . . . . .	107
3.2.2	Scelta dei sottoinsiemi da partizionare: sottoinsiemi di dimensioni più grandi. . . . .	113
3.2.3	Scelta dei sottoinsiemi da partizionare: sottoinsiemi più promettenti. . . . .	116
3.2.4	Algoritmo di Partizione con minimizzazioni locali (questa sezione non fa parte del programma). . . . .	122
3.2.5	L’Algoritmo Direct . . . . .	124
3.2.6	Estensione dell’Algoritmo Direct che utilizza una stima della costante di Lipschitz della funzione obiettivo . . . . .	132
3.3	Funzioni di tipo “Filled” . . . . .	136
3.3.1	Funzioni filled di tipo 1 . . . . .	139
3.3.2	Funzioni filled di tipo 2. . . . .	143

## Notazioni

$R^n$	spazio dei vettori $x$ a $n$ componenti reali;
$x \in R^n$	è inteso come vettore colonna;
$e_i \in R^n$	è il versore unitario $i$ -esimo, cioè il vettore che ha tutte componenti nulle escluso l' $i$ -esima che è uguale uno;
$R_+$	insieme dei reali non negativi;
$x^T$	vettore riga ottenuto come trasposto di $x$ ;
$(x)_i$	oppure $x_i$ indicano la $i$ -ma componente di $x$ (quindi $x_i \in R$ );
$x_k$	indica il $k$ -mo vettore di una successione (quindi $x_k \in R^n$ );
$\{x_k\}$	successione formata dai vettori $x_k$ ;
$\{x_k\}_K$	sottosuccessione definita dall' insieme (infinito) di indici $K$ ;
$\ x\ $	norma di $x$ ; in assenza di altre indicazioni, $\ x\ $ è la norma euclidea, ossia $\ x\  = \left(\sum_{i=1}^n x_i^2\right)^{1/2}$ , dove $x_i \in R$ sono le componenti di $x$ ;
$B(x^*; \rho)$	sfera aperta di raggio $\rho > 0$ con centro $x^* \in R^n$ , ossia: $B(x^*; \rho) = \{x \in R^n : \ x - x^*\  < \rho\}$ ;
$\nabla f(x)$	gradiente di una funzione $f : R^n \rightarrow R$ calcolato in $x$ ; $\nabla f(x)$ è inteso come vettore colonna con componenti $\frac{\partial f(x)}{\partial x_j}$ , $j = 1, \dots, n$ ;
$\nabla^2 f(x)$	matrice Hessiana ( $n \times n$ ) di una funzione $f : R^n \rightarrow R$ calcolata in $x$ , con componenti $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ , per $i, j = 1, \dots, n$ .
$Int(\mathcal{A})$	interno dell'insieme $\mathcal{A} \subseteq R^n$ .
$\partial(\mathcal{A})$	frontiera dell'insieme $\mathcal{A} \subseteq R^n$ .
$\mathcal{A} \setminus \mathcal{B}$	l'insieme definito da $\{x \in \mathcal{A} : x \notin \mathcal{B}\}$ .

# Capitolo 1

## Problemi di Ottimizzazione Nonlineare

Molti problemi che nascono nell'ambito dell'Ingegneria, dell'Economia e delle Scienze esatte e naturali possono essere rappresentati come problemi di ottimizzazione nonlineare. Questo motiva il sempre maggiore interesse dal punto di vista tecnico e scientifico verso lo studio e lo sviluppo di metodi che siano in grado di affrontare e risolvere questa classe di difficili problemi matematici.

### 1.1 Introduzione

Un problema di ottimizzazione consiste nel cercare di determinare dei punti appartenenti ad un insieme  $\mathcal{F}$  in cui una funzione  $f$  assume valori più bassi possibile. Tale problema viene rappresentato nella forma:

$$\begin{aligned} \min f(x) \\ x \in \mathcal{F}, \end{aligned} \tag{1.1}$$

dove

- la funzione  $f : \mathcal{F} \rightarrow R$  è detta *funzione obiettivo*;
- l'insieme  $\mathcal{F} \subseteq R^n$  è detto *insieme ammissibile*.

Per caratterizzare i possibili punti di soluzione del precedente problema di minimizzazione si possono introdurre le seguenti definizioni.

**Definizione 1.1.1** Un punto  $x^* \in \mathcal{F}$  si dice punto di minimo globale di  $f$  su  $\mathcal{F}$  se

$$f(x^*) \leq f(x), \quad \text{per ogni } x \in \mathcal{F}.$$

**Definizione 1.1.2** Un punto  $x^* \in \mathcal{F}$  si dice punto di minimo globale stretto di  $f$  su  $\mathcal{F}$  se

$$f(x^*) < f(x), \quad \text{per ogni } x \in \mathcal{F}, \quad x \neq x^*.$$

**Definizione 1.1.3** Un punto  $x^* \in \mathcal{F}$  si dice punto di minimo locale di  $f$  su  $\mathcal{F}$  se esiste un intorno  $B(x^*; \rho)$ , con  $\rho > 0$  tale che

$$f(x^*) \leq f(x), \quad \text{per ogni } x \in \mathcal{F} \cap B(x^*; \rho).$$

**Definizione 1.1.4** Un punto  $x^* \in \mathcal{F}$  si dice punto di minimo locale stretto di  $f$  su  $\mathcal{F}$  se esiste un intorno  $B(x^*; \rho)$ , con  $\rho > 0$  tale che

$$f(x^*) < f(x), \quad \text{per ogni } x \in \mathcal{F} \cap B(x^*; \rho), \quad x \neq x^*.$$

Si può notare che, se si ha un problema di massimizzazione cioè se si deve trovare un punto in cui la funzione  $f$  assume valore più alto possibile, ci si può sempre ricondurre a un problema di minimo, cambiando di segno la funzione obiettivo. Infatti un punto di massimo del problema

$$\max_{x \in \mathcal{F}} f(x)$$

è un punto  $x^* \in \mathcal{F}$  che, per definizione, soddisfa la seguente proprietà:

$$f(x^*) \geq f(x), \quad \text{per ogni } x \in \mathcal{F},$$

che è equivalente a:

$$-f(x^*) \leq -f(x), \quad \text{per ogni } x \in \mathcal{F},$$

da cui segue che  $x^*$  è anche un punto di minimo del problema

$$\min_{x \in \mathcal{F}} -f(x)$$

e risulta:

$$\max_{x \in \mathcal{F}} f(x) = -\min_{x \in \mathcal{F}} (-f(x)).$$

Perciò non si ha nessuna perdita di generalità a studiare ed affrontare solamente problemi di minimizzazione o, viceversa, solamente problemi di massimizzazione.

La natura del Problema (1.1) e, quindi, la sua difficoltà di risoluzione dipendono, ovviamente, dalle caratteristiche della funzione obiettivo e dalla struttura dell'insieme ammissibile. Usualmente un problema di ottimizzazione viene caratterizzato dal fatto che si abbia completa libertà o meno nella scelta del vettore  $x$ , infatti:

- è detto *problema di minimizzazione non vincolata* se  $\mathcal{F} = R^n$ , cioè se l'insieme ammissibile  $\mathcal{F}$  coincide con tutto lo spazio  $R^n$ , cioè:

$$\min_{x \in R^n} f(x) \tag{1.2}$$

- viene detto, invece, *problema di minimizzazione vincolata* un problema in cui  $\mathcal{F} \subset R^n$ .



Tuttavia, può essere considerato come un problema di minimizzazione non vincolato anche un qualsiasi problema in cui l'insieme ammissibile  $\mathcal{F}$  è un insieme aperto. Infatti, come nel caso in cui  $\mathcal{F} = R^n$ , i punti di minimo del problema possono essere caratterizzati esclusivamente dall'andamento della funzione obiettivo in un intorno del punto e non dal fatto che ci siano dei vincoli sulle variabili del problema. Perciò, per i problemi in cui l'insieme ammissibile è un insieme aperto, si adattano facilmente tutti i risultati e metodi proposti per il caso in cui  $\mathcal{F} = R^n$ .

Tra i problemi vincolati in cui  $\mathcal{F}$  è un insieme chiuso, la classe più comunemente considerata è quella in cui  $\mathcal{F}$  è descritto attraverso un *insieme finito di vincoli di eguaglianza e disequaglianza*:

$$\mathcal{F} = \{x \in R^n : g(x) \leq 0, \quad h(x) = 0\},$$

in cui  $g : R^n \rightarrow R^m$  e  $h : R^n \rightarrow R^p$  sono vettori di funzioni assegnate. Il problema di ottimo si può indicare, in tal caso, ponendo:

$$\begin{aligned} \min \quad & f(x) \\ & g(x) \leq 0 \\ & h(x) = 0. \end{aligned} \tag{1.3}$$

Nei paragrafi successivi, dopo aver richiamato le condizioni di esistenza delle soluzioni del Problema 1.1, vengono descritte alcune condizioni che cercano di dare delle caratterizzazioni di tali soluzioni.

## 1.2 Condizioni di esistenza

Nel cercare di risolvere il Problema (1.1) la prima difficoltà da affrontare è quella di capire se è ben posto, nel senso che potrebbe non esistere un punto in  $\mathcal{F}$  in cui la funzione  $f(x)$  assume valore più piccolo. Infatti, si potrebbe presentare una delle seguenti situazioni:

- l'insieme ammissibile  $\mathcal{F}$  potrebbe essere vuoto;
- l'insieme ammissibile  $\mathcal{F}$  potrebbe essere non vuoto ma la funzione obiettivo potrebbe essere illimitata inferiormente su  $\mathcal{F}$  ossia  $\inf_{x \in \mathcal{F}} f(x) = -\infty$ ;
- l'insieme ammissibile  $\mathcal{F}$  potrebbe essere non vuoto e la funzione obiettivo potrebbe essere limitata inferiormente su  $\mathcal{F}$  ma, anche in questo caso, *potrebbero non esistere punti di minimo globale* di  $f$  su  $\mathcal{F}$ ;

Una condizione *sufficiente* (ma non necessaria) per l'esistenza di un punto di minimo globale di un problema di ottimizzazione è quella espressa dalla proposizione seguente, che segue dal ben noto Teorema di Weierstrass.

**Proposizione 1.2.1** *Sia  $\mathcal{F} \subset R^n$  un insieme non vuoto e compatto. Sia  $f$  una funzione continua su  $\mathcal{F}$ . Allora esiste un punto di minimo globale di  $f$  in  $\mathcal{F}$ .*

La precedente proposizione si applica solamente alla classe dei problemi *vincolati* in cui l'insieme ammissibile è compatto. Per poter stabilire risultati di esistenza per problemi con insiemi ammissibili non compatti (in particolare nel caso in cui  $\mathcal{F} = R^n$ ) è necessario cercare di caratterizzare un qualche sottoinsieme di  $\mathcal{F}$  contenente le soluzioni ottime del problema.

A questo fine si introduce la definizione seguente.

**Definizione 1.2.2** *Sia  $\mathcal{F} \subseteq R^n$  e sia  $f : \mathcal{F} \rightarrow R$ ; si definisce insieme di livello di  $f$  su  $\mathcal{F}$  ogni insieme non vuoto del tipo:*

$$\mathcal{L}(\alpha) := \{x \in \mathcal{F} : f(x) \leq \alpha\},$$

in cui  $\alpha \in R$ .

In particolare, se  $x_0 \in \mathcal{F}$ , indichiamo con  $\mathcal{L}_0$  l'insieme di livello  $\mathcal{L}(f(x_0))$ , ossia:

$$\mathcal{L}_0 := \{x \in \mathcal{F} : f(x) \leq f(x_0)\}. \quad (1.4)$$

A questo punto, si può enunciare il risultato seguente che stabilisce una condizione sufficiente per l'esistenza di soluzioni globali di un problema di ottimizzazione facendo riferimento alla struttura degli insiemi di livello della funzione.

**Proposizione 1.2.3** *Sia  $\mathcal{F} \subseteq R^n$  e sia  $f$  una funzione continua su  $\mathcal{F}$ . Supponiamo che esista un insieme di livello di  $f$  su  $\mathcal{F}$  che sia non vuoto e compatto. Allora esiste un punto di minimo globale di  $f$  in  $\mathcal{F}$ .*

Nel caso generale, stabilire l'esistenza di un insieme di livello compatto può essere difficile. Tuttavia, in molti casi, si possono ottenere delle semplici condizioni per assicurare che *tutti* gli insiemi di livello siano compatti. In particolare la proposizione successiva fornisce una condizione necessaria e sufficiente (nota come condizione di *coercività*) perchè gli insiemi di livello di  $f$  su  $\mathcal{F}$  siano compatti.

**Proposizione 1.2.4** *. Sia  $\mathcal{F} \subseteq R^n$  e sia  $f$  una funzione continua definita su  $\mathcal{F}$ . Allora tutti gli insiemi di livello  $\mathcal{L}(\alpha) = \{x \in \mathcal{F} : f(x) \leq \alpha\}$  di  $f$  su  $\mathcal{F}$  sono compatti se e solo se le due seguenti condizioni sono soddisfatte:*

- (i) *se  $\{x_k\}$  è una sequenza di punti  $x_k \in \mathcal{F}$  tale che  $\lim_{k \rightarrow \infty} \|x_k\| = \infty$  allora segue che*

$$\lim_{k \rightarrow \infty} f(x_k) = \infty;$$

- (ii) *se  $\{x_k\}$  è una sequenza di punti  $x_k \in \mathcal{F}$  tale che  $\lim_{k \rightarrow \infty} x_k = \hat{x} \notin \mathcal{F}$ , allora segue che*

$$\lim_{k \rightarrow \infty} f(x_k) = \infty.$$

Le Proposizioni 1.2.3 and 1.2.4 forniscono delle condizioni sufficienti per l'esistenza delle soluzioni di un problema di minimizzazione in cui l'insieme ammissibile  $\mathcal{F}$  è un insieme aperto. In particolare si possono considerare due casi particolari che corrispondono alle situazioni di maggiore interesse.

Il primo dei due casi è quello in cui  $\mathcal{F} = R^n$ .

**Proposizione 1.2.5** *Sia  $f$  una funzione continua su  $R^n$  e si assuma che  $f$  sia coerciva su  $R^n$ , ossia che*

$$\lim_{k \rightarrow \infty} f(x_k) = \infty,$$

per ogni successione  $\{x_k\}$ , con  $x_k \in R^n$ , tale che  $\lim_{k \rightarrow \infty} \|x_k\| = \infty$ . Allora si ha:

- (i) tutti gli insiemi di livello  $\mathcal{L}(\alpha) = \{x \in R^n : f(x) \leq \alpha\}$  sono compatti;
- (ii) esiste un minimo globale di  $f$  su  $R^n$ .

Il secondo caso è quello in cui  $\mathcal{F}$  è un insieme limitato e aperto tale che  $f$  tende all'infinito al tendere verso la frontiera di  $\mathcal{F}$ . Per questo caso è possibile stabilire la seguente condizione la cui prova è simile a quella della precedente proposizione.

**Proposizione 1.2.6** *Sia  $\mathcal{F}$  un sottoinsieme limitato e aperto di  $R^n$  e sia  $f : \mathcal{F} \rightarrow R$  una funzione continua. Si supponga che*

$$\lim_{k \rightarrow \infty} f(x_k) = \infty,$$

per ogni successione  $\{x_k\}$ , con  $x_k \in \mathcal{F}$ , tale che  $\lim_{k \rightarrow \infty} x_k = \hat{x} \in \partial\mathcal{F}$ . Allora si ha:

- (i) tutti gli insiemi di livello  $\mathcal{L}(\alpha) = \{x \in \mathcal{F} : f(x) \leq \alpha\}$  sono compatti;
- (ii) esiste un minimo globale  $x^* \in \mathcal{F}$  di  $f$  su  $\mathcal{F}$ .

### 1.3 Condizioni di ottimalità locale

Purtroppo tutte le condizioni di ottimalità globali proposte in letteratura hanno un limitato interesse applicativo. Infatti, come sarà visto in seguito, esse sono legate al comportamento globale della funzione obiettivo sull'insieme ammissibile e, perciò, sono descritte necessariamente da condizioni complesse dal punto di vista computazionale.

Più utili dal punto di vista applicativo sono quelle che vengono dette *condizioni di ottimalità locale*. Queste non sono altro che delle condizioni *necessarie* (ma, in generale, non sufficienti) perchè un punto assegnato sia un punto di minimo locale (globale) di un problema di minimizzazione. Quindi, dal punto di vista teorico, non danno una caratterizzazione soddisfacente dei minimi locali (globali) del Problema 1.1 ma giocano un importante ruolo nella definizione di algoritmi di minimizzazione.

### 1.3.1 Condizioni di ottimalità per problemi nonvincolati

Si consideri per primo il caso di un problema di ottimizzazione non vincolata. Le due proposizioni successive descrivono le ben note condizioni necessarie di ottimalità per problemi di ottimizzazione non vincolati.

**Proposizione 1.3.1 (Condizione necessaria del primo ordine)** *Sia  $f$  continuamente differenziabile e sia  $\bar{x} \in R^n$  un punto di minimo locale (globale) del Problema (1.2). Allora*

$$\nabla f(\bar{x}) = 0. \quad (1.5)$$

**Proposizione 1.3.2 (Condizioni necessarie del secondo ordine)** *Sia  $f$  due volte continuamente differenziabile e sia  $\bar{x} \in R^n$  un punto di minimo locale (globale) del Problema (1.2). Allora*

$$\nabla f(\bar{x}) = 0, \quad (1.6)$$

$$d^T \nabla^2 f(\bar{x}) d \geq 0, \quad \text{per ogni } d \in R^n. \quad (1.7)$$

In modo da caratterizzare i punti che soddisfano le precedenti condizioni necessarie del primo ordine si introduce la seguente definizione.

**Definizione 1.3.3** *Sia  $f$  continuamente differenziabile su  $R^n$ , un punto  $\bar{x} \in R^n$  si dice punto di punto stazionario del Problema (1.2) se*

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

Perciò, da quanto detto, si ha che un punto di minimo locale (globale) del Problema (1.2) è necessariamente un punto stazionario del Problema (1.2).

### 1.3.2 Condizioni di ottimalità per problemi vincolati

Passando ai problemi di ottimizzazione vincolata vale la pena ricordare che, se l'insieme ammissibile  $\mathcal{F}$  è un insieme aperto, continuano a valere le precedenti condizioni di ottimalità e la precedente definizione di punto stazionario (basta cambiare  $R^n$  con  $\mathcal{F}$ ). Mentre la situazione è abbastanza più complessa nel caso in cui l'insieme ammissibile  $\mathcal{F}$  non è un insieme aperto. Infatti in questo caso la presenza di un minimo locale (globale) può nascere dall'azione combinata della funzione obiettivo e dei vincoli e questo naturalmente si riflette nelle condizioni di ottimalità.

Associato al Problema 1.3 si può introdurre la seguente definizione.

**Definizione 1.3.4** *Si definisce funzione Lagrangiana associata al Problema (1.3) la funzione  $L : R^n \times R^m \times R^q \rightarrow R$  data da:*

$$L(x, \lambda, \mu) := f(x) + \lambda^T g(x) + \mu^T h(x),$$

con  $\lambda \in R^m$  e  $\mu \in R^q$ .

Le prossime proposizioni descrivono le due condizioni necessarie di ottimalità maggiormente usate nell'ambito di problemi di ottimizzazione vincolati non lineari. Entrambe queste condizioni di ottimalità possono essere stabilite solamente nel caso che l'insieme ammissibile sia "sufficientemente regolare". Un modo per assicurare questa regolarità è quello di richiedere la seguente assunzione.

**Assunzione 1.3.1** *In ogni punto  $x \in \mathcal{F}$  i gradienti  $\nabla g_i(x)$ ,  $i \in \{i' : g_{i'}(x) = 0\}$ , e  $\nabla h_j(x)$ ,  $j = 1, \dots, q$ , sono linearmente indipendenti.*

Utilizzando la precedente assunzione si possono enunciare le seguenti proposizioni.

**Proposizione 1.3.5 (Condizioni di Kuhn-Tucker)** *Siano le funzioni  $f$ ,  $g_i$ ,  $i = 1, \dots, m$ ,  $h_j$ ,  $j = 1, \dots, q$  continuamente differenziabile e si supponga che Assunzione 1.3.1 sia verificata. Se  $\bar{x} \in R^n$  è un punto di minimo locale (globale) del Problema (1.3) allora esistono dei vettori  $\bar{\lambda} \in R^m$  e  $\bar{\mu} \in R^q$  tali che:*

$$\begin{aligned} \nabla_x L(\bar{x}, \bar{\lambda}, \bar{\mu}) &= 0, \\ g(\bar{x}) &\leq 0, \quad h(\bar{x}) = 0, \\ \bar{\lambda}_i &\geq 0, \quad g_i(\bar{x})\bar{\lambda}_i = 0, \quad i = 1, \dots, m. \end{aligned} \tag{1.8}$$

**Proposizione 1.3.6 (Condizioni di Kuhn-Tucker del secondo ordine)** *Siano le funzioni  $f$ ,  $g_i$ ,  $i = 1, \dots, m$ ,  $h_j$ ,  $j = 1, \dots, q$  due volte continuamente differenziabile e si supponga che Assunzione 1.3.1 sia verificata. Se  $\bar{x} \in R^n$  è un punto di minimo locale (globale) del Problema (1.3) allora esistono dei vettori  $\bar{\lambda} \in R^m$  e  $\bar{\mu} \in R^q$  tali che:*

$$\begin{aligned} \nabla_x L(\bar{x}, \bar{\lambda}, \bar{\mu}) &= 0, \\ g(\bar{x}) &\leq 0, \quad h(\bar{x}) = 0, \\ \bar{\lambda}_i &\geq 0, \quad g_i(\bar{x})\bar{\lambda}_i = 0, \quad i = 1, \dots, m. \end{aligned} \tag{1.9}$$

$$\begin{aligned} z^T \nabla^2 L(\bar{x}, \bar{\lambda}, \bar{\mu}) z &\geq 0, \quad \forall z \in R^n, z \neq 0 : \\ \nabla g_i(\bar{x})^T z &= 0, \quad \forall i \in \{i' : g_{i'}(\bar{x}) = 0\}, \\ \nabla h_j(\bar{x})^T z &= 0, \quad \forall j = 1, \dots, q. \end{aligned} \tag{1.10}$$

Analogamente al caso non vincolato si caratterizzano i punti che soddisfano le precedenti condizioni necessarie del primo ordine attraverso la seguente definizione.

**Definizione 1.3.7** *Siano le funzioni  $f$ ,  $g_i$ ,  $i = 1, \dots, m$ ,  $h_j$ ,  $j = 1, \dots, q$  continuamente differenziabile, il punto  $\bar{x} \in R^n$  è detto punto di Kuhn-Tucker del Problema di (1.3) se esistono dei vettori  $\bar{\lambda} \in R^m$  e  $\bar{\mu} \in R^q$ , detti moltiplicatori di Kuhn-Tucker, tali che:*

$$\begin{aligned} \nabla_x L(\bar{x}, \bar{\lambda}, \bar{\mu}) &= 0, \\ g(\bar{x}) &\leq 0, \quad h(\bar{x}) = 0, \\ \bar{\lambda}_i &\geq 0, \quad g_i(\bar{x})\bar{\lambda}_i = 0, \quad i = 1, \dots, m. \end{aligned}$$

Perciò, se l'Assunzione 1.3.1 è verificata, ogni punto di minimo locale (globale) del Problema (1.3) è necessariamente un punto di Kuhn-Tucker del Problema (1.3).

## 1.4 Problemi particolari di Ottimizzazione

In questa sezione vengono descritti alcuni esempi di classi di problemi di ottimizzazione le cui strutture particolari permettono di caratterizzare meglio, dal punto di vista matematico, i loro punti di minimo globale.

### 1.4.1 Problemi di programmazione convessa

Una classe importante dal punto di vista applicativo è quella dei problemi di programmazione convessa. Prima di descrivere questa particolare classe di problemi di minimizzazione, è necessario richiamare le seguenti definizioni.

**Definizione 1.4.1** *Dato un insieme  $C \subseteq R^n$ , si dice che  $C$  è un insieme convesso se comunque scelti due punti  $x, y \in C$  e comunque scelto un scalare  $\alpha \in [0, 1]$  si ha che*

$$\alpha x + (1 - \alpha)y \in C.$$

**Definizione 1.4.2** *Sia  $C \subseteq R^n$  un insieme convesso e sia  $f : C \rightarrow R$ . Si dice che  $f$  è convessa su  $C$  se comunque scelti due punti  $x, y \in C$  e comunque scelto un scalare  $\alpha \in [0, 1]$  si ha che*

$$f(\alpha x + (1 - \alpha)y) \leq \alpha f(x) + (1 - \alpha)f(y);$$

*si dice che  $f$  è strettamente convessa su  $C$  se comunque scelti due punti  $x, y \in C$ , con  $x \neq y$ , e comunque scelto un scalare  $\alpha \in (0, 1)$  si ha che*

$$f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha)f(y).$$

**Definizione 1.4.3** *Sia  $C \subseteq R^n$  un insieme convesso e sia  $f : C \rightarrow R$ . Si dice che  $f$  è concava su  $C$  se comunque scelti due punti  $x, y \in C$  e comunque scelto un scalare  $\alpha \in [0, 1]$  si ha che*

$$f(\alpha x + (1 - \alpha)y) \geq \alpha f(x) + (1 - \alpha)f(y);$$

*si dice che  $f$  è strettamente concava su  $C$  se comunque scelti due punti  $x, y \in C$ , con  $x \neq y$ , e comunque scelto un scalare  $\alpha \in (0, 1)$  si ha che*

$$f(\alpha x + (1 - \alpha)y) > \alpha f(x) + (1 - \alpha)f(y).$$

La seguente proposizione richiama alcune delle proprietà delle funzioni convesse.

**Proposizione 1.4.4** *Sia  $C \subseteq R^n$  un insieme convesso aperto. Se  $f$  è continuamente differenziabile su  $C$  allora:*

(i)  *$f$  è convessa su  $C$  se e solamente se per ogni  $x, y \in C$  si ha:*

$$f(y) - f(x) \geq \nabla f(x)^T (y - x);$$

(ii)  $f$  è strettamente convessa su  $C$  se e solamente se per ogni  $x, y \in C$ , con  $x \neq y$ , si ha:

$$f(y) - f(x) > \nabla f(x)^T(y - x).$$

Se  $f$  è due volte continuamente differenziabile su  $C$  allora:

(iii)  $f$  è convessa su  $C$  se e solamente se per ogni  $x \in C$  si ha:

$$d^T \nabla^2 f(x) d \geq 0 \quad \text{per ogni } d \in \mathbb{R}^n;$$

(iv)  $f$  è strettamente convessa su  $C$  se per ogni  $x \in C$  si ha:

$$d^T \nabla^2 f(x) d > 0 \quad \text{per ogni } d \in \mathbb{R}^n, d \neq 0.$$

Le Proprietà (i) e (ii) sono particolarmente significative, come si vedrà in seguito, per lo studio dei punti di minimo di questa classe particolare di funzioni. Le Proprietà (iii) e (iv) sono delle utili condizioni per identificare la convessità di una funzione.

Analogamente al caso di funzioni convesse si può stabilire la seguente proposizione che fornisce un aiuto a riconoscere le funzioni concave.

**Proposizione 1.4.5** *Sia  $C \subseteq \mathbb{R}^n$  un insieme convesso aperto. Se  $f$  continuamente differenziabile su  $C$  allora:*

(i)  $f$  è concava su  $C$  se e solamente se per ogni  $x, y \in C$  si ha:

$$f(y) - f(x) \leq \nabla f(x)^T(y - x);$$

(ii)  $f$  è strettamente concava su  $C$  se e solamente se per ogni  $x, y \in C$ , con  $x \neq y$ , si ha:

$$f(y) - f(x) < \nabla f(x)^T(y - x).$$

Se  $f$  è due volte continuamente differenziabile su  $C$  allora:

(iii)  $f$  è concava su  $C$  se e solamente se per ogni  $x \in C$  si ha:

$$d^T \nabla^2 f(x) d \leq 0 \quad \text{per ogni } d \in \mathbb{R}^n;$$

(iv)  $f$  è strettamente concava su  $C$  se per ogni  $x \in C$  si ha:

$$d^T \nabla^2 f(x) d < 0 \quad \text{per ogni } d \in \mathbb{R}^n, d \neq 0.$$

A questo punto si può introdurre la classe dei problema di programmazione convessa.

**Definizione 1.4.6** Si definisce problema di programmazione convessa un problema di minimizzazione del tipo:

$$\min_{x \in \mathcal{F}} f(x)$$

in cui  $\mathcal{F}$  è un insieme convesso e  $f$  è una funzione convessa su  $\mathcal{F}$  o, equivalentemente, un problema di massimizzazione del tipo:

$$\max_{x \in \mathcal{F}} f(x)$$

in cui  $\mathcal{F}$  è un insieme convesso e  $f$  è una funzione concava su  $\mathcal{F}$ .

Un esempio di classe di problemi convessi è l'insieme dei problemi di programmazione lineare.

I problemi di programmazione convessa godono di importanti proprietà descritte nei teoremi seguenti. Il primo dei quali mostra che nei problemi convessi non ci possono essere dei minimi locali che non sono minimi globali.

**Proposizione 1.4.7 (Coincidenza tra minimi locali e minimi globali)** Sia  $\mathcal{F} \subseteq \mathbb{R}^n$  un insieme convesso e  $f$  una funzione convessa (strettamente convessa) su  $\mathcal{F}$ . Allora ogni punto (un punto) di minimo locale di  $f$  su  $\mathcal{F}$  è anche (l'unico) punto di minimo globale.

Il secondo risultato riguarda la minimizzazione di una funzione convessa su tutto  $\mathbb{R}^n$  e stabilisce la coincidenza tra punti stazionari e minimi globali.

**Proposizione 1.4.8 (Coincidenza tra punti stazionari e minimi globali)** Sia  $f$  una funzione (strettamente) convessa e continuamente differenziabile su  $\mathbb{R}^n$ . Allora un punto stazionario di  $f$  su  $\mathbb{R}^n$  è un minimo globale (l'unico minimo globale) di  $f$  su  $\mathbb{R}^n$ .

L'ultimo risultato considera la coincidenza tra punti di Kuhn-Tucker e minimi globali.

**Proposizione 1.4.9 (Coincidenza tra punti di Kuhn-Tucker e minimi globali)** Sia dato il seguente problema vincolato

$$\begin{aligned} \min f(x) \\ g_i(x) \leq 0, \quad i = 1, \dots, m \\ a_j^T x - b_j = 0 \quad j = 1, \dots, q. \end{aligned} \tag{1.11}$$

e sia  $f$  una funzione (strettamente) convessa e continuamente differenziabile su  $\mathbb{R}^n$ , siano  $g_i$ ,  $i = 1, \dots, m$ , delle funzioni convesse e continuamente differenziabili su  $\mathbb{R}^n$ . Allora un punto di Kuhn-Tucker del Problema 1.11 è un minimo globale (l'unico minimo globale) del Problema 1.11.



### 1.4.2 Problemi di programmazione concava

Un'altra classe di problemi di minimizzazione particolarmente importante è quella dei problemi di programmazione concava. Infatti questi particolari problemi di ottimizzazione sono in grado di modellizzare numerosi problemi che nascono nel campo dell'economia. Inoltre è possibile dimostrare che, sotto opportune ipotesi, molti problemi di ottimizzazione combinatoria possono essere trasformati in problemi (continui) di programmazione concava.

Si definisce *problema di programmazione concava* un problema di *minimizzazione* del tipo:

$$\min_{x \in \mathcal{F}} f(x)$$

in cui  $\mathcal{F}$  è un *insieme convesso* e  $f$  è una *funzione concava* su  $\mathcal{F}$  o, equivalentemente, un problema di *massimizzazione* del tipo:

$$\max_{x \in \mathcal{F}} f(x)$$

in cui  $\mathcal{F}$  è un *insieme convesso* e  $f$  è una *funzione convessa* su  $\mathcal{F}$ .

È da notare che i problemi di programmazione lineare sono anche problemi di programmazione convessa.

I problemi di programmazione concava sono molto più “difficili” di quelli convessi. La difficoltà principale risiede nel fatto che i problemi concavi presentano normalmente molti punti di minimo locale che non sono punti di minimo globale.

Tuttavia, la particolare struttura della funzione obiettivo di questi problemi fornisce comunque informazioni importanti circa i suoi punti di minimo globale. Infatti la seguente proposizione dimostra che le soluzioni ottime dei problemi di programmazione concava, ove esistano, appartengono alla frontiera dell'insieme ammissibile.

**Teorema 1.4.10 (Assenza di soluzioni ottime interne)** *Sia  $\mathcal{F} \subseteq R^n$  un insieme convesso e sia  $f$  una funzione concava e non costante su  $\mathcal{F}$ . Allora, se esiste un punto di minimo globale di  $f$  su  $\mathcal{F}$ , questo appartiene alla frontiera di  $\mathcal{F}$ .*

Nel caso di minimizzazione di funzioni concave con vincoli lineari, vale un risultato simile al Teorema Fondamentale della Programmazione Lineare. Infatti ripetendo argomenti simili a quelli utilizzati nel Teorema Fondamentale della Programmazione Lineare si ha il seguente teorema.

**Teorema 1.4.11 (Soluzione ottima su un vertice)** *Sia  $\mathcal{F} \subseteq R^n$  un poliedro che ha almeno un vertice e sia  $f$  una funzione concava su  $\mathcal{F}$  che ammetta minimi globali su  $\mathcal{F}$ . Allora, esiste un punto di minimo globale di  $f$  su  $\mathcal{F}$  che coincide con un vertice del poliedro  $\mathcal{F}$ .*

Questo risultato mostra che, come nel caso dei problemi programmazione lineare, la ricerca di un minimo globale di una funzione concava su un simpleso si può ridurre al problema di minimizzare la funzione sull'insieme dei vertici del poliedro  $\mathcal{F}$ . Tuttavia, la nonlinearità della funzione obiettivo non permette di definire un algoritmo, analogo al Metodo del Simpleso per la Programmazione Lineare, che sia in grado di identificare efficientemente i vertici più promettenti trascurando già altri.

## 1.5 Condizioni di ottimalità globale

Nel seguito riportiamo alcuni esempi di tali caratterizzazioni matematiche dei minimi globali. Tuttavia, ogni trattazione teorica riguante i problemi di ottimizzazione globale, può essere effettuata se l'insieme ammissibile  $\mathcal{F}$  è sufficientemente regolare. Per assicurare questa regolarità viene spesso richiesta la seguente assunzione.

**Assunzione 1.5.1** *L'insieme ammissibile  $\mathcal{F} \subset R^n$  è un insieme non vuoto e compatto ed è tale che:*

$$\mathcal{F} = Cl(Int(\mathcal{F})), \quad (1.12)$$

dove  $Cl()$  indica la chiusura di un insieme e  $Int()$  l'interno di un insieme.

Spesso, in letteratura, insiemi che soddisfano all'Assunzione 1.5.1 vengono detti *insiemi robusti*.

Si può notare che la relazione (1.12) implica che un insieme robusto  $\mathcal{F}$  gode della seguente proprietà:

*comunque scelti  $\bar{x} \in \mathcal{F}$  e  $\bar{\varepsilon} > 0$  esiste un punto  $\tilde{x}$  tale che  $\tilde{x} \in B(\bar{x}, \bar{\varepsilon}) \cap Int(\mathcal{F})$ .*

Cioè in ogni intorno di un punto appartenente ad un insieme robusto esiste un punto interno all'insieme stesso. Questo, per esempio, esclude che l'insieme contenga punti isolati.

Nella figura (1.1) è riportato un esempio di un insieme che non soddisfa la relazione (1.12).

### 1.5.1 Condizione di ottimalità necessaria e sufficiente

Le condizioni di ottimalità globale devono essere in grado di identificare i minimi globali distinguendoli dai minimi locali e dagli altri punti stazionari. Perciò devono tener conto del comportamento globale della funzione obiettivo.

Una delle prime e delle più note condizioni di ottimalità globale è la seguente.

**Teorema 1.5.1** *Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1 e sia  $f$  una funzione continua su  $\mathcal{F}$ . Un punto  $x^* \in \mathcal{F}$  è un minimo globale di  $f$  su  $\mathcal{F}$  se e solamente se l'insieme di livello  $\mathcal{L}^* := \{x \in \mathcal{F} : f(x) < f(x^*)\}$  ha misura (di Lebesgue) nulla, cioè*

$$meas(\mathcal{L}^*) = 0.$$

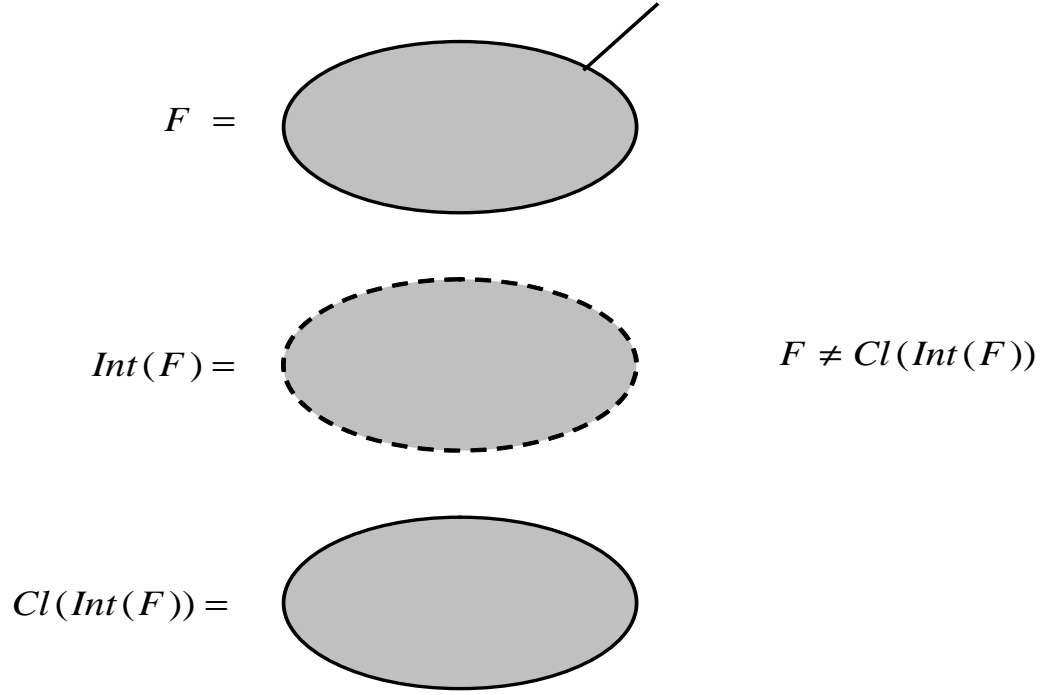


Figura 1.1: Esempio di un insieme che non è robusto.

**Prova.** Si assuma, per assurdo, che  $meas(\mathcal{L}^*) = 0$  e che  $x^*$  non sia un minimo globale di  $f$  su  $\mathcal{F}$ . Se  $x^*$  non fosse un minimo globale esisterebbe un punto  $\bar{x} \in \mathcal{F}$  tale che

$$f(\bar{x}) < f(x^*).$$

Per la continuità della funzione obiettivo seguirebbe che

$$f(x) < f(x^*),$$

per tutti i punti  $x$  appartenenti ad un intorno sufficientemente piccolo di  $\bar{x}$ . Utilizzando l'Assunzione 1.5.1, in questo intorno di  $\bar{x}$  ci sarebbe un punto interno all'insieme ammissibile, quindi esisterebbe un punto  $\tilde{x} \in Int(\mathcal{F})$  tale che:

$$f(\tilde{x}) < f(x^*).$$

Perciò esisterebbe un  $\tilde{\varepsilon} > 0$  tale che l'intorno  $B(\tilde{x}, \tilde{\varepsilon})$  di  $\tilde{x}$  sarebbe strettamente contenuto nell'insieme ammissibile  $\mathcal{F}$  e che, per ogni  $x' \in B(\tilde{x}, \tilde{\varepsilon})$ , si avrebbe:

$$f(x') < f(x^*).$$

Quindi, ricordando la definizione di  $\mathcal{L}^*$ , si otterrebbe

$$B(\tilde{x}, \tilde{\varepsilon}) \subset \mathcal{L}^*,$$

da cui si avrebbe l'assurdo

$$0 = \text{meas}(\mathcal{L}^*) > \text{meas}(B(\tilde{x}, \tilde{\varepsilon})) > 0.$$

Se, invece,  $x^*$  fosse un minimo globale di  $f$  su  $\mathcal{F}$  e si avesse che

$$\text{meas}(\mathcal{L}^*) > 0$$

allora l'insieme  $\mathcal{L}^*$  non sarebbe vuoto. Quindi, esisterebbe un  $\tilde{x} \in \mathcal{L}^*$ , cioè un punto  $\tilde{x} \in \mathcal{F}$  tale che:

$$f(\tilde{x}) < f(x^*)$$

e, quindi,  $x^*$  non sarebbe un minimo globale di  $f$  su  $\mathcal{F}$ . □

Nella figura (1.2) sono riportati due esempi di insiemi  $\mathcal{L}^*$ : il primo esempio si riferisce al caso in cui il punto  $x^*$  non è un minimo globale, il secondo invece descrive il caso in cui il punto  $x^*$  è il minimo globale della funzione.

Il precedente risultato è significativo dal punto di vista teorico. Infatti, descrivendo una condizione necessaria e sufficiente perchè un punto sia un minimo globale, caratterizza perfettamente le soluzioni del Problema 1.1. Tuttavia esso ha un utilizzo diretto limitato dal punto di vista applicativo. Infatti la determinazione della misura di un insieme  $n$  dimensionale ha un costo computazionale proibitivo e, quindi, il precedente teorema non può essere sfruttato computazionalmente.

Tuttavia questo risultato fornisce comunque informazioni utili dal punto di vista geometrico/metodologico. In particolare dai passi della sua dimostrazione si può provare la seguente proposizione che garantisce che l'insieme dei punti ammissibili che migliorano il valore della funzione obiettivo rispetto ad un valore calcolato in un punto non ottimo ha una forma "sufficientemente regolare". Come si vedrà nel seguito, tale risultato è il punto di partenza della maggior parte degli metodi di ottimizzazione globale di tipo probabilistico.

**Proposizione 1.5.2** *Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1 e sia  $f$  una funzione continua su  $\mathcal{F}$ . Allora il punto  $\bar{x} \in \mathcal{F}$  non è un minimo globale di  $f$  su  $\mathcal{F}$  se e solamente se l'insieme*

$$\{x \in \mathcal{F} : f(x) \leq f(\bar{x})\}$$

*contiene una ipersfera di raggio  $\varepsilon > 0$*

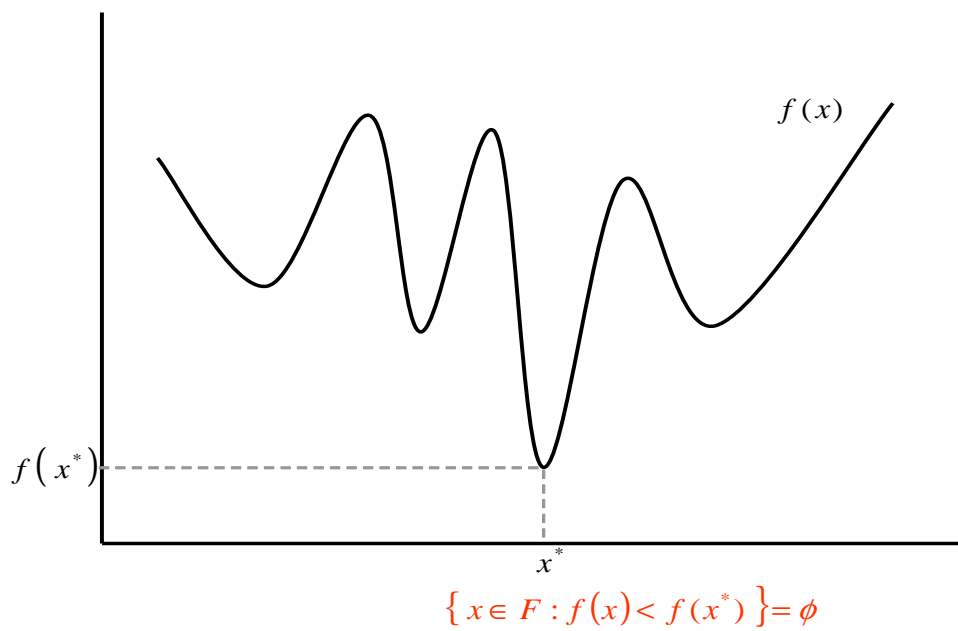
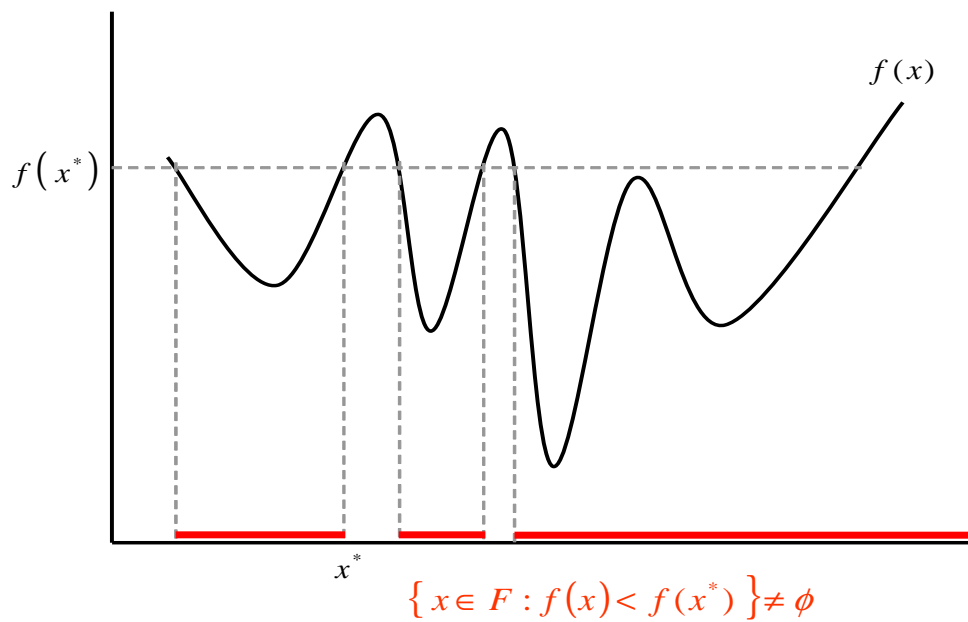


Figura 1.2: Insiemi  $\mathcal{L}^*$  nel caso in cui  $x^*$  non è minimo globale e nel caso che  $x^*$  è minimo globale.

## 1.5.2 Inviluppo convesso di una funzione

Una recente condizione di ottimalità si poggia sulla definizione di inviluppo convesso di una funzione.

**Definizione 1.5.3 (Inviluppo convesso di una funzione)** Sia  $\mathcal{F} \subseteq R^n$  un insieme convesso e sia  $f$  una funzione continua su  $\mathcal{F}$ , si definisce inviluppo convesso di  $f(x)$  su  $\mathcal{F}$  una funzione  $co(f(x))$  tale che:

- i)  $co(f(x))$  è convessa su  $\mathcal{F}$ ;
- ii)  $co(f(x)) \leq f(x)$  per ogni  $x \in \mathcal{F}$ ;
- iii) se  $\tilde{f}(x)$  è una funzione convessa su  $\mathcal{F}$  tale che  $\tilde{f}(x) \leq f(x)$  per ogni  $x \in \mathcal{F}$ , allora si ha che  $\tilde{f}(x) \leq co(f(x))$  per ogni  $x \in \mathcal{F}$ .

Dalla precedente definizione segue che la funzione  $co(f(x))$  è la migliore sottostima convessa della funzione  $f(x)$ .

La funzione  $co(f(x))$  presenta interessanti proprietà alcune di queste descritte nella seguente proposizione.

**Proposizione 1.5.4** Sia  $\mathcal{F} \subset R^n$  un insieme convesso non vuoto e compatto, sia  $f$  una funzione continua su  $\mathcal{F}$ . Allora ogni minimo globale di  $f(x)$  su  $\mathcal{F}$  è anche un minimo globale di  $co(f(x))$  su  $\mathcal{F}$  ed il valore ottimo di  $f(x)$  coincide con quello di  $co(f(x))$ .

**Prova (la prova non fa parte del programma di esame).** Sia  $x^* \in \mathcal{F}$  un minimo globale di  $f(x)$  su  $\mathcal{F}$ . Per la proprietà ii) della Definizione 1.5.3 si ha:

$$co(f(x^*)) \leq f(x^*).$$

Se fosse  $co(f(x^*)) < f(x^*)$  allora la seguente funzione

$$\tilde{f}(x) = \max\{f(x^*), co(f(x))\},$$

sarebbe una funzione convessa (in quanto il massimo di due funzioni convesse) tale che  $\tilde{f}(x) \neq co(f(x))$  e

$$co(f(x)) \leq \tilde{f}(x) \leq f(x), \quad \text{per ogni } x \in \mathcal{F}.$$

Questo, però, contraddirebbe la proprietà iii) della Definizione 1.5.3. Perciò si deve avere necessariamente che:

$$co(f(x^*)) = f(x^*) \tag{1.13}$$

Utilizzando la (1.13), la definizione della funzione  $\tilde{f}(x)$  e la proprietà iii) della Definizione 1.5.3, si ottiene che:

$$co(f(x^*)) = f(x^*) \leq \tilde{f}(x) \leq co(f(x)), \quad \text{per ogni } x \in \mathcal{F},$$

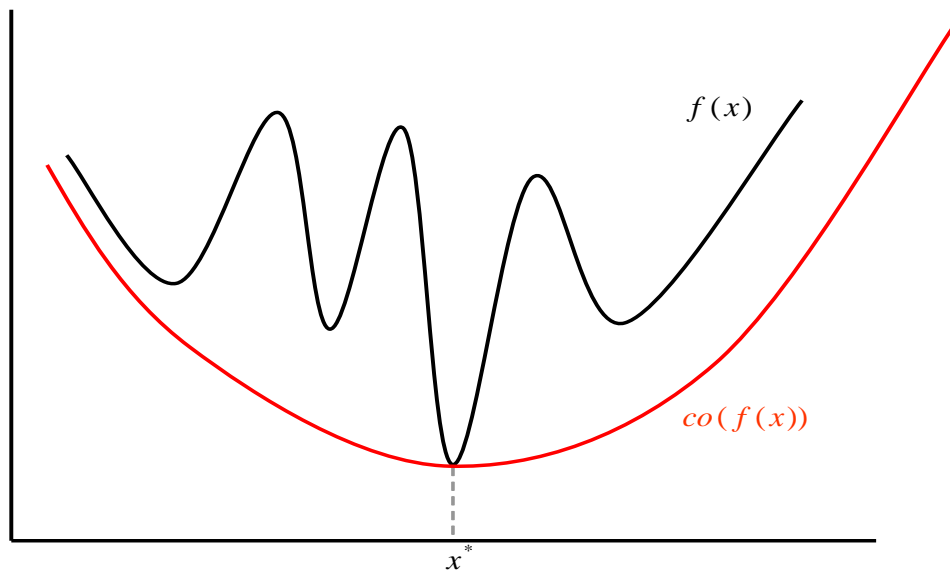


Figura 1.3: Esempio di involucro convesso di una funzione.

che implica che il punto  $x^*$  è anche un minimo globale della funzione  $co(f(x))$  su  $\mathcal{F}$ . Infine la (1.13) assicura anche che i valori ottimi della funzione  $f(x)$  e della funzione  $co(f(x))$  su  $\mathcal{F}$  coincidono.  $\square$

Un esempio di involucro convesso di una funzione è riportato nella figura (1.3).

La funzione  $co(f(x))$  può essere utilizzata per dare una ulteriore caratterizzazione dei minimi globali non vincolati di una funzione continuamente differenziabile.

**Teorema 1.5.5** *Sia  $f$  una funzione continuamente differenziabile su  $R^n$ . Un punto  $x^* \in R^n$  è un minimo globale di  $f$  su  $R^n$  se e solamente se le seguenti condizioni sono verificate:*

- i)  $\nabla f(x^*) = 0$ ;
- ii)  $co(f(x^*)) = f(x^*)$ .

Dai precedenti risultati emerge che la funzione  $co(f(x))$  potrebbe avere un ruolo significativo nella definizione di metodi di ottimizzazione globale. Purtroppo la sua utilizzazione è limitata per il fatto che, per problemi generali, non si ha una sua rappresentazione semplice. Infatti nel caso in cui la funzione  $f(x)$  sia continua su  $R^n$  e soddisfi ipotesi ragionevoli (al più il fatto che la funzione  $f(x)$  sia coercitiva, cioè  $\lim_{\|x\| \rightarrow \infty} f(x) = \infty$ ) si possono dare le seguenti espressioni equivalenti della funzione  $co(f(x))$ :

$$co(f(x)) = \sup \left\{ a^T x - b : a \in R^n, b \in R, a^T \tilde{x} - b \leq f(\tilde{x}) \quad \forall \tilde{x} \in R^n \right\}, \quad (1.14)$$

$$co(f(x)) = \inf \left\{ \sum_{i=1}^{n+1} \alpha_i f(x_i) : x_i \in R^n, \alpha_i \in R_+, \sum_{i=1}^{n+1} \alpha_i = 1, \sum_{i=1}^{n+1} \alpha_i x_i = x \right\} \quad (1.15)$$

$$co(f(x)) = \sup_p \left\{ p^T x - \sup_{\tilde{x}} \left\{ p^T \tilde{x} - f(\tilde{x}) \right\} \right\}. \quad (1.16)$$

La (1.14) mostra che in ogni punto la funzione  $co(f(x))$  è costituita dall'estremo superiore delle funzioni lineari che sottostimano la funzione  $f(x)$ .

La (1.15) mostra, invece, che il valore della funzione  $co(f(x))$  nel punto  $x$  è dato dall'estremo inferiore di particolari combinazioni dei valori della funzione obiettivo calcolati nei punti di vertice di semplici contenenti il punto  $x$ . I coefficienti di tali combinazione sono dati dai pesi che permettono di rappresentare il punto  $x$  in funzione dei vertici del semplice considerato.

Infine la (1.16) mostra che la funzione  $co(f(x))$  coincide con la funzione coniugata della funzione coniugata della  $f(x)$ . Infatti la funzione coniugata di  $f(x)$  è definita da:

$$f^*(p) = \sup_x \left\{ p^T x - f(x) \right\},$$

mentre la funzione coniugata di  $f^*(p)$  è data da:

$$f^{**}(x) = \sup_p \left\{ p^T x - f^*(p) \right\}.$$

Dalla (1.16) segue che

$$co(f(x)) = f^{**}(x).$$

### 1.5.3 Caratterizzazione dei minimi globali unici

Nel caso in cui una funzione  $f$  abbia un solo minimo globale sull'insieme ammissibile  $\mathcal{F}$ , si può caratterizzare ulteriormente tale minimo globale come descritto dal seguente teorema.

**Teorema 1.5.6** *Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1 e sia  $f$  continua su  $\mathcal{F}$ . Se la funzione  $f$  ha un unico minimo globale  $x^*$  su  $\mathcal{F}$  allora per  $i = 1, \dots, n$*

$$\lim_{k \rightarrow \infty} \frac{\int_{\mathcal{F}} x_i e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} = x_i^* \quad (1.17)$$

**Prova.** Per dimostrare il teorema basta provare che, per ogni  $i = 1, \dots, n$ , le sequenze  $\{\phi_k^i\}$  con

$$\phi_k^i = \frac{\int_{\mathcal{F}} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx}, \quad (1.18)$$



sono tali che

$$\lim_{k \rightarrow \infty} \phi_k^i = 0.$$

Cioè si deve dimostrare che comunque scelto un  $\varepsilon > 0$  esiste un indice  $k_\varepsilon$  tale che per tutti gli indici  $k \geq k_\varepsilon$  si ha:

$$\phi_k^i \leq \varepsilon. \quad (1.19)$$

Per qualsiasi fissato  $\varepsilon > 0$ , sia  $B_\varepsilon$  il seguente intorno aperto

$$B_\varepsilon = \{x \in \mathcal{F} : \|x - x^*\| < \frac{\varepsilon}{2}\}.$$

Utilizzando questo intorno, si può riscrivere la (1.18) nella seguente maniera:

$$\phi_k^i = \frac{\int_{B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} + \frac{\int_{\mathcal{F} \setminus B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx}. \quad (1.20)$$

Per quanto riguarda il primo termine della sommatoria si ha, ricordando la definizione di  $B_\varepsilon$ :

$$\frac{\int_{B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} \leq \frac{\varepsilon}{2} \frac{\int_{B_\varepsilon} e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} \leq \frac{\varepsilon}{2}. \quad (1.21)$$

Per quanto riguarda, invece, il secondo termine della somma in (1.20) può essere riscritto nella seguente forma:

$$\frac{\int_{\mathcal{F} \setminus B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} = \frac{\int_{\mathcal{F} \setminus B_\varepsilon} |x_i - x_i^*| e^{-k(f(x) - f(x^*))} dx}{\int_{\mathcal{F}} e^{-k(f(x) - f(x^*))} dx}. \quad (1.22)$$

Poichè, dall'Assunzione 1.5.1, l'insieme  $\mathcal{F}$  è un insieme compatto e poichè  $x^*$  è l'unico minimo globale di  $f$  su  $\mathcal{F}$ , esistono due costanti  $M > 0$  e  $\delta > 0$  tali che, per tutti  $x \in \mathcal{F} \setminus B_\varepsilon$ , si ha:

$$\begin{aligned} \|x - x^*\| &\leq M, \\ f(x) - f(x^*) &\geq \delta, \end{aligned}$$

per cui dalla (1.22) si ha:

$$\frac{\int_{\mathcal{F} \setminus B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} \leq M \frac{\int_{\mathcal{F} \setminus B_\varepsilon} e^{-k\delta} dx}{\int_{\mathcal{F}} e^{-k(f(x) - f(x^*))} dx} \leq M \frac{\text{meas}(\mathcal{F} \setminus B_\varepsilon) e^{-k\delta}}{\int_{\mathcal{F}} e^{-k(f(x) - f(x^*))} dx}. \quad (1.23)$$

Utilizzando la continuità della funzione  $f$  e sfruttando di nuovo il fatto che l'insieme  $\mathcal{F}$  soddisfa l'Assunzione 1.5.1 si ha che esiste un sottoinsieme  $W_\varepsilon$  tale che:

$$\begin{aligned} W_\varepsilon &\subset \mathcal{F}, & \text{meas}(W_\varepsilon) &> 0, \\ f(x) - f(x^*) &\leq \frac{\delta}{2} & \text{per tutti } x &\in W_\varepsilon. \end{aligned}$$

Utizzando questo sottoinsieme e la (1.23) si ottiene:

$$\frac{\int_{\mathcal{F} \setminus B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} \leq M \frac{\text{meas}(\mathcal{F} \setminus B_\varepsilon) e^{-k\delta}}{\int_{W_\varepsilon} e^{-k(\delta/2)} dx} \leq M \frac{\text{meas}(\mathcal{F} \setminus B_\varepsilon) e^{-k(\delta/2)}}{\text{meas}(W_\varepsilon)}, \quad (1.24)$$

da cui segue che esiste un indice  $k_\varepsilon$  tale che, per tutti gli indici  $k \geq k_\varepsilon$ , si ha:

$$\frac{\int_{\mathcal{F} \setminus B_\varepsilon} |x_i - x_i^*| e^{-kf(x)} dx}{\int_{\mathcal{F}} e^{-kf(x)} dx} \leq M \frac{\text{meas}(\mathcal{F} \setminus B_\varepsilon) e^{-k(\delta/2)}}{\text{meas}(W_\varepsilon)} \leq \frac{\varepsilon}{2}. \quad (1.25)$$

Concludendo la (1.20), (1.21) e la (1.25) provano la (1.19) e, quindi, dimostrano la tesi del teorema.  $\square$

## 1.6 Proprietà generali dei metodi di ottimizzazione globale

Come hanno mostrato le condizioni di ottimalità globale riportate nella sezione precedente, i punti di minimo globale di un generico problema di ottimizzazione non sono caratterizzabili matematicamente in modo semplice. Questo fatto rende difficile, da punto di vista teorico, la definizione di algoritmi di ottimizzazione in grado di determinare un punto di minimo globale. Infatti, la complessità delle condizioni di ottimalità globale porta sia all'impossibilità di sfruttare dal punto di vista algoritmico il fatto che un punto prodotto dall'algoritmo non è un minimo globale e sia alla difficoltà di definire dei criteri di arresto efficienti, cioè relativamente semplici ed affidabili.

Per cercare di superare le precedenti difficoltà gli algoritmi proposti in letteratura utilizzano due approcci differenti:

- algoritmi che utilizzano *informazioni globali* sul problema di ottimizzazione;
- algoritmi che utilizzano *informazioni locali* sul problema di ottimizzazione.

Gli algoritmi che utilizzano informazioni globali sono in grado di produrre sequenze di punti che hanno interessanti proprietà di convergenza. Tuttavia essi richiedono o ipotizzano la conoscenza di informazioni "a priori" sul problema da risolvere, per esempio: la struttura particolare del problema come la convessità o la concavità, la costante di Lipschitz della funzione obiettivo e dei vincoli, i limiti superiori sulle derivate prime o seconde delle funzioni che descrivono il problema, valore ottimo della funzione obiettivo, il numero di minimi globali. Purtroppo tali informazioni globali sono difficilmente disponibili nei problemi reali e questo, in qualche maniera, può limitare l'applicabilità di questa classe di metodi.

Gli algoritmi che utilizzano informazioni locali fanno riferimento, invece, a grandezze e quantità del problema di ottimizzazione facilmente ottenibili durante le iterazioni

dell'algoritmo, come, per esempio: i valori della funzione obiettivo e dei vincoli nei vari punti prodotti dall'algoritmo stesso, i valori delle derivate delle funzioni che descrivono il problema (se disponibili). Anche tutte le informazioni ottenibili da formule ed espressioni che utilizzano e combinano le precedenti quantità sono di tipo locale. Esempi di questo genere sono l'uso di formule di interpolazione o di estrapolazione, del punto prodotto dall'algoritmo in cui si è ottenute il più basso valore della funzione obiettivo, della massima pendenza della funzione obiettivo incontrata tra due punti. Non richiedendo informazioni "a priori" sul problema, questa classe di metodi ha una ampia applicabilità. Tuttavia, per cercare di avere una qualche proprietà di convergenza verso minimi globali, questi metodi devono estrarre durante le loro iterazioni delle informazioni sul comportamento globale della funzione obiettivo sull'insieme ammissibile  $\mathcal{F}$ . Questo viene fatto cercando di "campionare" (cioè valutare) la funzione obiettivo in un numero sufficientemente grande di punti appartenenti all'insieme ammissibile  $\mathcal{F}$ . A seconda di come viene effettuato questo campionamento, i metodi appartenenti a questa classe si dividono in:

- *metodi deterministici*, dove i punti in cui viene valutata la funzione obiettivo sono determinati utilizzano le informazioni sulla funzione già ottenute durante le iterazione dell'algoritmo;
- *metodi probabilistici*, dove i punti in cui la funzione viene campionata sono dei vettori aleatori distribuiti uniformemente o secondo leggi che tengono conto delle informazione estratte dall'algoritmo.

In entrambe queste classi di metodi, l'ideale sarebbe quello di eseguire un campionamento della funzione obiettivo che privilegiasse le zone in cui è localizzato un minimo globale e che fosse in grado di assicurare la convergenza dell'algoritmo verso minimi globali.

Nel seguito si descriveranno alcuni risultati che chiariscono il legame tra il modo di campionare la funzione obiettivo e le proprietà di convergenza dell'algoritmo verso i minimi globali.

### 1.6.1 Definizione e funzione di perturbazione

Per introdurre questi risultati è necessario definire formalmente quali sono le caratteristiche di un algoritmo che utilizza informazioni locali. Questo può essere fatto introducendo la seguente definizione.

**Definizione 1.6.1** *Si definisce algoritmo che utilizza informazioni locali un algoritmo che, ad ogni iterazione  $k$ , presenta le seguenti caratteristiche:*

- i) il nuovo punto prodotto  $x_{k+1}$  dipende solamente dalla funzione  $f : \mathcal{F} \rightarrow R$ , a cui è applicato l'algoritmo, e da un certo numero di punti  $\tilde{x}_k^0, \tilde{x}_k^1, \dots, \tilde{x}_k^{p_k}$ , in cui si sono estratte le informazioni sul problema da risolvere, cioè esiste un operatore  $D$  tale che*

$$x_{k+1} = D(f, \tilde{x}_k^0, \tilde{x}_k^1, \dots, \tilde{x}_k^{p_k});$$

ii) se esiste un insieme  $\mathcal{N} \subseteq \mathcal{F}$  e due funzioni  $f : \mathcal{F} \rightarrow R$  e  $\tilde{f} : \mathcal{F} \rightarrow R$  tali che:

$$f(x) = \tilde{f}(x), \quad \text{per ogni } x \in \mathcal{N}$$

e se i punti  $\tilde{x}_k^0, \tilde{x}_k^1, \dots, \tilde{x}_k^{p_k}$  utilizzati dall'algoritmo sono tali che:

$$\tilde{x}_k^i \in \mathcal{N}, \quad \text{per ogni } i = 0, \dots, p_k,$$

allora l'algoritmo produce lo stesso punto  $x_{k+1}$  sia se è applicato alla funzione  $f$  e sia se è applicato alla funzione  $\tilde{f}$ , cioè si ha

$$D(f, \tilde{x}_k^0, \tilde{x}_k^1, \dots, \tilde{x}_k^{p_k}) = D(\tilde{f}, \tilde{x}_k^0, \tilde{x}_k^1, \dots, \tilde{x}_k^{p_k}).$$

Per analizzare in generale le proprietà di convergenza di algoritmi di ottimizzazione globale è necessario richiamare la seguente proposizione.

**Proposizione 1.6.2** *Comunque scelti due scalari  $\varepsilon_1 > 0$  e  $\varepsilon_2 > 0$  con  $\varepsilon_1 > \varepsilon_2$ , si può definire una funzione  $s(\cdot; \varepsilon_1, \varepsilon_2) : R^n \rightarrow R$  infinitamente differenziabile tale che:*

$$\begin{aligned} s(x; \varepsilon_1, \varepsilon_2) &= 1, & \text{se } \|x\| &\leq \varepsilon_2, \\ 0 < s(x; \varepsilon_1, \varepsilon_2) &< 1, & \text{se } \varepsilon_2 < \|x\| < \varepsilon_1, \\ s(x; \varepsilon_1, \varepsilon_2) &= 0, & \text{se } \|x\| &\geq \varepsilon_1. \end{aligned}$$

**Prova.** Un modo per definire una funzione che soddisfa le proprietà descritte dalla proposizione è il seguente.

Per ogni  $\varepsilon_1 > 0$  e  $\varepsilon_2 > 0$ , con  $\varepsilon_1 > \varepsilon_2$ , si possono introdurre le seguenti funzioni infinitamente differenziabili:

$$s_1(x; \varepsilon_1) = \begin{cases} e^{-\frac{1}{\varepsilon_1^2 - \|x\|^2}} & \text{se } \|x\| < \varepsilon_1 \\ 0 & \text{se } \|x\| \geq \varepsilon_1 \end{cases}$$

$$s_2(x; \varepsilon_2) = \begin{cases} e^{-\frac{1}{\|x\|^2 - \varepsilon_2^2}} & \text{se } \|x\| > \varepsilon_2 \\ 0 & \text{se } \|x\| \leq \varepsilon_2. \end{cases}$$

Utilizzando le precedenti funzioni, si può definire la seguente funzione:

$$s(x, \varepsilon_1; \varepsilon_2) = \frac{s_1(x; \varepsilon_1)}{s_1(x; \varepsilon_1) + s_2(x; \varepsilon_2)}. \quad (1.26)$$

È facile verificare che la funzione  $s(x, \varepsilon_1; \varepsilon_2)$ , oltre ad essere infinitamente differenziabile, soddisfa anche le altre richieste della proposizione, infatti:

- se  $\|x\| \geq \varepsilon_1$  allora  $s_1(x; \varepsilon_1) = 0$  e  $s_2(x; \varepsilon_2) > 0$  da cui segue che  $s(x; \varepsilon_1, \varepsilon_2) = 0$ ;
- se  $\varepsilon_2 < \|x\| < \varepsilon_1$  allora  $s_1(x; \varepsilon_1) > 0$  e  $s_2(x; \varepsilon_2) > 0$  da cui segue che  $0 < s(x; \varepsilon_1, \varepsilon_2) < 1$ ;
- se  $\|x\| \leq \varepsilon_2$  allora  $s_1(x; \varepsilon_1) > 0$  e  $s_2(x; \varepsilon_2) = 0$  da cui segue che  $s(x; \varepsilon_1, \varepsilon_2) = 1$ .  $\square$

La precedente proposizione mostra che una qualsiasi funzione può essere perturbata arbitrariamente in un punto  $\tilde{x}$  lasciandola immutata in tutti i punti in cui è definita con l'esclusione di un prefissato intorno di un punto  $\tilde{x}$ . Per esempio, data una funzione  $f : R^n \rightarrow R$ , un punto  $\tilde{x} \in R^n$  ed un scalare  $\alpha \in R$ , la funzione

$$\tilde{f}(x) = f(x) + (\alpha - f(\tilde{x}))s(x - \tilde{x}; \varepsilon, \varepsilon/2)$$

coincide con la funzione  $f(x)$  per ogni  $x \in R^n \setminus \mathcal{B}(\tilde{x}; \varepsilon)$  ed è tale che  $\tilde{f}(\tilde{x}) = \alpha$ . Inoltre la nuova funzione  $\tilde{f}$  ha le stesse proprietà di regolarità della  $f$ , cioè se, per esempio,  $f$  è due volte continuamente differenziabile continua ad esserlo anche la  $\tilde{f}$ .

Da quanto osservato segue facilmente la seguente proposizione.

**Proposizione 1.6.3** *Comunque scelti una funzione  $f : \mathcal{F} \rightarrow R$ , un punto  $\tilde{x} \in \mathcal{F}$  ed uno scalare  $\varepsilon > 0$ , si può definire una nuova funzione  $\tilde{f} : \mathcal{F} \rightarrow R$  tale che:*

- ha le stesse proprietà di continuità di  $f$ ;
- $\tilde{f}(x) = f(x)$  per ogni  $x \in \mathcal{F} \setminus \mathcal{B}(\tilde{x}; \varepsilon)$ ;
- il punto  $\tilde{x} \in \mathcal{F}$  è l'unico minimo globale della funzione  $\tilde{f}$  su  $\mathcal{F}$ .

**Prova.** La dimostrazione segue osservando che un esempio di una funzione  $\tilde{f}$  che gode delle proprietà descritte dalla proposizione è il seguente:

$$\tilde{f}(x) = f(x) + (-\alpha e^{-\|x - \tilde{x}\|^2} - f(x))s(x - \tilde{x}; \varepsilon, \varepsilon/2),$$

dove  $\alpha$  è tale che

$$\alpha > \left| \min_{x \in \mathcal{F}} f(x) \right|.$$

$\square$

A questo punto è possibile analizzare le proprietà di convergenza degli algoritmi di ottimizzazione globale che utilizzano informazioni locali.

## 1.6.2 Proprietà di convergenza degli algoritmi di ottimizzazione globale deterministici

Il primo teorema considera una prima possibile proprietà di convergenza dei punti prodotti dall'algoritmo, cioè il caso in cui un minimo globale del problema possa coincidere con un punto prodotto dall'algoritmo oppure è un punto di accumulazione della sequenza prodotta.

**Teorema 1.6.4** Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1, sia  $\mathcal{C}$  l'insieme delle funzioni continue su  $\mathcal{F}$ . Per ogni funzione  $f \in \mathcal{C}$ , sia  $x^*$  un suo minimo globale su  $\mathcal{F}$  e sia  $\{x_k\}$  la sequenza di punti generata da un algoritmo di ottimizzazione globale che usa informazioni locali (Definizione 1.6.1) quando applicato alla minimizzazione di  $f$  su  $\mathcal{F}$ . Allora, per ogni funzione  $f \in \mathcal{C}$ , esiste un indice  $\bar{k}$  tale che  $x_{\bar{k}} = x^*$  oppure  $x^*$  è un punto di accumulazione della sequenza  $\{x_k\}$  se e solamente se, per ogni funzione  $f \in \mathcal{C}$ , i punti prodotti dall'algoritmo, al tendere di  $k$  all'infinito, formano un insieme denso su  $\mathcal{F}$  (cioè comunque scelti  $x \in \mathcal{F}$  e  $\varepsilon > 0$  esistono un  $\bar{k}$  ed un  $x_k$  tali che  $k \leq \bar{k}$  e  $x_k \in B(x; \varepsilon)$ ).

**Prova.** La sufficienza segue direttamente dalla definizione di insieme denso su  $\mathcal{F}$ . Infatti dalla densità segue che comunque scelto un  $\varepsilon > 0$  esistono un indice  $\bar{k}$  ed un punto  $x_k$  tali che  $k \leq \bar{k}$  e  $x_k \in B(x^*; \varepsilon)$ . Sia  $\{\varepsilon_i\}$  una sequenza tale che

$$\begin{aligned} \varepsilon_i > \varepsilon_{i+1} > 0 & \quad \text{per ogni } i, \\ \lim_{i \rightarrow \infty} \varepsilon_i = 0. \end{aligned}$$

Se si indica con  $k_i$  il più piccolo indice  $k \in \{0, 1, 2, \dots\}$  tale che

$$x_{k_i} \in B(x^*; \varepsilon_i),$$

segue che:

$$\lim_{i \rightarrow \infty} x_{k_i} = x^*.$$

Per dimostrare la necessità si supponga che esista una funzione  $f \in \mathcal{C}$  per cui l'insieme dei punti prodotti dall'algoritmo non formi un insieme denso su  $\mathcal{F}$ . Allora esisterebbe un punto  $\tilde{x} \in \mathcal{F}$  e un  $\tilde{\varepsilon} > 0$  tali che

$$x_k \notin B(\tilde{x}; \tilde{\varepsilon}), \quad \text{per tutti } k. \quad (1.27)$$

Sfruttando la Proposizione 1.6.2 si potrebbe costruire una funzione  $\tilde{f} \in \mathcal{C}$  che coinciderebbe con la funzione  $f$  al di fuori della sfera  $B(\tilde{x}; \tilde{\varepsilon})$  e che avrebbe il suo minimo globale  $\hat{x}$  all'interno della sfera  $B(\tilde{x}; \tilde{\varepsilon})$ .

Ricordando la Definizione 1.6.1, un algoritmo che usa informazioni locali produrrebbe una stessa sequenza di punti sia se applicato alla funzione  $f$  e sia se applicato alla funzione  $\tilde{f}$ . Perciò, tenendo conto della (1.27), non ci potrebbe essere un indice  $\bar{k}$  tale che  $x_{\bar{k}} = \tilde{x}$  nè  $\tilde{x}$  potrebbe essere un punto di accumulazione della sequenza  $\{x_k\}$  prodotta dall'algoritmo.  $\square$

La proprietà di convergenza considerata nel precedente teorema può non essere completamente soddisfacente. Infatti la mancanza di condizioni di ottimalità utilizzabili porta al fatto di non saper riconoscere un minimo globale nel caso coincidesse con un punto prodotto dall'algoritmo e di non saper individuare la sottosequenza che sta convergendo al minimo globale. Più interessante sarebbe stabilire che tutti i punti di accumulazione della sequenza prodotta sono dei minimi globali del problema. Purtroppo dal precedente teorema deriva un risultato negativo circa la possibilità di garantire questo tipo di convergenza, come descritto dalla seguente proposizione.

**Proposizione 1.6.5** *Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1, sia  $\mathcal{C}$  l'insieme delle funzioni continue su  $\mathcal{F}$ . Per ogni funzione  $f \in \mathcal{C}$  che non sia una costante, la sequenza di punti  $\{x_k\}$  generati da un algoritmo di ottimizzazione globale che usa informazioni locali (Definizione 1.6.1) non gode della proprietà che ogni suo punto di accumulazione è un minimo globale di  $f$  su  $\mathcal{F}$ .*

**Prova.** Se ogni suo punto di accumulazione della sequenza di punti  $\{x_k\}$  generati dall'algoritmo fosse un minimo globale di  $f$  su  $\mathcal{F}$  allora il Teorema 1.6.4 implicherebbe che i punti generati dall'algoritmo formerebbero un denso su  $\mathcal{F}$ . Dalla definizione di insieme denso si avrebbe che ogni punto dell'insieme  $\mathcal{F}$  sarebbe un punto di accumulazione della sequenza di punti  $\{x_k\}$  (basta ripetere la prima parte della prova del precedente Teorema facendo riferimento ad un qualsiasi punto  $x \in \mathcal{F}$ ). Si avrebbe, quindi, l'assurdo che ogni punto dell'insieme ammissibile  $\mathcal{F}$  sarebbe un minimo globale del problema.  $\square$

### 1.6.3 Proprietà di convergenza degli algoritmi di ottimizzazione globale probabilistici

Risultati analoghi a quelli ottenuti per algoritmi deterministici, si possono stabilire per gli algoritmi di ottimizzazione globale probabilistici che utilizzano informazioni locali. Infatti si può stabilire il seguente teorema che è l'analogo del Teorema 1.6.4.

**Teorema 1.6.6** *Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1, sia  $\mathcal{C}$  l'insieme delle funzioni continue su  $\mathcal{F}$ . Per ogni funzione  $f \in \mathcal{C}$ , sia  $x^*$  un suo minimo globale su  $\mathcal{F}$  e sia  $\{x_k\}$  la sequenza di punti aleatori generati da un algoritmo di ottimizzazione globale probabilistico che usa informazioni locali (Definizione 1.6.1) quando applicato alla minimizzazione di  $f$  su  $\mathcal{F}$ . Sia  $p \in (0, 1)$ . Allora, comunque scelta una funzione  $f \in \mathcal{C}$ , la probabilità che esiste un indice  $\bar{k}$  tale che  $x_{\bar{k}} = x^*$  oppure che  $x^*$  è un punto di accumulazione della sequenza  $\{x_k\}$  è maggiore di  $p$  se e solamente se, comunque scelta una funzione  $f \in \mathcal{C}$ , la probabilità che un qualsiasi punto di  $\mathcal{F}$  appartenga alla chiusura dei punti generati dall'algoritmo, al tendere di  $k$  all'infinito, è maggiore di  $p$ .*

L'analogo del Teorema 1.6.5 diventa, nel caso di algoritmi probabilistici, il seguente risultato.

**Teorema 1.6.7** *Sia  $\mathcal{F} \subset R^n$  un insieme che soddisfa l'Assunzione 1.5.1, sia  $\mathcal{C}$  l'insieme delle funzioni continue su  $\mathcal{F}$ . Allora comunque scelto un valore  $p \in (0, 1)$ , per ogni funzione  $f \in \mathcal{C}$  che non sia una costante, la probabilità che la sequenza di punti  $\{x_k\}$  generati da un algoritmo di ottimizzazione globale che usa informazioni locali (Definizione 1.6.1) non goda della proprietà che ogni suo punto di accumulazione è un minimo globale di  $f$  su  $\mathcal{F}$  è maggiore di  $p$ .*

Come notato per i Teoremi 1.6.4 e 1.6.5, anche i Teoremi 1.6.6 e 1.6.7 continuano a valere nel caso di funzioni continuamente differenziabili oppure nel caso di funzioni due volte continuamente differenziabili.

## Capitolo 2

# Metodi di ottimizzazione locale

In questo capitolo vengono introdotti e brevemente descritti quelli che vengono detti metodi di ottimizzazione *locale*. Per una trattazione completa, approfondita e rigorosa si rimanda il lettore intessato al libro:

L. Grippo, M. Sciandrone: "Metodi di Ottimizzazione Non Vincolata". Springer-Verlag Italia 2011.

In questo capitolo si seguirà l'approccio proposto nel precedente libro riportando brevemente alcuni dei risultati principali.

Nel seguito si considererà una particolare classe di problemi, detti problemi di ottimizzazione *continuamente differenziabili*, che presentano le seguenti caratteristiche:

- la funzione obiettivo  $f$  è almeno *continuamente differenziabile*;
- nel caso di problemi vincolati, le funzioni  $g_i$ ,  $i = 1, \dots, m$ , e  $h_j$ ,  $j = 1, \dots, p$ , che descrivono l'insieme ammissibile, sono almeno *continuamente differenziabili*

### 2.1 Introduzione

Gli algoritmi proposti per risolvere problemi di ottimizzazione consentono di determinare dei punti  $x^* \in \Omega$ , dove  $\Omega$  è un certo insieme di *punti desiderati*. Gli algoritmi di ottimizzazione sono classificati, oltre per la classe di problemi a cui si applicano, anche per il particolare insieme  $\Omega$  a cui si riferiscono. In particolare se  $\Omega$  è costituito da tutti i minimi globali del Problema 1.1, i corrispondenti algoritmi vengono detti *algoritmi di ottimizzazione globale*. Purtroppo la definizione di algoritmi generali di ottimizzazione globale è uno degli argomenti più difficili nel campo dell'ottimizzazione ed è tuttora un argomento di ricerca aperto.

La maggior parte dei metodi ed algoritmi proposti in letteratura appartengono alla classe chiamata *metodi di ottimizzazione locale* oppure, più semplicemente, *metodi locali*. Questa classe di metodi, partendo da un punto iniziale  $x_0 \in R^n$ , cercano di



produrre una sequenza di punti  $\{x_k\}$  che abbia una "qualche proprietà di convergenza" verso punti di minimo locale del problema. Questi metodi cercano di sfruttare tutte le informazioni locali che possono essere estratte dal problema (per esempio: calcolo delle derivate prime della funzione obiettivo e dei vincoli, calcolo delle derivate seconde, valutazione della funzione obiettivo e dei vincoli in punti vicini al punto corrente) e di utilizzare il fatto che una qualsiasi funzione può essere approssimata localmente abbastanza bene da una funzione lineare o da una funzione quadratica. In realtà, in assenza di particolari proprietà della funzione obiettivo e dei vincoli, le sequenze di punti prodotte da questi metodi non presentano "proprietà di convergenza" verso dei minimi locali del problema, ma piuttosto verso punti che soddisfano delle condizioni necessarie di ottimo locale (viste nella sezione 1.4.2). Questi punti, pur essendo solamente dei candidati ad essere dei minimi locali, vengono accettati in questa classe di algoritmi come *soluzioni locali* del Problema (1.1).

### L'insieme $\Omega$ dei metodi di ottimizzazione locale per problemi non vincolati

Normalmente, nei metodi locali per problemi non vincolati,  $\Omega$  è costituito da punti che soddisfano le *condizioni necessarie di ottimalità del primo ordine* (cioè quelli che sono stati definiti punti stazionari) per il Problema (1.2) ed in cui il valore della funzione obiettivo non è peggiorato.

$$\Omega := \{x \in R^n : f(x) \leq f(x_0), \quad \nabla f(x) = 0\}.$$

In pratica, se il punto iniziale  $x_0$  non è un punto stazionari i metodi locali sono in grado di garantire che  $f(x) < f(x_0)$ .

Recentemente sono stati proposti in letteratura dei metodi che permettono di determinare dei punti che soddisfano le *condizioni necessarie di ottimalità del secondo ordine* (cioè punti che sono dei "migliori" candidati ad essere dei minimi locali del problema di partenza). In questo caso l'insieme  $\Omega$  è dato da:

$$\Omega := \{x \in R^n : f(x) \leq f(x_0), \quad \nabla f(x) = 0, \quad d^T \nabla^2 f(x) d \geq 0, \quad \forall d \in R^n\}.$$

### L'insieme $\Omega$ dei metodi di ottimizzazione locale per problemi vincolati

Analogamente al caso non vincolato, la maggior parte dei metodi locali per problemi vincolati fanno riferimento ad un insieme  $\Omega$  che è costituito da punti che soddisfano le *condizioni necessarie di ottimalità del primo ordine* (cioè quelli che sono stati definiti punti di Kuhn-Tucker) per il Problema (1.3). Inoltre se il punto di partenza appartiene all'insieme ammissibile tali metodi permettono di determinare un punto di Kuhn-Tucker in cui valore della funzione obiettivo non è peggiore di quello di partenza. Formalmente si ha:

$$\Omega := \{x \in R^n : x \text{ è un punto di Kuhn-Tucker e } f(x) \leq f(x_0) \text{ se } x_0 \in \mathcal{F}\}.$$

Anche nel campo dei metodi locali per problemi vincolati sono stati proposti recentemente degli algoritmi che cercano di produrre punti dei punti che soddisfano le *condizioni necessarie di ottimalità del secondo ordine* per il Problema (1.3), in questi casi si

ha:

$$\Omega := \{x \in R^n : x \text{ soddisfa le condizioni necessarie di ottimalità} \\ \text{del secondo ordine e } f(x) \leq f(x_0) \text{ se } x_0 \in \mathcal{F}\}.$$

## 2.2 Convergenza globale

Un algoritmo o metodo di ottimizzazione produce, a partire da una stima iniziale  $x_0$ , una sequenza di punti  $\{x_k\}$  che può essere considerata una sequenza di stime successive di un punto appartenente all'insieme desiderato  $\Omega$ . Dal punto di vista applicativo si è interessati a considerare metodi di ottimizzazione che sono *affidabili* ed *efficienti*. L'affidabilità di un algoritmo viene comunemente associata al concetto di *convergenza globale* dell'algoritmo. Mentre la sua efficienza viene associata al concetto di *rapidità* (*o velocità*) di convergenza della sequenza dei punti prodotta dall'algoritmo.

**Definizione 2.2.1** *Un algoritmo o un metodo di ottimizzazione si dice globalmente convergente se, qualunque sia il punto di partenza  $x_0$ , la sequenza di punti  $\{x_k\}$  che produce soddisfa una delle seguenti proprietà:*

- (i) *esiste un punto  $x_\nu$  tale che  $x_\nu \in \Omega$ ;*
- (ii) *la successione  $\{x_k\}$  converge ad un punto di  $\Omega$ ;*
- (iii) *esiste un punto di accumulazione di  $\{x_k\}$  ed ogni punto di accumulazione di  $\{x_k\}$  appartiene ad  $\Omega$ ;*
- (iv) *esiste un punto di accumulazione di  $\{x_k\}$  che appartiene ad  $\Omega$ .*

Nel caso (i) l'algoritmo *termina* dopo un numero finito di passi determinando un punto di  $\Omega$ . In questo caso si dice che l'algoritmo ha *convergenza finita*. Purtroppo solamente per classi molto particolari di problemi si possono definire algoritmi che presentano questa proprietà (per esempio quelli per minimizzare funzioni quadratiche convesse). Generalmente i metodi proposti hanno, al più, proprietà di tipo asintotico del tipo (ii), (iii) o (iv). Ovviamente, la proprietà (iv) è quella a cui corrisponde la nozione più debole di convergenza tra quelle considerate; la proprietà (iv) tuttavia può essere già sufficiente, dal punto di vista pratico, ad assicurare un comportamento soddisfacente dell'algoritmo. Infatti essa assicura che, dopo aver effettuato un numero sufficientemente grande di iterazioni, si è in grado di ottenere una buona stima di un punto di  $\Omega$ .

È da sottolineare che nella Definizione 1.1 il termine *globale* deriva dal fatto che una delle le proprietà di convergenza (i)-(iv) deve valere *comunque* si fissi il punto iniziale  $x_0$ .

**Definizione 2.2.2** *Se un algoritmo di ottimizzazione produce una sequenza di punti  $\{x_k\}$  che soddisfa una delle proprietà di convergenza (i)-(iv) solamente se il punto iniziale  $x_0$  appartiene ad un intorno opportuno di un punto di  $\Omega$ , l'algoritmo si dice localmente convergente.*

Generalmente, nel caso di algoritmi localmente convergenti, l'intorno a partire dal quale convergono non è conosciuto *a priori* ma se ne conosce solamente l'esistenza.

## 2.3 Rapidità di convergenza

Per quanto riguarda lo studio della caratterizzazione della rapidità di convergenza di un algoritmo di ottimizzazione, esso viene effettuato, per semplicità, *supponendo* che la sequenza  $\{x_k\}$  sia convergente ad un punto  $x^* \in \Omega$ .

Una seconda semplificazione, che comunemente viene fatta, deriva dal fatto che normalmente si può escludere il caso in cui si ha convergenza finita, cioè che esiste un valore finito di  $k$  per cui  $x_k = x^*$ . Infatti algoritmi che convergono in un numero finito di iterazioni si riferiscono a problemi di ottimizzazione molto particolari e la loro efficienza può essere naturalmente misurata dal numero di iterazioni necessarie ad ottenere il punto  $x^*$ . Diversa è la situazione nel caso generale di algoritmi che producono sequenze infinite di punti  $\{x_k\}$ . Una valutazione della efficienza di questi algoritmi implica necessariamente lo studio e la caratterizzazione del comportamento asintotico della sequenze  $\{x_k\}$  prodotta al tendere di  $k$  all'infinito.

In letteratura, la rapidità di convergenza di un algoritmo è stata valutata utilizzando diversi tipi di misure. Il criterio più diffuso è quello di considerare l'andamento della norma della differenza tra il punto  $x_k$ , ottenuto all'iterazione  $k$ -esima, ed il punto  $x^*$ , cioè l'errore commesso  $e_k$  dato da:

$$e_k = \|x_k - x^*\|.$$

La rapidità di convergenza di un algoritmo può quindi essere definita caratterizzando la velocità con cui  $e_k$  tende a zero e questo, generalmente, è stato fatto confrontando le norme dei  $e_{k+1}$  e  $e_k$ .

**Definizione 2.3.1** Sia  $x_k \rightarrow x^*$ . Se esistono  $\bar{k} \geq 0$ ,  $p > 0$  e  $\alpha \geq 0$  tali che per  $k \geq \bar{k}$  si ha:

$$\|x_{k+1} - x^*\| \leq \alpha \|x_k - x^*\|^p$$

allora si dice che la  $\{x_k\}$  ha rapidità di convergenza di ordine  $p$ .

In pratica i casi più interessanti sono  $p = 1$  e  $p = 2$ , in quanto la maggior parte degli algoritmi proposti in letteratura presentano rapidità di convergenza che ricadono in questi due casi.

- Se  $p = 2$  nella Definizione 2.3.1, allora si dice che la sequenza  $\{x_k\}$  ha rapidità di convergenza *quadratica*.
- Se  $p = 1$  nella Definizione 2.3.1, allora si dice che la sequenza  $\{x_k\}$  ha rapidità di convergenza
  - *lineare* se  $\alpha \in [0, 1)$ ,

- *sublineare* se  $\alpha \in [1, \infty)$ .

Tra la rapidità di convergenza quadratica e quella lineare si può introdurre un livello intermedio utilizzando la seguente definizione

**Definizione 2.3.2** *Sia  $x_k \rightarrow x^*$ . Se esistono un  $\bar{k} \geq 0$  ed una sequenza di scalari positivi  $\{\alpha_k\}$ , con  $\alpha_k \rightarrow 0$ , tali che per  $k \geq \bar{k}$  si ha:*

$$\|x_{k+1} - x^*\| \leq \alpha_k \|x_k - x^*\|$$

*allora si dice che la sequenza  $\{x_k\}$  ha rapidità di convergenza superlineare.*

## 2.4 Metodi di ottimizzazione non vincolata

In questa sezione si descriveranno molto brevemente alcune delle classi più importanti di algoritmi per affrontare un problema di minimizzazione non vincolata, ovvero di un problema del tipo:

$$\begin{aligned} \min f(x) \\ x \in \mathbb{R}^n, \end{aligned} \tag{2.1}$$

dove  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  è una funzione continuamente differenziabile.

### 2.4.1 Introduzione

Il punto di partenza degli algoritmi di ottimizzazione locale sono le condizioni di ottimalità descritte da Proposizione 1.3.1 e da Proposizione 1.3.2. In particolare le loro dimostrazioni ci indicano chiaramente una via (algoritmica) per migliorare la funzione obiettivo in ogni punto che non soddisfa una delle due condizioni descritte dalle due proposizioni.

Tutto questo può essere formalizzato introducendo la seguente definizione:

**Definizione 2.4.1** *Si definisce insieme delle direzioni di discesa della funzione  $f$  nel punto  $\bar{x}$  il seguente insieme  $D(\bar{x})$ :*

$$D(\bar{x}) = \{d \in \mathbb{R}^n : \exists \delta > 0 \text{ per cui } f(\bar{x} + \alpha d) < f(\bar{x}), \forall \alpha \in (0, \delta) \}$$

Per quanto riguarda il caso di punti che non soddisfano la condizione di ottimalità descritta dalla Proposizione 1.3.1 si può stabilire il seguente risultato.

**Proposizione 2.4.2** Sia  $f \in C^1(\mathbb{R}^n)$ . Se il punto  $\bar{x} \in \mathbb{R}^n$  non è un punto stazionario del Problema (2.1) allora il seguente insieme di direzioni  $D_s(\bar{x})$  nel punto  $\bar{x}$

$$D_s(\bar{x}) = \{d \in \mathbb{R}^n : \nabla f(\bar{x})^T d < 0\} \quad (2.2)$$

non è vuoto e si ha:

$$D_s(\bar{x}) \subseteq D(\bar{x}) \quad (2.3)$$

**Prova.** Il Teorema della Media implica

$$f(\bar{x} + \alpha d) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T d + r(\alpha), \quad \alpha > 0$$

con

$$\lim_{\alpha \rightarrow 0} \frac{r(\alpha)}{\alpha} = 0.$$

Ricordando che  $\nabla f(\bar{x})^T d < 0$ , si ha:

$$f(\bar{x} + \alpha d) = f(\bar{x}) - \alpha |\nabla f(\bar{x})^T d| + r(\alpha) = f(\bar{x}) - \alpha \left( |\nabla f(\bar{x})^T d| - \frac{r(\alpha)}{\alpha} \right).$$

Da cui segue l'esistenza di uno scalare  $\delta > 0$  tale

$$f(\bar{x} + \alpha d) < f(\bar{x}), \quad \forall \alpha \in (0, \delta),$$

Quindi si ottiene che  $d \in D(\bar{x})$ . □

Un risultato analogo si può ottenere considerando punti che non soddisfano le condizioni della Proposizione 1.3.2.

Questi risultati costituiscono le basi per definire dei metodi di ottimizzazione non vincolata in quanto forniscono l'indicazione che, in un punto che non soddisfa le condizioni di ottimalità, si possono determinare delle particolari direzioni lungo le quali la funzione obiettivo diminuisce.

Le precedenti considerazioni motivano il fatto che una struttura generale di un algoritmo di ottimizzazione non vincolata possa essere la seguente:

**Schema generale di un algoritmo.**

*Passo 1:* Si sceglie un punto iniziale  $x_0 \in \mathbb{R}^n$  e si pone  $k = 0$ .

*Passo 2:* Se  $x_k$  è un punto stazionario stop.

*Passo 3:* Si determina un vettore  $d_k \in \mathbb{R}^n$ , detto *direzione di ricerca*.

*Passo 4:* Si determina uno scalare  $\alpha_k \in \mathbb{R}$ , detto *passo (o spostamento)* lungo  $d_k$ .

*Passo 5:* Si produce un nuovo punto

$$x_{k+1} = x_k + \alpha_k d_k,$$

si pone  $k = k + 1$  e si ritorna al Passo 2.

Una tale struttura evidenzia che l'idea di fondo di un metodo di ottimizzazione non vincolata è quella di cercare di trovare un'approssimazione del minimo della funzione obiettivo attraverso una sequenza di minimizzazioni "più semplici". Infatti al Passo 3 la direzione  $d_k$  viene usualmente calcolata facendo riferimento alla minimizzazione di una funzione  $\eta(d)$  che rappresenta una *approssimazione* della funzione  $f(x_k + d)$  pensata come funzione della sola variabile  $d \in R^n$ . Ovviamente  $\eta(d)$  viene scelta in maniera tale da poter determinare il suo minimo in maniera semplice. Una volta determinata la direzione  $d_k$ , nel Passo 4 si calcola lo spostamento  $\alpha_k$  in maniera tale che sia una *approssimazione* di un minimo della funzione  $\phi(\alpha) := f(x_k + \alpha d_k)$  che dipende dalla sola variabile scalare  $\alpha \in R$ .

*Convergenza di un algoritmo di ottimizzazione non vincolata.*

Il primo problema da affrontare, quando si vuole dimostrare la convergenza globale di un algoritmo, è quello di assicurare che la sequenza di punti  $\{x_k\}$  prodotta abbia *almeno un punto di accumulazione*. A questo fine, normalmente, si *assume* di conoscere un punto  $x_0 \in R^n$  tale che *l'insieme di livello*

$$\mathcal{L}_0 := \{x \in R^n : f(x) \leq f(x_0)\}$$

sia compatto.

Poiché la minima richiesta rivolta ad un algoritmo di minimizzazione è quella di migliorare la stima iniziale  $x_0$ , si ha che tutti i punti prodotti  $x_k$  devono necessariamente soddisfare la condizione  $f(x_k) \leq f(x_0)$ . Quindi segue che tutti i punti della successione rimangono nell'insieme compatto  $\mathcal{L}_0$  e questo implica l'esistenza di almeno un punto di accumulazione.

Una volta considerato il problema dell'esistenza di punti di accumulazione, il passo successivo per stabilire la convergenza globale dell'algoritmo è quello di stabilire se la sequenza di punti  $\{x_k\}$  prodotta soddisfa una delle proprietà (i)-(iv) della Definizione 2.2.1

Molti risultati generali sono stati proposti in letteratura per caratterizzare la convergenza dei metodi di ottimizzazione non vincolata. Qui se ne riporta uno che, nella sua semplicità, mette comunque in evidenza che la convergenza globale di un algoritmo può essere assicurata richiedendo ipotesi riconducibili a condizioni da imporre sulla scelta della direzione di ricerca  $d_k$  e del passo  $\alpha_k$ .

**Proposizione 2.4.3** *Sia  $\{x_k\}$  la successione prodotta dall'algoritmo, si supponga che  $\mathcal{L}_0$  sia compatto.*

*Siano verificate le seguenti condizioni:*

- i)  $f(x_{k+1}) \leq f(x_k)$  per ogni  $k$ ;*
- ii) se  $\nabla f(x_k) \neq 0$  per ogni  $k$ , si ha:*

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0; \tag{2.4}$$

iii) esistono due costanti  $c > 0$  e  $q \geq 0$  tali che, per  $d_k \neq 0$  si ha:

$$\frac{\nabla f(x_k)^T d_k}{\|d_k\|} \leq -c \|\nabla f(x_k)\|^q. \quad (2.5)$$

Allora, o esiste un indice  $\nu \geq 0$  tale che  $x_\nu \in \mathcal{L}_0$  e  $\nabla f(x_\nu) = 0$ , oppure viene prodotta una successione infinita tale che:

- (a)  $\{x_k\}$  rimane in  $\mathcal{L}_0$  ed ammette punti di accumulazione;
- (b) la successione  $\{f(x_k)\}$  converge;
- (c)  $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$ , cioè ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  è tale che  $\nabla f(\bar{x}) = 0$ .

Come detto in precedenza le ipotesi (i)-(iii) della Proposizione 2.4.3 possono essere riconducibili a condizioni sulla direzione  $d_k$  e sul passo  $\alpha_k$ .

- L'ipotesi i) può essere soddisfatta scegliendo valori per  $\alpha_k$  sufficientemente piccoli ed una direzione  $d_k$  tale che soddisfa la seguente condizione:

$$\nabla f(x_k)^T d_k < 0$$

che garantisce che, per valori abbastanza piccoli di  $\alpha$ , si abbia  $f(x_k + \alpha_k d_k) < f(x_k)$ .

- L'ipotesi ii) può essere soddisfatta utilizzando delle semplici tecniche di *ricerca unidimensionale* per il calcolo di  $\alpha_k$ .
- L'ipotesi iii) pone delle condizioni sulla scelta della direzione  $d_k$ . Infatti, ricordando che il coseno dell'angolo  $\theta_k$  tra il gradiente  $\nabla f(x_k)$  e la direzione  $d_k$  è definito da:

$$\cos \theta_k = \frac{\nabla f(x_k)^T d_k}{\|d_k\| \|\nabla f(x_k)\|}.$$

l'ipotesi iii) implica che  $\cos \theta_k < 0$  che, geometricamente, equivale a richiedere che la direzione  $d_k$  formi sempre un angolo acuto con la direzione dell'antigradiente  $-\nabla f(x_k)$ .

La Proposizione 2.4.3 può essere utilizzata per assicurare la convergenza globale di molte classi di metodi di ottimizzazione non vincolata. Infatti le richieste sulla scelta del passo  $\alpha_k$  possono essere soddisfatte molto facilmente (come si vedrà nella prossima sezione) ed inoltre non è difficile trovare delle direzioni che soddisfano l'ipotesi iii).

L'esempio più immediato è quello di considerare delle direzioni del tipo

$$d_k = -H_k \nabla f(x_k)$$

dove  $H_k$  è una matrice definita positiva, con massimo autovalore  $\lambda_M(H_k)$  e minimo autovalore  $\lambda_m(H_k)$  tali che, per ogni  $k$ , verificano  $M \geq \lambda_M(H_k) \geq \lambda_m(H_k) \geq m > 0$ , con  $M \geq m > 0$ .

In tal caso si ha che la iii) della Proposizione 2.4.3 è verificata con  $q = 1$  e  $c = m/M$ . Infatti usando la seguente relazione

$$\|d_k\| = \|H_k \nabla f(x_k)\| \leq \lambda_M(H_k) \|\nabla f(x_k)\| \leq M \|\nabla f(x_k)\|$$

si ha:

$$\nabla f(x_k)^T d_k = -\nabla f(x_k)^T H_k \nabla f(x_k) \leq -\lambda_m(H_k) \|\nabla f(x_k)\|^2 \leq -\frac{m}{M} \|\nabla f(x_k)\| \|d_k\|$$

Un altro modo (molto diffuso nei metodi tipo-Newton) per assicurare la condizione (iii) della Proposizione 2.4.3 è quella di mostrare che, per ogni  $k$ , esistono numeri  $c_1 > 0$ ,  $c_2 > 0$ ,  $q_1 \geq 3$  e  $q_2 \geq 2$  tali che la direzione  $d_k$  verifica le seguenti condizioni:

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^{q_1}, \quad \|d_k\|^{q_2} \leq c_2 \|\nabla f(x_k)\|. \quad (2.6)$$

Tuttavia, benché le sue condizioni sembrano essere alquanto generali, la Proposizione 2.4.3 si è rilevata inadeguata a stabilire la convergenza globale di classi importanti di algoritmi di minimizzazione non vincolata come, ad esempio metodi del tipo del gradiente coniugato o i metodi Quasi Newton. Questo è dovuto al fatto che, in questi casi, risulta difficile stabilire la validità della condizione iii) di Proposizione 2.4.3. Infatti in tutti gli algoritmi in cui la direzione  $d_k$  viene scelta in base ad informazioni relative alle iterazioni precedenti, non sembra possibile cercare di soddisfare una condizione, come la iii) di Proposizione 2.4.3, che si basa su grandezze e variabili relative esclusivamente alla iterazione corrente.

È nata quindi l'esigenza di dare dei criteri di convergenza globale che richiedano condizioni più deboli sulla direzione di ricerca  $d_k$ . Questo può essere fatto a prezzo, però, di richiedere condizioni più forti sulla scelta del passo  $\alpha_k$  come mostra la seguente proposizione.

**Proposizione 2.4.4** *Sia  $\{x_k\}$  la successione prodotta dall'algoritmo e si supponga che  $\mathcal{L}_0$  sia compatto e che  $d_k \neq 0$  per  $\nabla f(x_k) \neq 0$ . Supponiamo che valgano le condizioni:*

- i)  $f(x_{k+1}) \leq f(x_k)$  per ogni  $k$ ;
- ii) se  $\nabla f(x_k) \neq 0$  per ogni  $k$ , si ha:

$$\sum_{k=0}^{\infty} \left( \frac{\nabla f(x_k)^T d_k}{\|d_k\|} \right)^2 < \infty; \quad (2.7)$$

- iii) se  $\nabla f(x_k) \neq 0$  per ogni  $k$  allora si ha:

$$\sum_{k=0}^{\infty} \left( \frac{\nabla f(x_k)^T d_k}{\|d_k\| \|\nabla f(x_k)\|} \right)^2 = \infty. \quad (2.8)$$



Allora, o esiste un indice  $\nu \geq 0$  tale che  $x_\nu \in \mathcal{L}_0$  e  $\nabla f(x_\nu) = 0$ , oppure viene prodotta una successione infinita tale che:

- (a)  $\{x_k\}$  rimane in  $\mathcal{L}_0$  ed ammette punti di accumulazione;
- (b) la successione  $\{f(x_k)\}$  converge;
- (c)  $\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$ , cioè esiste un punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  tale che  $\nabla f(\bar{x}) = 0$ .

È da notare, innanzitutto, che il risultato di convergenza ottenuto è più debole rispetto al precedente: infatti, mentre nella Proposizione 2.4.3 si riusciva a stabilire che tutti i punti di accumulazione della sequenza  $\{x_k\}$  sono dei punti stazionari, nella Proposizione 2.4.4 si riesce a dimostrare solamente che un punto di accumulazione di  $\{x_k\}$  è un punto stazionario.

Come detto precedentemente la condizione iii) di Proposizione 2.4.3 è più stringente della iii) di Proposizione 2.4.4. Inversamente, la ii) di Proposizione 2.4.4 implica la ii) di Proposizione 2.4.3.

## 2.4.2 Algoritmi di ricerca unidimensionale

Gli *algoritmi di ricerca unidimensionale* sono delle tecniche che permettono di calcolare il passo  $\alpha_k$  da effettuare lungo una assegnata direzione. Come visto nella precedente sezione, si può dire che l'obiettivo di queste tecniche è quello di effettuare una minimizzazione della funzione

$$\phi(\alpha) := f(x_k + \alpha d_k),$$

Tuttavia, benché  $\phi$  è funzione della sola variabile  $\alpha$ , una sua minimizzazione è, in generale, molto dispendiosa specialmente in problemi in cui  $f$  non è convessa. D'altra parte, l'esperienza computazionale mostra che non esistono particolari vantaggi nel cercare di determinare  $\alpha_k$  come una buona stima del minimo di  $\phi(\alpha)$ .

Tenendo conto delle precedenti osservazioni, un metodo di ricerca unidimensionale deve cercare di determinare in maniera semplice un  $\alpha_k$  che, pur essendo una stima molto approssimata di un minimo di  $\phi$ , garantisca comunque all'algoritmo di minimizzazione opportune proprietà di convergenza. In pratica, un algoritmo di ricerca unidimensionale può essere definito utilizzando determinati *criteri di accettabilità*. Tali criteri, essendo molto semplici, possono essere soddisfatti, in generale, dopo un numero molto limitato di tentativi. Tuttavia i valore che vengono accettati per  $\alpha_k$  sono tali da garantire che sono soddisfatte le ipotesi che servono a stabilire la convergenza globale dell'algoritmo di ottimizzazione (per esempio le ipotesi i) e ii) della Proposizione 2.4.3 oppure le ipotesi i) e ii) della Proposizione 2.4.4).

### Criterio di Armijo, condizioni di Goldstein

Uno dei modi più semplici per assicurare il soddisfacimento delle ipotesi i) e ii) della Proposizione 2.4.3 è quello di imporre che il valore  $\alpha_k$  soddisfi il seguente criterio di Armijo.

*Criterio di Armijo*

- Dati  $a > 0$ ,  $\delta \in (0, 1)$ ,  $\gamma \in (0, 1/2)$ .
- Si sceglie  $\alpha_k = \delta^h a$  dove  $h$  è il più piccolo intero non negativo tale che:

$$f(x_k + \delta^h a d_k) \leq f(x_k) + \gamma \delta^h a \nabla f(x_k)^T d_k. \quad (2.9)$$

**Osservazione.** Dal punto di vista geometrico, la precedente condizione impone di scegliere  $\alpha_k$  come il più grande valore di  $\alpha$  nell'insieme:

$$A = \{\alpha : \alpha = \delta^j a, j = 0, 1, \dots\}$$

per cui il valore della funzione  $\phi(\alpha_k)$  sia al di sotto del valore della retta passante per  $(0, \phi(0))$  e avente pendenza  $\gamma \phi'(0) = \gamma \nabla f(x_k)^T d_k$ .

**Osservazione.** Un differente modo di interpretare il metodo di Armijo è quello di considerarlo come un criterio che sceglie il passo  $\alpha_k$  valutando l'affidabilità del modello locale lineare della funzione. Infatti se si introduce il modello lineare

$$\psi_k(\alpha) = f(x_k) + \alpha \nabla f(x_k)^T d_k$$

la condizione (2.9) può essere riscritta

$$\begin{aligned} f(x_k + \delta^h a d_k) - f(x_k) &\leq \gamma (f(x_k) + \delta^h a \nabla f(x_k)^T d_k - f(x_k)) \\ f(x_k + \delta^h a d_k) - f(x_k) &\leq \gamma (\psi_k(\delta^h a) - \psi_k(0)) \\ \frac{f(x_k + \delta^h a d_k) - f(x_k)}{\psi_k(\delta^h a) - \psi_k(0)} &\geq \gamma. \end{aligned}$$

Quindi il criterio di Armijo ritiene accettabile un passo  $\alpha = \delta^h a$  se il rapporto tra la variazione della funzione e la variazione della sua approssimazione lineare supera la soglia di  $\gamma$ .

Teoricamente e computazionalmente questo controllo è molto significativo perchè (come si vedrà successivamente) le direzioni  $d_k$  usate vengono determinate sfruttando i modelli locali (lineari, quadratici) della funzione obiettivo. In sintesi, la condizione (2.9) controlla se, in corrispondenza di uno spostamento  $\alpha = \delta^h a$ , il modello lineare è una approssimazione sufficientemente buona della funzione data. In caso contrario lo spostamento  $\alpha = \delta^h a$  sarebbe ad una distanza in cui anche la direzione  $d_k$  scelta potrebbe essere non significativa.

Il seguente algoritmo permette di determinare un passo  $\alpha_k$  che soddisfa il Criterio di Armijo

**Algoritmo di Armijo.**

*Passo 0.* Dati  $x_k, d_k, \nabla f(x_k) \in \mathbb{R}^n$ ,  $\delta \in (0, 1)$ ,  $\gamma \in (0, 1)$ ,  $\alpha_0 = a > 0$ , si pone  $i = 0$ .

*Passo 1.* Si calcola  $f(x_k + \alpha_i d_k)$ .

*Passo 2.* Se  $f(x_k + \alpha_i d_k) > f(x_k) + \alpha_i \gamma \nabla f(x_k)^T d_k$

si pone  $\alpha_{i+1} = \delta \alpha_i$ ,  $i = i + 1$  e si va al Passo 1.

*Passo 3.* Si pone  $\alpha_k = \alpha_i$  e Stop.

Le proprietà dell'Algoritmo di Armijo sono descritte nella proposizione seguente.

**Proposizione 2.4.5** *Supponiamo che l'insieme di livello  $\mathcal{L}_0$  sia compatto e supponiamo che valgano le condizioni:*

*i)  $\nabla f(x_k)^T d_k < 0$  per ogni  $k$ ,*

*ii) esistono delle costanti  $c > 0$  e  $q > 0$  per cui si ha per ogni  $k$ :*

$$\|d_k\| \geq c \left( \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right)^q.$$

*Allora l'algoritmo termina dopo un numero finito di passi producendo un valore  $\alpha_k > 0$  che soddisfa il criterio di Armijo e la successione definita da  $x_{k+1} = x_k + \alpha_k d_k$  soddisfa le condizioni:*

*(a)  $f(x_{k+1}) < f(x_k)$ ;*

*(b)  $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0$ .*

**Prova.** Prima di tutto si dimostra che l'algoritmo termina dopo un numero finito di passi producendo un  $\alpha_k = \delta^h a$  che soddisfa il criterio di Armijo.

Se la procedura cicla tra il Passo 1 ed il Passo 2 esisterebbe una sequenza di scalari  $\{\alpha_i\}$  tale che

$$\lim_{i \rightarrow \infty} \alpha_i = 0,$$

$$\frac{f(x_k + \alpha_i d_k) - f(x_k)}{\alpha_i} \geq \gamma \nabla f(x_k)^T d_k,$$

Utilizzando il Teorema della media

$$f(x_k + \alpha_i d_k) = f(x_k) + \alpha_i \nabla f(x_k + \theta_i \alpha_i d_k)^T d_k,$$

con  $\theta_i \in (0, 1)$ , si otterrebbe:

$$\nabla f(x_k + \theta_i \alpha_i d_k)^T d_k \geq \gamma \nabla f(x_k)^T d_k.$$

che, facendo tendere  $i \rightarrow \infty$  e ricordando l'assunzione i), si arriverebbe all'assurdo che entrambe le seguenti relazioni sarebbero vere:

$$\begin{aligned} (1 - \gamma) \nabla f(x_k)^T d_k &\geq 0, \\ \nabla f(x_k)^T d_k &< 0. \end{aligned}$$

Ora si può passare alla dimostrazione dei punti (a) e (b).

Dalla condizione (2.9) su  $\alpha_k$  e dal fatto che  $\nabla f(x_k)^T d_k < 0$  si ha che:

$$f(x_{k+1}) \leq f(x_k) - \alpha_k \gamma |\nabla f(x_k)^T d_k|, \quad (2.10)$$

da cui segue che il punto (a).

Dalla (2.10) si ha anche che

$$x_k \in \mathcal{L}_0, \quad \text{per ogni } k, \quad (2.11)$$

da cui, ricordando l'ipotesi di compattezza dell'insieme  $\mathcal{L}_0$ , si ha che la sequenza  $\{x_k\}$  è limitata e che ammette almeno un punto di accumulazione.

La (2.10) e la compattezza dell'insieme  $\mathcal{L}_0$  implicano anche che la sequenza di scalari  $\{f(x_k)\}$  è non crescente e limitata inferiormente. Perciò la sequenza  $\{f(x_k)\}$  ammette un limite  $\bar{f}$ , cioè:

$$\lim_{k \rightarrow \infty} f(x_k) = \bar{f}. \quad (2.12)$$

Dal precedente limite e dalla (2.10) si ottiene:

$$\lim_{k \rightarrow \infty} \alpha_k \gamma |\nabla f(x_k)^T d_k| \leq \lim_{k \rightarrow \infty} (f(x_k) - f(x_{k+1})) = 0,$$

da cui segue che:

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} = 0. \quad (2.13)$$

Se, per assurdo, il punto b) della proposizione non fosse vero e ricordando l'assunzione ii), il limite (2.13) implicherebbe l'esistenza di un insieme infinito di indici  $K$  e di sottosequenze  $\{x_k\}_K$  e  $\{\alpha_k\}_K$  tali che:

$$\begin{aligned} \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} &\geq \varepsilon > 0, & \text{per ogni } k \in K, \\ \|d_k\| &\geq c\varepsilon^q > 0, & \text{per ogni } k \in K, \\ \lim_{k \rightarrow \infty, k \in K} \alpha_k &= 0. \end{aligned}$$

Poichè  $x_k \in \mathcal{L}_0$ , per ogni  $k \in K$ , e che  $\mathcal{L}_0$  è un insieme compatto esisterebbero un sottoinsieme infinito di indici  $K' \subseteq K$  e tre sottosequenze  $\{x_k\}_{K'}$ ,  $\{\frac{d_k}{\|d_k\|}\}_{K'}$  e  $\{\alpha_k\}_{K'}$  tali che:

$$\lim_{k \rightarrow \infty, k \in K'} x_k = \bar{x}, \quad (2.14)$$

$$\lim_{k \rightarrow \infty, k \in K'} \frac{d_k}{\|d_k\|} = \bar{d}, \quad (2.15)$$

$$\lim_{k \rightarrow \infty, k \in K'} \alpha_k = 0, \quad (2.16)$$

$$\lim_{k \rightarrow \infty, k \in K'} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \nabla f(\bar{x})^T \bar{d} = -\eta < 0. \quad (2.17)$$

Dalla (2.16) si avrebbe che, per valori  $k \in K'$  sufficientemente grandi,  $\alpha_k < a$ . Per questi valori di  $k$ , le condizioni su  $\alpha_k$  del criterio di Armijo implicherebbero:

$$f(x_k + \frac{\alpha_k}{\delta} d_k) > f(x_k) + \frac{\alpha_k}{\delta} \gamma \nabla f(x_k)^T d_k, \quad (2.18)$$

Ricordando di nuovo il Teorema della media si avrebbe che:

$$f(x_k + \frac{\alpha_k}{\delta} d_k) = f(x_k) + \frac{\alpha_k}{\delta} \nabla f(x_k + \theta_k \frac{\alpha_k}{\delta} d_k)^T d_k,$$

con  $\theta_k \in (0, 1)$ .

Applicando questo teorema alla (2.18) si avrebbe per valori  $k \in K'$  sufficientemente grandi:

$$\nabla f(x_k + \theta_k \frac{\alpha_k}{\delta} d_k)^T d_k > \gamma \nabla f(x_k)^T d_k,$$

da cui seguirebbe:

$$\nabla f(x_k + \theta_k \frac{\alpha_k}{\delta} d_k)^T \frac{d_k}{\|d_k\|} > \gamma \frac{\nabla f(x_k)^T d_k}{\|d_k\|}. \quad (2.19)$$

Facendo il limiti per  $k \rightarrow \infty$  e  $k \in K'$ , la (2.19) porterebbe a:

$$\nabla f(\bar{x})^T \bar{d} \geq \gamma \nabla f(\bar{x})^T \bar{d}, \quad (2.20)$$

che contraddirebbe la (2.17) ed l'ipotesi che  $\gamma \in (0, 1)$ . □

Osserviamo innanzitutto che l'ipotesi ii) della precedente proposizione non è particolarmente restrittiva. Infatti, come si è visto precedentemente, per soddisfare la condizione iii) di Proposizione 2.4.3, si sceglie una direzione che verifica

$$\nabla f(x_k)^T d_k \leq -c_1 \|\nabla f(x_k)\|^{q_1}$$

La precedente relazione implica la ii) di Proposizione 2.4.5. Infatti, dalla disuguaglianza Schwarz, si ha:

$$\left( \frac{|\nabla f(x_k)^T d_k|}{\|d_k\|} \right)^{q_1} \leq \|\nabla f(x_k)\|^{q_1} \leq \frac{1}{c_1} |f(x_k)^T d_k| \leq \frac{M}{c_1} \|d_k\|,$$

dove  $M$  è il massimo di  $\|\nabla f(x_k)\|$  su  $\mathcal{L}_0$ .

Da quanto visto un algoritmo di ricerca unidimensionale che utilizza il criterio di Armijo inizia la ricerca di  $\alpha_k$  a partire da un valore costante  $a$  prefissato e calcola la funzione in tutti i punti  $a\delta^j$  con  $j = 0, 1, \dots$ , fino a trovare un valore che soddisfi la (2.9). Un tale algoritmo è sicuramente molto adatto per i metodi tipo-Newton in cui iniziare la ricerca unidimensionale a partire dal valore costante  $a = 1$  è un requisito essenziale per assicurarne la rapidità di convergenza superlineare. Per altri metodi (per esempio metodi del gradiente, metodi delle direzioni coniugate) partire da un valore di prova iniziale  $\alpha = a$  costante per ogni iterazione può richiedere un numero elevato di valutazioni della funzione durante la ricerca unidimensionale in quanto il valore  $a$ , non avendo nessuna motivazione teorica, può produrre spostamenti troppo grandi (cioè  $a$  può essere molto distante da un minimo di  $\phi(\alpha)$ ); da qui la necessità di dover ridurre molte volte il valore di  $\alpha$  secondo il fattore costante  $\delta$ .

Per questi metodi è preferibile far partire la ricerca unidimensionale da una stima di un minimo di  $\phi(\alpha)$  ottenuta utilizzando delle approssimazioni quadratiche o cubiche della funzione  $\phi(\alpha)$  e quindi variabile con  $k$ . Da cui, per questi casi, è necessario definire criteri di accettabilità che risultino applicabili anche a partire da stime iniziali non costanti.

Un esempio di un criterio di questo tipo è costituito dalla seguente variazione del criterio Armijo.

*Criterio di Armijo generalizzato*

- Dato  $\gamma \in (0, 1/2)$ .
- Si sceglie  $\alpha_k > 0$  tale che:

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \alpha_k \gamma \nabla f(x_k)^T d_k \\ f(x_k + 2\alpha_k d_k) &> f(x_k) + 2\alpha_k \gamma \nabla f(x_k)^T d_k \end{aligned}$$

Il precedente criterio permette di utilizzare stime iniziali del passo che posso variare ogni iterazione, però richiede almeno due calcoli di funzione.

Nel caso in cui si è in grado di produrre dei valori iniziali del parametro  $\alpha$  efficienti, è preferibile utilizzare le seguenti condizioni di Goldstein che, nel caso più favorevole usano un solo calcolo di funzione obiettivo.

*Condizioni di Goldstein*

- Dati  $0 < \gamma_1 < \gamma_2 < 1/2$ .
- Si sceglie  $\alpha_k$  tale da soddisfare:

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + \gamma_1 \alpha_k \nabla f(x_k)^T d_k \\ f(x_k + \alpha_k d_k) &\geq f(x_k) + \gamma_2 \alpha_k \nabla f(x_k)^T d_k. \end{aligned}$$

Da un punto di vista geometrico, questa condizione impone di scegliere come valore di  $\alpha_k$  un punto tale che il corrispondente valore di  $\phi$  sia compreso tra le due rette con pendenza, rispettivamente  $\gamma_1 \nabla f(x_k)^T d_k$  e  $\gamma_2 \nabla f(x_k)^T d_k$  passanti per il punto  $(0, \phi(0))$ .

Anche, per il criterio di Armijo generalizzato e per le condizioni di Goldstein si possono dimostrare le proprietà a) e b) della Proposizione 2.4.5. Inoltre, per entrambe, non è necessario richiedere l'assunzione ii).

### Condizioni di Wolfe

In certi casi è necessario scegliere il passo  $\alpha_k$  in maniera più precisa rispetto al criterio di Armijo o alle condizioni di Goldstein. A questo fine si possono considerare le condizioni di Wolfe che sono dei criteri di accettabilità che impongono condizioni sulla derivata della funzione  $\phi(\alpha)$  nel punto  $\alpha_k$ .

*Condizioni deboli di Wolfe*

- Dati  $\gamma_1 \in (0, 1/2)$  e  $\gamma_2 \in (\gamma_1, 1)$ .
- Si sceglie  $\alpha_k$  tale da soddisfare:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma_1 \alpha_k \nabla f(x_k)^T d_k \quad (2.21)$$

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \gamma_2 \nabla f(x_k)^T d_k. \quad (2.22)$$

*Condizioni forti di Wolfe*

- Dati  $\gamma_1 \in (0, 1/2)$  e  $\gamma_2 \in (\gamma_1, 1)$ .
- Si sceglie  $\alpha_k$  tale da soddisfare:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \gamma_1 \alpha_k \nabla f(x_k)^T d_k \quad (2.23)$$

$$|\nabla f(x_k + \alpha_k d_k)^T d_k| \leq \gamma_2 |\nabla f(x_k)^T d_k|. \quad (2.24)$$

Analogamente al criterio di Armijo o alle condizioni di Goldstein, le precedenti condizioni di Wolfe richiedono una sufficiente riduzione della funzione  $\phi(\alpha)$  (attraverso le (2.21) e (2.23)). La (2.22) richiede inoltre che la tangente alla curva  $\phi(\alpha)$  in  $\alpha_k$  abbia pendenza positiva ( $\phi(\alpha)$  crescente) oppure abbia pendenza negativa ma minore, in valore assoluto, di  $\gamma_2 |\nabla f(x_k)^T d_k|$ . Il criterio forte con la (2.24) richiede invece che la pendenza sia in valore assoluto minore di  $\gamma_2 |\nabla f(x_k)^T d_k|$ . In altre parole questo equivale a richiedere che  $\alpha_k$  sia scelto in una zona in cui  $\phi(\alpha)$  è sufficientemente piatta.

Per entrambi i criteri di Wolfe si possono stabilire le stesse proprietà.

**Proposizione 2.4.6** *Supponiamo che l'insieme di livello  $\mathcal{L}_0$  sia compatto e che  $\nabla f(x_k)'d_k < 0$  per ogni  $k$ . Allora, esiste un intervallo  $[\alpha_l, \alpha_u]$ , con  $0 \leq \alpha_l < \alpha_u$  tale che ogni  $\alpha_k \in [\alpha_l, \alpha_u]$  soddisfa il criterio debole (forte) di Wolfe e la successione definita da  $x_{k+1} = x_k + \alpha_k d_k$  soddisfa le condizioni:*

(a)  $f(x_{k+1}) < f(x_k)$ ;

(b)  $\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)'d_k}{\|d_k\|} = 0$ ;

(c) *se esiste una costante di Lipschitz  $L > 0$  tale che, per ogni  $x, y \in \mathcal{L}_0$  risulti*

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|,$$

*allora si ha anche:*

$$\sum_{k=0}^{\infty} \left( \frac{\nabla f(x_k)'d_k}{\|d_k\|} \right)^2 < \infty.$$

Il punto (c) della precedente proposizione mostra che i criteri di Wolfe, a differenza delle altre tecniche, sono in grado di soddisfare l'ipotesi ii) della Proposizione 2.4.4.

### 2.4.3 Metodo del gradiente

Come si è detto precedentemente, normalmente la direzione di ricerca  $d_k$  viene determinata considerando una approssimazione della funzione obiettivo. Nel metodo del gradiente si fa riferimento una *approssimazione lineare* di  $f(x_k + d)$  pensata come funzione del solo vettore  $d$ .

Avendo supposto che  $f$  è continuamente differenziabile, si può scrivere:

$$f(x_k + d) = f(x_k) + \nabla f(x_k)'d + \beta_1(x_k, d),$$

con

$$\lim_{\|d\| \rightarrow 0} \frac{\beta_1(x_k, d)}{\|d\|} = 0.$$

L'idea del metodo del gradiente è quella di approssimare la funzione  $f(x_k + d)$  con la funzione  $\psi_k(d)$  data da:

$$\psi_k(d) := f(x_k) + \nabla f(x_k)'d,$$

e di scegliere come direzione di ricerca  $d_k$  quella direzione che minimizza la  $\psi_k(d)$  nella sfera di raggio unitario. In altre parole,  $d_k$  è la soluzione del seguente problema:

$$\begin{aligned} \min \psi_k(d), \\ \|d\| = 1, \end{aligned}$$



che è equivalente a

$$\begin{aligned} \min \quad & \nabla f(x_k)^T d, \\ & \|d\| = 1. \end{aligned}$$

Ricordando la disuguaglianza di Schwarz, la soluzione al precedente problema è data da  $d_k^* = -\nabla f(x_k)/\|\nabla f(x_k)\|$  e quindi il metodo del gradiente può essere descritto da:

$$x_{k+1} = x_k - \tilde{\alpha}_k \frac{\nabla f(x_k)}{\|\nabla f(x_k)\|},$$

che, ridefinendo il passo lungo la direzione ( $\alpha := \tilde{\alpha}/\|\nabla f(x_k)\|$ ), può essere riscritto come

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k).$$

### Convergenza globale

Poiché nel metodo del gradiente si ha  $d_k = -\nabla f(x_k)$ , si può notare subito che è soddisfatta la condizione iii) di Proposizione 2.2.1. Quindi la convergenza globale di questo metodo può essere caratterizzata dal seguente risultato.

**Proposizione 2.4.7** *Sia  $\{x_k\}$  la successione prodotta dal metodo del gradiente e si assuma che  $\mathcal{L}_0$  sia compatto. Si supponga inoltre che la ricerca unidimensionale per calcolare  $\alpha_k$  sia tale da soddisfare le condizioni:*

- i)  $f(x_{k+1}) < f(x_k)$  se  $\nabla f(x_k) \neq 0$ ;
- ii) se  $\nabla f(x_k) \neq 0$  per ogni  $k$ , si ha:

$$\lim_{k \rightarrow \infty} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0.$$

Allora, o esiste un indice  $\nu \geq 0$  tale che  $x_\nu \in \mathcal{L}_0$  e  $\nabla f(x_\nu) = 0$ , oppure viene prodotta una successione infinita tale che:

- a)  $\{x_k\}$  rimane in  $\mathcal{L}_0$  ed ammette punti di accumulazione;
- b) ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  soddisfa  $\nabla f(\bar{x}) = 0$ ;
- c) ogni punto di accumulazione è un punto stazionario che non è un massimo locale di  $f$ .

In base alla proposizione precedente, per ottenere un algoritmo globalmente convergente basta associare al metodo del gradiente una qualsiasi delle ricerche unidimensionali descritte nella precedente sezione, in quanto tutte soddisfano le proprietà (i) e (ii) di Proposizione 2.4.7.

## Rapidità di convergenza

Purtroppo il difetto del metodo del gradiente è la sua rapidità di convergenza. Infatti si può dimostrare che anche nel caso ideale in cui la funzione da minimizzare sia una funzione *quadratica convessa* ed il passo  $\alpha_k$  sia calcolato minimizzando esattamente la funzione scalare  $\phi(\alpha) = f(x_k + \alpha d_k)$  esistono punti iniziali a partire dai quali il metodo del gradiente produce una sequenza di punti  $\{x_k\}$  per cui la rapidità di convergenza è lineare.

In particolare si ha il seguente risultato.

**Proposizione 2.4.8** *Sia data funzione*

$$f(x_k) = \frac{1}{2} x^T Q x,$$

dove  $Q$  è una matrice simmetrica definita positiva.

Il metodo del gradiente definito con  $\alpha_k$  ottenuto minimizzando esattamente la funzione  $f(x_k + \alpha d_k)$ , cioè:

$$\alpha_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T Q d_k} = \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^T Q \nabla f(x_k)}$$

converge al minimo  $x^* = 0$  di  $f(x)$  e si ha:

$$\|x_{k+1} - x^*\| \leq \left(\frac{\lambda_M}{\lambda_m}\right)^{1/2} \left(\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m}\right) \|x_k - x^*\|,$$

in cui  $\lambda_M$  e  $\lambda_m$  sono, rispettivamente, il massimo ed il minimo autovalore di  $Q$ .

Il precedente risultato mostra che il metodo del gradiente nel caso di funzioni quadratiche e convesse ha una rapidità di convergenza che è almeno lineare e che dipende dal rapporto  $\lambda_M/\lambda_m$  tra il massimo ed il minimo autovalore della matrice Hessiana di  $f(x)$ . Quindi ci si può aspettare che la rapidità di convergenza del metodo del gradiente peggiori, in genere, al crescere della differenza tra  $\lambda_M$  e  $\lambda_m$ , ossia all'aumentare del *mal condizionamento* della matrice  $Q$ .

Questa analisi, svolta in un semplice caso quadratico convesso, dà delle indicazioni negative sul comportamento locale del metodo gradiente nel caso di funzioni non quadratiche. L'esperienza di calcolo ha, infatti, confermato queste conclusioni in quanto il metodo del gradiente è apparso complessivamente un metodo molto semplice di minimizzazione non vincolata ma poco efficiente.

## Motodi del gradiente non monotoni

Per ottenere informazioni sulla rapidità di convergenza ed ottenere, nella precedente proposizione, si è scelto il passo  $\alpha_k$  nella maniera che sembrava essere la migliore possibile, cioè minimizzando esattamente lungo l'antigradiente. Tutte le conclusioni ottenute e le successive implementazioni del metodo del gradiente sono state influenzate da questa scelta.

La seguente proposizione descrive un caso in cui si fa una diversa scelta per il passo  $\alpha_k$ .

**Proposizione 2.4.9** *Sia data funzione*

$$f(x_k) = \frac{1}{2}x^T Qx,$$

dove  $Q$  è una matrice simmetrica definita positiva.

Il metodo del gradiente in cui si abbia

$$\alpha_k = \frac{1}{\lambda_{k+1}}$$

dove  $\lambda_i, i = 1, \dots, n$  sono gli autovalori della matrice  $Q$ , converge al minimo  $x^*$  di  $f(x)$  al più in  $n$  passi.

Quindi, a differenza della Proposizione 2.4.8, in questo caso si ottiene la convergenza finita dell'algorithmo qualunque sia il punto di partenza. Questa differenza fra i risultati ottenuti utilizzando due scelte diverse per il passo  $\alpha_k$  ci porta alle seguenti osservazioni:

- Determinare il passo  $\alpha_k$  cercando di minimizzare la funzione  $f(x_k - \alpha \nabla f(x_k))$  può non essere la scelta migliore;
- scelte differenti possono far migliorare l'efficienza del metodo del gradiente nel caso di funzioni non quadratiche;
- in particolare bisognerebbe determinare  $\alpha_k$  cercando di "estrarre" informazioni del secondo ordine della funzione obiettivo.

Recentemente, ispirandosi ai metodi Quasi-Newton (vedere Sezione 2.4.5), si sono proposte delle nuove regole per la scelta di  $\alpha$  non più legate alla minimizzazione lungo la direzione di ricerca. Una di queste consiste nel determinare  $\alpha_k$  in maniera da minimizzare la funzione:

$$\| (x_k - x_{k-1}) - \alpha (\nabla f(x_k) - \nabla f(x_{k-1})) \|^2$$

cioè

$$\alpha_k = \frac{(x_k - x_{k-1})^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{(\nabla f(x_k) - \nabla f(x_{k-1}))^T (\nabla f(x_k) - \nabla f(x_{k-1}))}.$$

Queste nuove versioni del metodo del gradiente, anche se non impongono più una diminuzione monotona della funzione obiettivo, fanno migliorare in maniera significativa l'efficienza computazionale del metodo del gradiente.

## Metodi del gradiente che non richiedono il calcolo della funzione

I recenti sviluppi riguardanti i sistemi ad apprendimento richiedono la definizione di metodi di ottimizzazione in grado di affrontare problemi in cui la valutazione esatta della funzione obiettivo può essere estremamente costosa dal punto di vista computazionale. Per questo motivo sono diventate di interesse alcune versioni del metodo del gradiente che non richiedono la valutazione della funzione obiettivo.

Un esempio di un tale algoritmo è descritto nella seguente proposizione

**Proposizione 2.4.10** *Si assuma che  $\mathcal{L}_0$  sia compatto e che il gradiente  $\nabla f$  soddisfi una condizione di Lipschitz, cioè che esista una costante  $L$  tale che per ogni  $x, y \in \mathcal{L}_0$  si abbia:*

$$\|\nabla f(x_k) - \nabla f(y)\| \leq L\|x - y\|.$$

Sia  $\{x_k\}$  la successione prodotta dal seguente metodo:

$$x_{k+1} = x_k - \bar{\alpha} \nabla f(x_k)$$

con il passo  $\bar{\alpha}$  che soddisfa:

$$\epsilon \leq \bar{\alpha} \leq \frac{2 - \epsilon}{L} \quad \text{con} \quad \epsilon > 0.$$

Allora, o esiste un indice  $\nu \geq 0$  tale che  $x_\nu \in \mathcal{L}_0$  e  $\nabla f(x_\nu) = 0$ , oppure viene prodotta una successione infinita tale che:

- a)  $\{x_k\}$  rimane in  $\mathcal{L}_0$  ed ammette punti di accumulazione;
- b) ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  soddisfa  $\nabla f(\bar{x}) = 0$ ;
- c) ogni punto di accumulazione è un punto stazionario che non è un massimo locale di  $f$ .

**Prova.** Il Teorema della media nella forma di integrale permette di scrivere:

$$f(x_{k+1}) - f(x_k) = \int_0^1 \nabla f(x_k + \lambda(x_{k+1} - x_k))^T (x_{k+1} - x_k) d\lambda,$$

da cui:

$$f(x_{k+1}) - f(x_k) \leq \int_0^1 \|\nabla f(x_k + \lambda(x_{k+1} - x_k)) - \nabla f(x_k)\| \|x_{k+1} - x_k\| d\lambda + \nabla f(x_k)^T (x_{k+1} - x_k),$$

Dalla assunzione di Lipschitzianet  del gradiente si ha:

$$\begin{aligned} f(x_{k+1}) - f(x_k) &\leq \nabla f(x_k)^T (x_{k+1} - x_k) + L\|x_{k+1} - x_k\|^2 \int_0^1 \lambda d\lambda \\ &= \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &= -\bar{\alpha}\|\nabla f(x_k)\|^2 + \bar{\alpha}^2 \frac{L}{2}\|\nabla f(x_k)\|^2 \\ &= -\bar{\alpha} \left(1 - \bar{\alpha} \frac{L}{2}\right) \|\nabla f(x_k)\|^2, \end{aligned}$$

che implica  $f(x_{k+1}) \leq f(x_k)$  per ogni  $k$ , per la scelta  $0 < \bar{\alpha} < \frac{2}{L}$ . Quindi il punto a) è provato.

L'assunzione di compattezza dell'insieme  $\mathcal{L}_0$  e il punto a) assicurano che la sequenza non crescente  $\{f(x_k)\}$  è limitata inferiormente e, quindi, convergente. Questo fatto e l'ultima disuguaglianza precedente provano i punti b) e c).  $\square$

L'algoritmo descritto dalla precedente proposizione è interessante per la sua capacità di convergere a punti stazionari senza utilizzare i valori della funzione obiettivo. Purtroppo presenta il difetto che la scelta giusta del passo  $\bar{\alpha}$  richiede il valore della costante di Lipschitz che normalmente non è nota. In pratica al posto della costante  $L$  si usano dei valori che dovrebbero essere del suo sovrastime. Tuttavia, non avendo idea di quanto possa valere  $L$ , si è costretti ad usare valori molto grandi che producono dei passi  $\bar{\alpha}$  molto piccoli e che possono rendere inefficiente l'algoritmo.

La seguente proposizione indica che si possono definire algoritmi globalmente convergenti che non utilizzano la funzione obiettivo e che non usano la conoscenza della costante di Lipschitz. Questo risultato è ottenuto richiedendo che i passi  $\alpha_k$  siano scelti con delle leggi opportune.

**Proposizione 2.4.11** *Si assuma che il gradiente  $\nabla f$  soddisfi una condizione di Lipschitz, cioè esista una costante  $L$  tale che per ogni  $x, y \in R^n$  si abbia:*

$$\|\nabla f(x_k) - \nabla f(y)\| \leq L\|x - y\|.$$

Sia  $\{x_k\}$  la successione prodotta dal seguente metodo:

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

dove i passi  $\alpha_k$  soddisfano:

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty.$$

Allora, o  $\lim_{k \rightarrow \infty} f(x_k) = -\infty$  oppure ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  è un punto stazionario di  $f$ .

## Metodi del gradiente con errore

Un ulteriore passo verso la definizione di metodi in grado di affrontare i difficili problemi di ottimizzazione che nascono dalle recenti sfide applicative è quello di definire e studiare metodi che possa superare l'ulteriore difficoltà di non conoscere con esattezza il gradiente della funzione obiettivo. Cioè metodi che utilizzano un "gradiente con errore". Le basi teoriche per definire tali algoritmi sono le seguenti due proposizioni

**Proposizione 2.4.12** *Si assuma che il gradiente  $\nabla f$  soddisfi una condizione di Lipschitz, cioè che esista una costante  $L$  tale che per ogni  $x, y \in R^n$  si abbia:*

$$\|\nabla f(x_k) - \nabla f(y)\| \leq L\|x - y\|.$$

Sia  $\{x_k\}$  la successione prodotta dal seguente metodo:

$$x_{k+1} = x_k - \bar{\alpha}(\nabla f(x_k) + e_k)$$

dove i vettori  $e_k$  soddisfano per ogni  $k$ :

$$\|e_k\| \leq \bar{\alpha}C, \quad \text{con} \quad C > 0$$

dove il passo  $\bar{\alpha}$  soddisfa:

$$\epsilon \leq \bar{\alpha} \leq \frac{2 - \epsilon}{L} \quad \text{con} \quad \epsilon > 0. \quad (2.25)$$

Allora, o  $\lim_{k \rightarrow \infty} f(x_k) = -\infty$  oppure ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  soddisfa la seguente proprietà

$$\|\nabla f(\bar{x})\| \leq B\bar{\alpha}, \quad \text{con} \quad B > 0.$$

Confrontando le tesi della precedente proposizione con quella di Proposizione 2.4.10 si nota che la presenza dell'errore nel calcolo del gradiente produce un peggioramento significativo delle proprietà teoriche dell'algoritmo.

**Proposizione 2.4.13** *Si assuma che il gradiente  $\nabla f$  soddisfi una condizione di Lipschitz, cioè che esista una costante  $L$  tale che per ogni  $x, y \in R^n$  si abbia:*

$$\|\nabla f(x_k) - \nabla f(y)\| \leq L\|x - y\|.$$

Sia  $\{x_k\}$  la successione prodotta dal seguente metodo:

$$x_{k+1} = x_k - \alpha_k(\nabla f(x_k) + e_k)$$

dove i vettori  $e_k$  soddisfano per ogni  $k$ :

$$\|e_k\| \leq \alpha_k(C + D\|\nabla f(x_k)\|), \quad \text{con} \quad C, D > 0$$

dove i passi  $\alpha_k$  soddisfano:

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \alpha_k^2 < \infty. \quad (2.26)$$

Allora, o  $\lim_{k \rightarrow \infty} f(x_k) = -\infty$  oppure ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  è un punto stazionario di  $f$ .

Confrontando la precedente proposizione con Proposizione 2.4.11 si può notare che la presenza del rumore nel calcolo del gradiente non fa variare le proprietà teoriche dell'algoritmo. Questo è dovuto alle particolari richieste sulla scelta dei passi  $\alpha_k$  che rendono particolarmente robusto il metodo proposto.

Un esempio importante di un problema di ottimo per cui non si può calcolare il gradiente esatto della funzione obiettivo, nasce nell'addestramento dei sistemi ad apprendimento. In questa classe di problemi usualmente si deve minimizzare una funzione obiettivo che ha la seguente struttura:

$$f(x) = \sum_{i=1}^M f_i(x), \quad (2.27)$$

dove  $M$  può essere un numero enorme.

Il fatto che la funzione obiettivo possa essere costituita dalla somma di un numero molto grande di funzioni, rende estremamente costosa la sua valutazione e praticamente impossibile il calcolo esatto del suo gradiente. Per questa classe di problemi si usa la seguente variazione del metodo del gradiente in cui, in ogni iterazione, viene valutato solamente il gradiente di una funzione della sommatoria.

*Metodo del gradiente incrementale*

$$x_{k+1} = x_k - \alpha_k \sum_{i=1}^M \nabla f_i(z_k^i),$$

dove

$$\begin{aligned} z_k^1 &= x_k, \\ z_k^{i+1} &= z_k^i - \alpha_k \nabla f_i(z_k^i), \quad i = 1, \dots, M-1. \end{aligned}$$

Il precedente metodo ricade nella classe dei metodi del gradiente con errore dove:

$$e_k = \sum_{i=1}^M (\nabla f_i(x_k) - \nabla f_i(z_k^i)).$$

Se si assume che tutti i gradienti  $\nabla f_i$ , con  $i = 1, \dots, M$  soddisfino una condizione di Lipschitz, cioè che esista una costante  $L$  tale che per ogni  $x, y \in \mathbb{R}^n$  si abbia:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L\|x - y\|, \quad i = 1, \dots, M,$$

dalle Proposizioni 2.4.12 e 2.4.13 si possono derivare delle versioni del metodo del gradiente incrementale globalmente convergenti.

In particolare se si assume che esiste una costante  $C > 0$  per cui, per ogni  $x \in \mathbb{R}^n$ ,

$$\|\nabla f_i(x)\| \leq C, \quad i = 1, \dots, M,$$

il metodo del gradiente incrementale con  $\alpha_k = \bar{\alpha}$ , con  $\bar{\alpha}$  che soddisfa (2.25), ricade nelle ipotesi del Proposizione 2.4.12.

Invece, se si assume che esistono delle costanti  $C > 0$  e  $D > 0$  per cui, per ogni  $x \in R^n$ ,

$$\|\nabla f_i(x)\| \leq C + D\|\nabla f(x)\|, \quad i = 1, \dots, M,$$

il metodo del gradiente incrementale con i passi  $\alpha_k$  che soddisfano le condizioni (2.26), ricade nelle ipotesi del Proposizione 2.4.13.

#### 2.4.4 Metodo di Newton

Se la funzione  $f$  è due volte continuamente differenziabile e se  $x_k$  è un punto dato, si può scrivere:

$$f(x_k + d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d + \beta_2(x_k, d), \quad (2.28)$$

con

$$\lim_{\|d\| \rightarrow 0} \frac{\beta_2(x_k, d)}{\|d\|^2} = 0.$$

L'approccio del metodo di Newton è quello di cercare di determinare il minimo di una funzione due volte continuamente differenziabile costruendo una successione di punti ottenuti minimizzando ad ogni passo l'*approssimazione quadratica* della funzione  $f$  data da:

$$\phi_k(d) := f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T \nabla^2 f(x_k) d. \quad (2.29)$$

Tale funzione quadratica  $\phi_k$ , in base alla (2.28), può essere considerata una buona approssimazione di  $f(x_k + d)$ .

Quindi, a partire da  $x_0$ , il metodo di Newton è definito dall'iterazione:

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k). \quad (2.30)$$

dove il vettore  $d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ , detto *direzione di Newton*, è un punto stazionario della funzione quadratica  $\phi_k(d)$ . Inoltre, se la matrice Hessiana  $\nabla^2 f(x_k)$  è definita positiva, allora la direzione di Newton è il punto di minimo della funzione quadratica  $\phi_k(d)$ .

##### 2.4.4.1 Convergenza locale e rapidità di convergenza

Il fatto di utilizzare le informazioni derivanti dalle derivate del primo e del secondo ordine permette al metodo di Newton di avere localmente ottime proprietà di convergenza. Infatti, utilizzando queste informazioni è possibile costruire ed utilizzare la funzione quadratica  $\phi_k(d)$  che, in un intorno sufficientemente piccolo di un minimo della funzione obiettivo, approssima ottimamente il comportamento della funzione obiettivo. La seguente proposizione mostra le ottime proprietà locali di convergenza del metodo di Newton.

**Proposizione 2.4.14** *Sia  $f$  una funzione due volte continuamente differenziabile su  $R^n$ . Supponiamo inoltre che valgano le condizioni seguenti:*



- i) esiste un  $x^* \in R^n$  tale che  $\nabla f(x^*) = 0$ ;
- ii) la matrice Hessiana  $\nabla^2 f(x^*)$  è non singolare;
- iii) esiste una costante  $L > 0$  tale che, per ogni  $x, y \in R^n$ , si abbia

$$\left\| \nabla^2 f(x) - \nabla^2 f(y) \right\| \leq L \|x - y\|.$$

Allora esiste una sfera aperta  $\mathcal{B}(x^*; \varepsilon) := \{x \in R^n : \|x - x^*\| < \varepsilon\}$ , tale che, se  $x_0 \in \mathcal{B}(x^*; \varepsilon)$ , la successione  $\{x_k\}$  generata dal metodo di Newton a partire da  $x_0$  rimane in  $\mathcal{B}(x^*; \varepsilon)$  e converge a  $x^*$  con rapidità di convergenza quadratica.

**Dim.** Poiché  $\nabla^2 f(x^*)$  è non singolare e  $\nabla^2 f(x)$  è continua su  $R^n$ , è possibile trovare un  $\varepsilon_1 > 0$  e un  $\mu > 0$  tali che  $\mathcal{B}(x^*; \varepsilon_1)$  e che, per ogni  $x \in \mathcal{B}(x^*; \varepsilon_1)$  risulti:

$$\left\| \nabla^2 f(x)^{-1} \right\| \leq \mu.$$

Sia ora

$$\varepsilon < \min [\varepsilon_1, 1/\mu L]$$

e supponiamo che sia  $x_k \in \mathcal{B}(x^*; \varepsilon)$ .

Essendo per ipotesi  $\nabla f(x^*) = 0$ , possiamo riscrivere la (2.30) nella forma:

$$x_{k+1} - x^* = -\nabla^2 f(x_k)^{-1} [ -\nabla^2 f(x_k)(x_k - x^*) + \nabla f(x_k) - \nabla f(x^*) ],$$

da cui segue:

$$\begin{aligned} \|x_{k+1} - x^*\| &\leq \left\| \nabla^2 f(x_k)^{-1} \right\| \left\| -\nabla^2 f(x_k)(x_k - x^*) + \nabla f(x_k) - \nabla f(x^*) \right\| \\ &\leq \mu \left\| -\nabla^2 f(x_k)(x_k - x^*) + \nabla f(x_k) - \nabla f(x^*) \right\|. \end{aligned} \quad (2.31)$$

Poiché  $\nabla f$  è differenziabile, si ha:

$$\nabla f(x_k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^* + \lambda(x_k - x^*))(x_k - x^*) d\lambda,$$

e quindi, per la (2.31) si può scrivere:

$$\|x_{k+1} - x^*\| \leq \mu \left\| \int_0^1 [\nabla^2 f(x^* + \lambda(x_k - x^*)) - \nabla^2 f(x_k)](x_k - x^*) d\lambda \right\|.$$

Dalla disuguaglianza precedente si ottiene:

$$\|x_{k+1} - x^*\| \leq \mu \int_0^1 \left\| \nabla^2 f(x^* + \lambda(x_k - x^*)) - \nabla^2 f(x_k) \right\| d\lambda \|x_k - x^*\|$$

e quindi, tenendo conto dell'ipotesi (iii) si ha:

$$\|x_{k+1} - x^*\| \leq \mu L \|x_k - x^*\|^2. \quad (2.32)$$

Poiché si è assunto  $x_k \in \mathcal{B}(x^*, \varepsilon)$  e si è scelto  $\varepsilon < 1/\mu L$ , dalla (2.32) segue

$$\|x_{k+1} - x^*\| \leq \mu L \|x_k - x^*\|^2 \leq \mu L \varepsilon \|x_k - x^*\| < \|x_k - x^*\|. \quad (2.33)$$

Ciò implica  $x_{k+1} \in \mathcal{B}(x^*, \varepsilon)$  e di conseguenza, per induzione, si ha  $x_k \in \mathcal{B}(x^*, \varepsilon)$  per ogni  $k$ .

Applicando ripetutamente la (2.33) si ha anche:

$$\|x_k - x^*\| \leq (\mu L \varepsilon)^k \|x_0 - x^*\|,$$

da cui segue, essendo  $\mu L \varepsilon < 1$ , che  $x_k \rightarrow x^*$ . La (2.32) implica allora che la rapidità di convergenza è quadratica.  $\square$

**Osservazione.** Il metodo di Newton presenta anche una interessante proprietà di invarianza rispetto alla scalatura delle variabili.

Supponiamo di effettuare un cambio di variabili del tipo

$$y = Tx$$

dove  $T$  è una matrice invertibile.

Nel nuovo spazio di variabili la funzione da minimizzare, il suo gradiente ed Hessiano diventano:

$$\tilde{f}(y) = f(T^{-1}y), \quad \nabla \tilde{f}(y) = T^{-1} \nabla f(T^{-1}y), \quad \nabla^2 \tilde{f}(y) = T^{-1} \nabla^2 f(T^{-1}y) T^{-1}.$$

Una iterazione del metodo di Newton rispetto al nuovo vettore di variabili  $y$  assume la seguente forma:

$$y_{k+1} = y_k - \nabla^2 \tilde{f}(y_k)^{-1} \nabla \tilde{f}(y_k) = y_k - T \nabla^2 f(T^{-1}y_k)^{-1} T T^{-1} \nabla f(T^{-1}y_k).$$

Se dal punto  $y_{k+1}$  si ritorna allo spazio delle variabili  $x$  si ha:

$$x_{k+1} = T^{-1}y_{k+1} = T^{-1}y_k - \nabla^2 f(T^{-1}y_k)^{-1} \nabla f(T^{-1}y_k) = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k),$$

che coincide con il punto fornito dalla iterazione del metodo di Newton fatta direttamente nello spazio delle  $x_k$ .

#### 2.4.4.2 Modifiche globalmente convergenti del metodo di Newton

La precedente Proposizione 2.4.14 mette in evidenza sia i pregi e sia i difetti del metodo di Newton. Infatti da una parte ci indica che questo metodo ha in pratica la migliore rapidità di convergenza, dall'altra però, mostra che nella sua forma pura, non può essere utilizzato come algoritmo di minimizzazione globalmente convergente. Infatti ci sono i seguenti problemi da affrontare:

- $\nabla^2 f(x_k)$  può essere singolare e, quindi, la direzione di Newton può non essere definita in  $x_k$ ;

- il punto iniziale  $x_0$  può non appartenere all'insieme  $\mathcal{B}(x^*; \varepsilon)$  e, di conseguenza, la successione prodotta dal metodo di Newton può non convergere;
- la Proposizione 2.4.14 mostra che il metodo di Newton converge a partire da un intorno di un qualsiasi punto stazionario in cui  $\nabla^2 f(x_k)$  non è singolare, quindi si può avere convergenza verso massimi locali.

Un argomento importante nel campo dell'ottimizzazione non vincolata è quello di definire degli algoritmi di minimizzazione che superino le precedenti difficoltà, e che quindi siano globalmente convergenti pur preservando contemporaneamente le buone caratteristiche di rapidità di convergenza del metodo di Newton. In particolare è utile dare la seguente definizione:

**Definizione 2.4.15** *Sia  $f$  due volte continuamente differenziabile e supponiamo che l'insieme di livello  $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$  sia compatto. Diremo che l'algoritmo definito dall'iterazione:*

$$x_{k+1} = x_k + s_k, \quad k = 0, 1 \dots$$

*(con  $s_k = 0$  se e solo se  $\nabla f(x_k) = 0$ ), è una modifica globalmente convergente del metodo di Newton se valgono le seguenti proprietà:*

- (i) *se  $\{x_k\}$  è infinita ogni punto di accumulazione di  $\{x_k\}$  è un punto stazionario di  $f$  appartenente a  $\mathcal{L}_0$ ;*
- (ii) *nessun punto di accumulazione di  $\{x_k\}$  è un punto di massimo locale di  $f$ ;*
- (iii) *se  $\{x_k\}$  converge ad un punto di minimo locale  $x^*$  di  $f$  e  $\nabla^2 f(x)$  soddisfa le ipotesi della proposizione 2.4.14, esiste un  $k^*$  tale che, per ogni  $k \geq k^*$  si ha:  $s_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ .*

La maggior parte delle modifiche globalmente convergenti del metodo di Newton proposte in letteratura ricadono nelle seguenti due classi di metodi:

- metodi che utilizzano ricerche unidimensionali;
- metodi della regione di confidenza (trust region).

#### **- Metodi che utilizzano ricerche unidimensionali**

Uno dei modi più immediati per realizzare una modifica globalmente convergente del metodo di Newton è quello di far riferimento ai criteri di convergenza globale descritti precedentemente.

Seguendo questa impostazione, il primo passo da fare è quello di introdurre di nuovo il parametro scalare  $\alpha_k$  che definisce il passo lungo la direzione (ciò equivale a porre  $s_k := \alpha_k d_k$  nella Definizione 2.28). Il secondo è quello di perturbare la direzione di Newton se non soddisfa le condizioni di convergenza. In particolare si può stabilire il seguente risultato.

**Proposizione 2.4.16** *Sia  $f$  due volte continuamente differenziabile e supponiamo che l'insieme di livello  $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$  sia compatto. L'algoritmo definito dall'iterazione:*

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots,$$

*è una modifica globalmente convergente del metodo di Newton se valgono le seguenti proprietà:*

- (i) *il passo  $\alpha_k$  viene calcolato utilizzando il metodo di Armijo assumendo come valore di prova iniziale  $a = 1$ ;*
- (ii) *la direzione  $d_k$  soddisfa le condizioni (iii) di Proposizione 2.4.3 e (i)-(ii) di Proposizione 2.4.5;*
- (iii) *se  $\{x_k\}$  converge ad un punto di minimo locale  $x^*$  di  $f(x)$  e  $\nabla^2 f(x)$  soddisfa le ipotesi della Proposizione 2.4.14, esiste un  $k^*$  tale che, per ogni  $k \geq k^*$  si ha:*  

$$d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k).$$

Il risultato precedente, mentre definisce esattamente il modo di calcolare il passo  $\alpha_k$ , lascia abbastanza libertà nella scelta della direzione di ricerca  $d_k$ . In letteratura sono stati proposti vari criteri per calcolare  $d_k$  in maniera tale da soddisfare le condizioni (iii) di Proposizione 2.4.3 e (i)-(ii) di Proposizione 2.4.5;

Uno dei modi più semplici di determinare una direzione  $d_k$  che permetta di definire una modifica globalmente convergente del metodo di Newton, è quella di utilizzare la direzione dell'antigradiente quando la direzione di Newton non soddisfa opportune condizioni. In particolare è possibile definire il seguente schema che fornisce una  $d_k$  che soddisfa le condizioni (iii) di Proposizione 2.4.3 e (i)-(ii) di Proposizione 2.4.5;

*Modifica della direzione di Newton*

- *Dati:  $c_1 > 0$ ,  $c_2 > 0$ ,  $c_3 > 0$ ,  $q_1 > 0$ ,  $q_2 \geq 2$  e  $q_3 \geq 3$ .*
- *Se vale una delle seguenti relazioni:*

$$\begin{aligned} |Det(\nabla^2 f(x_k))| &\leq c_1 \|\nabla f(x_k)\|^{q_1}, \\ -\nabla f(x_k)' [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) &> -c_3 \|\nabla f(x_k)\|^{q_3}, \\ \left\| [\nabla^2 f(x_k)]^{-1} \nabla f(x_k) \right\|^{q_2} &> c_2 \|\nabla f(x_k)\|, \end{aligned}$$

*si pone  $d_k = -\nabla f(x_k)$ ;*

- *altrimenti si pone  $d_k = -[\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$ .*

**- Metodi della regione di confidenza (Trust Region)**

In questa classe di metodi si determina *contemporaneamente* sia la direzione e sia l'ampiezza dello spostamento lungo questa la direzione. Analogamente al metodo di Newton puro, i metodi Trust Region determinano, ad ogni iterazione, un vettore  $s_k$  minimizzando il modello quadratico  $\phi_k(s)$  della funzione obiettivo  $f$ . A differenza del metodo di Newton questa minimizzazione non viene effettuata su tutto  $R^n$  ma solo su una regione sferica di centro  $x_k$  in cui si suppone che il modello quadratico sia una buona approssimazione di  $f$ .

Molti algoritmi di tipo Trust Region sono stati proposti in letteratura, lo schema generale iniziale di questa classe di metodi è il seguente.

*Schema generale di un algoritmo Trust Region*

*Passo 1:* Dati  $x_0 \in R^n$ ,  $a_0 > 0$ ,  $0 < c_1 < c_2 < 1$ ,  $0 < \gamma_1 < \gamma_2 < 1$ ,  $\gamma_3 > 1$  e  $k=0$ .

*Passo 2:* Se  $x_k$  è un punto stazionario stop

*Passo 3:* Sia  $s_k \in R^n$  il minimo di

$$\begin{aligned} \min \phi_k(s) \\ \|s\| \leq a_k. \end{aligned} \quad (2.34)$$

dove

$$\phi_k(s) = f(x_k) + \nabla f(x_k)'s + \frac{1}{2}s'\nabla^2 f(x_k)s.$$

*Passo 4:* Si controlla se  $\phi_k(s)$  in  $\|s\| \leq a_k$  è una buona approssimazione di  $f(x_k + s)$ ;

questo controllo viene effettuato confrontando la riduzione effettiva

$$\Delta f_k(s_k) = f(x_k) - f(x_k + s_k)$$

e la riduzione prevista

$$\Delta \phi_k(s_k) = f(x_k) - \phi_k(s_k);$$

che si hanno passando da  $x_k$  a  $x_k + s_k$ ;

- se  $\Delta f_k(s_k) < c_1 \Delta \phi_k(s_k)$

si diminuisce  $a_{k+1}$  scegliendo  $a_{k+1} \in [\gamma_1 a_k, \gamma_2 a_k]$ ,

si risolve nuovamente il problema (2.34) ponendo  $x_{k+1} = x_k$ ,

$k=k+1$  e si torna la Passo 3.

- se  $\Delta f_k(s_k) \geq c_1 \Delta \phi_k(s_k)$

si pone  $a_{k+1} = a_k$ ,  $x_{k+1} = x_k + s_k$ ,  $k=k+1$  e si torna la Passo 2.

(se  $\Delta f_k(s_k) \geq c_2 \Delta \phi_k(s_k)$ , cioè  $\Delta f_k(s_k)$  è “abbastanza” più grande

di  $\Delta q_k(s_k)$  allora all'iterazione successiva si può aumentare  $a_{k+1}$ , ponendo  $a_{k+1} = \gamma_3 a_k$ .)

Dal punto di vista geometrico si può osservare che:

- se  $a_k \rightarrow 0$  allora la direzione di  $s_k$  tende alla direzione di  $-\nabla f(x_k)$ ;
- se  $\nabla^2 f(x_k)$  è definita positiva allora per valori crescenti di  $a_k$  la direzione di  $s_k$  tende alla direzione di  $-\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$ .

Quindi, l'algoritmo Trust Region fornisce uno spostamento che può essere visto come una combinazione tra uno spostamento lungo l'antigradiente, che tende a garantire la convergenza globale, ed uno spostamento lungo la direzione di Newton che cerca di assicurare una buona rapidità di convergenza.

Per quanto riguarda le proprietà di convergenza dello schema di algoritmo precedentemente descritto si ha il seguente risultato.

**Proposizione 2.4.17** *Sia  $f$  due volte continuamente differenziabile e sia l'insieme di livello  $\mathcal{L}_0 = \{x \in R^n : f(x) \leq f(x_0)\}$  compatto. L'algoritmo Trust Region precedentemente descritto è una modifica globalmente convergente del metodo di Newton.*

*Inoltre, se la successione di punti prodotta dall'algoritmo converge ad un punto  $x^*$  allora la matrice Hessiana in  $x^*$  è semidefinita positiva.*

Dal risultato precedente si può osservare che i metodi Trust Region presentano proprietà più forti rispetto alle modifiche del metodo di Newton viste nel paragrafo precedente. Infatti, come mostra la precedente proposizione, essi convergono a punti che soddisfano le *condizioni necessarie del secondo ordine*.

Purtroppo il prezzo da pagare per ottenere queste buone proprietà di convergenza è alto. Infatti ad ogni passo è necessario risolvere *esattamente* il problema (2.34) e questo è estremamente oneroso specialmente nel caso in cui la matrice Hessiana  $\nabla^2 f(x_k)$  non è definita positiva.

Allo scopo di superare questa difficoltà si sono proposti dei metodi Trust Region che risolvono solo *in maniera approssimata* il sottoproblema (2.34).

Questi metodi sono tipicamente più efficienti dal punto di computazionale rispetto all'algoritmo originario ma la maggior parte di loro perde la proprietà di convergere a punti che soddisfano le *condizioni necessarie del secondo ordine* (cioè convergono ai punti stazionari).

### 2.4.4.3 Metodo di Newton “inesatto”

Nel metodo di Newton per determinare la direzione di ricerca è necessario risolvere esattamente ad ogni iterazione il seguente sistema lineare

$$\nabla^2 f(x_k)d + \nabla f(x_k) = 0. \quad (2.35)$$

Per problemi di *dimensioni elevate* la risoluzione esatta del sistema (2.35) può essere *troppo oneroso* se non addirittura praticamente impossibile. Quindi un problema importante da affrontare è quello di capire se è possibile, risolvendo *approssimativamente* il sistema (2.35), ottenere una direzione che permetta di definire un algoritmo di minimizzazione con una buona rapidità di convergenza. Una interessante risposta a questa questione è data dalla seguente proposizione.

**Proposizione 2.4.18** *Sia  $f$  due volte continuamente differenziabile e sia  $\{x_k\}$  la successione generata dall'iterazione*

$$x_{k+1} = x_k + d_k, \quad k = 0, 1, \dots,$$

dove  $d_k$  è una soluzione approssimata del sistema lineare (2.35), cioè un vettore che soddisfa la (2.35) a meno di un residuo  $r_k$  dato da:

$$r_k = \nabla^2 f(x_k)d + \nabla f(x_k).$$

Se la successione di punti prodotta dall'algoritmo converge ad un punto  $x^*$ , se sono soddisfatte le ipotesi della Proposizione 2.4.14 e se

$$\lim_{k \rightarrow \infty} \frac{\|r_k\|}{\|\nabla f(x_k)\|} = 0,$$

allora la successione  $\{x_k\}$  converge superlinearmente ad  $x^*$ .

Il risultato precedente mostra che, se nel calcolare la direzione di Newton si commette un errore che tende a zero “abbastanza” velocemente all'avvicinarsi ad un punto stazionario, si continua a mantenere una buona rapidità di convergenza. Infatti la rapidità di convergenza superlineare, sebbene inferiore a quella quadratica, assicura comunque un comportamento molto soddisfacente dal punto di vista pratico.

La Proposizione 2.4.18 è la base per definire quella classe di modifiche globalmente convergenti del metodo di Newton dette *metodi di Newton troncato*. Tali metodi permettono di affrontare e risolvere efficientemente problemi a grandi dimensioni e saranno brevemente descritti in seguito nella sezione riguardante i metodi delle direzioni coniugate.

#### 2.4.4.4. Metodi di globalizzazione non monotoni

Recentemente un nuovo approccio per definire delle nuove modifiche globalmente convergenti del metodo di Newton è stato proposto in letteratura. Questo nuovo filone è nato dall'osservazione il metodo di Newton nella sua forma pura (2.4.2) è frequentemente molto più efficiente delle varie modifiche proposte. Per esempio si può considerare la seguente funzione

$$f(x) = 10^8(x_2 - x_1^2)^2 + (1 - x_1)^2,$$

e come punto di partenza  $x_0 = (-1.2, 1)$ . In questo caso il metodo di Newton puro converge in 3 iterazioni mentre una delle più efficienti modifiche convergenti del metodo di Newton (implementata nella routine E04LBF della libreria NAG) converge in 1631 iterazioni (e 2447 calcoli di funzione).

Una della cause di questa differenza di efficienza fra il metodo di Newton puro e le sue modifiche globalmente convergenti è il fatto che quest'ultime producono una successione di punti a cui corrispondono necessariamente valori *monotonicamente decrescenti* della funzione obiettivo. Questa diminuzione del valore della funzione obiettivo ad ogni iterazione è ottenuta

- controllando il passo  $\alpha_k$  lungo la direzione di ricerca nell'approccio che utilizza le ricerche unidimensionali;
- controllando il raggio  $a_k$  della regione di confidenza nell'approccio trust region.

In entrambi i casi si è spesso (specialmente quando si vuole minimizzare funzioni molto non lineari) costretti ad scegliere valori "piccoli" per il passo  $\alpha_k$  o per il raggio  $a_k$  e, quindi, ad effettuare spostamenti alquanto diversi da quelli che farebbe il metodo di Newton puro. Questo fenomeno porta sicuramente a conseguenze negative in un intorno di un punto stazionario che soddisfa le ipotesi della Proposizione 2.4.14 Infatti questa proposizione assicura che esiste un intorno (di cui non si sa a priori l'estensione e che non è detto che sia piccolo) in cui il metodo di Newton puro converge ed ha una rapidità di convergenza quadratica. Perciò, in questo intorno, l'imporre una decrescita monotona della funzione obiettivo può essere una richiesta contrastante con l'efficienza del metodo, perché potrebbe portare a non effettuare lo stesso spostamento del metodo di Newton puro.

Per superare le difficoltà descritte, recentemente sono stati proposti degli algoritmi che, pur assicurando la convergenza globale, evitano però di richiedere che i valori della funzione obiettivo diminuiscano ad ogni iterazione. La maggior parte di questi nuovi algoritmi ricadono nella classe delle modifiche globalmente convergenti di Newton che utilizzano le ricerche unidimensionali.

Uno dei primi metodi proposti si basa su un criterio accettazione del passo  $\alpha_k$  che permette che la funzione possa aumentare durante un numero di passi prefissato. In particolare si può definire il seguente algoritmo.

#### Criterio di Armijo non monotono



- Dati  $a > 0$ ,  $\delta \in (0, 1)$ ,  $\gamma \in (0, 1/2)$  ed un intero  $M$ .
- Si sceglie  $\alpha_k = \delta^h a$  dove  $h$  è il più piccolo intero nonnegativo tale che:

$$f(x_k + \alpha_k d_k) \leq \max_{0 \leq j \leq M} [f(x_{k-j})] + \gamma \alpha_k \nabla f(x_k)' d_k.$$

Si può notare che il precedente criterio consente che  $f(x_{k+1})$  possa essere maggiore di  $f(x_k)$  e, perciò, che possa essere accettato più facilmente il passo unitario lungo la direzione di Newton.

Si può dimostrare che se si utilizza il precedente criterio di Armijo non monotono (con  $a = 1$ ) con una direzione di ricerca che soddisfa le condizioni (iii) di Proposizione 2.4.3 e (ii) di Proposizione 2.4.5 si ottiene un metodo che è una modifica globalmente convergente del metodo di Newton.

Benché l'utilizzazione di una ricerca unidimensionale di tipo nonmonotono in una modifica del metodo di Newton abbia fatto migliorare i risultati numerici rispetto ai metodi monotoni, si è notato che ulteriori miglioramenti possono essere ottenuti “concedendo” ancora più libertà al metodo di Newton. Per esempio ottimi risultati possono essere ottenuti se non si effettuano controlli sul valore della funzione obiettivo per un numero prefissato di iterazioni, quindi scegliendo sempre il passo unitario in queste iterazioni. Questo comportamento sembra indicare che è utile cercare di utilizzare il più possibile il metodo di Newton puro anche se non si è nella sua regione di convergenza. Una possibile spiegazione di questo fatto è che, in generale, la direzione di ricerca contiene *più informazioni* sul problema rispetto ai criteri di scelta dello spostamento. Infatti:

- la direzione di Newton utilizza  $\nabla^2 f(x_k)$  e  $\nabla f(x_k)$  che sono equivalenti a  $n(n+3)/2$  calcoli di funzioni;
- le tecniche che determinano gli spostamenti  $\alpha_k$  lungo la direzione di ricerca si basano su singoli calcoli di funzione.

Però, dall'altra parte, una scelta “sbagliata” del passo  $\alpha_k$  influenza pesantemente l'entità dello spostamento effettuato dall'algoritmo.

Quindi in base alle precedenti considerazioni, una possibilità per sviluppare dei nuovi algoritmi di ottimizzazione più efficienti è quella di proporre dei metodi di globalizzazione che determinino il passo  $\alpha_k$  cercando di sfruttare più informazioni sul problema. Utilizzando non solamente i valori della funzione obiettivo ma anche altre grandezze che possano caratterizzare il comportamento dell'algoritmo.

### 2.4.5 Metodi Quasi-Newton

I *metodi Quasi-Newton* nascono dall'esigenza di definire dei metodi efficienti di ottimizzazione non vincolata senza richiedere la conoscenza delle derivate seconde della funzione obiettivo. Tali metodi infatti presentano le seguenti caratteristiche:

- utilizzano soltanto la conoscenza delle derivate prime;
- rappresentano delle approssimazioni del metodo di Newton che conservano (sotto appropriate ipotesi) una rapidità di convergenza superlineare.

Da quanto visto nella sezione precedente, il metodo di Newton è dato dalla seguente iterazione:

$$x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k).$$

Un iterazione di un generico metodo Quasi-Newton è descritta da:

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k), \quad (2.36)$$

in cui  $B_k$  è una matrice aggiornata iterativamente ed  $\alpha_k$  è, al solito, lo spostamento lungo la direzione di ricerca. Affinché un metodo descritto dall'iterazione (2.36) possa ereditare l'ottima rapidità di convergenza del metodo di Newton è naturale richiedere che la matrice  $B_k$  approssimi (in un qualche senso) la matrice Hessiana.

La scelta più immediata è quella di richiedere che le matrici  $B_k$  siano delle *approssimazioni consistenti* della matrice Hessiana, cioè che abbiano la proprietà che, se la successione di punti  $\{x_k\}$  converge ad un punto stazionario  $x^*$  (con la matrice Hessiana non singolare), allora:

$$\lim_{k \rightarrow \infty} \|B_k - \nabla^2 f(x^*)\| = 0.$$

I metodi del tipo (2.36) che utilizzano matrici che godono della precedente proprietà sono detti *metodi tipo-Newton*.

La richiesta che la successione di matrici  $\{B_k\}$  siano delle approssimazioni consistenti della matrice Hessiana può essere alquanto pesante e, ai fini di mantenere una rapidità di convergenza superlineare, può essere indebolita in maniera significativa. Infatti ci si può ispirare al seguente risultato.

**Proposizione 2.4.19** *Sia  $f(x)$  due volte continuamente differenziabile, sia  $\{B_k\}$  una successione di matrici non singolari, sia  $\{x_k\}$  data da:*

$$x_{k+1} = x_k - B_k^{-1} \nabla f(x_k)$$

*e sia  $\{x_k\}$  convergente al punto  $x^*$  dove  $\nabla^2 f(x^*)$  è non singolare. Allora  $\{x_k\}$  converge superlinearmente ad  $x^*$  e risulta  $\nabla f(x^*) = 0$  se e solo se:*

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(x^*)) (x_{k+1} - x_k)\|}{\|x_{k+1} - x_k\|} = 0.$$

Quindi, il risultato precedente indica che, al fine di avere una buona rapidità di convergenza, una matrice  $\tilde{B}$  può essere considerata una buona "approssimazione" di  $\nabla^2 f$  se

$$\tilde{B}(x_{k+1} - x_k) \approx \nabla^2 f(x_k)(x_{k+1} - x_k),$$

che è una relazione vettoriale e, perciò, molto più debole della richiesta che  $\tilde{B} \approx \nabla^2 f(x_k)$  che è una relazione matriciale.

Ricordando i teoremi della media, si ha che se il punto  $x_{k+1}$  è sufficientemente vicino a  $x_k$  allora:

$$\nabla^2 f(x_k)(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k),$$

Quindi una matrice  $\tilde{B}$  può essere considerata una “approssimazione” di  $\nabla^2 f$  se soddisfa la condizione:

$$\tilde{B}(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k).$$

In linea con la precedente considerazione, i metodi Quasi-Newton producono ad ogni iterazione una matrice  $B_{k+1}$  che soddisfa la seguente *equazione di Quasi-Newton*:

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k). \quad (2.37)$$

Un modo equivalente per giustificare il fatto di richiedere le matrici prodotte soddisfino l'equazione Quasi-Newton (2.37) è quello di far riferimento nuovamente al modello quadratico. Infatti è facile verificare che se la funzione obiettivo fosse una funzione quadratica, con matrice Hessiana  $Q$ , si avrebbe:

$$Q(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k).$$

Quindi l'imporre l'equazione Quasi-Newton, equivale a richiedere che la matrice prodotta soddisfi una relazione verificata dalla matrice Hessiana nel caso in cui la funzione obiettivo fosse quadratica.

Come accennato nei metodi Quasi-Newton viene prodotta una sequenza di matrici  $\{B_k\}$ . In particolare, alla generica  $k$ -esima iterazione, la matrice  $B_{k+1}$  viene determinata aggiornando  $B_k$  con dei termini di correzione in modo tale che la matrice ottenuta soddisfi l'equazione Quasi-Newton riferita ai punti  $x_{k+1}$  e  $x_k$ , cioè:

$$\begin{aligned} B_{k+1} &= B_k + \Delta B_k, \\ \gamma_k &= (B_k + \Delta B_k) \delta_k. \end{aligned} \quad (2.38)$$

dove, per semplicità si pone:

$$\delta_k = x_{k+1} - x_k, \quad \gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k).$$

Le formule del tipo (2.38), in cui si cerca di approssimare la matrice Hessiana, vengono dette *formule dirette*.

Un approccio alternativo nell'ambito dei metodi Quasi-Newton è quello di produrre una matrice  $H_{k+1}$  che cerchi di approssimare l'inversa della matrice Hessiana. In questo caso la formula Quasi-Newton da soddisfare è la seguente:

$$H_{k+1}(\nabla f(x_{k+1}) - \nabla f(x_k)) = x_{k+1} - x_k.$$

Coerentemente, l'iterazione di un metodo Quasi-Newton che segue questa impostazione è definita da

$$x_{k+1} = x_k - \alpha_k H_k \nabla f(x_k),$$

e la matrice  $H_{k+1}$  è ottenuta utilizzando le seguenti formule, dette *formule inverse*:

$$\begin{aligned} H_{k+1} &= H_k + \Delta H_k, \\ (H_k + \Delta H_k) \gamma_k &= \delta_k. \end{aligned}$$

I vari metodi Quasi-Newton finora proposti differiscono fra loro essenzialmente per le formule usate nella definizione della matrici di aggiornamento  $\Delta B_k$  o  $\Delta H_k$ . La loro caratteristica comune è che sia  $\Delta B_k$  che  $\Delta H_k$  vengono ottenuti con la sola valutazione del gradiente nel punto  $x_{k+1}$  e quindi senza un eccessivo aumento del costo computazionale rispetto ad un qualsiasi algoritmo tipo gradiente.

Un esempio di una formula di aggiornamento diretta particolarmente utilizzata (detta *formula BFGS*) è la seguente:

$$B_{k+1} = B_k + \frac{\gamma_k \gamma_k^T}{\delta_k^T \gamma_k} - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k},$$

mentre, riguardo le formule di aggiornamento inverse, un esempio particolarmente significativo è il seguente:

$$H_{k+1} = H_k + \frac{\delta_k \delta_k^T}{\delta_k^T \gamma_k} - \frac{H_k \gamma_k \gamma_k^T H_k}{\gamma_k^T H_k \gamma_k}.$$

Per quanto riguarda la scelta del passo  $\alpha_k$ , motivazioni di sia di tipo teorico che di tipo pratico indicano che, nei metodi Quasi-Newton, la tecnica più adeguata è quella di calcolare il passo in modo da soddisfare le condizioni di Wolfe (si veda la Sezione 2.4.2).

### - Metodi Quasi-Newton a memoria limitata

I metodi Quasi-Newton a memoria limitata sono stati proposti per estendere l'approccio dei metodi Quasi-Newton anche a problemi a grandi dimensioni. L'idea di questi nuovi metodi è quella di costruire una direzione che approssimi la direzione di Newton utilizzando solamente le informazioni ottenute nelle ultime iterazioni. Tali informazioni vengono memorizzate utilizzando un numero limitato di vettori.

Un metodo a memoria limitata, che si è rivelato particolarmente efficiente, è quello che viene chiamato *metodo BFGS a memoria limitata*. Questo metodo è molto simile al metodo BFGS standard, la sola differenza è nella matrice di aggiornamento. Infatti, se si sceglie  $B_0 = I$ , al posto di

$$B_k = I + \sum_{i=0}^{k-1} \left[ \frac{\gamma_i \gamma_i'}{\delta_i' \gamma_i} - \frac{B_i \delta_i \delta_i' B_i}{\delta_i' B_i \delta_i} \right],$$

si utilizza (per  $k \geq m + 1$ )

$$\tilde{B}_k = I + \sum_{i=k-m-1}^{k-1} \left[ \frac{\gamma_i \gamma_i'}{\delta_i' \gamma_i} - \frac{B_i \delta_i \delta_i' B_i}{\delta_i' B_i \delta_i} \right].$$

la direzione  $d_k = -\tilde{B}_k^{-1}\nabla f(x_k)$  si può calcolare eseguendo solamente pochi prodotti scalari.

I risultati numerici ottenuti con questi metodi hanno mostrato che è sufficiente scegliere  $m = 5$  per ottenere risultati comparabili con il metodo BFGS completo. Un tale valore per  $m$  permette di risolvere efficientemente anche problemi a grosse dimensioni. Inoltre l'esperienza numerica sembra indicare che, sorprendentemente, il comportamento di questo metodo non migliora in maniera significativa per valori più grandi della memoria  $m$ . Tale comportamento continua a presentarsi anche al crescere delle dimensioni dei problemi da minimizzare.

Per quanto riguarda le proprietà teoriche del metodo BFGS a memoria limitata, si è provato che questo metodo è globalmente e linearmente convergente per funzioni convesse.

## 2.4.6 Metodo del gradiente coniugato

Il metodo del gradiente coniugato è stato studiato e sviluppato per affrontare principalmente problemi di ottimizzazione non vincolata a grosse dimensioni. Per tali problemi non è ammissibile far uso di operazioni matriciali e richiedere la conoscenza della matrice Hessiana. Inoltre è necessario ridurre il più possibile il numero di vettori utilizzati dall'algoritmo per limitare il più possibile l'occupazione di memoria del calcolatore. Quindi il metodo maggiormente applicabile per questa classe di problemi sembrerebbe essere il metodo del gradiente. Purtroppo, per quanto detto nella sezione 2.4.3, le versioni tradizionali di tale metodo possono essere alquanto inefficienti e quindi non utilizzabili in pratica per risolvere problemi di ottimizzazione non vincolata a grosse dimensioni.

Per cercare di avere una efficienza migliore rispetto al metodo del gradiente tradizionale, il metodo di tipo gradiente coniugato prendono come punto di partenza il fatto di approssimare una generica funzione obiettivo  $f(x_k + d)$  con il suo modello quadratico  $\phi_k(d)$  (definito dalla (2.29) in Sezione 2.4.4) e di cercare di minimizzare questa funzione quadratica senza dover fare l'inversione della matrice Hessiana, come nel caso del metodo di Newton.

Per affrontare questa sfida uno strumento essenziale sono le *direzioni coniugate* rispetto alla matrice Hessiana di  $f(x)$ .

### - Direzioni coniugate

**Definizione 2.4.20** *Assegnata una matrice  $Q$  simmetrica, due vettori non nulli  $d_i, d_j \in \mathbb{R}^n$  si dicono coniugati rispetto a  $Q$  (oppure  $Q$ -coniugati) se risulta:*

$$d_i^T Q d_j = 0.$$

Dalla precedente definizione segue immediatamente la seguente proposizione.

**Proposizione 2.4.21** *Siano  $d_0, \dots, d_m \in R^n$  vettori non nulli mutuamente coniugati rispetto ad una matrice  $Q$  simmetrica definita positiva. Allora  $d_0, \dots, d_m$  sono linearmente indipendenti.*

Dopo aver introdotto le direzioni coniugate si può ritornare al problema di voler minimizzare, senza fare inversioni di matrici, una funzione quadratica del tipo:

$$f(x) = \frac{1}{2}x^T Qx + c^T x, \quad (2.39)$$

in cui  $Q$  è una matrice simmetrica definita positiva.

Se si conoscono  $n$  direzioni  $d_0, \dots, d_{n-1}$  coniugate rispetto a  $Q$ , si può effettuare la seguente trasformazione dello spazio delle variabili:

$$x = \sum_{i=0}^{n-1} \alpha_i d_i.$$

Nel nuovo spazio la funzione obiettivo assume la forma:

$$\tilde{f}(\alpha) = \sum_{i=0}^{n-1} \left[ \frac{1}{2} \alpha_i^2 d_i^T Q d_i + \alpha_i d_i^T c \right] = \sum_{i=0}^{n-1} \tilde{f}_i(\alpha_i),$$

cioè diventa la somma di  $n$  funzioni quadratiche ciascuna di una sola variabile scalare  $\alpha_i$ .

Quindi utilizzando  $n$  direzioni coniugate si può trasformare un problema di ottimizzazione su  $R^n$  in  $n$  problemi separati di ottimizzazione su  $R$ . Questo fa capire che, utilizzando le direzioni coniugate, si può determinare il minimo di una funzione quadratica strettamente convessa in un numero finito di iterazioni. Più formalmente si ha il seguente risultato.

**Proposizione 2.4.22** *Sia  $f(x)$  data dalla (2.39) e sia  $d_0, d_1, \dots, d_{n-1}$  un insieme di vettori non nulli e mutuamente coniugati rispetto a  $Q$ . Si definisca l'algoritmo*

$$x_{k+1} = x_k + \alpha_k d_k,$$

in cui  $x_0 \in R^n$  è un punto iniziale arbitrario ed  $\alpha_k$  è scelto in modo da minimizzare  $f(x)$  lungo  $d_k$ , ossia:

$$\alpha_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T Q d_k} = -\frac{(Qx_k + c)^T d_k}{d_k^T Q d_k}.$$

Allora esiste  $m \leq n - 1$  tale che  $x_{m+1}$  coincide con il punto di minimo  $x^*$  di  $f(x)$ .

## - Metodo del gradiente coniugato nel caso quadratico

Da quanto visto nella sezione precedente una funzione quadratica strettamente convessa può essere minimizzata molto efficientemente se si conoscono  $n$  direzioni coniugate rispetto alla sua matrice Hessiana. Quindi il problema si sposta a quello di trovare in maniera “semplice” delle direzioni coniugate.

Questo problema, sempre nel caso di funzioni quadratiche, può essere risolto utilizzando *il metodo del gradiente coniugato* che fornisce, attraverso un processo iterativo, delle direzioni coniugate.

Sia  $x_0 \in R^n$  un punto arbitrario allora ogni iterazione del metodo del gradiente coniugato è data da:

$$d_k = \begin{cases} -\nabla f(x_k), & \text{se } k = 0, \\ -\nabla f(x_k) + \beta_k d_{k-1}, & \text{se } k \geq 1; \end{cases}$$

$$\beta_k = \frac{\nabla f(x_k)^T Q d_{k-1}}{d_{k-1}^T Q d_{k-1}}$$

$$x_{k+1} = x_k + \alpha_k d_k;$$

$$\alpha_k = -\frac{\nabla f(x_k)^T d_k}{d_k^T Q d_k}$$

Per tale metodo si può provare il seguente risultato.

**Proposizione 2.4.23** *Sia data la funzione*

$$f(x) = \frac{1}{2} x^T Q x + c^T x.$$

- (a) *Se la matrice  $Q$  è definita positiva allora l'algoritmo del gradiente coniugato produce delle direzioni  $d_0, \dots, d_k$  che sono  $Q$ -coniugate e determina, in al più  $n$  iterazioni, il punto di minimo  $x^*$  della funzione  $f(x)$ .*
- (b) *Se la matrice  $Q$  è semidefinita positiva e se la funzione  $f(x)$  ha un punto di minimo allora l'algoritmo del gradiente coniugato produce delle direzioni  $d_0, \dots, d_k$  che sono  $Q$ -coniugate e determina, in al più  $n - 1$  iterazioni, un punto di minimo  $x^*$  della funzione  $f(x)$ .*

## - Metodo del gradiente coniugato nel caso non quadratico

Il metodo del gradiente coniugato, come descritto nel paragrafo precedente, permette di trovare il minimo di una funzione quadratica strettamente convessa senza effettuare inversioni di matrici. Se si vuole utilizzare questo metodo per minimizzare funzioni generali a grandi dimensioni, bisogna risolvere i seguenti problemi:

- il calcolo del passo  $\alpha_k$ : in generale non è possibile nel caso di funzioni non quadratiche calcolare il passo  $\alpha_k$  minimizzando esattamente la funzione obiettivo.
- Il calcolo dello scalare  $\beta_k$ : infatti nella espressione di  $\beta_k$  compare esplicitamente la matrice Hessiana della funzione obiettivo e quindi si richiedono operazioni matriciali che limitano fortemente le dimensioni del problema da minimizzare.

Il primo dei due ostacoli viene superato determinando il passo  $\alpha_k$  utilizzando uno degli algoritmi di ricerca unidimensionale viste nella sezione 2.4.2, in particolare quello che si avvicina di più ad una minimizzazione esatta lungo la direzione di ricerca.

Per quanto riguarda il secondo ostacolo da risolvere, si può notare che, nel caso quadratico, si possono ricavare diverse espressioni di  $\beta_k$  che sono tutte fra di loro equivalenti. In alcune di queste non compare più la matrice Hessiana della funzione obiettivo e, quindi, possono essere utilizzate nel caso di problemi a grandi dimensioni. Tuttavia è da notare che queste espressioni diverse di  $\beta_k$  sono equivalenti solo nel caso quadratico e, perciò, danno luogo ad algoritmi diversi per il calcolo della direzione nel caso non quadratico. In definitiva un esempio di un metodo del gradiente coniugato nel caso non quadratico può essere descritto nella seguente maniera.

- dato un qualsiasi punto iniziale  $x_0 \in R^n$  allora per ogni iterazione  $k$  si pone:

$$d_k = \begin{cases} -\nabla f(x_k), & \text{se } k = 0, \\ -\nabla f(x_k) + \beta_k d_{k-1}, & \text{se } k \geq 1; \end{cases}$$

$$x_{k+1} = x_k + \alpha_k d_k;$$

- dove

- lo scalare  $\alpha_k$  viene usualmente calcolato utilizzando le condizioni forti di Wolfe (si veda la Sezione 2.4.2) oppure altre condizioni che tendono ad assicurare buone proprietà alla successiva direzione;
- lo scalare  $\beta_k$  è dato normalmente da una delle seguenti due formule (equivalenti nel caso di funzioni quadratiche):

$$\beta_k = \frac{\|\nabla f(x_k)\|^2}{\|\nabla f(x_{k-1})\|^2}, \quad \beta_k = \frac{\nabla f(x_k)^T (\nabla f(x_k) - \nabla f(x_{k-1}))}{\|\nabla f(x_{k-1})\|^2}.$$



Grazie alla utilizzazione di particolari tecniche di per il calcolo del passo  $\alpha_k$  si riesce a garantire ai metodi del gradiente coniugato soddisfacenti proprietà di convergenza globale.

Per quanto riguarda la rapidità di convergenza di questi metodi, la maggior parte di questi dei risultati proposti riguardano il caso in cui le ricerche unidimensionali sono esatte. Il risultato più significativo stabilisce che, in un intorno di un punto stazionario che soddisfa le condizioni della Proposizione 2.4.14, la sequenza di punti prodotta soddisfa la seguente relazione:

$$\|x_{k+n} - x^*\| = O(\|x_k - x^*\|^2).$$

Cioè, al meglio, questi metodi presentano una rapidità di convergenza quadratica ogni  $n$  passi. Questo riflette il fatto che  $n$  iterazioni di un metodo del gradiente coniugato permettono di minimizzare una funzione quadratica strettamente convessa e, quindi, equivalgono approssimativamente ad una iterazione del metodo di Newton.

## 2.4.7 Metodi di Newton troncato

I metodi di Newton Troncato appartengono alla classe delle modifiche globalmente convergenti del metodo di Newton e sono particolarmente adatti per affrontare problemi a grandi dimensioni.

Tali metodi si basano sul fatto che, se la matrice Hessiana  $\nabla^2 f(x_k)$  è definita positiva allora, determinare la direzione di Newton equivale a minimizzare la funzione quadratica:

$$\phi_k(d) = f(x_k) + \nabla f(x_k)'d + \frac{1}{2}d^T \nabla^2 f(x_k)d$$

cioè a trovare una direzione tale che

$$\nabla \phi_k(d) = \nabla^2 f(x_k)d + \nabla f(x_k) = 0. \quad (2.40)$$

La cosa importante è che la Proposizione 2.4.18 indica che una “buona” direzione  $d_k$  può essere ottenuta minimizzando in maniera approssimata la funzione  $\phi_k(d)$ . Infatti, per avere un algoritmo superlinearmente convergente, è sufficiente ottenere una direzione  $d_k$  tale che:

$$\|\nabla \phi_k(d_k)\| \leq \eta_k \|\nabla f(x_k)\| \quad (2.41)$$

con  $\eta_k \rightarrow 0$ . Il soddisfare la condizione (2.41) equivale al fatto che la direzione  $d_k$  soddisfa il sistema (2.40) a meno di un errore  $r_k = \|\nabla \phi_k(d_k)\|$  per cui vale:

$$\lim_{k \rightarrow \infty} \frac{\|r_k\|}{\|\nabla f(x_k)\|} = \lim_{k \rightarrow \infty} \frac{\|\nabla \phi_k(d_k)\|}{\|\nabla f(x_k)\|} = 0.$$

Nella maggior parte dei metodi di Newton troncato il calcolo della direzione  $d_k$  viene effettuato applicando il metodo del gradiente coniugato (vedere sezione 2.4.6) per minimizzare in maniera approssimata la funzione quadratica  $\phi_k(d)$ . Tale scelta presenta i seguenti vantaggi:

- in molti casi una direzione  $d_k$  che soddisfa un criterio del tipo (2.41) può essere ottenuta con questo metodo in poche iterazioni;
- se  $\|\nabla f(x_k)\|$  è grande il gradiente coniugato fornisce una  $d_k$  che tende a coincidere con l'antigradiente della funzione obiettivo, mentre se  $\|\nabla f(x_k)\|$  è piccola la  $d_k$  calcolata tende a coincidere con la direzione di Newton; quindi il metodo del gradiente coniugato fornisce una direzione che impone la convergenza globale se si è distanti da un punto stazionario, mentre, se si è vicini, produce una direzione che assicura una buona rapidità di convergenza;
- se  $\nabla^2 f(x_k)$  non è definita positiva, la minimizzazione di  $\phi_k(d)$  con il gradiente coniugato usualmente si arresta fornendo una direzione di discesa a curvatura negativa.

Il metodo di Newton Troncato, proposto per primo in letteratura, può essere descritto dal seguente schema.

- dato un qualsiasi punto iniziale  $x_0 \in R^n$  allora per ogni iterazione  $k$  si pone:

$$x_{k+1} = x_k + \alpha_k d_k;$$

- dove

- lo scalare  $\alpha_k$  viene calcolato con Criterio di Armijo ponendo  $a = 1$  (vedere la Sezione 2.4.2);

- direzione  $d_k$  è calcolata dal seguente algoritmo:

P0: dati  $\varepsilon_1 > 0$   $\varepsilon_2 > 0$ , si pone  $p_0 = 0$ ,  $s_0 = -\nabla \phi_k(p_0) = -\nabla f(x_k)$  e  $i = 0$ .

Se  $s_0^T \nabla^2 f(x_k) s_0 < \varepsilon_1 \|s_0\|^2$  si pone  $d_k = -\nabla f(x_k)$  e Stop;

P1: si pone

$$p_{i+1} = p_i + \alpha_i s_i; \quad \alpha_i = -\frac{\nabla \phi_k(p_i)^T s_i}{s_i^T \nabla^2 f(x_k) s_i}$$

P2: se

$$\|\nabla \phi_k(p_{i+1})\| \leq \frac{1}{k+1} \varepsilon_2 \|\nabla f(x_k)\|$$

si pone  $d_k = p_{i+1}$  e Stop; altrimenti si pone  $i = i + 1$ ;

P3: si pone

$$s_i = -\nabla \phi_k(p_i) + \beta_i s_{i-1}, \quad \beta_i = \frac{\nabla \phi_k(p_i)^T \nabla^2 f(x_k) s_{i-1}}{s_{i-1}^T \nabla^2 f(x_k) s_{i-1}}$$

P4: se

$$s_i^T \nabla^2 f(x_k) s_i < \varepsilon_1 \|s_i\|^2$$

si pone  $d_k = p_i$  e Stop; altrimenti si torna a P1;

### 2.4.8 Metodi che non usano le derivate

Tutti i metodi descritti precedentemente si basano sulla conoscenza del gradiente della funzione obiettivo. Come visto, il gradiente della funzione obiettivo ha giocato un ruolo importante sia nell'individuazione di una direzione lungo la quale la funzione diminuisce che nella determinazione di un efficiente spostamento da effettuare lungo una direzione di discesa.

Purtroppo, varie classi di importanti problemi applicativi particolarmente complessi possono essere modellizzati come problemi di ottimizzazione in cui non sono disponibili le informazioni del primo ordine della funzione obiettivo e dei vincoli. Per esempio, nei problemi di programmazione matematica che nascono nell'ambito della progettazione ottima, non sono note le espressioni analitiche della funzione obiettivo o dei vincoli del problema. In questi casi si possono ottenere delle approssimazioni dei valori di queste funzioni attraverso dei complessi codici di simulazione o dei processi di misurazione.

La non conoscenza delle rappresentazioni analitiche delle funzioni che caratterizzano questi particolari problemi di ottimo implica l'impossibilità di calcolarne le derivate prime e, quindi, i rispettivi gradienti.

I valori delle derivate prime di una funzione possono essere approssimate con buona precisione sfruttando la seguente relazione:

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x + \eta e_i) - f(x)}{\eta} \quad (2.42)$$

dove  $e_i$  è il versore  $i$ -esimo (cioè un vettore che tutte le componenti uguali a zero con l'eccezione della  $i$ -esima che è uguale a uno) e  $\eta$  è uno scalare sufficientemente piccolo.

Tuttavia questa possibilità non può essere sfruttata nella precedente classe di problemi applicativi in quanto le valutazioni di funzione ottenute possono essere affette da errori di approssimazione o di misura. Se si indica con  $\omega_1$  e  $\omega_2$  gli errori associati al calcolo di  $f(x + \eta e_i)$  e  $f(x)$ , si ottiene la seguente relazione:

$$\frac{(f(x + \eta e_i) + \omega_1) - (f(x) + \omega_2)}{\eta} \approx \frac{\partial f(x)}{\partial x_i} + \frac{\omega_1 - \omega_2}{\eta},$$

dove l'ultimo termine può assumere valori significativi (per il fatto che  $\eta$  deve essere piccolo) e, quindi, rendere imprecisa la stima della derivata parziale di  $f$  rispetto a  $x_i$ .

Da quanto detto, risulta evidente l'importanza dal punto di vista applicativo di studiare e proporre algoritmi di ottimizzazione che usino solamente il valore della funzione obiettivo e dei vincoli. Per poter avere buone proprietà teoriche, tali metodi devono essere in grado di sostituire le informazioni derivanti dalla conoscenza del gradiente che guidano la convergenza della sequenza dei punti prodotta da un algoritmo di ottimizzazione.

Le componenti del gradiente forniscono una importante misura della sensibilità della funzione al variare delle corrispondenti variabili (si veda l'equazione (2.42)). Grazie a

questa caratteristica il gradiente di una funzione permette di determinare efficientemente delle direzioni di ricerca.

Gli algoritmi che non usano le derivate determinano delle direzioni di discesa valutando la funzione obiettivo su particolari insiemi di direzioni. Molto frequentemente gli insiemi di direzioni usate da tali algoritmi soddisfano la seguente assunzione.

**Assunzione 2.4.1** *L'insieme di direzioni  $d_1, d_2, \dots, d_q$ , con  $\|d_i\| = 1$  per ogni  $i = 1, \dots, q$ , è tale che per ogni  $d \in R^n$  esistono degli scalari  $\beta_1, \beta_2, \dots, \beta_q \in R_+$  tali che*

$$d = \sum_{i=1}^q \beta_i d_i.$$

Esempi di direzioni che soddisfano la precedente assunzione sono i seguenti.

$$\begin{aligned} d_1 = e_1, \quad d_2 = -e_1, \quad d_3 = e_2, \quad d_4 = -e_2, \quad \dots, \quad d_{2n-1} = e_n, \quad d_{2n} = -e_n; \\ d_1 = e_1, \quad d_2 = e_2, \quad d_3 = e_3, \quad \dots, \quad d_n = e_n, \quad d_{n+1} = -\frac{e}{\|e\|}, \end{aligned}$$

dove  $e \in R^n$  è un vettore le cui componenti sono uguali a uno.

L'interesse di insieme di direzioni che soddisfano la precedente assunzione è descritto nella seguente proposizione.

**Proposizione 2.4.24** *Sia  $d_1, d_2, \dots, d_q$  un insieme di direzioni che soddisfa l'Assunzione 2.4.1 allora  $\bar{x}$  non è un punto stazionario se e solamente se esiste almeno una direzione  $d_h$  appartenente all'insieme considerato tale che*

$$\nabla f(\bar{x})^T d_h < 0.$$

La precedente proposizione che garantisce che, in corrispondenza di un punto non stazionario, almeno una delle direzioni di un insieme che soddisfa l'Assunzione 2.4.1 è di discesa.

Identificata una direzione di discesa si deve affrontare il problema della scelta di un passo efficiente  $\alpha_k$  da effettuare lungo questa direzione. Le tecniche di linesearch viste precedentemente sfruttavano la conoscenza del gradiente. Tuttavia è possibile definire tecniche con proprietà teoriche simili che utilizzano solamente i valori della funzione obiettivo.

Un esempio di un criterio di questo tipo è costituito dalla seguente versione del criterio Armijo che non utilizza il gradiente.

*Criterio di Armijo senza derivate*

- Dati  $x \in R^n$ ,  $d \in R^n$  con  $\|d\| = 1$ ,  $\gamma \in (0, 1)$ .

- Si sceglie  $\alpha > 0$  tale che:

$$\begin{aligned} f(x + \alpha d) &\leq f(x) - \gamma \alpha^2 \\ f(x + 2\alpha d) &> f(x) - \gamma (2\alpha)^2 \end{aligned}$$

A questo punto si sono introdotti tutti gli strumenti per riportare un esempio di un algoritmo di minimizzazione che non usa le derivate.

**Algoritmo che non use le derivate.**

*Dati:*  $x_0 \in R^n$ ,  $d_1 \in R^n, \dots, d_q \in R^n$ , con  $\|d\| = 1$ ,  $\tilde{\alpha}_0^1 > 0, \dots, \tilde{\alpha}_0^q > 0$ ,  $\gamma \in (0, 1)$   
e  $k = 0$ ;

*Passo 1:* si pone  $i = 1$  e  $y_k^1 = x_k$ ;

*Passo 2:* se  $f(y_k^i + \tilde{\alpha}_k^i d_i) \leq f(y_k^i) - \gamma (\tilde{\alpha}_k^i)^2$  allora si calcola un  $\alpha_k^i$  che soddisfa il Criterio di Armijo senza derivate e si pone  $\tilde{\alpha}_{k+1}^i = \alpha_k^i$ ;  
altrimenti si pone  $\alpha_k^i = 0$  e  $\tilde{\alpha}_{k+1}^i = 0.5 \tilde{\alpha}_k^i$ ;

*Passo 3:* si pone  $y_k^{i+1} = y_k^i + \alpha_k^i d_i$ ;

*Passo 4:* se  $i < q$  si pone  $i = i + 1$  e si ritorna al Passo 2.

*Passo 5:* si determina  $x_{k+1}$  tale che  $f(x_{k+1}) \leq f(y_k^{i+1})$ , si pone  $k = k + 1$  e si ritorna al Passo 1.

Riguardo al precedente algoritmo si possono fare due precisazioni:

- l'algoritmo produce le sequenze di scalari  $\{\tilde{\alpha}_k^i\}$ , con  $i = 1, \dots, q$ , ad ogni iterazione  $k$ , lo scalare  $\tilde{\alpha}_k^i$  rappresenta il passo che viene effettuato lungo la direzione  $d_i$  nel Passo 2 per capire se tale direzione è di discesa. Questi scalari  $\{\tilde{\alpha}_k^i\}$  vengono aggiornati durante le iterazioni dell'algoritmo in modo da cercare di estrarre la variabilità delle funzione obiettivo lungo la corrispondente direzione;
- nel Passo 5 l'algoritmo accetta come  $x_{k+1}$  un qualsiasi punto ottenuto da tecniche di approssimazione varie purché produca un miglioramento della funzione obiettivo rispetto al punto  $y_k^{i+1}$  prodotto dall'algoritmo. Se nessuna tecnica di approssimazione è disponibile allora si può porre  $x_{k+1} = y_k^{i+1}$ .

**Proposizione 2.4.25** *Sia  $\{x_k\}$  la successione prodotta dall'algoritmo, si supponga che  $\mathcal{L}_0$  sia compatto. Supponiamo che l'insieme di direzioni  $d_1, d_2, \dots, d_q$  soddisfi l'Assunzione 2.4.1.*

*Allora, o esiste un indice  $\nu \geq 0$  tale che  $x_\nu \in \mathcal{L}_0$  e  $\nabla f(x_\nu) = 0$ , oppure viene prodotta una successione infinita tale che:*

- (a)  $\{x_k\}$  rimane in  $\mathcal{L}_0$  ed ammette punti di accumulazione;
- (b) la successione  $\{f(x_k)\}$  converge;
- (c)  $\lim_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$ , cioè ogni punto di accumulazione  $\bar{x}$  di  $\{x_k\}$  è tale che  $\nabla f(\bar{x}) = 0$ .

## 2.5 Metodi di ottimizzazione vincolata

### 2.5.1 Introduzione

Come già detto, se si deve risolvere un problema di ottimizzazione vincolata, la situazione è molto diversa a seconda che l'insieme ammissibile  $\mathcal{F}$  è un insieme chiuso oppure che è aperto.

Infatti se  $\mathcal{F}$  è un *insieme chiuso* può accadere che la soluzione del problema cada proprio sulla frontiera dell'insieme ammissibile. Normalmente, in questi casi, la presenza di un minimo è dovuto all'azione congiunta della funzione obiettivo ed dei vincoli rendendo assolutamente inadeguati i risultati e le tecniche proposte nel campo dell'ottimizzazione non vincolata.

Diversa è la situazione se  $\mathcal{F}$  è un *insieme aperto*. In questo caso, se esiste una soluzione questa è localmente un minimo non vincolato della funzione obiettivo e, perciò, continuano ad essere valide tutte le condizioni di ottimalità ottenute nel caso non vincolato oltre al fatto che, in un intorno della soluzione, molti degli algoritmi proposti per risolvere problemi non vincolati possono essere applicati direttamente.

Sempre nel caso in cui  $\mathcal{F}$  è aperto, il Problema (1.1) si semplifica drasticamente se si conosce un punto  $\tilde{x}$  interno all'insieme ammissibile e se è compatto l'insieme di livello

$$\mathcal{L}_{\mathcal{F}}(\tilde{x}) := \{x \in \mathcal{F} : f(x) \leq f(\tilde{x})\}.$$

In questo caso, una soluzione del problema può essere ottenuta, a partire dal punto  $\tilde{x}$ , adattando facilmente uno dei metodi o algoritmi proposti per risolvere problemi di minimizzazione non vincolata. L'unica modifica da effettuare è quella di introdurre (per esempio nelle minimizzazioni unidimensionali) dei controlli sul fatto che i punti prodotti non escano dall'insieme ammissibile.

Quindi, in conclusione, si può dire che un problema vincolato *reale* è quello in cui l'insieme ammissibile è un insieme chiuso. Invece, un problema vincolato in cui l'insieme ammissibile è un insieme aperto, si conosce un punto ammissibile ed l'insieme di livello  $\mathcal{L}_{\mathcal{F}}(\tilde{x})$  è un insieme compatto è assolutamente *equivalente* ad un problema di minimizzazione non vincolato.

Nel seguito considereremo problemi vincolati nella seguente forma:

$$\begin{aligned} \min \quad & f(x) \\ & g(x) \leq 0 \\ & h(x) = 0. \end{aligned} \tag{2.43}$$

in cui  $g : R^n \rightarrow R^m$  e  $h : R^n \rightarrow R^p$  sono vettori di funzioni assegnate.

Dato un punto ammissibile  $x$  gli indici dei vincoli attivi saranno identificati dal seguente insieme:

$$I(x) = \{i : g_i(x) = 0\}.$$

Risolvere un problema di ottimizzazione vincolata presenta un ordine di difficoltà maggiore del risolvere un problema di ottimizzazione non vincolato. Infatti, in generale, un problema vincolato può essere visto come la somma di due sottoproblemi alquanto difficili:

- quello di determinare dei punti ammissibili;
- quello di minimizzare la funzione obiettivo.

Perciò un generico algoritmo di ottimizzazione vincolata deve essere in grado di risolvere contemporaneamente i due precedenti sottoproblemi. In pratica, ad ogni iterazione, deve riuscire a bilanciare l'esigenza di far migliorare il valore della funzione obiettivo con quella di controllare la violazione dei vincoli. In particolare, trovare un punto che soddisfa determinati vincoli è equivalente ad un problema di ottimizzazione globale, cioè quello di minimizzare una funzione che pesa la violazione dei vincoli. Anche nei casi fortunati in cui si conoscano dei punti ammissibili od in cui l'insieme ammissibile abbia una struttura particolarmente semplice, la sola presenza di vincoli sulla variabili rende più difficile il problema di minimizzare la funzione obiettivo. Una conferma di questo fatto può aversi, per esempio, osservando le condizioni di ottimalità: quelle per problemi vincolati sono alquanto più complesse ed articolate di quelle per problemi non vincolati.

A causa delle difficoltà descritte, lo studio di metodi per l'ottimizzazione vincolata ha avuto uno sviluppo molto più lento e meno armonioso rispetto a quello per l'ottimizzazione non vincolata. Tuttavia, recentemente, vari algoritmi per problemi vincolati sono stati proposti in letteratura. Se si trascurano tutti quei metodi che sfruttano una qualche particolarità presente nella struttura del problema vincolato (per esempio, vincoli lineari, programmazione convessa, programmazione quadratica), nel proporre la maggior parte degli algoritmi per problemi vincolati generali, si sono seguiti tradizionalmente due approcci:

- risolvere il problema originale risolvendo una *sequenza* di problemi vincolati *più semplici*;
- *trasformare* il problema originale vincolato in una *sequenza* di problemi *non vincolati* oppure in un *singolo* problema *non vincolato*.

Nel ambito del primo approccio, i metodi i più considerati e più studiati sono *metodi di programmazione quadratica ricorsiva* (metodi RQP).

Per quanto riguarda il secondo approccio i metodi più promettenti si basano sulla possibilità di risolvere un problema vincolato attraverso una minimizzazione non vincolata di particolari funzioni continuamente differenziabile.

Nel seguito si decriverà brevemente le idee su cui si basano i metodi RQP e le funzioni di penalità.

## 2.5.2 Metodi sequenziali

Questi metodi traggono ispirazione da quelli non vincolati nel senso che cercano di determinare delle direzioni efficienti per il problema di ottimizzazione da affrontare.

Riguardo i problemi di ottimizzazione vincolati l'obiettivo è quella di determinare delle direzioni che tengano conto della necessità sia di diminuire la funzione obiettivo e sia di rispettare i vincoli.

Nel seguito di questa sezione saranno accennati due approcci che possono essere visti come l'estensioni del metodo dell'antigradiente e del metodo di Newton proposti per problemi non vincolati.

### - Metodi del primo ordine

Analogamente al caso dei metodi di ottimizzazione non vincolata, la base di questo approccio è la possibilità di dimostrare che in ogni punto, che non soddisfa le condizioni di ottimalità per problemi di ottimizzazione vincolata, si possono determinare delle direzioni lungo le quali esistono degli spostamenti che producono una diminuzione della funzione obiettivo senza perturbare eccessivamente l'ammissibilità.

In particolare, riguardo punti ammissibili che non soddisfano le condizioni di Kuhn-Tucker, si ha il seguente teorema:

**Teorema 2.5.1** *Siano  $f \in C^1(\mathbb{R}^n)$ ,  $g_i \in C^1(\mathbb{R}^n)$ , per ogni  $i = 1, \dots, m$ ,  $h_j \in C^1(\mathbb{R}^n)$ , per ogni  $j = 1, \dots, p$ . Sia  $\bar{x}$  un punto ammissibile che non è un punto di Kuhn-Tucker ed in cui sia verificata l'ipotesi di Mangasarian-Fromowitz, cioè non esistono degli scalari  $\alpha_i \geq 0$ ,  $i \in I(\bar{x})$ , e  $\beta_j$ ,  $j = 1, \dots, p$ , non tutti nulli per cui ha:*

$$\sum_{i \in I(\bar{x})} \alpha_i \nabla g_i(\bar{x}) + \sum_{j=1}^p \beta_j \nabla h_j(\bar{x}) = 0.$$

Allora si ha che i seguenti insiemi:

$$\begin{aligned} D_s(\bar{x}) &= \{d \in \mathbb{R}^n : \nabla f(\bar{x})^T d < 0\} \\ F_s(\bar{x}) &= \{d \in \mathbb{R}^n : \nabla g_i(\bar{x})^T d < 0 \quad i \in I(\bar{x})\} \\ H(\bar{x}) &= \{d \in \mathbb{R}^n : \nabla h_j(\bar{x})^T d = 0, \quad j = 1, \dots, p\}, \end{aligned}$$

hanno intersezione non vuota.

Inoltre, per ogni  $d \in D_s(\bar{x}) \cap F_s(\bar{x}) \cap H(\bar{x})$  esiste un  $\delta > 0$  tale che per ogni  $\alpha \in (0, \delta)$  si ha:

$$\begin{aligned} f(\bar{x} + \alpha d) &< f(\bar{x}), \\ g_i(\bar{x} + \alpha d) &\leq 0, \quad i = 1, \dots, m, \\ h_j(\bar{x} + \alpha d) &= r_j(\bar{x}, \alpha d) \quad j = 1, \dots, p, \end{aligned}$$

con  $r_j(\bar{x}, \alpha d) = 0$  se  $h_j(x)$  è una funzione lineare oppure  $\lim_{\alpha \rightarrow 0} r_j(\bar{x}, \alpha d)/\alpha = 0$ .



I risultati precedenti mostrano che, in corrispondenza di un punto ammissibile che non soddisfa le condizioni di ottimalità, esistono delle direzioni lungo le quali ci sono degli spostamenti che

- producono una diminuzione della funzione obiettivo,
- non violano i vincoli di disuguaglianza ed i vincoli di uguaglianza lineari,
- producono violazioni "contenute" dei vincoli di uguaglianza non lineari.

Molte classi di metodi per problemi vincolati si basano su un uso efficiente di tali direzioni che vengono determinate risolvendo sequenze di problemi del tipo:

$$\begin{aligned} \min_d \quad & \nabla f(x_k)^T d \\ & \nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad i = 1, \dots, m, \\ & \nabla h_j(x_k)^T d + h_j(x_k) = 0 \quad j = 1, \dots, p. \end{aligned} \quad (2.44)$$

#### - Metodi di programmazione quadratica ricorsiva

L'idea da cui hanno origine *i metodi di programmazione quadratica ricorsiva (RQP)* è quella di estendere al caso vincolato il metodo di Newton. Come visto precedentemente il metodo di Newton determina la direzione di ricerca minimizzando l'approssimazione quadratica della funzione obiettivo. L'estensione naturale di questo approccio è quello di determinare una direzione  $d_k$  risolvendo il seguente problema:

$$\begin{aligned} \min_d \quad & \frac{1}{2} d^T \nabla^2 f(x_k) d + \nabla f(x_k)^T d + f(x_k) \\ & \frac{1}{2} d^T \nabla^2 g_i(x_k) d + \nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad i = 1, \dots, m, \\ & \frac{1}{2} d^T \nabla^2 h_j(x_k) d + \nabla h_j(x_k)^T d + h_j(x_k) = 0 \quad j = 1, \dots, p. \end{aligned} \quad (2.45)$$

Il precedente problema, che costituisce l'approssimazione quadratica del Problema (1.3), presenta la difficoltà di avere dei vincoli quadratici e, quindi, di non poter essere risolto facilmente. Da qui la necessità di approssimare il problema originale con un problema più trattabile dal punto di vista computazionale. L'idea è stata quella di utilizzare come approssimazione del Problema (1.3) un *problema di programmazione quadratica*, cioè un problema di minimizzazione vincolata in cui la funzione obiettivo è quadratica ed i vincoli sono lineari. In particolare si può notare che, se la tripla  $(d^*, \eta^*, \rho^*)$  è un punto stazionario della funzione Lagrangiana del Problema (2.45), è anche un punto stazionario della funzione Lagrangiana del seguente problema:

$$\begin{aligned} \min_d \quad & \frac{1}{2} d^T \nabla_x^2 L(x_k, \eta^*, \rho^*) d + \nabla f(x_k)^T d + f(x_k) \\ & \nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad i = 1, \dots, m, \\ & \nabla h_j(x_k)^T d + h_j(x_k) = 0 \quad j = 1, \dots, q. \end{aligned} \quad (2.46)$$

Questo fatto suggerisce che può essere possibile ottenere una buona approssimazione del Problema (1.3) anche utilizzando un problema di programmazione quadratica, in cui i vincoli sono una approssimazione lineare dei vincoli non lineari di partenza. Questo può essere ottenuto, purché la funzione obiettivo del problema approssimante, oltre ad essere costituita dal modello quadratico della funzione obiettivo del problema originario, include anche delle informazioni del secondo ordine dei vincoli.

Seguendo quanto visto, l'approccio dei metodi di programmazione quadratica ricorsiva è quello:

- di risolvere un problema vincolato risolvendo una sequenza di *problemi di programmazione quadratica*;
- di sfruttare il fatto che, sotto opportune ipotesi, un problema di programmazione quadratica può essere *risolto efficientemente* (cioè in un numero finito di passi).

Nella sua forma originale un metodo RQP può essere descritto dal seguente schema.

*Schema originale di un algoritmo RQP*

- Siano  $x_0$ ,  $\lambda_0$  e  $\mu_0$  delle stime iniziali del punto soluzione del problema vincolato e dei suoi moltiplicatori associati.
- Alla  $k$ -esima iterazione, data la tripla  $(x_k, \lambda_k, \mu_k)$ , si trova il minimo  $d_k$  del seguente problema ed i suoi moltiplicatori associati  $(\eta_k, \rho_k)$ :

$$\begin{aligned} \min_d \quad & \frac{1}{2} d^T \nabla_x^2 L(x_k, \lambda_k, \mu_k) d + \nabla f(x_k)^T d + f(x_k) \\ & \nabla g_i(x_k)^T d + g_i(x_k) \leq 0 \quad i = 1, \dots, m, \\ & \nabla h_j(x_k)^T d + h_j(x_k) = 0 \quad j = 1, \dots, q. \end{aligned} \quad (2.47)$$

- Si pone:

$$\begin{aligned} x_{k+1} &= x_k + d_k, \\ \lambda_{k+1} &= \eta_k, \\ \mu_{k+1} &= \rho_k. \end{aligned}$$

Il sottoproblema (2.47) deriva dal Problema (2.46) in cui si sono sostituiti nella funzione obiettivo i moltiplicatori ottimi  $(\eta^*, \rho^*)$  del Problema (2.45) con le stime correnti  $(\lambda_k, \mu_k)$ . Nonostante questa ulteriore approssimazione il sottoproblema (2.47) continua ad avere con il problema originale dei forti legami. Infatti si può dimostrare che:

- l'insieme ammissibile del sottoproblema quadratico (QP<sub>k</sub>) è un'approssimazione lineare di quello del problema originale;

- la funzione Lagrangiana del sottoproblema quadratico  $L_Q(d, \eta, \rho)$  è un'approssimazione quadratica della funzione Lagrangiana del Problema (1.3), infatti, se si pone  $d_\eta := \eta - \lambda_k$  e  $d_\rho := \rho - \mu_k$ , si ha:

$$L_Q(d, d_\eta, d_\rho) = L(x_k, \lambda_k, \mu_k) + \nabla L(x_k, \lambda_k, \mu_k)^T (d, d_\eta, d_\rho) \quad (2.48) \\ + \frac{1}{2} (d, d_\eta, d_\rho)^T \nabla^2 L(x_k, \lambda_k, \mu_k) (d, d_\eta, d_\rho).$$

Perciò, in base ai precedenti legami, un'iterazione del metodo RQP equivale a

- effettuare un passo del *metodo di Newton* per la funzione Lagrangiana;
- forzare l'*ammissibilità* rispetto alla linearizzazione dell'insieme ammissibile.

Infatti determinare una tripla di Kuhn-Tucker del sottoproblema (2.47) equivale, da una parte, a trovare un punto stazionario del modello quadratico (2.48) della funzione Lagrangiana del Problema (1.3), e, dall'altra, a determinare una direzione di ricerca che soddisfa le approssimazioni lineari dei vincoli.

### - Convergenza locale

Come visto precedentemente, i metodi di programmazione quadratica ricorsiva sono stati proposti cercando di estendere l'approccio del metodo di Newton al caso di problemi di minimizzazione vincolata. Il frutto di questo tentativo è il fatto che questi metodi presentano proprietà di convergenza locale simili a quelle del metodo di Newton per problemi non vincolati. Uno dei risultati che riguardano le proprietà locali di questi metodi è descritto dalla seguente proposizione.

**Proposizione 2.5.2** *Siano  $f, g_i, i = 1, \dots, m$  e  $h_j, j = 1, \dots, q$  delle funzioni due volte continuamente differenziabili. Siano le matrici Hessiane  $\nabla^2 f, \nabla^2 g_i, i = 1, \dots, m$ , e  $\nabla^2 h_j, j = 1, \dots, q$ , Lipschitz continue. Sia  $(x^*, \lambda^*, \mu^*)$  una tripla di Kuhn-Tucker in cui sono verificate le seguenti ipotesi:*

- i) i gradienti dei vincoli attivi sono linearmente indipendenti;*
- ii) sono soddisfatte le condizioni sufficienti del secondo ordine;*
- iii) è soddisfatta la stretta complementarità.*

*Allora esiste un intorno aperto  $\mathcal{D}$  di  $(x^*, \lambda^*, \mu^*)$  tale che, se  $(x_0, \lambda_0, \mu_0) \in \mathcal{D}$ , la successione  $\{x_k, \lambda_k, \mu_k\}$  prodotta dal algoritmo RQP precedentemente descritto è ben definita (cioè i sottoproblemi (2.47) ammettono soluzione) e converge a  $(x^*, \lambda^*, \mu^*)$  con rapidità di convergenza quadratica.*

### - Convergenza globale

La precedente proposizione mostra che i metodi di programmazione quadratica ricorsiva, sotto opportune ipotesi, hanno buone proprietà di convergenza in un intorno di un minimo del problema vincolato. Per quanto riguarda l'utilizzazione pratica di questi algoritmi bisogna tener conto che, quando non si conosce un punto di partenza  $x_0$  "sufficientemente" vicino ad un minimo, si devono affrontare le seguenti difficoltà:

- l'algoritmo può non essere definito, cioè il sottoproblema quadratico può *non ammettere soluzione*;
- la sequenza di punti  $\{x_k\}$  prodotta può *non convergere*.

Per quanto riguarda il primo dei due problemi, una possibile causa può essere il fatto che la funzione obiettivo del sottoproblema (2.47) è *illimitata inferiormente* sull'insieme ammissibile linearizzato. Questa situazione può essere evitata utilizzando un'approssimazione  $B_k$  della matrice Hessiana  $\nabla_x^2 L(x_k, \lambda_k, \mu_k)$  che sia definita positiva. Ben più difficile è il caso in cui invece *l'insieme ammissibile linearizzato è vuoto*. Per superare questo secondo ostacolo, l'approccio seguito è quello di risolvere in questi casi, un nuovo *sottoproblema quadratico a dimensioni più grandi* che però ammette sempre una soluzione.

Per quanto riguarda il problema della non convergenza della successione  $\{x_k\}$  prodotta dagli algoritmi di programmazione quadratica ricorsiva, ci si ispira alle tecniche proposte per assicurare la convergenza globale del metodo di Newton. L'idea è quella di controllare lo spostamento da effettuare lungo la direzione prodotta dal sottoproblema quadratico, cioè il nuovo punto dovrebbe essere prodotto da una iterazione del tipo:

$$x_{k+1} = x_k + \alpha_k d_k,$$

dove il passo  $\alpha_k$  dovrebbe essere calcolato utilizzando una ricerca unidimensionale che valuti la *bontà* del nuovo punto prodotto. Come accennato precedentemente, la difficoltà dei problemi di minimizzazione vincolati risiede nel fatto che la misura della bontà di un nuovo punto prodotto deve tener conto contemporaneamente di una possibile diminuzione della funzione obiettivo e di un possibile miglioramento (o mantenimento) della ammissibilità del punto prodotto.

Un modo naturale per valutare se un punto è migliore dell'altro è quello di utilizzare una *funzione di penalità* che, come si vedrà nella prossima sezione, permette di pesare contemporaneamente le due diverse esigenze presenti in un problema di ottimizzazione vincolata.

### 2.5.3 Funzioni di penalità

Un modo per affrontare un problema di ottimizzazione vincolata è quello di cercare di trasformarlo nella *minimizzazione non vincolata* di una particolare funzione detta normalmente *funzione di penalità* o *funzione di merito*. In generale le funzioni di penalità vengono costruite aggiungendo alla funzione obiettivo un termine che *penalizza* la violazione dei vincoli. In letteratura sono state proposte numerose funzioni di merito che si

differenziano tra di loro nel modo di penalizzare la violazione dei vincoli. Tali differenze si riflettono nelle proprietà teoriche che le varie funzioni di penalità presentano.

Nel seguito si descriveranno brevemente alcune funzioni di penalità che non richiede la conoscenza di un punto strettamente interno all'insieme ammissibile o la possibilità di sfruttare qualche particolarità della sua struttura.

### - Funzioni di penalità sequenziali esterne

Queste funzioni sono del tipo:

$$P_S(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \phi_S(x),$$

dove  $\varepsilon$  è un scalare positivo detto *parametro di penalità* o *coefficiente di penalità* e  $\phi_S$  è la funzione detta *termine di penalità*. Questa funzione  $\phi_S$  serve a misurare la violazione dei vincoli e soddisfa le seguenti proprietà:

- i)  $\phi_S(x)$  è una funzione *continuamente differenziabile*;
- ii)  $\phi_S(x) = 0$  se  $x \in \mathcal{F}$ ;
- iii)  $\phi_S(x) > 0$  se  $x \notin \mathcal{F}$ .

Un esempio di funzione che soddisfa le precedenti proprietà è la seguente:

$$\phi_S(x) = \sum_{i=1}^m \max\{0, g_i(x)\}^2 + \sum_{j=1}^q h_j(x)^2.$$

Le funzioni di penalità che appartengono a questa classe hanno il pregio di essere *molto semplici*. Dal punto di vista teorico si può dimostrare che, se  $x_\varepsilon$  è il minimo di  $P_S(x; \varepsilon)$ , allora per valori di  $\varepsilon$  che tendono a zero si ha che i punti  $x_\varepsilon$  tendono ad una soluzione del problema vincolato. Un difetto di questa classe di funzioni di penalità è quello di essere *non esatte*, cioè non esiste nessun valore di  $\varepsilon$  per cui è possibile dimostrare che il problema vincolato di partenza è equivalente a minimizzare in maniera non vincolata una di queste funzioni  $P_S(x)$ .

Una ulteriore limitazione di queste funzioni di merito è dovuta al fatto che non si può stabilire nessuna relazione tra i suoi punti stazionari ed il problema vincolato di partenza. Questo fatto costituisce un grosso problema dal punto di vista computazionale in quanto gli algoritmi di minimizzazione non vincolata permettono di determinare solamente dei punti stazionari della funzione di penalità.

Nonostante questi limiti questa classe di funzioni di penalità permettono di definire uno dei primi metodi proposti in letteratura in questo ambito (chiamato Metodo delle Penalità Sequenziali). Anche se non è sicuramente molto efficiente, questo metodo ha un certo interesse per la sua semplicità e per le discrete proprietà teoriche.

Tale metodo consiste trasformare il problema di partenza in una sequenza di minimizzazioni (approssimate) non vincolate della seguente funzione di penalità:

$$P(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \sum_{i=1}^m \max\{0, g_i(x)\}^2 + \frac{1}{\varepsilon} \sum_{j=1}^p h_j(x)^2, \quad (2.49)$$

L'idea di fondo del Metodo delle Penalità Sequenziali è quella di produrre una sequenza di punti  $\{x_k\}$  attraverso delle minimizzazioni approssimate della funzione  $P(x, \varepsilon_k)$ . In particolare, al crescere del numero di iterazioni, ciascun punto  $x_k$  della sequenza approssima sempre meglio un punto stazionario della funzione  $P(x, \varepsilon_k)$ .

Nelle iterazioni iniziali, al parametro di penalità  $\varepsilon_k$  che compare nella funzione  $P(x, \varepsilon_k)$  vengono assegnati valori relativamente grandi. L'idea è che, in queste iterazione, la funzione obiettivo del problema vincolato abbia nella espressione della funzione  $P(x, \varepsilon_k)$  un peso relativamente grande.

All'aumentare della iterazioni i punti  $x_k$  vengono ottenuto minimizzando in maniera sempre più precisa la funzione  $P(x, \varepsilon_k)$  in cui si scengono valori del parametro di penalità  $\varepsilon_k$  sempre più piccoli in modo da aumentare, sempre di più il peso della violazione dei vincoli. In questa modo si cerca di garantire di ottenere, al limite, punti che soddisfano tutti i vincoli e che stiano in zone in cui la funzione obiettivo assume valori relativamente piccoli.

Il Metodo delle Penalità Sequenziali può essere descritto dai seguenti passi.

### Metodo delle Penalità Sequenziali.

*Passo 0.* Dati  $x_0 \in R^n$ ,  $\theta_1, \theta_2, \theta_3 \in (0, 1)$ ,  $\varepsilon_{-1} = \varepsilon_0 > 0$  e  $\delta_0 > 0$ , si pone  $k = 0$ .

*Passo 1.* Si pone

$$\begin{aligned}(\lambda_k)_i &= \frac{2}{\varepsilon_{k-1}} \max\{0, g_i(x_k)\}, \quad i = 1, \dots, m, \\(\mu_k)_j &= \frac{2}{\varepsilon_{k-1}} h_j(x_k), \quad j = 1, \dots, p,\end{aligned}$$

se  $(x_k, \lambda_k, \mu_k)$  soddisfa le condizioni di KKT Stop.

*Passo 2.* A partire da punto  $x_k$ , si calcola il punto  $x_{k+1} \in R^n$  tale tale che

$$\|\nabla P((x_{k+1}; \varepsilon_k))\| \leq \delta_k.$$

*Passo 3.* Se

$$\sum_{i=1}^m \max\{0, g_i(x_{k+1})\}^2 + \sum_{j=1}^p h_j(x_{k+1})^2 \leq \theta_1 \left( \sum_{i=1}^m \max\{0, g_i(x_k)\}^2 + \sum_{j=1}^p h_j(x_k)^2 \right),$$

si pone  $\varepsilon_{k+1} = \varepsilon_k$  altrimenti si pone  $\varepsilon_{k+1} = \theta_2 \varepsilon_k$ .

*Passo 4.* Si pone  $\delta_{k+1} = \theta_3 \delta_k$ ,  $k = k + 1$  e si ritorna al Passo 1.

Nel Passo 1 si determinano della stime dei moltiplicatori di KKT e si controlla se  $x_k$  ottenuto soddisfa le condizioni di KKT.

Nel passo 2 si effettua una minimizzazione non vincolata di  $P(x, \varepsilon_k)$ . Tale minimizzazione viene effettuata in maniera approssimata, infatti viene arrestata appena si ottiene un punto  $x_{k+1}$  in cui la norma del gradiente di  $P$  è inferiore o uguale al parametro  $\delta_k$ . Un tale punto può essere determinato, in un numero finito di iterazioni, da un qualsiasi metodo di ottimizzazione non vincolata, tra cui il Metodo del Gradiente descritto precedentemente.

Il Passo 3 è dedicato all'aggiornamento del parametro di penalità  $\varepsilon_k$ . Se nel nuovo punto determinato  $x_{k+1}$  la violazione dei vincoli non è migliorata in maniera significativa rispetto al punto ottenuto all'iterazione precedente, il parametro di penalità viene diminuito nella iterazione successiva. Altrimenti rimane invariato.

Nell'ultimo passo viene diminuito il parametro  $\delta_k$  che determina errore accettato nell'approssimare un punto stazionario della funzione  $P(x, \varepsilon_k)$  attraverso una sua minimizzazione non vincolata.

Il seguente teorema descrive le proprietà asintotiche dei punti prodotti dal precedente metodo. Le ipotesi di questo teorema necessitano della definizione dell'insieme degli

indici dei vincoli attivi o violati, cioè dell'insieme:

$$\tilde{I}(\bar{x}) = \{i = 1, \dots, m : g_i(\bar{x}) \geq 0\}. \quad (2.50)$$

**Teorema 2.5.3** *Siano  $f \in C^1(\mathbb{R}^n)$  e  $g_i \in C^1(\mathbb{R}^n)$ , per ogni  $i = 1, \dots, m$ , e  $h_j \in C^1(\mathbb{R}^n)$ , per ogni  $j = 1, \dots, p$ . Sia  $\{x_k\}$  la sequenza di punti prodotta dal Metodo delle Penalità Sequenziali. Sia  $\bar{x}$  un punto di accumulazione della sequenza  $\{x_k\}$  che soddisfa la seguente ipotesi:*

- non esistono degli scalari  $\alpha_i \geq 0$ ,  $i \in \tilde{I}(\bar{x})$ , e  $\beta_j$ ,  $j = 1, \dots, p$ , non tutti nulli per cui si ha:

$$\sum_{i \in \tilde{I}(\bar{x})} \alpha_i \nabla g_i(\bar{x}) + \sum_{j=1}^p \beta_j \nabla h_j(\bar{x}) = 0.$$

Allora  $\bar{x}$  è un punto di Karush-Kuhn-Tucker.

**Prova.** Sia  $\{x_k\}_K$  la sottosequenza tale che

$$\lim_{k \rightarrow \infty, k \in K} x_k = \bar{x}. \quad (2.51)$$

La tesi del teorema segue dalla dimostrazione dei seguenti tre punti:

- i) il punto  $\bar{x}$  appartiene all'insieme ammissibile del problema, cioè  $\bar{x} \in \mathcal{F}$ ;
- ii) le sequenze di vettori  $\{\lambda_k\}_K$  e  $\{\mu_k\}_K$  generate dall'algoritmo al Passo 1 sono limitate;
- iii) il punto  $\bar{x}$  soddisfa le condizioni di Karush-Kuhn-Tucker del problema.

Punto i). Se esiste un  $\bar{k}$  tale che, per ogni  $k \geq \bar{k}$ , si ha  $\varepsilon_k = \varepsilon > 0$  significa che per ogni  $k \geq \bar{k}$  il test al Passo 3 è soddisfatto, quindi

$$\sum_{i=1}^m \max\{0, g_i(x_k)\}^2 + \sum_{j=1}^p h_j(x_k)^2 \leq \theta_1^{k-\bar{k}} \left( \sum_{i=1}^m \max\{0, g_i(x_{\bar{k}})\}^2 + \sum_{j=1}^p h_j(x_{\bar{k}})^2 \right),$$

Da questo, tenendo conto che  $\theta_1 \in (0, 1)$  segue che:

$$\lim_{k \rightarrow \infty} \sum_{i=1}^m \max\{0, g_i(x_k)\}^2 + \sum_{j=1}^p h_j(x_k)^2 = 0,$$

da cui segue che:

$$\begin{aligned} \lim_{k \rightarrow \infty, k \in K} \sum_{i=1}^m \max\{0, g_i(x_k)\} &= \sum_{i=1}^m \max\{0, g_i(\bar{x})\} = 0, \\ \lim_{k \rightarrow \infty, k \in K} \sum_{j=1}^p h_j(x_k) &= \sum_{j=1}^p h_j(\bar{x}) = 0. \end{aligned}$$



Le precedenti relazioni provano che  $\bar{x} \in \mathcal{F}$ .

Si consideri ora il caso in cui la sequenza di scalari  $\{\varepsilon_k\}$  è tale che

$$\lim_{k \rightarrow \infty} \varepsilon_k = 0.$$

Dal Passo 2 si ha:

$$\left\| \varepsilon_{k-1} \nabla f(x_k) + 2 \sum_{i=1}^m \max\{0, g_i(x_k)\} \nabla g_i(x_k) + 2 \sum_{j=1}^p h_j(x_k) \nabla h_j(x_k) \right\| \leq \varepsilon_{k-1} \delta_{k-1}.$$

Facendo i limiti per  $k \rightarrow \infty$  e  $k \in K$  si ottiene

$$\sum_{i=1}^m \max\{0, g_i(\bar{x})\} \nabla g_i(\bar{x}) + \sum_{j=1}^p h_j(\bar{x}) \nabla h_j(\bar{x}) = 0,$$

che può essere riscritta:

$$\sum_{i \in \bar{I}(\bar{x})} \max\{0, g_i(\bar{x})\} \nabla g_i(\bar{x}) + \sum_{j=1}^p h_j(\bar{x}) \nabla h_j(\bar{x}) = 0,$$

La precedente uguaglianza e l'ipotesi fatta di regolarità dei vincoli nel punto  $\bar{x}$  implicano che:

$$\max\{0, g_i(\bar{x})\} = 0, \quad i = 1, \dots, m, \quad h_j(\bar{x}) = 0, \quad j = 1, \dots, p,$$

che assicurano nuovamente che  $\bar{x} \in \mathcal{F}$ .

Punto ii). Sia

$$V_k = \sqrt{\sum_{i=1}^m (\lambda_k)_i^2 + \sum_{j=1}^p (\mu_k)_j^2}. \quad (2.52)$$

Se, per assurdo, le sequenze di vettori  $\{\lambda_k\}_K$  e  $\{\mu_k\}_K$  fossero illimitate, esisterebbero delle sottosequenze  $\{\lambda_k\}_{\tilde{K}}$  e  $\{\mu_k\}_{\tilde{K}}$ , con  $\tilde{K} \subseteq K$  tali che:

$$\lim_{k \rightarrow \infty, k \in \tilde{K}} V_k = \infty. \quad (2.53)$$

Dalla definizione di  $V_k$  si avrebbe che le sottosequenze  $\{\lambda_k/V_k\}_{\tilde{K}}$  e  $\{\mu_k/V_k\}_{\tilde{K}}$  sarebbero limitate quindi esisterebbe un sottoinsieme di indici  $\hat{K} \subseteq \tilde{K}$  tale che:

$$\lim_{k \rightarrow \infty, k \in \hat{K}} \frac{(\lambda_k)_i}{V_k} = \tilde{\lambda}_i \geq 0, \quad i = 1, \dots, m, \quad (2.54)$$

$$\lim_{k \rightarrow \infty, k \in \hat{K}} \frac{(\mu_k)_j}{V_k} = \tilde{\mu}_j, \quad j = 1, \dots, p. \quad (2.55)$$

Inoltre dalla definizione di  $\lambda_k$  e dalla (2.52) si avrebbe anche:

$$\lim_{k \rightarrow \infty, k \in \hat{K}} \frac{(\lambda_k)_i}{V_k} = \tilde{\lambda}_i = 0, \quad i \notin \tilde{I}(\bar{x}), \quad (2.56)$$

$$\sqrt{\sum_{i=1}^m \tilde{\lambda}_i^2 + \sum_{j=1}^p \tilde{\mu}_j^2} = 1. \quad (2.57)$$

Il Passo 2 implicherebbe:

$$\left\| \frac{1}{V_k} \nabla P(x_k; \varepsilon_{k-1}) \right\| = \left\| \frac{\nabla f(x_k)}{V_k} + \sum_{i=1}^m \frac{(\lambda_k)_i}{V_k} \nabla g_i(x_k) + \sum_{j=1}^p \frac{(\mu_k)_j}{V_k} \nabla h_j(x_k) \right\| \leq \frac{\delta_{k-1}}{V_k}.$$

Facendo i limiti per  $k \rightarrow \infty$  e  $k \in \hat{K}$  e ricordando la (2.56) si avrebbe:

$$\sum_{i \in \tilde{I}(\bar{x})} \tilde{\lambda}_i \nabla g_i(\bar{x}) + \sum_{j=1}^p \tilde{\mu}_j \nabla h_j(\hat{x}) = 0. \quad (2.58)$$

La (2.54), la (2.57), la (2.55) e la (2.58) negherebbero l'ipotesi di regolarità dei vincoli nel punto  $\bar{x}$ .

Punto iii). Dal Passo 2 e dalla definizione del vettore  $\lambda_k$  si ha:

$$\left\| \nabla f(x_k) + \sum_{i=1}^m (\lambda_k)_i \nabla g_i(x_k) + \sum_{j=1}^p (\mu_k)_j \nabla h_j(x_k) \right\| \leq \delta_{k-1}, \quad (2.59)$$

$$\lambda_k^T g(x_k) \geq 0, \quad (2.60)$$

$$\lambda_k \geq 0. \quad (2.61)$$

Poichè le sottosequenze  $\{\lambda_k\}_K$  e  $\{\mu_k\}_K$  sono limitate esiste un sottoinsieme di indici  $\bar{K} \subseteq K$  tale che:

$$\lim_{k \rightarrow \infty, k \in \bar{K}} x_k = \bar{x} \in \mathcal{F}, \quad (2.62)$$

$$\lim_{k \rightarrow \infty, k \in \bar{K}} \lambda_k = \bar{\lambda} \geq 0, \quad (2.63)$$

$$\lim_{k \rightarrow \infty, k \in \bar{K}} \mu_k = \bar{\mu}, \quad (2.64)$$

$$\lim_{k \rightarrow \infty, k \in \bar{K}} \delta_{k-1} = 0. \quad (2.65)$$

Dalle (2.59)-(2.65) si ottiene:

$$\nabla f(\bar{x}) + \nabla g(\bar{x}) \bar{\lambda} + \nabla h(\bar{x}) \bar{\mu} = 0, \quad (2.66)$$

$$\bar{\lambda}^T g(\bar{x}) \geq 0, \quad (2.67)$$

$$\bar{\lambda} \geq 0, \quad (2.68)$$

$$g(\bar{x}) \leq 0, \quad (2.69)$$

$$h(\bar{x}) = 0. \quad (2.70)$$

La (2.68) e la (2.69) implicano che  $\bar{\lambda}^T g(\bar{x}) \leq 0$  che, insieme alla (2.67), da

$$\bar{\lambda}^T g(\bar{x}) = 0. \quad (2.71)$$

In conclusione si ottiene che le (2.66), (2.68), (2.69), (2.70), (2.71) mostrano che il punto  $\bar{x}$  è un punto di Karush-Kuhn-Tucker del problema.  $\square$

**- Funzioni di penalità esatte non differenziabili (questa sezione non fa parte del programma di esame)**

Queste funzioni di merito hanno una struttura del tipo:

$$P_N(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \phi_N(x),$$

dove, in questo caso, la funzione  $\phi_N$  ha le seguenti caratteristiche:

- i)  $\phi_N(x)$  non è una funzione continuamente differenziabile;
- ii)  $\phi_N(x) = 0$  se  $x \in \mathcal{F}$ ;
- iii)  $\phi_N(x) > 0$  se  $x \notin \mathcal{F}$ .

In letteratura sono state proposte numerose funzioni  $\phi_N(x)$  che soddisfano i precedenti punti (i)-(iii), una delle più note è la seguente:

$$\phi_N(x) = \sum_{i=1}^m \max\{0, g_i(x)\} + \sum_{j=1}^q |h_j(x)|.$$

Il fatto che in questa classe di funzioni di merito si rilassi la richiesta che il termine di penalità  $\phi_N(x)$  sia continuamente differenziabile permette di superare le limitazioni delle funzioni di penalità sequenziali. Infatti le funzioni di penalità non differenziabili sono *esatte*, in quanto, sotto opportune ipotesi, è possibile dimostrare che, per valori sufficientemente piccoli del parametro di penalità  $\varepsilon$ , ogni minimo globale (locale) del problema vincolato è un minimo globale (locale) non vincolato di  $P_N(x; \varepsilon)$  e viceversa. Inoltre, una relazione analoga vale anche tra i punti di Kuhn-Tucker del problema originale ed i punti critici della funzione non differenziabile  $P_N(x; \varepsilon)$ .

Naturalmente la limitazione principale nell'utilizzazione di questa classe di funzioni di penalità risiede nella loro non differenziabilità che le rende alquanto *difficili da minimizzare*.

**- Funzioni Lagrangiane aumentate (questa sezione non fa parte del programma di esame)**

A differenza delle funzioni di merito viste precedentemente, in questa classe di funzioni l'idea è quella di trasformare un problema di minimizzazione vincolato su  $R^n$  in uno di minimizzazione non vincolato su  $R^{n+m+q}$ . In particolare la funzione di merito da minimizzare assume la struttura:

$$L_A(x, \lambda, \mu; \varepsilon) = f(x) + \frac{1}{\varepsilon} \phi_E(x, \lambda, \mu; \varepsilon),$$

dove il termine di penalità  $\phi_E$ , oltre ad essere funzione della tripla  $(x, \lambda, \mu)$ , dipende anche dal parametro di penalità  $\varepsilon$ . Nella funzione Lagrangiana aumentata classica la funzione  $\phi_E$  soddisfa le seguenti proprietà:

i)  $\phi_E(x, \lambda, \mu; \varepsilon)$  è una funzione *continuamente differenziabile* nella tripla  $(x, \lambda, \mu)$ ;

ii) per ogni  $\lambda$  e  $\mu$  si ha

$$\lim_{\varepsilon \rightarrow 0} \phi_E(x, \lambda, \mu; \varepsilon) = \phi_S(x),$$

dove  $\phi_S$  è il termine di penalità usato nelle funzioni di penalità sequenziali esterne;

iii) se  $x \in \mathcal{F}$  allora  $\phi_E(x, \lambda, \mu; \varepsilon) = 0$  se e solamente se  $\lambda_i \geq 0$  e  $\lambda_i g_i(x) = 0$  per ogni  $i = 1, \dots, m$ ;

iv) se  $(x^*, \lambda^*, \mu^*)$  è una tripla di Kuhn-Tucker allora per ogni  $\varepsilon$  si ha:

$$\frac{1}{\varepsilon} \nabla \phi_E(x^*, \lambda^*, \mu^*; \varepsilon) = -\nabla f(x^*).$$

In particolare la funzione Lagrangiana aumentata tradizionale proposta la seguente funzione:

$$\begin{aligned} \phi_E(x, \lambda, \mu; \varepsilon) = & \sum_{i=1}^m \left[ \varepsilon \lambda_i \max\left\{g_i(x), \frac{-\varepsilon \lambda_i}{2}\right\} + \max\left\{g_i(x), \frac{-\varepsilon \lambda_i}{2}\right\}^2 \right] \\ & + \sum_{j=1}^q \left[ \varepsilon \mu_j h_j(x) + h_j(x)^2 \right]. \end{aligned}$$

La proprietà (i) assicura ovviamente che la funzione di penalità risultante sia continuamente differenziabile. La (ii) implica che, quando il coefficiente di penalità tende a zero, la funzione  $\phi_E$  tende al termine di penalità utilizzato nelle funzioni di penalità sequenziali che pesa la violazione dei vincoli. La proprietà (iii) mostra che il termine  $\phi_E$  è in grado di forzare la positività del moltiplicatore  $\lambda$  e la complementarità (cioè  $g_i(x)\lambda_i = 0$ ). L'ultima proprietà assicura che ogni una tripla di Kuhn-Tucker  $(x^*, \lambda^*, \mu^*)$  è un punto stazionario della funzione Lagrangiana aumentata.

Sotto opportune ipotesi e per valori sufficientemente piccoli del parametro di penalità  $\varepsilon$ , si ha anche che ogni minimo globale (locale) del problema vincolato è un minimo globale (locale) non vincolato della funzione  $L_A(x, \lambda^*, \mu^*; \varepsilon)$ . Purtroppo quest'ultimo risultato non può essere applicato direttamente per trasformare il problema vincolato originale in uno vincolato perché richiederebbe la conoscenza dei moltiplicatori di Kuhn-Tucker  $\lambda^*$  e  $\mu^*$ . Tuttavia la precedente proprietà delle funzioni Lagrangiane aumentate hanno ispirato la definizione di una classe di algoritmi, detti *metodi dei moltiplicatori*, che si basano su una sequenza di minimizzazioni, rispetto alla variabile  $x$ , di funzioni del tipo  $L_A(x, \lambda_k, \mu_k; \varepsilon_k)$ , dove le stime dei moltiplicatori  $(\lambda_k, \mu_k)$  e il parametro di penalità  $\varepsilon_k$  vengono aggiornati iterativamente secondo opportune regole.

**- Funzioni di penalità esatte continuamente differenziabili (questa sezione non fa parte del programma di esame).**

L'idea base su cui si fondano funzioni di penalità esatte continuamente differenziabili è quella di *approssimare* in una funzione Lagrangiana aumentata i moltiplicatori di Kuhn-Tucker  $(\lambda^*, \mu^*)$  attraverso delle funzioni della sola variabile  $x$ . Più precisamente la struttura base di una funzione di penalità di questo tipo è la seguente:

$$P_E(x; \varepsilon) = f(x) + \frac{1}{\varepsilon} \phi_E(x, \lambda(x), \mu(x); \varepsilon),$$

dove il termine di penalità è lo stesso di quello utilizzato nella funzione Lagrangiana aumentata vista precedentemente, mentre  $\lambda(x)$  e  $\mu(x)$  sono delle particolari funzioni tali che:

- i)  $\lambda(\cdot) : R^n \rightarrow R^m$  e  $\mu(\cdot) : R^n \rightarrow R^q$ ;
- ii) se  $(x^*, \lambda^*, \mu^*)$  è una tripla di Kuhn-Tucker allora  $\lambda(x^*) = \lambda^*$  e  $\mu(x^*) = \mu^*$ ;
- iii)  $\lambda(\cdot)$  e  $\mu(\cdot)$  sono funzioni *continuamente differenziabili*.

In letteratura tutte le funzioni che godono delle precedenti proprietà vengono dette *funzioni moltiplicatrici*.

Grazie alla proprietà del termine di penalità  $\phi_E$  e delle funzioni  $\lambda(x)$  e  $\mu(x)$ , le funzioni di penalità di questa classe presentano delle buone proprietà di esattezza. Infatti, sotto opportune ipotesi e per valori sufficientemente piccoli di  $\varepsilon$ , è possibile stabilire una *corrispondenza biunivoca* tra i minimi globali (locali) e punti di Kuhn-Tucker del problema vincolato originario e i minimi globali (locali) e i punti stazionari della funzione *continuamente differenziabile*  $P_E(x; \varepsilon)$ .

**- Funzioni Lagrangiane aumentate esatte (questa sezione non fa parte del programma di esame)**

Le funzioni di penalità esatte continuamente differenziabili si basano fortemente sull'utilizzazione delle funzione moltiplicatrici  $\lambda(x)$  e  $\mu(x)$ . Purtroppo il calcolo di tali funzioni può essere *troppo oneroso* se il numero di vincoli del problema è *grande*. Le funzioni Lagrangiane aumentate esatte permettono di trasformare un problema vincolato in uno non vincolato evitando l'utilizzazione di funzioni moltiplicatrici. Il prezzo da pagare è che il problema non vincolato da risolvere è un problema sullo *spazio esteso*  $R^{n+m+q}$ . Analogamente alle funzioni di penalità esatte continuamente differenziabili, le funzioni Lagrangiane aumentate esatte traggono ispirazione dalle funzioni Lagrangiane aumentate descritte. Infatti la forma originale di una funzioni di merito di questa classe è la seguente:

$$L_E(x, \lambda, \mu; \varepsilon) = f(x) + \frac{1}{\varepsilon} \phi_E(x, \lambda, \mu; \varepsilon) + \phi_L(x, \lambda, \mu),$$

dove, nuovamente, il termine di penalità  $\phi_E$  è quello presente nelle funzioni di Lagrangiane aumentate mentre il termine nuovo  $\phi_L$  ha le seguenti caratteristiche:

- i)  $\phi_L(x, \lambda, \mu)$  è una funzione *continuamente differenziabile* nella tripla  $(x, \lambda, \mu)$ ;

ii) se  $(x^*, \lambda^*, \mu^*)$  è una tripla di Kuhn-Tucker del Problema (1.3) allora

$$\phi_L(x^*, \lambda, \mu) = 0$$

*se e solamente se  $\lambda = \lambda^*$  e  $\mu = \mu^*$ .*

Quindi a differenza delle funzioni Lagrangiane aumentate, quelle esatte hanno un termine  $\phi_L$  la cui minimizzazione forza le variabili  $\lambda$  e  $\mu$  a coincidere con i moltiplicatori di Kuhn-Tucker del problema vincolato.

Le funzioni Lagrangiane esatte si possono stabilire le stesse proprietà di esattezza delle funzioni di penalità esatte continuamente differenziabili.

## Capitolo 3

# Metodi di ottimizzazione globale

La maggior parte dei metodi proposti in letteratura considerano dei problemi di ottimizzazione globale che hanno la seguente struttura.

$$\begin{aligned} \min f(x) \\ x \in \mathcal{D}, \end{aligned} \tag{3.1}$$

dove la funzione  $f : \mathcal{D} \rightarrow R$  è almeno continua e l'insieme  $\mathcal{D} \subseteq R^n$  è dato da:

$$\mathcal{D} = \{x \in R^n : l \leq x \leq u\}, \tag{3.2}$$

dove  $l \in R^n$  e  $u \in R^n$  sono due vettori dati.

L'insieme dei punti di minimo globale del precedente problema è rappresentato da:

$$X^* = \{x^* \in \mathcal{D} : f(x^*) \leq f(x), \text{ per tutti } x \in \mathcal{D}\}, \tag{3.3}$$

mentre il valore ottimo della funzione obiettivo è rappresentato da:

$$f^* = f(x^*), \quad \text{con } x^* \in X^*. \tag{3.4}$$

Molto spesso si ipotizza anche che i due vettori siano stati scelti in maniera tale che i minimi globali della funzione  $f$  su  $\mathcal{D}$  siano interni all'insieme  $\mathcal{D}$ .

In altre parole i problemi globali maggiormente affrontati sono quelli non vincolati oppure quelli che hanno vincoli molto semplici che impongono solamente dei limiti superiori ed inferiori sulle componenti del vettore delle variabili.

Molto più limitatata è stata l'attività di ricerca verso problemi di ottimizzazione globale con vincoli più complessi. Questo è dovuto alla complessità di questi problemi vincolati che rende inapplicabile gran parte dei risultati e metodi proposti per i problemi del tipo (3.1). Comunque, da punto di vista pratico e teorico, l'uso di funzioni di penalità esatte o lagrangiani aumentati esatti può costituire uno strumento efficace per affrontare problemi di ottimizzazione globale vincolata.

Una descrizione completa ed approfondita dei molti metodi proposti per risolvere problemi di ottimizzazione del tipo (3.1) richiederebbe una trattazione molto complessa e molto lunga. Quindi, nei capitoli successivi, allo scopo di fornire un'introduzione all'argomento, ci si limiterà a considerare solamente alcune classi particolari di metodi di ottimizzazione globale. La scelta si è rivolta principalmente verso metodi che, da una parte, fossero relativamente semplici da descrivere e particolarmente facili da realizzare e che, dall'altra parte, avessero le potenzialità per affrontare anche problemi di ottimizzazione globale derivanti da applicazioni ingegneristiche "difficili", cioè quelle in cui la funzione obiettivo è rappresentata da una "black box" ed il numero di variabili non è particolarmente piccolo.

### 3.1 Metodi Probabilistici

Questo capitolo è dedicato ad introdurre l'approccio probabilistico per la soluzione di problemi di ottimizzazione globale del tipo (3.1). Dopo aver richiamato quale è l'idea da cui parte tale approccio, vengono brevemente descritti alcuni algoritmi di tipo probabilistico: tra cui i metodi di tipo multistart, i metodi detti di "simulated annealing" e metodi che usano popolazioni di punti.

#### 3.1.1 Algoritmi che utilizzano punti scelti a caso su $\mathcal{D}$ .

Uno dei punti di partenza dei metodi di tipo probabilistico è il seguente risultato.

**Proposizione 3.1.1** *Sia  $\{x_k\}$  una sequenza di punti aleatori scelti a caso su  $\mathcal{D}$ , cioè vettori generati con una distribuzione di probabilità uniforme che, per ogni sottoinsieme  $\mathcal{A}$  tale che  $\text{meas}(\mathcal{A}) > 0$  (con  $\text{meas}(\cdot)$  misura di Lebesgue di un insieme), associa una misura di probabilità*

$$m = \frac{\text{meas}(\mathcal{A})}{\text{meas}(\mathcal{D})}, \quad (3.5)$$

*Allora per ogni sottoinsieme  $\mathcal{A}$  di  $\mathcal{D}$  tale che  $\text{meas}(\mathcal{A}) > 0$ , si ha:*

$$\lim_{k \rightarrow \infty} \text{Prob}\left\{X_k \cap \mathcal{A} \neq \emptyset\right\} = 1, \quad (3.6)$$

con  $X_k = \{x_0, \dots, x_k\}$ .

**Prova.** Dalla (3.5) si ha:

$$1 > m > 0.$$

Lo scalare  $m$  dà la probabilità che un punto, scelto a caso su  $\mathcal{D}$ , cada nell'insieme  $\mathcal{A}$ . Di conseguenza la probabilità che, generati  $k$  vettori con una distribuzione uniforme su  $\mathcal{D}$ , almeno uno cada in  $\mathcal{A}$  è data da:

$$\text{Prob}\left\{X_k \cap \mathcal{A} \neq \emptyset\right\} = 1 - (1 - m)^k$$

da cui segue:

$$\lim_{k \rightarrow \infty} \text{Prob}\left\{X_k \cap \mathcal{A} \neq \emptyset\right\} = 1. \quad \square$$



### - Campionamento uniforme della funzione obiettivo

Il precedente risultato indica che, se  $\mathcal{A}$  è un qualsiasi intorno di un minimo globale  $x^*$  di  $f(x)$  su  $\mathcal{D}$  e se si generano molti punti a caso, allora uno di questi punti generati cade, molto probabilmente, vicino al minimo globale  $x^*$ . Questo suggerisce un primo semplice algoritmo di ottimizzazione globale.

#### Algoritmo di Campionamento Uniforme.

*Passo 0:* si genera a caso un punto  $x_0$  su  $\mathcal{D}$ , si pone  $x_0^* = x_0$  e  $k = 1$ ;

*Passo 1:* si genera a caso un punto  $x_k$  su  $\mathcal{D}$ ;

*Passo 2:* se  $f(x_k) < f(x_{k-1}^*)$  allora si pone  $x_k^* = x_k$

altrimenti si pone  $x_k^* = x_{k-1}^*$ ;

*Passo 3:* si pone  $k = k + 1$  e si ritorna al Passo 1.

Utilizzando la precedente Proposizione 3.1.1, si può stabilire la seguente proprietà di convergenza dei punti prodotti dal precedente algoritmo.

**Proposizione 3.1.2** *Sia  $\{x_k^*\}$  la sequenza di punti aleatori prodotti dall'Algoritmo di Campionamento Uniforme. Allora, comunque scelto un  $\varepsilon > 0$ , si ha:*

$$\lim_{k \rightarrow \infty} \text{Prob} \left\{ x_k^* \in \{x \in \mathcal{D} : f(x) \leq f^* + \varepsilon\} \right\} = 1. \quad (3.7)$$

dove  $f^*$  indica il valore ottimo della  $f$  su  $\mathcal{D}$  (definito dalla (3.4)).

**Prova.** Se si definisce l'insieme  $\mathcal{A}$  nella seguente maniera:

$$\mathcal{A} = \{x \in \mathcal{D} : f(x) \leq f^* + \varepsilon\},$$

si può notare che

$$\text{Prob} \left\{ x_k^* \in \{x \in \mathcal{D} : f(x) \leq f^* + \varepsilon\} \right\} = \text{Prob} \left\{ X_k \cap \mathcal{A} \neq \emptyset \right\},$$

dove  $X_k = \{x_0, \dots, x_k\}$  è l'insieme di punti generati a caso su  $\mathcal{D}$  dall'algoritmo.

La dimostrazione della (3.7) della Proposizione segue utilizzando la Proposizione 3.1.1 e ricordando che, dal Teorema 1.5.1, si ha che  $\text{meas}(\mathcal{A}) > 0$ .  $\square$

A partire dalla precedente proposizione si può anche dimostrare che la sequenza dei valori della funzione obiettivo  $\{f(x_k^*)\}$  prodotta dal precedente algoritmo converge con probabilità uno al valore attimo  $f(x^*)$  con  $x^* \in X^*$ .

### - Metodi di tipo multistart

Questi metodi si basano sull'idea di utilizzare la generazione di punti a caso su  $\mathcal{D}$  per determinare un punto in un intorno di un minimo globale e di utilizzare qualche algoritmo di minimizzazione locale per determinare efficientemente un minimo globale. Quindi questa classe di metodi possono essere visti o come un tentativo di globalizzare i metodi di ottimizzazione locale oppure come un tentativo di migliorare l'efficienza del Algoritmo di Campionamento Uniforme sfruttando il fatto che molti algoritmi locali possono essere attratti da un minimo globale, cioè soddisfano la seguente assunzione

**Assunzione 3.1.1** *Se  $\{x_k\}$  è la sequenza di punti generata dall'algoritmo e se  $x^*$  è un minimo globale di  $f$  su  $R^n$ , esiste un  $\varepsilon > 0$  tale che se, per un indice  $\bar{k}$ , si ha  $x_{\bar{k}} \in B(x^*; \varepsilon)$  allora:*

- (i)  $x_k \in B(x^*; \varepsilon)$  per tutti i  $k \geq \bar{k}$ ;
- (ii)  $\lim_{k \rightarrow \infty} x_k = x^*$ .

L'algoritmo di tipo multistart di partenza è il seguente.

#### **Algoritmo Multistart 1.**

*Passo 0:* si genera a caso un punto  $x_0$  su  $\mathcal{D}$ , si pone  $x_0^* = x_0$  e  $k = 1$ ;

*Passo 1:* si genera a caso un punto  $x_k$  su  $\mathcal{D}$ ;

*Passo 2:* partendo da  $x_k$  si applica un algoritmo di ottimizzazione locale ottenendo  $y_k$ ;

*Passo 3:* se  $f(y_k) < f(x_{k-1}^*)$  allora si pone  $x_k^* = y_k$

altrimenti si pone  $x_k^* = x_{k-1}^*$ ;

*Passo 4:* si pone  $k = k + 1$  e si ritorna al Passo 1.

Naturalmente l'uso di un algoritmo deterministico locale può far migliorare la proprietà di convergenza dei punti prodotti

**Proposizione 3.1.3** *Se esiste un minimo globale  $x^*$  non vincolato della funzione  $f$  su  $\mathcal{D}$  per cui l'algoritmo di minimizzazione locale (utilizzato al Passo 2) soddisfa l'Assunzione 3.1.1 allora la sequenza  $\{x_k^*\}$  di punti aleatori prodotti dall'Algoritmo di Multistart 1 è tale che:*

$$\lim_{k \rightarrow \infty} \text{Prob} \left\{ x_k^* \in X^* \right\} = 1, \quad (3.8)$$

dove  $X^*$  è l'insieme dei minimi globali del Problema (3.1).

**Prova.** Segue, nuovamente, dalla Proposizione 3.1.1. Infatti

$$Prob\left\{x_k^* \in X^*\right\} \geq Prob\left\{x_k^* = x^*\right\},$$

dove  $x^*$  è il minimo globale considerato nell'enunciato della proposizione.

Quindi, si può scegliere

$$\mathcal{A} = B(x^*, \varepsilon),$$

dove  $B(x^*, \varepsilon)$  è l'intorno descritto nella Assunzione 3.1.1. Poichè, per ipotesi, l'algoritmo di minimizzazione locale soddisfa la Assunzione 3.1.1, nel momento in cui si genera un punto  $x_k$  tale che  $x_k \in B(x^*, \varepsilon)$  si ha  $y_k = x^*$ . Perciò si ha che:

$$Prob\left\{x_k^* = x^*\right\} = Prob\left\{X_k \cap \mathcal{A} \neq \emptyset\right\},$$

dove  $X_k = \{x_0, \dots, x_k\}$  è l'insieme di punti generati a caso su  $\mathcal{D}$  dall'algoritmo e, perciò la tesi della proposizione segue dalla Proposizione 3.1.1.  $\square$

L'Algoritmo Multistart 1 è stato il primo ad essere proposto in letteratura ed è tuttora abbastanza usato; tuttavia presenta il difetto che può richiedere un numero molto grande di minimizzazioni locali. Per superare questo difetto sono state proposte delle variazioni dell'Algoritmo Multistart 1 che cercano di ridurre il numero di minimizzazioni locali senza variarne le proprietà.

### **Algoritmo Multistart 2.**

*Passo 0:* si genera a caso un punto  $x_0$  su  $\mathcal{D}$ , si pone  $x_0^* = x_0$  e  $k = 1$ ;

*Passo 1:* si genera a caso un punto  $x_k$  su  $\mathcal{D}$ ;

*Passo 2:* se  $f(x_{k-1}^*) < f(x_k)$  allora si pone  $x_k^* = x_{k-1}^*$  e si va al Passo 4;

*Passo 3:* partendo da  $x_k$  si applica un algoritmo di ottimizzazione locale ottenendo  $x_k^*$ ;

*Passo 4:* si pone  $k = k + 1$  e si ritorna al Passo 1.

Nel precedente algoritmo le minimizzazioni locali non vengono applicate se il valore della funzione obiettivo nel punto generato a caso è peggiore del miglior valore della funzione obiettivo ottenuto dall'algoritmo. Poichè le minimizzazioni locali producono dei punti in cui il valore della funzione obiettivo è non peggiore rispetto al valore che si ha nel punto iniziale, ne deriva che ogni applicazione di tali minimizzazioni produce un punto che può essere considerato come nuova stima  $x_k^*$ . Purtroppo la strategia adottata dal precedente algoritmo può portare ad effettuare poche minimizzazioni locali. L'algoritmo seguente cerca di seguire una strategia intermedia tra i due precedenti algoritmi multistart.

**Algoritmo Multistart 3.**

*Passo 0:* si genera a caso un punto  $x_0$  su  $\mathcal{D}$ , si pone  $x_0^* = x_0$  e  $k = 1$ ;

*Passo 1:* si generano a caso  $N_k^1$  punti su  $\mathcal{D}$ ;

*Passo 2:* si scelgono  $N_k^2 < N_k^1$  punti “più promettenti” e da ciascuno di questi si effettua una minimizzazione locale;

*Passo 3:* fra i punti  $y_k^1, y_k^2, \dots, y_k^{N_k^2}$  ottenuti da queste minimizzazioni locali si sceglie quello  $y_k$  a cui corrisponde il valore della funzione obiettivo più piccolo, cioè

$$f(y_k) = \min_{1 \leq i \leq N_k^2} f(y_k^i);$$

*Passo 4:* se  $f(y_k) < f(x_{k-1}^*)$  allora si pone  $x_k^* = y_k$

altrimenti si pone  $x_k^* = x_{k-1}^*$ ;

*Passo 5:* si pone  $k = k + 1$  e si ritorna al Passo 1.

L’idea del precedente algoritmo è quella di ridurre il più possibile il numero delle minimizzazioni locali utilizzando le informazioni che si possono ottenere sulla funzione andando a valutarne il valore su un certo numero di punti scelti a caso. Nel Passo 2 si cerca di raggruppare (usualmente utilizzando le tecniche della “cluster analysis”) i punti generati al Passo 1 in maniera tale che ogni gruppo sia costituito da punti che appartengono alla regione di attrazione dello stesso minimo locale. Dopodiché per ogni gruppo si effettua una sola minimizzazione locale.

**3.1.2 Metodi di tipo “simulated annealing”**

Una evoluzione ulteriore dei metodi descritti nella precedente sezione è quella di cercare di definire dei nuovi algoritmi di ottimizzazione globale basati sull’uso di punti generati in maniera da cercare di sfruttare le informazioni sul problema ottenute durante le iterazioni dell’algoritmo.

**- Distribuzioni non uniformi di punti**

Una osservazione importante nell’ambito dei metodi di ottimizzazione globale probabilistici riguarda il fatto che le proprietà di convergenza stabilite dalla Proposizione 3.1.2 e dalla Proposizione 3.1.3 non dipendono in maniera esclusiva dal fatto che i punti  $x_0, \dots, x_k$  sono generati con distribuzione uniforme su  $\mathcal{D}$ , cioè sono realizzazioni di variabili aleatorie destribuite con funzioni densità di probabilità costanti su  $\mathcal{D}$ . Infatti sia la Proposizione 3.1.2 e che la Proposizione 3.1.3 seguono dalla Proposizione 3.1.1 ma quest’ultima può essere generalizzata nella seguente proposizione.

**Proposizione 3.1.4** Sia  $\{x_k\}$  una sequenza di punti aleatori generati secondo delle distribuzioni di probabilità a cui sono associate delle misure di probabilità  $m_k(\cdot)$  tali che, per ogni sottoinsieme  $\mathcal{A}$  di  $\mathcal{D}$  con  $\text{meas}(\mathcal{A}) > 0$ , soddisfano:

$$\prod_{k=0}^{\infty} (1 - m_k(\mathcal{A})) = 0. \quad (3.9)$$

Allora per ogni sottoinsieme  $\mathcal{A}$  di  $\mathcal{D}$  con  $\text{meas}(\mathcal{A}) > 0$  si ha:

$$\lim_{k \rightarrow \infty} \text{Prob}\{X_k \cap \mathcal{A} \neq \emptyset\} = 1, \quad (3.10)$$

con  $X_k = \{x_0, \dots, x_k\}$ .

**Prova.** Per definizione di misura di probabilità e dalle ipotesi della proposizione,  $m_k(\mathcal{A})$  è la probabilità che un punto generato  $x_k$  cada nell'insieme  $\mathcal{A}$ . Di conseguenza la probabilità che, tra i vettori generati  $x_0, \dots, x_k$ , almeno uno cada in  $\mathcal{A}$  è data da:

$$\text{Prob}\{X_k \cap \mathcal{A} \neq \emptyset\} = 1 - \prod_{i=0}^k (1 - m_i(\mathcal{A}))$$

da cui, usando la (3.9), segue:

$$\lim_{k \rightarrow \infty} \text{Prob}\{X_k \cap \mathcal{A} \neq \emptyset\} = 1. \quad \square$$

### - Algoritmi di tipo “simulated annealing”

La Proposizione 3.1.4 mette in evidenza la possibilità di definire dei nuovi algoritmi probabilistici che usino punti generati secondo funzioni densità di probabilità non costanti su  $\mathcal{D}$ . Questo fatto può essere sfruttato per cercare di utilizzare funzioni densità di probabilità che siano maggiormente concentrate intorno ai minimi globali, cioè che rendano più probabile la generazione di punti più vicini ai minimi globali.

Un esempio di una funzione densità di probabilità con queste caratteristiche può essere ottenuto traendo ispirazione dai risultati della meccanica statistica.

Si consideri un sistema fisico composto da un numero molto grande di particelle della stessa natura, si indichi con  $s$  lo stato del sistema e con  $E(s)$  l'energia associata a questo stato. Se il sistema fisico è in una situazione di equilibrio termico allora la funzione densità di probabilità, relativa al fatto che il sistema sia nello stato  $s$ , è proporzionale a

$$e^{-\frac{E(s)}{\mathcal{K}T}}$$

ove  $\mathcal{K}$  è la costante di Boltzman e  $T$  la temperatura.

Parlando in maniera approssimata, è noto che, se si abbassa la temperatura, diventano sempre più probabili gli stati a bassa energia, fino alla situazione limite per cui  $T \rightarrow 0$

e gli unici stati possibili sono quelli ad energia nulla. Quindi, in linea teorica, si può immaginare un sistema fittizio che associa ad ogni suo stato  $x$ , l'energia:

$$E(x) = f(x) - f^* \geq 0$$

dove  $f^*$  è, come al solito, il minimo globale della  $f(x)$ . Ora, se si simulasse il comportamento di questo sistema e se si facesse tendere a zero la temperatura, diventerebbero sempre più probabili gli stati  $x^*$  del sistema che corrispondono a basse energia fino ad arrivare alla situazione limite che:

$$E(x^*) = f(x^*) - f^* = 0.$$

Per avvalorare ulteriormente questa idea si può far riferimento al Teorema 1.5.6. Infatti, nel caso in cui il minimo globale  $x^*$  su  $\mathcal{D}$  sia unico, questo teorema implica (ponendo  $\mathcal{K} = 1$  e  $k = 1/T$ ):

$$x_i^* = \lim_{T \rightarrow 0} \frac{\int_{\mathcal{D}} x_i e^{-f(x)/T} dx}{\int_{\mathcal{D}} e^{-f(x)/T} dx} = \lim_{T \rightarrow 0} \frac{\int_{\mathcal{D}} x_i e^{-(f(x)-f^*)/T} dx}{\int_{\mathcal{D}} e^{-(f(x)-f^*)/T} dx}, \quad \text{per } i = 1, \dots, n. \quad (3.11)$$

Le precedenti uguaglianze possono essere riscritte come:

$$x_i^* = \lim_{T \rightarrow 0} \int_{\mathcal{D}} x_i p_T(x) dx = \lim_{T \rightarrow 0} \bar{x}_i(T) \quad (3.12)$$

dove

$$p_T(x) = \frac{e^{-f(x)/T}}{\int_{\mathcal{D}} e^{-f(x)/T} dx} = \frac{e^{-(f(x)-f^*)/T}}{\int_{\mathcal{D}} e^{-(f(x)-f^*)/T} dx}, \quad (3.13)$$

è una densità di probabilità e dove  $\bar{x}_i(T)$  sono i valori medi di variabili aleatorie che sono distribuite secondo la densità di probabilità  $p_T(x)$ .

Quindi, l'idea su cui si basano i metodi di ottimizzazione globale di tipo “simulated annealing” è quella di simulare dei vettori aleatori distribuiti secondo la densità di probabilità  $p_T(x)$  data dalla (3.13). La relazione (3.12) indica che, al diminuire di  $T$ , i vettori  $x$  generati su  $\mathcal{D}$  secondo la funzione densità di probabilità  $p_T(x)$  si avvicinano, in media, al punto di minimo globale della  $f(x)$  su  $\mathcal{D}$ .

Un elemento essenziale allo sviluppo dei metodi “simulated annealing” è il fatto che punti distribuiti secondo la funzione densità di probabilità  $p_T(x)$  posso essere generati abbastanza facilmente utilizzando il metodo di Von Neumann. Tale metodo si può applicare a tutte le funzioni densità di probabilità  $p(x)$  che hanno la seguente struttura:

$$p(x) = Cs(x), \quad (3.14)$$

dove  $C$  è una costante positiva e  $s : \mathcal{D} \rightarrow R_+$  è tale che  $0 < s(x) \leq 1$ , per tutti  $x \in \mathcal{D}$ . Utilizzando la definizione (3.13), si può scrivere:

$$p_T(x) = \frac{e^{-(f(x)-f^*)/T}}{\int_{\mathcal{D}} e^{-(f(x)-f^*)/T} dx} = \tilde{C} \tilde{s}(x),$$

dove

$$\tilde{C} = 1 / \left( \int_{\mathcal{D}} e^{-(f(x)-f^*)/T} dx \right) \quad \text{e} \quad \tilde{s}(x) = e^{-(f(x)-f^*)/T}.$$

Quindi la funzione  $p_T(x)$  ha la struttura (3.14) e, perciò, si può utilizzare il metodo di Von Neumann.

*Sia  $T$  uno scalare positivo, sia  $\alpha$  uno scalare scelto a caso su  $[0, 1]$  e sia  $x$  un vettore scelto a caso su  $\mathcal{D}$ , se*

$$\alpha \leq e^{-(f(x)-f^*)/T}, \quad (3.15)$$

*allora il punto  $x$  è una realizzazione della variabile aleatoria distribuita secondo la funzione densità di probabilità*

$$p_T(x) = \frac{e^{-(f(x)-f^*)/T}}{\int_{\mathcal{D}} e^{-(f(x)-f^*)/T} dx}. \quad (3.16)$$

Purtroppo, per avere la struttura (3.14), la funzione densità di probabilità  $p_T(x)$  deve far comparire nella sua espressione il valore ottimo  $f^*$  della funzione obiettivo. Per la maggior parte dei problemi di ottimo globale questa quantità non è nota “a priori”, quindi viene normalmente sostituita con una sua sovrastima  $\hat{f}$ . Quindi la tecnica di generazione di un nuovo punto viene adattata di conseguenza ottenendo un punto che è la realizzazione di una nuova variabile aleatoria.

*Sia  $T$  uno scalare positivo, sia  $\hat{f}$  tale  $\hat{f} \geq f^*$ , sia  $\alpha$  uno scalare scelto a caso su  $[0, 1]$  e sia  $x$  un vettore scelto a caso su  $\mathcal{D}$ , se*

$$\alpha \leq e^{-[f(x)-\hat{f}]_+/T}, \quad (3.17)$$

*dove  $[f(x)-\hat{f}]_+ = \max\{0, f(x)-\hat{f}\}$ , allora il punto  $x$  è una realizzazione della variabile aleatoria distribuita secondo la funzione densità di probabilità  $\hat{p}_T(x)$  data*

$$\hat{p}_T(x) = \frac{e^{-[f(x)-\hat{f}]_+/T}}{\int_{\mathcal{D}} e^{-[f(x)-\hat{f}]_+/T} dx}. \quad (3.18)$$

La nuova funzione densità di probabilità  $\hat{p}_T(x)$  può svolgere un ruolo simile a quello della funzione densità di probabilità originaria  $p_T(x)$ . Infatti, al diminuire della temperatura  $T$ , invece di concentrarsi solamente intorno ai minimi globali si concentra intorno a tutti i punti  $x \in \mathcal{D}$  tali che  $f(x) \leq \hat{f}$ . Se  $\hat{f}$  è il miglior valore della funzione obiettivo

ottenuto, la funzione densità di probabilità  $\hat{p}_T(x)$  assegna maggiore probabilità alle regioni in cui si ha, comunque, un miglioramento della funzione obiettivo.

Una possibile utilizzazione di punti generati con la distribuzione con funzione densità di probabilità  $\hat{p}_T(x)$  è descritta nel seguente algoritmo.

**Algoritmo Simulated Annealing.**

*Passo 0:* si sceglie una temperatura iniziale  $T_0$ , si genera a caso un punto  $x_0$  su  $\mathcal{D}$ , si pone  $x_0^* = x_0$  e  $k = 1$ ;

*Passo 1:* si generano a caso un punto  $x_k$  su  $\mathcal{D}$  ed uno scalare  $\alpha_k$  su  $[0, 1]$ ;

*Passo 2:* se  $\alpha_k > e^{-[f(x_k) - f(x_{k-1}^*)]_+ / T_{k-1}}$  allora si pone  $x_k^* = x_{k-1}^*$  e si va al Passo 6;

*Passo 3:* partendo da  $x_k$  si applica un algoritmo di ottimizzazione locale ottenendo  $y_k$ ;

*Passo 4:* se  $f(y_k) < f(x_{k-1}^*)$  allora si pone  $x_k^* = y_k$

altrimenti si pone  $x_k^* = x_{k-1}^*$ ;

*Passo 5:* si calcola la  $T_k \in (0, T_{k-1}]$ ;

*Passo 6:* si pone  $k = k + 1$  e si ritorna al Passo 1.

Il precedente algoritmo può essere considerato una via di mezzo tra l'Algoritmo Multistart 1, dove le minimizzazioni locali vengono applicate a partire da ogni punto generato a caso su  $\mathcal{D}$ , e l'Algoritmo Multistart 2, dove le minimizzazioni locali vengono applicate solamente a partire dai punti generati a caso su  $\mathcal{D}$  in cui si ha un miglioramento della funzione obiettivo. Grazie al test al Passo 2, nell'Algoritmo Simulated Annealing, le minimizzazioni vengono applicate a partire sia dai punti generati a caso su  $\mathcal{D}$  in cui si ha un miglioramento della funzione obiettivo e sia da punti generati a caso su  $\mathcal{D}$  in cui si ha un peggioramento della funzione obiettivo ma che possono essere considerati delle realizzazioni di variabili aleatorie distribuite secondo la funzione densità di probabilità  $\hat{p}_T(x)$ .

Per quanto riguarda le proprietà di convergenza delle sequenze dei punti generati dall'Algoritmo Simulated Annealing si può stabilire il seguente risultato.

**Proposizione 3.1.5** *Se la sequenza di temperature  $\{T_k\}$ , utilizzate nell'Algoritmo Simulated Annealing, è tale che le funzioni densità  $\hat{p}_{T_k}(x)$  definiscono una sequenza di misure di probabilità  $\{m_k(\cdot)\}$  che soddisfano le ipotesi di Proposizione 3.1.4 e se esiste un minimo globale  $x^*$  non vincolato della funzione  $f$  su  $\mathcal{D}$  per cui l'algoritmo di minimizzazione locale (utilizzato al Passo 2) soddisfa la Assunzione 3.1.1, allora la sequenza  $\{x_k^*\}$  di punti aleatori prodotti dall'Algoritmo Simulated Annealing è tale che:*

$$\lim_{k \rightarrow \infty} \text{Prob} \left\{ x_k^* \in X^* \right\} = 1, \quad (3.19)$$



dove  $X^*$  è l'insieme dei minimi globali del Problema (3.1).

**Prova.** La dimostrazione della proposizione segue utilizzando gli stessi ragionamenti fatti per dimostrare la Proposizione 3.1.3, relativa alle proprietà di convergenza dell'Algoritmo Multistart 1, con la sola differenza di usare la Proposizione 3.1.4 invece della Proposizione 3.1.1  $\square$

Una condizione sufficiente a garantire che la sequenza delle temperature  $T_k$  prodotte dall'algoritmo siano tali da soddisfare l'ipotesi della precedente proposizione è che esista un  $\varepsilon > 0$  tale che

$$T_k \geq \varepsilon, \quad \text{per ogni } k.$$

Condizioni più deboli possono essere determinate notando che, per dimostrare il limite (3.19), è sufficiente che le misure di probabilità derivanti dalle funzioni densità di probabilità  $\hat{p}_T(x)$  soddisfino la (3.9) in corrispondenza solamente ad insiemi  $\mathcal{A}$  con  $\text{meas}(\mathcal{A}) > 0$  e contenuti nell'intorno  $B(x^*; \varepsilon)$  descritto nella Assunzione 3.1.1.

La scelta della sequenza di temperature  $\{T_k\}$ , oltre ad essere legata alla convergenza dell'algoritmo, influenza anche l'efficienza computazione dell'algoritmo. Infatti per temperature più basse si tenderà a produrre punti che hanno maggiore probabilità di essere una buona approssimazione di un minimo globale. Tuttavia questi punti saranno più difficili da generare (si veda il test al Passo 2). Viceversa, a temperature più alte, si genereranno più facilmente i punti ma che, però, avranno meno probabilità di essere buone approssimazioni di un minimo globale.

### 3.1.3 Metodi che usano “popolazioni” di punti (questa sezione non fa parte del programma di esame).

A partire dall'approccio probabilistico si sono sviluppati vari metodi di ottimizzazione globale di tipo euristico. Cioè metodi per cui non si può stabilire nessuna proprietà teorica di convergenza. Questi metodi nascono dall'esigenza di affrontare problemi di ottimizzazione reali. Infatti, per quanto visto nel capitolo precedente (si veda il Teorema 1.6.4 ed il Teorema 1.6.6) e per quanto descritto nei paragrafi precedenti di questo capitolo, le proprietà di convergenza di un metodo sono legate al fatto che, al crescere delle iterazione dell'algoritmo, la funzione obiettivo è valutata “praticamente” su tutto l'insieme ammissibile. Tuttavia seguire, anche in maniera approssimata, questo approccio può essere assolutamente impraticabile nella maggior parte dei casi reali. Per questo motivo sono stati proposti dei metodi che si basano sui seguenti passi:

- ottenere delle informazioni sul comportamento globale della funzione obiettivo sull'insieme ammissibile effettuando inizialmente un campionamento finito della funzione obiettivo su un insieme, detto *popolazione*, iniziale di punti.
- migliorare iterativamente i punti che costituiscono la popolazione attraverso la sostituzione dei punti in cui il valore della funzione obiettivo è più alta con dei nuovi punti ottenuti attraverso delle minimizzazioni locali “approssimate”;

- ripetere il precedente procedimento di miglioramento fino a quando la popolazione non si è “concentrata” intorno ai minimi globali della funzione obiettivo.

Tra le classi di metodi che si basano sull’uso di popolazioni di punti quelle più note e più utilizzate sono:

- *gli algoritmi genetici;*
- *gli algoritmi evolutivi;*
- *metodi di tipo “swarm”.*

Per brevità, nel seguito ci si limiterà a richiamare gli approcci dei citati metodi. Questa scelta è dovuta dalla difficoltà di dare delle trattazioni generali degli algoritmi genetici, degli algoritmi evolutivi e dei metodi di tipo “swarm” in grado di rappresentare le varie scelte algoritmiche particolari.

### - Algoritmi genetici

Nel determinare i nuovi punti della famiglia, gli algoritmi genetici cercano di trarre ispirazione dai processi evolutivi biologici. Infatti negli algoritmi genetici inizialmente proposti le variabili del problema venivano codificate attraverso stringhe di binari, i nuovi punti venivano generati attraverso la ripetizione delle seguenti due operazioni “genetiche”::

- *l’operazione di “crossover”* che consiste nel selezionare a caso due punti della popolazione (punti genitori) e tagliando le stringhe che li rappresentano in corrispondenza di uno stesso indice scelto a caso; i punti figli derivavano dallo scambio delle parti delle stringhe dei genitori (figura 3.1);
- *l’operazione di “mutation”* in cui un vettore della popolazione viene “mutato” cambiandogli una componente della stringa scelta a caso (figura 3.2).

Gli algoritmi genetici più recenti non utilizzano la codifica binaria ma cercano di adattare le due precedenti operazioni alla codifica originaria delle variabili del problema di ottimizzazione da risolvere. Esempi di adattamento delle operazioni di “crossover” e di “mutation” al caso in cui i vettori hanno la loro codifica originaria sono le seguenti:

- *l’operazione di “crossover”* se  $x \in \mathcal{D}$  e  $y \in \mathcal{D}$  sono dei vettori genitori scelti dalla popolazione, le loro componenti vengono mischiate per dare vita a dei nuovi vettori figli, nel seguente modo:

$$\begin{aligned}(\tilde{x})_i &= \alpha^i(x)_i + (1 - \alpha^i)(y)_i, & i = 1, \dots, n, \\(\tilde{y})_i &= \alpha^i(y)_i + (1 - \alpha^i)(x)_i, & i = 1, \dots, n,\end{aligned}$$

dove gli scalari  $\alpha^i$  sono scelti a caso su  $[-0.5, 1.5]$ ;

- l'operazione di "mutation" un vettore della popolazione  $x \in \mathcal{D}$  viene "mutato" perturbargli una sola componente, cioè viene generato il nuovo punto  $\hat{x}$  tale che:

$$\begin{aligned}(\hat{x})_i &= (x)_i + \hat{\alpha}(u - l)_i, \\(\hat{x})_j &= (x)_j, \quad j = 1, \dots, n, \quad j \neq i,\end{aligned}$$

dove l'indice  $i$  è scelto a caso e  $\hat{\alpha} = 0.1$  oppure è scelto a caso tra  $[-0.1, 0.1]$ .

### - Algoritmi evolutivi

La caratteristica che distingue gli algoritmi evolutivi dagli algoritmi genetici è il fatto che, ad ogni iterazione, questi algoritmi cercano di aggiornare tutti i punti della popolazione.

Più in particolare, essi cercano di sostituire ogni punto  $x \in \mathcal{D}$  della popolazione con un nuovo punto  $y \in \mathcal{D}$  ottenuto dalla combinazione di operazioni di "crossover" e di "mutation". Se nel nuovo punto  $y$  si ha un miglioramento della funzione obiettivo allora il punto di partenza  $x$  viene sostituito dal nuovo punto  $y$ , altrimenti nella popolazione viene mantenuto il punto  $x$ .

Le tecniche con cui vengono questi nuovi punti variano in maniera significativa da algoritmo ad algoritmo. Un esempio di una tale tecnica è il seguente:

scelto un punto  $x \in \mathcal{D}$  della popolazione il nuovo punto  $y \in \mathcal{D}$  si determina a partire da  $x$  e da un punto  $\hat{x} \in \mathcal{D}$  ottenuto dalla seguente operazione:

$$\hat{x} = x^a + \hat{\alpha}(x^b - x^c),$$

dove  $\hat{\alpha} \in (0, 1]$  e  $x^a, x^b, x^c$  sono scelti a caso nella popolazione di punti;

il nuovo punto  $y$  è dato, quindi, dalle seguenti relazioni:

$$\begin{aligned}(y)_i &= (\hat{x})_i \quad \text{se } \alpha^i \leq 0.5 \quad \text{oppure } i = \hat{i}, \\(y)_i &= (x)_i \quad \text{se } \alpha^i > 0.5 \quad \text{e } i \neq \hat{i},\end{aligned}$$

dove l'indice  $\hat{i}$  è scelto a caso tra  $\{1, \dots, n\}$  e gli scalari  $\alpha^i, i = 1, \dots, n$ , sono scelti a caso tra  $[0, 1]$ .

Negli algoritmi evolutivi più recenti si cerca di migliorare ogni punto della popolazione sfruttando anche il più possibile tutte le informazioni sulla funzione obiettivo contenute nei punti presenti nella popolazione.

### - Metodi di tipo "swarm"

I metodi di tipo "swarm" hanno in comune con gli algoritmi evolutivi l'idea di cercare aggiornare, ad ogni iterazione, tutti i punti della popolazione. Ma, a differenza di quest'ultimi, non traggono ispirazione dai processi evolutivi ma piuttosto dal comportamento degli sciami (swarm). Infatti essi interpretano i vettori che formano la famiglia

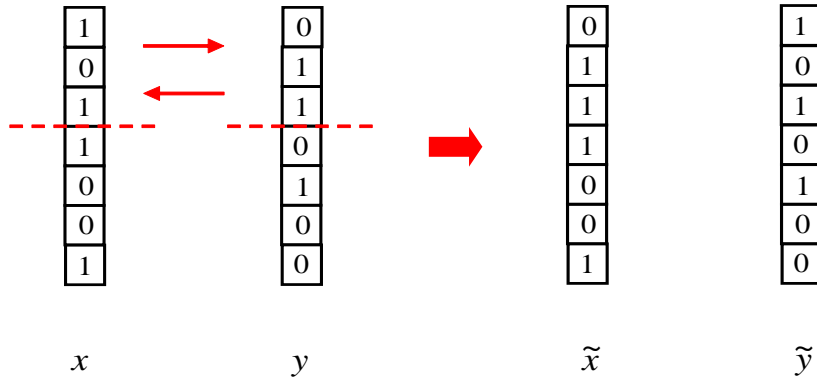


Figura 3.1: Esempio dell'operazione di "crossover".

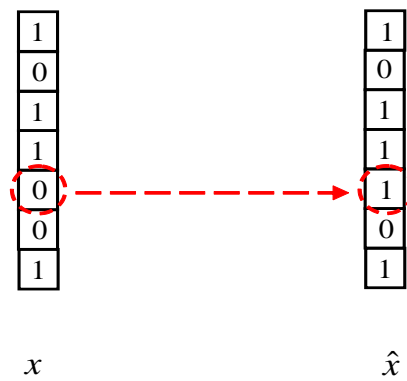


Figura 3.2: Esempio dell'operazione di "mutation".

all'iterazione  $k$ -esima come punti che identificano la posizione dei membri di uno sciame all'istante  $k$ -esimo. Quindi al variare delle iterazione, l'evoluzione della popolazione è associata al movimento complessivo dello sciame.

Lo stato dello sciame al generico istante  $k$ -esimo è identificato dalle “posizioni” degli elementi dello sciame

$$S_k = \{x_k^1, \dots, x_k^m\},$$

e dalle corrispondenti velocità

$$V_k = \{v_k^1, \dots, v_k^m\}.$$

Nel definire le posizioni degli elementi all'istante successivo  $k + 1$  (cioè nel definire il nuovo insieme  $S_{k+1}$ ), questi metodi cercano di simulare il comportamento in natura dei membri di uno sciame. Infatti il movimento di ciascun elemento tiene conto della esperienza individuale (che può essere rappresentata dal miglior punto incontrato durante il movimento del  $i$ -esimo membro dello sciame) e dalla esperienza globale dello sciame (che può essere rappresentata dal miglior punto incontrato durante il movimento di tutti i membri dello sciame).

In particolare, un esempio di formule utilizzate dai metodi di tipo “swarm” per descrivere il movimento della famiglia di punti è il seguente:

$$\begin{aligned} v_{k+1}^i &= v_k^i + cr_k^1(\bar{x}_k^i - x_k^i) + cr_k^2(x_k^* - x_k^i), & i = 1, \dots, m, \\ x_{k+1}^i &= x_k^i + v_{k+1}^i, & i = 1, \dots, m. \end{aligned}$$

dove

$$\begin{aligned} f(\bar{x}_k^i) &= \min_{h=1, \dots, k} f(x_h^i), \\ f(x_k^*) &= \min_{i=1, \dots, m} f(\bar{x}_k^i), \end{aligned}$$

dove  $c$  è una costante positiva, chiamata costante di accelerazione,  $r_k^1$  e  $r_k^2$  sono dei scalari scelti a caso su  $[0, 1]$ .

Nei metodi di tipo “swarm” più recenti, per controllare meglio l'evoluzione delle velocità ed evitare che siano usati vettori  $v_k^i$  troppo grandi, vengono introdotti ulteriori coefficienti nella formula che fornisce le nuove velocità:

$$v_{k+1}^i = \chi(wv_k^i + c^1 r_k^1(\bar{x}_k^i - x_k^i) + c^2 r_k^2(x_k^* - x_k^i)), \quad i = 1, \dots, m.$$

dove  $w$  è una costante chiamata inerzia,  $c^1$  e  $c^2$  sono costanti positive chiamate coefficiente cognitivo e coefficiente sociale,  $\chi$  è una costante chiamata fattore di costrizione.

## 3.2 Metodi che utilizzano partizioni dell'insieme ammissibile

In questo capitolo viene brevemente considerata una classe di metodi deterministici di ottimizzazione globale che si basano sull'idea di cercare di produrre sequenze di

partizioni dell'insieme ammissibile che tendano ad essere più addensate nelle regioni in cui ci sono i minimi globali della funzione obiettivo.

Per descrivere in maniera formale questa nuova classe di metodi è necessario introdurre la definizione di partizione di un insieme.

**Definizione 3.2.1 (Partizione di un insieme)** *Dato un insieme  $\mathcal{F} \subseteq R^n$ , una collezione di insiemi  $\{\mathcal{F}^i : i \in I\}$  è una partizione dell'insieme  $\mathcal{F}$  se si ha:*

$$\mathcal{F} = \bigcup_{i \in I} \mathcal{F}^i,$$

$$\mathcal{F}^i \cap \mathcal{F}^j = \partial \mathcal{F}^i \cap \partial \mathcal{F}^j, \quad \forall i, j \in I, \quad i \neq j;$$

dove  $\partial \mathcal{F}^j$  indica la frontiera dell'insieme  $\mathcal{F}^j$ .

### 3.2.1 Schema generale di un metodo che usa partizioni dell'insieme ammissibile

Nel caso particolare in cui l'insieme ammissibile è costituito dall'insieme  $\mathcal{D} = \{x \in R^n : l \leq x \leq u\}$  un modo naturale di partizionarlo è quello di suddividerlo in sottoinsiemi che hanno la stessa struttura. Seguendo questa linea, in questo capitolo, si considerano metodi che hanno come punto caratterizzante il fatto che, ad ogni iterazione, viene prodotta una partizione  $\{\mathcal{D}^i : i \in I_k\}$  dell'insieme ammissibile  $\mathcal{D}$  con

$$\mathcal{D}^i = \{x \in R^n : l^i \leq x \leq u^i\}, \quad i \in I_k.$$

Un modello generale di un algoritmo che genera una sequenza di partizioni dell'insieme ammissibile può essere descritto dal seguente schema.

### Algoritmo di Partizione

*Passo 0:* Si pone  $\mathcal{D}^0 = \mathcal{D}$ ,  $l^0 = l$ ,  $u^0 = u$ ,  $I_0 = \{0\}$  e  $k = 0$ ;

*Passo 1:* data la partizione  $\{\mathcal{D}^i : i \in I_k\}$  di  $\mathcal{D}$  con

$$\mathcal{D}^i = \{x \in R^n : l^i \leq x \leq u^i\}, \quad \text{per ogni } i \in I_k,$$

si identifica un sottoinsieme di indici  $I_k^* \subseteq I_k$ ;

si pone  $\bar{I}^0 = I_k$ ,  $\hat{I}^0 = I_k^*$  e  $p = 0$

*Passo 2:* si sceglie un indice  $h \in \hat{I}^p$  e si partiziona  $\mathcal{D}^h$  in  $m \geq 2$  sottointervalli

$$\mathcal{D}^{h_1}, \mathcal{D}^{h_2}, \dots, \mathcal{D}^{h_m}.$$

*Passo 3:* si pone:

$$\begin{aligned} \bar{I}^{p+1} &= \bar{I}^p \cup \bigcup_{j=1, \dots, m} \{h_j\} \setminus \{h\}, \\ \hat{I}^{p+1} &= \hat{I}^p \setminus \{h\}, \end{aligned}$$

se  $\hat{I}^{p+1} \neq \emptyset$  si pone  $p = p + 1$  e si torna al Passo 2;

*Passo 4:* si definisce la nuova partizione  $\{\mathcal{D}^i : i \in I_{k+1}\}$  con

$$I_{k+1} = \bar{I}^{p+1},$$

si pone  $k = k + 1$  e si torna al Passo 1.

Al Passo 1, data una partizione dell'insieme ammissibile, l'algoritmo seleziona (secondo un criterio non ancora specificato) un certo numero di sottoinsiemi (identificati dall'insieme di indici  $I_k^*$ ) appartenenti alla partizione. Nel passo 2 e nel passo 3 ognuno dei sottointervalli selezionati viene partizionato (utilizzando una tecnica anche essa non specificata) in un numero  $m \geq 2$  di sottoinsiemi.

Per effettuare un'analisi di alcune proprietà generali del precedente algoritmo è necessario caratterizzare le tecniche di partizione utilizzabili attraverso il fatto che siano in grado di soddisfare la seguente ipotesi.

**Assunzione 3.2.1** *Esistono dei scalari  $\varepsilon_1$ ,  $\varepsilon_2$  e  $\varepsilon_3$ , con  $0 < \varepsilon_1 < \varepsilon_2 < 1$  e  $\varepsilon_3 \in (0, 1]$ , tali che ogni insieme  $\mathcal{D}^h$ ,  $h \in I_k^*$ , selezionato dall'algoritmo in una qualsiasi iterazione  $k$ , è partizionato in  $m$  sottoinsiemi  $\mathcal{D}^{h_j}$ ,  $j = 1, \dots, m$  che soddisfano:*

$$\varepsilon_1 \|u^h - l^h\| \leq \|u^{h_j} - l^{h_j}\| \leq \varepsilon_2 \|u^h - l^h\|, \quad j = 1, \dots, m; \quad (3.20)$$

$$\frac{(u^{h_j} - l^{h_j})_{i_{\min}}}{(u^{h_j} - l^{h_j})_{i_{\max}}} \geq \varepsilon_3 \frac{(u^h - l^h)_{i_{\min}}}{(u^h - l^h)_{i_{\max}}}. \quad (3.21)$$

dove, per un qualsiasi vettore  $v \in R^n$ ,  $i_{\min}$  e  $i_{\max}$  sono dati da  $(v)_{i_{\min}} = \min_{i=1, \dots, n}(v)_i$  e  $(v)_{i_{\max}} = \max_{i=1, \dots, n}(v)_i$

A differenza degli algoritmi di minimizzazione locale, che sono caratterizzati dalla sequenza di punti che generano, l'evoluzione dell'Algoritmo di Partizione è rappresentata dallo sviluppo dei sottoinsiemi generati. Perciò le sue proprietà teoriche possono essere analizzate caratterizzando le sequenze di insiemi che vengono prodotte al tendere all'infinito delle iterazioni dell'algoritmo.

Queste sequenze di insiemi possono essere individuate facendo corrispondere ad ogni sottoinsieme  $\mathcal{D}^{i_k}$ , con  $i_k \in I_k$ , della partizione dell'insieme  $\mathcal{D}$  della generica iterazione  $k$ -esima, un particolare sottoinsieme  $\mathcal{D}^{i_{k-1}}$ , con  $i_{k-1} \in I_{k-1}$  della partizione della iterazione precedente. In particolare, l'insieme  $\mathcal{D}^{i_{k-1}}$  è definito nelle seguente maniera:

- se il sottoinsieme  $\mathcal{D}^{i_k}$  nasce da un un processo di divisione allora  $\mathcal{D}^{i_{k-1}}$  è il sottoinsieme che lo ha generato;
- se, invece, il sottoinsieme  $\mathcal{D}^{i_k}$  non deriva da un un processo di partizione allora si pone  $\mathcal{D}^{i_{k-1}} = \mathcal{D}^{i_k}$ .

Ripetendo lo stesso procedimento, si può associare all'insieme  $\mathcal{D}^{i_{k-1}}$  un insieme  $\mathcal{D}^{i_{k-2}}$ , con  $i_{k-2} \in I_{k-2}$ , della partizione della iterazione  $k-2$ , a quest'ultimo si può associare uno della partizione della iterazione  $k-3$  e così via fino ad arrivare all'insieme  $\mathcal{D}^{i_0} = \mathcal{D}$ . Quindi, ad ogni sottoinsieme  $\mathcal{D}^{i_k}$ , con  $i_k \in I_k$ , della partizione della iterazione  $k$ -esima si può far corrispondere la seguente collezione di insiemi

$$\mathcal{D}^{i_0}, \mathcal{D}^{i_1}, \dots, \mathcal{D}^{i_{k-1}}, \mathcal{D}^{i_k},$$

Al crescere delle iterazioni dell'algoritmo, la precedente collezione di insiemi produce una *sequenza annidata*  $\{\mathcal{D}^{i_k}\}$  di insiemi cioè una sequenza tale che, per ogni  $k$ , si ha:

$$\mathcal{D}^{i_{k+1}} \subseteq \mathcal{D}^{i_k}.$$

Inoltre, ad ogni iterazione, le istruzioni del Passo 2 implicano che un determinato intervallo viene suddiviso in un numero prefissato di sottointervalli e, quindi, il numero di sottoinsiemi dell'insieme ammissibile aumenta di una quantità fissa. Tra le sequenze annidate prodotte dall'algoritmo alcune sono particolarmente significative, in particolare si ha la sequenza definizione.

**Definizione 3.2.2** Una sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione è detta strettamente annidata se

$$\mathcal{D}^{i_{k+1}} \subset \mathcal{D}^{i_k}, \quad \text{per un numero infinito di volte.} \quad (3.22)$$

Perciò, per le sequenze non strettamente annidate, esiste un indice  $k_0$  dopo il quale il sottoinsieme  $\mathcal{D}^{i_{k_0}}$  non viene più suddiviso, cioè:

$$\mathcal{D}^{i_k} = \mathcal{D}^{i_{k_0}}, \quad \text{per tutti } k \geq k_0. \quad (3.23)$$



Mentre per quelle strettamente annidate il processo di divisione continua un numero infinto di volte.

Le tre seguenti proposizioni descrivono delle caratterizzazioni matematiche delle sequenze di insiemi strettamente annidate prodotte dall'Algoritmo di Partizione.

**Proposizione 3.2.3** *Se l'Assunzione 3.2.1 è soddisfatta, una sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione è strettamente annidata se e solamente se*

$$\lim_{k \rightarrow \infty} \|u^{i_k} - l^{i_k}\| = 0. \quad (3.24)$$

**Prova.** Sia  $\{\mathcal{D}^{i_k}\}$  una sequenza prodotta dall'Algoritmo di Partizione. Dalla definizione della sequenza  $\{\mathcal{D}^{i_k}\}$  e dalle istruzioni dell'algoritmo, ogni volta che accade che

$$\mathcal{D}^{i_k} \subset \mathcal{D}^{i_{k-1}},$$

significa che, nell'iterazione k-1, il sottoinsieme  $\mathcal{D}^{i_k}$  è stato generato dalla partizione dell'insieme  $\mathcal{D}^{i_{k-1}}$ . Dall'Assunzione 3.2.1 si ha

$$\varepsilon_1 \|u^{i_{k-1}} - l^{i_{k-1}}\| \leq \|u^{i_k} - l^{i_k}\| \leq \varepsilon_2 \|u^{i_{k-1}} - l^{i_{k-1}}\|, \quad (3.25)$$

Ricordando nuovamente la definizione della sequenza di insiemi  $\{\mathcal{D}^{i_k}\}$  si ha che, per  $j = 1, \dots, k$ ,

$$\mathcal{D}^{i_j} \subset \mathcal{D}^{i_{j-1}} \quad \text{oppure} \quad \mathcal{D}^{i_j} = \mathcal{D}^{i_{j-1}}.$$

Applicando ripetutamente la (3.25) si ha

$$(\varepsilon_1)^{p_k} \|u^0 - l^0\| \leq \|u^{i_k} - l^{i_k}\| \leq (\varepsilon_2)^{p_k} \|u^0 - l^0\|, \quad (3.26)$$

dove  $p_k$  indica quante volte il processo di divisione del Passo 2 è intervenuto per generare l'insieme  $\mathcal{D}^{i_k}$  oppure, equivalentemente, quante volte si ha la stretta inclusione tra due insiemi successivi nella sequenza  $\mathcal{D}^{i_0}, \mathcal{D}^{i_1}, \dots, \mathcal{D}^{i_{k-1}}, \mathcal{D}^{i_k}$ .

Ora, se la sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'algoritmo è strettamente annidata segue, per definizione, che

$$\lim_{k \rightarrow \infty} p_k = \infty,$$

da cui, utilizzando la (3.26) e il fatto che  $\varepsilon_2 \in (0, 1)$ , si ottiene che

$$\lim_{k \rightarrow \infty} \|u^{i_k} - l^{i_k}\| \leq \lim_{k \rightarrow \infty} (\varepsilon_2)^{p_k} \|u^0 - l^0\| = 0.$$

Viceversa, se la sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'algoritmo soddisfa il limite (3.24), dalla (3.26) segue che

$$\lim_{k \rightarrow \infty} (\varepsilon_1)^{p_k} \|u^0 - l^0\| \leq \lim_{k \rightarrow \infty} \|u^{i_k} - l^{i_k}\| = 0.$$

La precedente relazione e il fatto che  $\varepsilon_1 \in (0, 1)$  implicano che

$$\lim_{k \rightarrow \infty} p_k = \infty,$$

da cui segue che la sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'algoritmo è strettamente annidata.  $\square$

**Proposizione 3.2.4** *Se l'Assunzione 3.2.1 è soddisfatta, una sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione è strettamente annidata se e solamente se*

$$\lim_{k \rightarrow \infty} u^{i_k} = \bar{x} \quad (3.27)$$

$$\lim_{k \rightarrow \infty} l^{i_k} = \bar{x}; \quad (3.28)$$

oppure, equivalentemente

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\}.$$

**Proposizione 3.2.5** *Se l'Assunzione 3.2.1 è soddisfatta, una sequenza  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione è strettamente annidata e  $\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\}$  se e solamente se, per ogni  $\varepsilon > 0$ , esiste un indice  $\bar{k}$  tale che per ogni  $k \geq \bar{k}$  si ha:*

$$\mathcal{D}^{i_k} \subset B(\bar{x}; \varepsilon). \quad (3.29)$$

Dopo aver descritto alcune proprietà delle sequenze di insiemi strettamente annidate prodotte dall'Algoritmo di Partizione, nella seguente proposizione si affronta il problema di garantire che tale algoritmo produca almeno una di queste sequenze.

**Proposizione 3.2.6** *Se l'Assunzione 3.2.1 è soddisfatta, l'Algoritmo di Partizione genera almeno una sequenza strettamente annidata di insiemi  $\{\mathcal{D}^{i_k}\}$ .*

**Prova.** La dimostrazione segue ricordando i seguenti due punti caratterizzanti l'Algoritmo di Partizione:

- alla generica iterazione  $k$ -esima, l'algoritmo ha prodotto una partizione  $\{\mathcal{D}^i, i \in I_k\}$  dell'insieme ammissibile  $\mathcal{D}$  tale che:

$$\mathcal{D} = \bigcup_{i \in I_k} \mathcal{D}^i, \quad \text{Int}(\mathcal{D}^i) \cap \text{Int}(\mathcal{D}^j) = \emptyset, \quad \forall i, j \in I_k, \quad i \neq j; \quad (3.30)$$

- ripetendo gli argomenti utilizzati nella dimostrazione della Proposizione 3.2.3 segue che l'applicazione ripetuta dell'Assunzione 3.2.1 implica, per ogni  $i \in I_k$ ,

$$(\varepsilon_1)^{p_i} \|u^0 - l^0\| \leq \|u^i - l^i\| \leq (\varepsilon_2)^{p_i} \|u^0 - l^0\|; \quad (3.31)$$

$$\frac{(u^i - l^i)_{i_{\min}}}{(u^i - l^i)_{i_{\max}}} \geq (\varepsilon_3)^{p_i} \frac{(u^0 - l^0)_{i_{\min}}}{(u^0 - l^0)_{i_{\max}}}. \quad (3.32)$$

dove  $p_i$  indica il numero di volte in cui processo di divisione è intervenuto per generare l'insieme  $\mathcal{D}^i$  a partire dall'insieme iniziale  $\mathcal{D}$ .

Dalle (3.31) si ha

$$\sqrt{n}(u^i - l^i)_{i_{\max}} \geq \|u^i - l^i\| \geq (\varepsilon_1)^{p_i} \|u^0 - l^0\|.$$

che utilizzata nella (3.32) implica

$$(u^i - l^i)_{i_{\min}} \geq (\varepsilon_1 \varepsilon_3)^{p_i} \frac{(u^0 - l^0)_{i_{\min}}}{(u^0 - l^0)_{i_{\max}}} \frac{\|u^0 - l^0\|}{\sqrt{n}}. \quad (3.33)$$

Ora, si supponga per assurdo che l'algoritmo non generi nessuna sequenza strettamente annidata. Questo implicherebbe l'esistenza di un valore  $\bar{p}$  tale che, per qualsiasi  $k$ , limiterebbe superiormente i numeri di processi di divisione che avrebbero generato i sottoinsiemi  $\mathcal{D}^i$ , con  $i \in I_k$ , della partizione di  $\mathcal{D}$  nella iterazione  $k$ -esima. Cioè, per ogni  $k$  e per ogni  $i \in I_k$ , si avrebbe che:

$$p_i \leq \bar{p}. \quad (3.34)$$

Le (3.33) e (3.34) implicherebbero che, per ogni  $k$  e per ogni  $i \in I_k$ ,

$$(u^i - l^i)_{i_{\min}} \geq (\varepsilon_1 \varepsilon_3)^{\bar{p}} \frac{(u^0 - l^0)_{i_{\min}}}{(u^0 - l^0)_{i_{\max}}} \frac{\|u^0 - l^0\|}{\sqrt{n}}.$$

da cui se ne ricaverebbe l'esistenza di una costante  $\bar{\delta}$  tale che, per ogni  $k$  e per ogni  $i \in I_k$ , risulterebbe:

$$meas(\mathcal{D}^i) \geq \bar{\delta}. \quad (3.35)$$

Dalle istruzioni dell'algoritmo si ha che, ad ogni iterazione, viene generata una nuova partizione con un numero di sottoinsiemi che è aumentato di  $m \geq 2$  elementi rispetto alla partizione precedente. Quindi, per  $k$  tendente all'infinito, il numero di sottoinsiemi che costituiscono le partizioni tende all'infinito. Questo può essere espresso equivalentemente con

$$\lim_{k \rightarrow \infty} |I_k| = \infty, \quad (3.36)$$

dove  $|I_k|$  indica la cardinalità dell'insieme  $I_k$ , cioè il numero di indici che lo compongono.

Le (3.30), (3.35) e (3.36) produrrebbero un assurdo con la compattezza dell'insieme ammissibile  $\mathcal{D}$ . Infatti le (3.30), (3.35) e (3.36) implicherebbero che un insieme compatto può essere partizionato in un numero arbitrariamente grande di insiemi che hanno misura uniformemente diversa da zero.  $\square$

I risultati descritti in questa sottosezione derivano scegliendo la tecnica di partizione utilizzata nello schema generale dell'Algoritmo di Partizione in maniera che l'Assunzione 3.2.1 sia soddisfatta.

Le prossime due sottosezioni sono dedicate all'analisi del comportamento asintotico dell'algoritmo nel caso di due differenti approcci per la scelta dei sottoinsiemi da partizionare.

### 3.2.2 Scelta dei sottoinsiemi da partizionare: sottoinsiemi di dimensioni più grandi.

Una possibile caratteristica su cui possono basarsi i metodi che utilizzano partizioni è quella di cercare di localizzare i minimi globali tentando di analizzare il comportamento della funzione obiettivo su insiemi sempre più piccoli. Infatti al diminuire delle dimensioni dell'insieme, le informazioni locali della funzioni diventano sempre più attendibili. Seguendo questa impostazione, la scelta dei sottoinsiemi da partizionare deve garantire che, al crescere delle iterazioni, le dimensioni di tutti gli insiemi che costituiscono la partizione tendano a diminuire. In questa sottosezione viene mostrato che questa caratteristica può essere garantita richiedendo che almeno uno dei sottoinsiemi di dimensione massima è partizionato, ad ogni iterazione.

Per formalizzare questa modalità di scelta dei sottoinsiemi da partizionare ed effettuare l'analisi teorica è necessario introdurre alcune nuove notazioni. Data una partizione  $\{\mathcal{D}^i : i \in I_k\}$  si identifica l'ampiezza massima dei sottointervalli con:

$$d_k^{max} = \max_{i \in I_k} \|u^i - l^i\|, \quad (3.37)$$

e l'insieme degli indici dei sottoinsiemi che hanno ampiezza massima con:

$$I_k^{max} = \{i \in I_k : \|u^i - l^i\| = d_k^{max}\}. \quad (3.38)$$

A questo punto si può introdurre la seguente assunzione.

**Assunzione 3.2.2** *Esiste un infinito sottoinsieme di indici  $K \subseteq \{1, 2, 3, \dots\}$ , tale che  $I_k^{max} \cap I_k^* \neq \emptyset$  per ogni  $k \in K$ .*

La proposizione seguente mostra che, se almeno un sottoinsieme di dimensione massima viene selezionato in un sottoinsieme infinito di iterazione, il processo di partizione di ogni sottoinsieme generato non viene interrotto mai.

**Proposizione 3.2.7** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.2 siano soddisfatte. Siano  $\{\mathcal{D}^i, i \in I_k\}$  la partizione generata dall'Algoritmo di Partizione all'iterazione  $k$ -esima e  $d_k^{max}$  la quantità data dalla (3.37). Allora*

$$\lim_{k \rightarrow \infty} d_k^{max} = 0. \quad (3.39)$$

**Prova (la prova non fa parte del programma di esame).** Dalle istruzioni dell'Algoritmo di Partizione si ha che la sequenza di scalari  $\{d_k^{max}\}$  è non crescente e limitata inferiormente. Perciò la sequenza  $\{d_k^{max}\}$  è convergente. Si supponga, per assurdo, che il limite (3.39) non sia verificato e che, quindi, esista un valore  $\bar{\delta} > 0$  tale che, per tutti  $k$ , si abbia:

$$d_k^{max} > \bar{\delta}. \quad (3.40)$$

Dalla compattezza dell'insieme  $\mathcal{D}$  si ha che il numero di sottointervalli che hanno ampiezza

$$\|u^i - l^i\| > \bar{\delta}$$

è limitato superiormente da un numero  $N$ . Infatti, in caso contrario, si avrebbe la contraddizione che l'insieme compatto  $\mathcal{D}$  potrebbe essere partizionato in un numero infinito di sottointervalli con misura diversa da zero. In fatti per ogni sottointervallo con ampiezza della diagonale maggiori della quantità  $\bar{\delta}$  si avrebbe, ricordando la (3.32),

$$(u^i - l^i)_{i_{\min}} \geq \varepsilon_3^{p_i} \frac{(u^0 - l^0)_{i_{\min}} \|u^i - l^i\|}{(u^0 - l^0)_{i_{\max}} \sqrt{n}} \geq \bar{\delta}. \quad (3.41)$$

in cui si tenuto in conto che  $\sqrt{n}(u^i - l^i)_{i_{\max}} \geq \|u^i - l^i\|$ .

Per ogni iterazione  $k$ , si può definire il seguente insieme di indici

$$\tilde{I}_k(\bar{\delta}) = \{i \in I_k : \|u^i - l^i\| > \bar{\delta}\},$$

cioè l'insieme degli indici dei sottointervalli che hanno lunghezza della diagonale strettamente maggiore della costante  $\bar{\delta}$ . Da quanto detto precedentemente, si ha che il numero di questi indici è limitato superiormente da un numero  $N$ , cioè

$$|\tilde{I}_k(\bar{\delta})| \leq N, \quad \text{per ogni } k.$$

Ricordando la definizione dell'insieme  $I_k^{max}$  (data dalla (3.38)), si ha anche:

$$I_k^{max} \subseteq \tilde{I}_k(\bar{\delta}).$$

Poichè si è supposto che sia vera l'Assunzione 3.2.2, l'insieme di iterazioni  $K = \{k : I_k^{max} \cap I_k^* \neq \emptyset\}$  è infinito. Per ogni  $k \in K$  esiste un indice  $i_k^{max} \in I_k^{max} \cap I_k^*$ . Quindi l'insieme  $\mathcal{D}^{i_k^{max}}$  è partizionato dall'algoritmo dando luogo a dei nuovi sottointervalli  $\mathcal{D}^{h_j}$ ,  $j = 1, \dots, m$ , che, ricordando l'Assunzione 3.2.1, hanno ampiezze che soddisfano la seguente relazione:

$$\|u^{h_j} - l^{h_j}\| \leq \varepsilon_2 \|u^{i_k^{max}} - l^{i_k^{max}}\| = \varepsilon_2 d_k^{max}, \quad j = 1, \dots, m \quad (3.42)$$

con  $\varepsilon_2 \in (0, 1)$ .

Dati due indici di iterazioni  $k < \bar{k}$  si definisce con  $K(k, \bar{k})$  l'insieme di iterazioni che appartengono a  $K$  e che sono comprese tra  $k$  e  $\bar{k}$ , cioè:

$$K(k, \bar{k}) = \{h \in K : k \leq h \leq \bar{k}\}.$$

Poichè  $K$  è infinito esiste un indice  $\bar{k}(N)$  tale che tra  $k$  e  $\bar{k}(N)$  ci sono  $N$  iterazioni che appartengono a  $K$ , cioè:

$$|K(k, \bar{k}(N))| = N.$$

Nelle  $N$  iterazioni  $h \in K(k, \bar{k}(N))$ , dalla (3.40) seguirebbe che

$$d_h^{max} > \bar{\delta}, \quad h \in K(k, \bar{k}(N)) \quad (3.43)$$

ma questa relazione implicherebbe che, in queste  $N$  iterazioni, la Procedura di Partizione sarebbe stata applicata a tutti gli insiemi  $\mathcal{D}^i$ ,  $i \in \tilde{I}_k(\bar{\delta})$ , oppure che sarebbe stata

applicata a due insiemi strettamente annidati tra di loro. In entrambi i casi, ricordando la (3.42), si avrebbe:

$$d_{k(N)}^{\max} \leq \varepsilon_2 d_k^{\max}.$$

Ripetendo gli stessi ragionamenti e definendo con  $\bar{k}(pN)$  l'indice tale che:

$$|K(k, \bar{k}(pN))| = pN.$$

Assumendo vera la (3.40), per le iterazioni  $h \in K(k, \bar{k}(pN))$ , si otterrebbe che:

$$d_{\bar{k}(pN)}^{\max} \leq (\varepsilon_2)^p d_k^{\max}, \quad \text{con} \quad p = 1, 2, \dots$$

che, per valori sufficientemente grandi di  $p$  implicherebbe che

$$d_{\bar{k}(pN)}^{\max} < \bar{\delta}$$

che produrrebbe una contraddizione con la (3.40).  $\square$

Dalla precedente proposizione segue che l'Algoritmo di Partizione, con la precedente richiesta sulla scelta dei sottoinsiemi da partizionare, genera un insieme di punti che, al crescere delle iterazioni, tende a diventare un insieme denso su  $\mathcal{D}$ , come è formalizzato dal seguente risultato.

**Proposizione 3.2.8** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.2 siano soddisfatte. Allora si ha che:*

- i) tutte le sequenze di insiemi  $\{\mathcal{D}^{i_k}\}$  generate dall'Algoritmo di Partizione sono strettamente annidate;*
- ii) per ogni  $\tilde{x} \in \mathcal{D}$  l'Algoritmo di Partizione genera una sequenza di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata tale che*

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

**Prova.** *Punto i).* Per ogni sequenza di insiemi  $\{\mathcal{D}^{i_k}\}$  generata dall'Algoritmo di Partizione, dalla definizione di  $d_k^{\max}$  e dalla Proposizione 3.2.7, si ottiene che:

$$\lim_{k \rightarrow \infty} \|u^{i_k} - l^{i_k}\| \leq \lim_{k \rightarrow \infty} d_k^{\max} = 0,$$

da cui, utilizzando la Proposizione 3.2.3, si ha che la sequenza è strettamente annidata.

*Punto ii).* Poichè l'algoritmo genera una sequenza di partizioni  $\{\mathcal{D}^i : i \in I_k\}$  dell'insieme ammissibile  $\mathcal{D}$ , in ogni iterazione è possibile identificare un sottoinsieme  $\mathcal{D}^{\ell_k}$ , con  $\ell_k \in I_k$ , che contiene il punto  $\tilde{x}$ , cioè i cui estremi soddisfano, per ogni  $k$ :

$$l^{\ell_k} \leq \tilde{x} \leq u^{\ell_k}. \quad (3.44)$$

Dalla Proposizione 3.2.8 si ha che la sequenza di insiemi  $\{\mathcal{D}^{\ell_k}\}$  è strettamente annidata. Facendo i limiti, per  $k$  tendente all'infinito, dei tre termini delle disequazioni (3.44) e ricordando la Proposizione 3.2.4 segue la dimostrazione della proposizione.  $\square$

Dal precedente risultate seguono i seguenti corollari.

**Corollario 3.2.9** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.2 siano soddisfatte. Sia  $\{D^i, i \in I_k\}$  la partizione generata dall'Algoritmo di Partizione all'iterazione  $k$ -esima. Per ogni  $\bar{x} \in \mathcal{D}$  e per ogni  $\varepsilon > 0$  esiste una iterazione  $k$  dell'Algoritmo di Partizione in cui c'è un indice  $h_k \in I_k$  tale che:*

$$\mathcal{D}^{h_k} \subset B(\bar{x}; \varepsilon).$$

**Corollario 3.2.10** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.2 siano soddisfatte. Sia  $\{D^i, i \in I_k\}$  la partizione generata dall'Algoritmo di Partizione all'iterazione  $k$ -esima. Per ogni minimo globale  $x^*$  di  $f(x)$  su  $\mathcal{D}$  e per ogni  $\varepsilon > 0$  esiste una iterazione  $k$  dell'Algoritmo di Partizione in cui c'è un indice  $h_k \in I_k$  tale che:*

$$\mathcal{D}^{h_k} \subset B(x^*; \varepsilon).$$

### 3.2.3 Scelta dei sottoinsiemi da partizionare: sottoinsiemi più promettenti.

Nella sottosezione precedente si è considerato il caso in cui l'Algoritmo di Partizione, nella scelta dei sottoinsiemi da partizionare, includesse uno di dimensioni massime. In questa sottosezione si analizza una strategia diversa in cui l'idea base è quella di selezionare i sottoinsiemi considerati più promettenti. I punti caratterizzanti di questa scelta dei sottoinsiemi da partizionare sono i seguenti:

- data una partizione  $\{D^i : i \in I_k\}$  dell'insieme ammissibile  $\mathcal{D}$ ;
- per ogni sottoinsieme  $D^i$ ,  $i \in I_k$ , viene calcolato uno scalare  $R_k^i$  che fornisce una stima del valore minimo che assume la funzione obiettivo nel sottoinsieme considerato;
- vengono determinati gli insiemi  $\mathcal{D}^h$ , a cui corrispondono gli scalari  $R_k^h$  a valore più basso;
- uno dei precedenti insiemi viene suddiviso ulteriormente (in alcuni algoritmi vengono suddivisi tutti).

Per incominciare ad analizzare formalmente questa seconda strategia di scelta dei sottoinsiemi da partizionare è utile introdurre la seguente ipotesi.

**Assunzione 3.2.3** *Per ogni  $k$ , si ha  $I_k^* = \{h\}$  dove l'indice  $h$  è tale che*

$$R_k^h = \min_{i \in I_k} R_k^i,$$

*con i valori d'interesse  $R_k^i$ ,  $i \in I_k$ , che soddisfano a:*

- i) per ogni sequenza strettamente annidata di insiemi  $\{D^{i_k}\}$ , con  $\bigcap_{k=0}^{\infty} D^{i_k} = \{\bar{x}\}$ , si ha*

$$\lim_{k \rightarrow \infty} R_k^{i_k} = f(\bar{x});$$

ii) in corrispondenza ad un minimo globale  $x^* \in X^*$  esiste un indice  $\bar{k}$  tale che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha che, per ogni  $k \geq \bar{k}$ ,

$$R_k^{j_k} \leq f(x^*).$$

La seguente proposizione caratterizza il comportamento dell'Algoritmo di Partizione nel caso in cui utilizzi, come strategia di scelta dell'insieme da partizionare, quella descritta dall'Assunzione 3.2.3. In particolare essa mostra che ogni sequenza strettamente annidata prodotta dall'algoritmo "converge" verso un minimo globale della funzione obiettivo.

**Proposizione 3.2.11** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.3 siano soddisfatte. Allora, per ogni sequenza strettamente annidata di insiemi  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione, si ha che*

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} \subseteq X^*. \quad (3.45)$$

(Dove  $X^*$  è l'insieme dei minimi globali della funzione  $f$  sull'insieme ammissibile  $\mathcal{D}$ .)

**Prova.** Si procede assumendo, per assurdo, che l'Algoritmo di Partizione produca una sequenza strettamente annidata di insiemi per cui non valga la (3.45), cioè una sequenza per cui si abbia:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

con  $\bar{x} \notin X^*$ .

Sia  $K \subseteq \{1, 2, \dots\}$  il sottoinsieme degli indici delle iterazioni dell'algoritmo in cui l'insieme  $\mathcal{D}^{i_k}$  viene suddiviso. Dalle istruzioni dell'algoritmo si ha che, per  $k \in K$ :

$$R_k^{i_k} = \min_{i \in I_k} R_k^i. \quad (3.46)$$

Per ogni  $k \in K$ , sia  $\mathcal{D}_k^{j_k}$  l'insieme che contiene il minimo globale  $x^* \in X^*$  considerato nel punto ii) dell'Assunzione 3.2.3. Dalla (3.46) si ha che, per ogni  $k \in K$ :

$$R_k^{i_k} \leq R_k^{j_k}. \quad (3.47)$$

L'assunzione ii) implica l'esistenza di un indice  $\bar{k}$  tale che, per tutti gli indici  $k \in K$  e  $k \geq \bar{k}$ , si ha:

$$R_k^{i_k} \leq R_k^{j_k} \leq f(x^*). \quad (3.48)$$

Facendo tendere  $k$  all'infinito ed utilizzando l'ipotesi i), si ottiene dalla (3.48):

$$\lim_{k \in K, k \rightarrow \infty} R_k^{i_k} = f(\bar{x}) \leq f(x^*),$$

che produrrebbe un assurdo con il fatto che  $\bar{x} \notin X^*$ . □

Le proprietà dell'Algoritmo di Partizione posso essere completate richiedendo la seguente assunzione che è un rafforzamento dell'Assunzione 3.2.3.



**Assunzione 3.2.4** Per ogni  $k$  si ha  $I_k^* = \{h\}$  dove l'indice  $h$  è tale che

$$R_k^h = \min_{i \in I_k} R_k^i,$$

con i valori d'interesse  $R_k^i$ ,  $i \in I_k$ , che soddisfano a:

i) per ogni sequenza strettamente annidata di insiemi  $\{\mathcal{D}^{i_k}\}$ , con  $\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\}$ , si ha

$$\lim_{k \rightarrow \infty} R_k^{i_k} = f(\bar{x});$$

ii) per ogni minimo globale  $x^* \in X^*$  esistono una costante  $\delta > 0$  ed un indice  $\bar{k}$  tali che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha che, per ogni  $k \geq \bar{k}$ ,

$$R_k^{j_k} \leq f(x^*) - \delta \|u^{j_k} - l^{j_k}\|.$$

Utilizzando questa nuova assunzione si può stabilire la seguente proposizione che assicura che ogni minimo globale "attrae" una sequenza di insiemi strettamente annidata di insiemi prodotta dall'algoritmo .

**Proposizione 3.2.12** Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.4 siano soddisfatte. Allora, per ogni minimo globale  $x^* \in X^*$ , esiste una sequenza strettamente annidata di insiemi  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione tale che:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{x^*\}. \quad (3.49)$$

**Prova.** Si procede assumendo, per assurdo, che esista un minimo globale  $x^* \in X^*$  per cui l'Algoritmo di Partizione non produca una sequenza strettamente annidata di insiemi per cui valga la (3.49). In altre parole si assume che, per ogni sequenza strettamente annidata di insiemi  $\{\mathcal{D}^{i_k}\}$  prodotta dall'Algoritmo di Partizione si abbia:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

con  $\bar{x} \neq x^*$ .

Dalla precedente relazione si avrebbe, come prima conseguenza, che la sequenza di sottoinsiemi  $\{\mathcal{D}^{j_k}\}$ , considerati nel punto ii) dell'Assunzione 3.2.4 e contenenti il minimo globale  $x^*$ , non sarebbe strettamente annidata e, quindi, esisterebbero uno scalare  $\varepsilon > 0$  ed un indice  $\tilde{k}$  tali che per ogni  $k \geq \tilde{k}$  si avrebbe:

$$\|u^{j_k} - l^{j_k}\| \geq \varepsilon. \quad (3.50)$$

Sia  $\{\mathcal{D}^{i_k}\}$  una sequenza strettamente annidata di insiemi prodotta dall'Algoritmo di Partizione e sia  $K \subseteq \{1, 2, \dots\}$  il sottoinsieme degli indici delle iterazioni dell'algoritmo

in cui l'insieme  $\mathcal{D}^{i_k}$  viene suddiviso, cioè  $i_k \in I_k^*$ , per ogni  $k \in K$ . Quindi, per ogni  $k \in K$ , si ha che:

$$R_k^{i_k} = \min_{i \in I_k} R_k^i. \quad (3.51)$$

Il punto ii) dell'Assunzione 3.2.4, (3.50) e (3.51) implicano che per tutti gli indici  $k \in K$  e  $k \geq \max\{\bar{k}, \tilde{k}\}$ , si ha:

$$R_k^{i_k} \leq R_k^{j_k} \leq f(x^*) - \delta \|u^{j_k} - l^{j_k}\| \leq f(x^*) - \delta \varepsilon. \quad (3.52)$$

Facendo tendere  $k$  all'infinito ed utilizzando il punto i) dell'Assunzione 3.2.4, si ottiene dalla (3.52):

$$\lim_{k \in K, k \rightarrow \infty} R_k^{i_k} = f(\bar{x}) \leq f(x^*) - \delta \varepsilon,$$

che produrrebbe un assurdo con il fatto che  $x^* \in X^*$ . □

Unendo la Proposizione 3.2.11 e la Proposizione 3.2.12 con la Proposizione 3.2.5 si ottengono i seguenti corollari

**Corollario 3.2.13** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.3 siano soddisfatte. Sia  $\{D^i, i \in I_k\}$  la partizione generata dall'Algoritmo di Partizione all'iterazione  $k$ -esima. Esiste un minimo globale  $x^*$  di  $f(x)$  su  $\mathcal{D}$  tale che, per ogni  $\varepsilon > 0$ , esiste una iterazione  $k$  dell'Algoritmo di Partizione in cui c'è un indice  $h_k \in I_k$  tale che:*

$$\mathcal{D}^{h_k} \subset B(x^*; \varepsilon).$$

**Corollario 3.2.14** *Si supponga che l'Assunzione 3.2.1 e l'Assunzione 3.2.4 siano soddisfatte. Sia  $\{D^i, i \in I_k\}$  la partizione generata dall'Algoritmo di Partizione all'iterazione  $k$ -esima. Per ogni minimo globale  $x^*$  di  $f(x)$  su  $\mathcal{D}$  e per ogni  $\varepsilon > 0$  esiste una iterazione  $k$  dell'Algoritmo di Partizione in cui c'è un indice  $h_k \in I_k$  tale che:*

$$\mathcal{D}^{h_k} \subset B(x^*; \varepsilon).$$

Scelta dei scalari  $R_k^i$  che utilizzano una sovrastima della costante di Lipschitz della funzione obiettivo.

Efficienti valori per gli scalari  $R_k^i$  può essere ottenuti facilmente nel caso di in cui sia nota una sovrastima della costante di Lipschitz della funzione da minimizzare.

Prima di tutto può essere utile richiamare la definizione di funzione Lipschitziana.

**Definizione 3.2.15** *Dato un insieme  $\mathcal{F} \subseteq R^n$ , una funzione  $f : \mathcal{F} \rightarrow R$  si dice Lipschitziana su  $\mathcal{F}$ , se esiste una costante  $L > 0$  (chiamata costante di Lipschitz) tale che per tutti  $x, y \in \mathcal{F}$  si ha*

$$|f(y) - f(x)| \leq L \|y - x\|. \quad (3.53)$$

La richiesta che la funzione obiettivo sia Lipschitziana non è una ipotesi particolarmente restrittiva. Per esempio, la seguente proposizione mostra che, su un insieme convesso e compatto, una funzione continuamente differenziabile è sicuramente Lipschitziana.

**Proposizione 3.2.16** *Sia  $\mathcal{F} \subset \mathbb{R}^n$  un insieme convesso e compatto e sia  $f : \mathcal{F} \rightarrow \mathbb{R}$  una funzione continuamente differenziabile. Allora la funzione  $f$  è Lipschitziana su  $\mathcal{F}$ .*

**Prova.** Ricordando il Teorema della Media si ha che, comunque scelti  $y, x \in \mathcal{F}$ :

$$f(y) = f(x) + \nabla f(x + \theta(y - x))^T (y - x),$$

con  $\theta \in [0, 1]$ . Dalla uguaglianza precedente si ottiene:

$$|f(y) - f(x)| \leq \tilde{L} \|y - x\|,$$

dove la costante  $\tilde{L}$  è data da

$$\tilde{L} = \max_{x \in \mathcal{F}} \|\nabla f(x)\|,$$

ed è ben definita dalla continuità di  $\nabla f$  e dalla compattezza del insieme  $\mathcal{F}$ .  $\square$

La precedente proposizione si applica in particolare alla classe di problemi di ottimizzazione che si sta trattando, cioè

$$\begin{aligned} \min f(x) \\ x \in \mathcal{D}, \end{aligned} \tag{3.54}$$

dove  $\mathcal{D} = \{x \in \mathbb{R}^n : l \leq x \leq u\}$ .

Nel seguito di questa sezione si assume che la funzione obiettivo  $f$  sia Lipschitziana sull'insieme ammissibile  $\mathcal{D}$  e che  $L$  sia la sua costante di Lipschitz o un sua sovrastima.

Dalla Definizione 3.2.15 si ha che una funzione Lipschitziana sull'insieme  $\mathcal{D}$  soddisfa contemporaneamente le due seguenti relazioni

$$f(y) \geq f(x) - L \|y - x\| \tag{3.55}$$

$$f(y) \leq f(x) + L \|y - x\| \tag{3.56}$$

per ogni  $x, y \in \mathcal{D}$ .

La prima delle due precedenti relazioni può essere sfruttata per definire gli scalari  $R_k^i$  associati ai vari intervalli  $\{D^i, \text{ con } i \in I_k\}$ . L'idea è quella di considerare il punto centrale  $x^i$  dell'insieme  $\mathcal{D}^i$  e di usare la sottostima della funzione obiettivo data da:

$$\hat{f}^i(x, L) = f(x^i) - L \|x - x^i\|.$$

In particolare, per ogni  $i \in I_k$ , si può fare le seguenti scelte:

$$R_k^i = R^i = \min_{x \in \mathcal{D}^i} \hat{f}^i(x, L) = f(x^i) - \frac{L}{2} \|u^i - l^i\|, \tag{3.57}$$

L'ultima uguaglianza segue dal fatto che per ogni  $x \in \mathcal{D}^i$  si ha:

$$\begin{aligned} \|x - x^i\| &= \sqrt{\sum_{i=1}^n \left(x_i - \frac{u_i + l_i}{2}\right)^2} = \frac{1}{2} \sqrt{\sum_{i=1}^n \left((x_i - l_i)^2 + (x_i - u_i)^2 + 2(x_i - l_i)(x_i - u_i)\right)} \\ &\leq \frac{1}{2} \sqrt{\sum_{i=1}^n \left((x_i - l_i)^2 + (u_i - x_i)^2 + 2(x_i - l_i)(u_i - x_i)\right)} = \frac{1}{2} \|u^i - l^i\|. \end{aligned}$$

In questo contesto si ha la seguente proposizione.

**Proposizione 3.2.17** *Se l'Assunzione 3.2.1 è soddisfatta allora:*

- se  $L$  è la costante di Lipschitz della funzione  $f$ , gli scalari  $R_k^i$  definiti dalla (3.57) soddisfano l'Assunzione 3.2.3;
- se  $L$  è una sovrastima della costante di Lipschitz della funzione  $f$ , gli scalari  $R_k^i$  soddisfano l'Assunzione 3.2.4

**Prova.** Riguardo i punti i) di Assunzione 3.2.3 e di Assunzione 3.2.4, la Proposizione 3.2.4 assicura che ogni sequenza  $\{\mathcal{D}^{i_k}\}$  strettamente annidata prodotta dall'Algoritmo di Partizione soddisfa:

$$\begin{aligned} \lim_{k \rightarrow \infty} u^{i_k} &= \bar{x}; \\ \lim_{k \rightarrow \infty} l^{i_k} &= \bar{x}; \\ \lim_{k \rightarrow \infty} x^{i_k} &= \bar{x}. \end{aligned}$$

che implicano

$$\lim_{k \rightarrow \infty} R_k^i = \lim_{k \rightarrow \infty} f(x^i) - \frac{L}{2} \|u^i - l^i\| = f(\bar{x}),$$

da cui seguono i punti i) di Assunzione 3.2.3 e di Assunzione 3.2.4.

Sia  $x^*$  un minimo globale del problema e sia  $\{\mathcal{D}^{j_k}\}$  il sottoinsieme che contiene  $x^*$  alla  $k$ -esima iterazione. Se  $L$  è la costante di Lipschitz della funzione allora si ha:

$$f(x^*) \geq f(x^{j_k}) - L \|x^* - x^{j_k}\| \geq \min_{x \in \mathcal{D}^{j_k}} f(x^{j_k}) - L \|x - x^{j_k}\| = f(x^{j_k}) - \frac{L}{2} \|u^{j_k} - l^{j_k}\| = R_k^{j_k},$$

che dimostra il punto ii) dell'Assunzione 3.2.3.

Si consideri ora il caso in cui la costante  $L$  è una sovrastima della costante di Lipschitz cioè si abbia:

$$L = \tilde{L} + \delta,$$

dove  $\tilde{L}$  è la vera costante di Lipschitz. In questo caso si ha:

$$f(x^*) - \frac{\delta}{2} \|u^{j_k} - l^{j_k}\| \geq \min_{x \in \mathcal{D}^{j_k}} \left[ f(x^{j_k}) - \tilde{L} \|x - x^{j_k}\| \right] - \frac{\delta}{2} \|u^{j_k} - l^{j_k}\| = f(x^{j_k}) - \frac{L}{2} \|u^{j_k} - l^{j_k}\| = R_k^{j_k},$$

da cui segue il punto ii) dell'Assunzione 3.2.4.  $\square$

Quando la costante di Lipschitz della funzione obiettivo non è nota, si può cercare di utilizzare la valutazione della funzione obiettivo effettuate durante le iterazioni dell'algoritmi per cercare di estrarre informazioni sul comportamento globale della funzione obiettivo. In questo caso, per ogni  $i \in I_k$ , si possono definire i seguenti scalari:

$$R_k^i = f(x^i) - \frac{L_k^i}{2} \|u^i - l^i\|. \quad (3.58)$$

Ripetendo i passi della dimostrazione di Proposizione 3.2.17 e ricordando l'Assunzione 3.2.4 si stabilisce la seguente proposizione che indica che per assicurare buone proprietà di convergenza è sufficiente essere in grado di stimare dopo un numero finito di iterazioni le costanti di Lipschitz locali della funzione in regioni vicino ai minimi globali.

**Proposizione 3.2.18** *Se l'Assunzione 3.2.1 e se le sequenze  $\{L_k^i\}$  sono limitate per ogni  $i \in I_k$ , allora:*

- se in corrispondenza ad un minimo globale  $x^* \in X^*$  esiste un indice  $\bar{k}$  tale che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha per ogni  $k \geq \bar{k}$ ,

$$f(x^{j_k}) - \frac{L_k^{j_k}}{2} \|u^{j_k} - l^{j_k}\| \leq f(x^*),$$

allora ogni sequenze di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata prodotta dall'algoritmo è tale che:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} \subseteq X^*;$$

- se per ogni minimo globale  $x^* \in X^*$  esistono una costante  $\delta > 0$  ed un indice  $\bar{k}$  tali che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha per ogni  $k \geq \bar{k}$ ,

$$f(x^{j_k}) - \frac{L_k^{j_k}}{2} \|u^{j_k} - l^{j_k}\| < f(x^*) - \delta \|u^{j_k} - l^{j_k}\|,$$

allora, per ogni  $x^* \in X^*$ , l'algoritmo produce una sequenza di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata tale che

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{x^*\}.$$

### 3.2.4 Algoritmo di Partizione con minimizzazioni locali (questa sezione non fa parte del programma).

Analogamente a quanto avvenuto nel caso degli algoritmi probabilistici anche gli algoritmi che partizionano l'insieme ammissibile possono sfruttare al loro interno l'efficienza dei metodi di ottimizzazione locale in modo da migliorare le loro proprietà teoriche e

la loro efficienza computazionale. In particolare il seguente algoritmo è un esempio di un Algoritmo di Partizione che utilizza minimizzazioni locali.

**Algoritmo di Partizione Multistart.**

*Passo 0:* Si pone  $\mathcal{D}^0 = \mathcal{D}$ ,  $l^0 = l$ ,  $u^0 = u$ ,  $I_0 = \{0\}$ , si sceglie un punto  $x_0 \in \mathcal{D}^0$ , si pone  $x_0^* = x_0$  e  $k = 0$ ;

*Passo 1:* data la partizione  $\{\mathcal{D}^i : i \in I_k\}$  di  $\mathcal{D}$ ;  
 si identifica un sottoinsieme di indici  $I_k^* \subseteq I_k$ ,  
 si pone  $\bar{I}^0 = I_k$ ,  $\hat{I}^0 = I_k^*$  e  $p = 0$ ;

*Passo 2:* si sceglie un indice  $h \in \hat{I}^p$  e si partiziona  $\mathcal{D}^h$  in  $m \geq 2$  sottointervalli

$$\mathcal{D}^{h_1}, \mathcal{D}^{h_2}, \dots, \mathcal{D}^{h_m},$$

si scelgono i punti  $x^{h_j} \in \mathcal{D}^{h_j}$ , per  $j = 1, \dots, m$ ;

*Passo 3:* a partire da ogni punto  $x^{h_j}$ ,  $j = 1, \dots, m$ , si effettuano delle minimizzazioni locali ottenendo  $y^{h_j}$ ,  $j = 1, \dots, m$ ;

si sceglie  $y^*$  tale che  $f(y^*) = \min_{j=1, \dots, m} f(y^{h_j})$ ;

se  $f(y^*) < f(x_k^*)$  si pone  $x_{k+1}^* = y^*$ ; altrimenti si pone  $x_{k+1}^* = x_k^*$ ;

*Passo 4:* si pone:

$$\bar{I}^{p+1} = \bar{I}^p \bigcup_{j=1, \dots, m} \{h_j\} \setminus \{h\},$$

$$\hat{I}^{p+1} = \hat{I}^p \setminus \{h\},$$

se  $\hat{I}^{p+1} \neq \emptyset$  si pone  $p = p + 1$  e si torna al Passo 2;

*Passo 5:* si definisce la nuova partizione  $\{\mathcal{D}^i : i \in I_{k+1}\}$  con

$$I_{k+1} = \bar{I}^{p+1};$$

si pone  $k = k + 1$  e si torna al Passo 1.

Come accenato, l'utilizzazione dei metodi di ottimizzazione locale permette di realizzare degli algoritmi con interessanti proprietà teoriche. In particolare, per quando riguarda il precedente algoritmo, ha si il seguente risultato.

**Proposizione 3.2.19** *Si supponga che siano verificate l'Assunzione 3.2.1 ed una tra l'Assunzione 3.2.2 e l'Assunzione 3.2.4. Se esiste un minimo globale  $x^*$  della funzione  $f$  su  $\mathcal{D}$  per cui l'algoritmo di minimizzazione locale (utilizzato al Passo 3) soddisfa la Assunzione 3.1.1 allora esiste una iterazione  $\bar{k}$  in cui l'Algoritmo di Partizione*

Multistart produce un punto  $x_{\bar{k}}^*$  tale che:

$$x_{\bar{k}}^* \in X^*. \quad (3.59)$$

**Prova.** la prova segue direttamente dal Corollario 3.2.10 oppure dal Corollario 3.2.14 e dalla la Assunzione 3.1.1 dell'algorithmo locale.  $\square$

### 3.2.5 L'Algorithmo Direct

Da quanto visto nella sezione precedente è sufficiente ottenere, entro un numero sufficientemente grande di iterazioni, una sovrastima della costante di Lipschitz della funzione obiettivo in un intorno di un minimo globale per garantire la convergenza di un metodo di partizione verso il minimo globale. Tuttavia, anche quest'ultima richiesta pone dei limiti sulla garanzia di ottenere un minimo globale utilizzando questi algoritmi. Questo motiva il tentativo di definire degli algoritmi che cerchino di sfruttare l'ipotesi che la funzione è Lipschitziana senza richiedere, però, nessuna informazione sulla costante di Lipschitz. Tra questi metodi uno dei più significativi ed importanti è quello chiamato Algorithmo Direct.

L'Algorithmo Direct, come gli algoritmi diagonali, partiziona l'insieme ammissibile  $\mathcal{D}$  in un numero crescente di sottointervalli  $\mathcal{D}^i = \{x \in R^n : l^i \leq x \leq u^i\}$  con  $i \in I_k$  (da cui il nome "dividing rectangles"). Ad ogni iterazione, un certo numero di sottointervalli vengono giudicati di interesse e, di conseguenza, vengono ulteriormente suddivisi. La maggiore novità dell'Algorithmo Direct è il modo con cui giudica l'interesse di un sottointervallo. Infatti gli algoritmi diagonali scelgono delle stime delle costanti di Lipschitz locali e, sfruttando la Lipschitzianità della funzione obiettivo, selezionano il sottointervallo di maggiore interesse. Invece l'Algorithmo di Direct seleziona ogni sottointervallo per cui esiste un valore della costante di Lipschitz della funzione obiettivo che rende il sottointervallo considerato il più interessante.

In maniera simile a quanto fatto nella sezione precedente, il punto di partenza di questo algoritmo è di misurare l'interesse di un intervallo  $\mathcal{D}^i$  calcolando il valore che assume ai suoi punti estremi la sottostima della funzione obiettivo

$$\hat{f}^i(x, \bar{L}) = f(x^i) - \bar{L}\|x - x^i\|,$$

dove  $x^i$  è il punto centrale dell'insieme  $\mathcal{D}^i$  e  $\bar{L}$  è una stima della costante di Lipschitz della funzione obiettivo (figura (3.3)). Gli intervalli con valori più bassi sono, quindi, considerati di maggiore interesse.

Tuttavia i valori assunti agli estremi dei sottoinsiemi dalle funzioni  $\hat{f}^i(x, \bar{L})$  dipendono dal particolare valore della stima  $\bar{L}$ . Infatti per valori molto grandi di  $\bar{L}$  sono giudicati interessanti i sottoinsiemi di dimensioni massime (figura (3.4)). Per valori molto piccoli di  $\bar{L}$  sono giudicati interessanti i sottoinsieme a cui corrispondono i valori più piccoli di  $f(x^i)$  (figura (3.5)). Tutti gli altri sottointervalli possono essere divisi in due gruppi: il primo costituito da sottoinsiemi per cui esiste un valore di  $\bar{L}$  per cui sono considerati interessanti (figura (3.6)) e il secondo costituito da sottoinsiemi per cui non esiste un valore di  $\bar{L}$  per cui sono considerati interessanti (sottoinsieme  $\mathcal{D}^3$  della figura (3.6)).

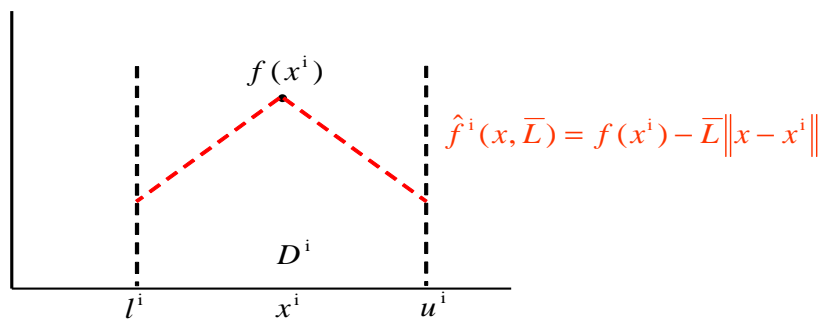


Figura 3.3: Esempio della sottostima  $\hat{f}(x, \bar{L})$ .

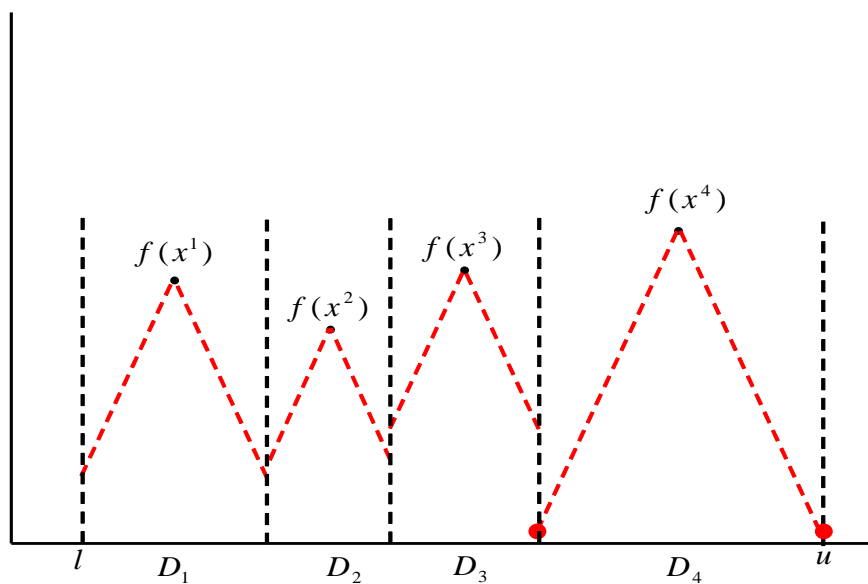


Figura 3.4: Esempio di valutazioni dei sottoinsiemi per  $\bar{L}$  grande.



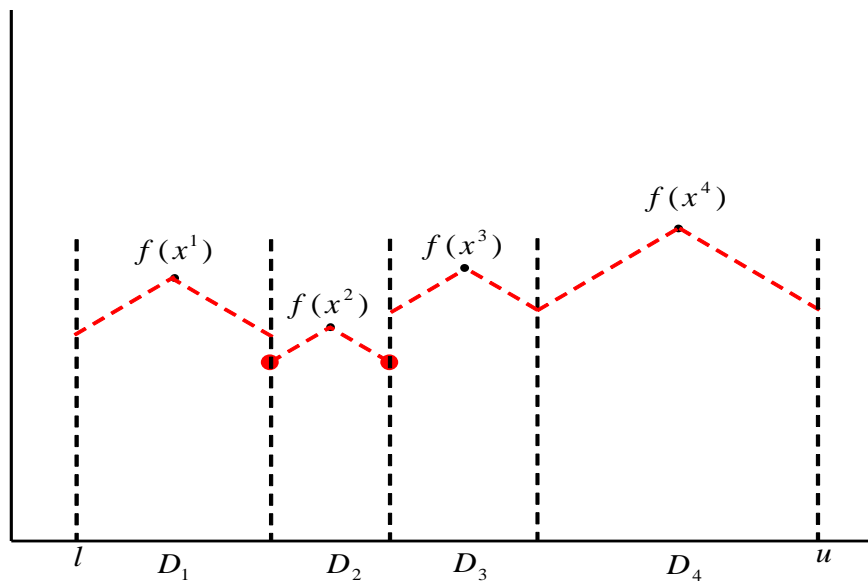


Figura 3.5: Esempio di valutazioni dei sottoinsiemi per  $\bar{L}$  piccolo.

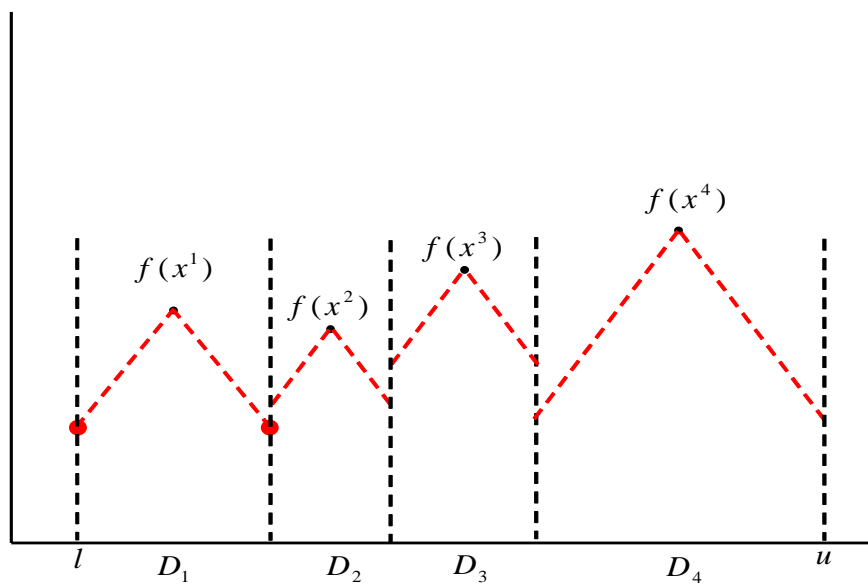


Figura 3.6: Esempio di valutazioni dei sottoinsiemi per  $\bar{L}$  medio.

Formalmente l'Algoritmo Direct usa la seguente definizione di sottointervalli potenzialmente ottimi.

**Definizione 3.2.20** *Data la partizione  $\{\mathcal{D}^i : i \in I\}$  dell'insieme  $\mathcal{D}$ , dove, per ogni  $i \in I$*

$$\mathcal{D}^i = \{x \in R^n : l^i \leq x \leq u^i\}, \quad x^i = \frac{u^i + l^i}{2},$$

e sia

$$f_{min} = \min_{i \in I} f(x^i).$$

Un sottointervallo  $\mathcal{D}^h$ ,  $h \in I$ , è detto potenzialmente ottimo se, scelto un parametro  $\varepsilon > 0$ , esiste una costante  $\bar{L}^h > 0$  tale che:

$$f(x^h) - \frac{\bar{L}^h}{2} \|u^h - l^h\| \leq f(x^i) - \frac{\bar{L}^h}{2} \|u^i - l^i\|, \quad \text{per tutti } i \in I, \quad (3.60)$$

$$f(x^h) - \frac{\bar{L}^h}{2} \|u^h - l^h\| \leq f_{min} - \varepsilon |f_{min}|, \quad (3.61)$$

Introdotta la precedente definizione, è possibile descrivere formalmente l'Algoritmo Direct.

**Algoritmo Direct.**

*Passo 0:* Si pone  $\mathcal{D}^0 = \mathcal{D}$ ,  $l^0 = l$ ,  $u^0 = u$ ,  $x^0 = (l^0 + u^0)/2$ ,  $I_0 = \{0\}$  e  $k = 0$ ;

*Passo 1:* data la partizione  $\{\mathcal{D}^i : i \in I_k\}$  di  $\mathcal{D}$

dato l'insieme di punti  $\{x^i = (l^i + u^i)/2 : i \in I_k\}$

si identifica il sottoinsieme di indici  $I_k^* \subseteq I_k$  tale che:

$$I_k^* = \{i \in I_k : \mathcal{D}^i \text{ è potenzialmente ottimo}\},$$

si pone  $\bar{I}^0 = I_k$ ,  $\hat{I}^0 = I_k^*$  e  $p = 0$

*Passo 2:* si sceglie un indice  $h \in \hat{I}^p$  e si determinano:

$$\begin{aligned} \delta &= \max_{1 \leq j \leq n} (u^h - l^h)_j, \\ J &= \{j = 1, \dots, n : (u^h - l^h)_j = \delta\} \\ m &= |J| \quad (\text{dove } |J| \text{ è la cardinalità dell'insieme } J); \end{aligned}$$

*Passo 3:* si determinano i  $2m + 1$  punti dati da:

$$x^{h_0} = x^h, \quad x^{h_j} = x^h + \frac{\delta}{3}e_j, \quad x^{h_{j+m}} = x^h - \frac{\delta}{3}e_j, \quad \text{per ogni } j \in J;$$

*Passo 4:* attraverso la *Procedura di Partizione* si partiziona l'insieme  $\mathcal{D}^h$  attraverso i  $2m + 1$  sottointervalli  $\mathcal{D}^{h_0}, \mathcal{D}^{h_j}, \mathcal{D}^{h_{j+m}}$ , con  $j \in J$ , che hanno come punti centrali i vettori  $x^{h_0}, x^{h_j}, x^{h_{j+m}}$ , con  $j \in J$ ;

*Passo 5:* si pone:

$$\begin{aligned} \bar{I}^{p+1} &= \bar{I}^p \cup \{h_0\} \cup \bigcup_{j \in J} \{h_j, h_{j+m}\} \setminus \{h\}, \\ \hat{I}^{p+1} &= \hat{I}^p \setminus \{h\}, \end{aligned}$$

se  $\hat{I}^{p+1} \neq \emptyset$  si pone  $p = p + 1$  e si torna al Passo 2;

*Passo 6:* si definisce la nuova partizione  $\{\mathcal{D}^i : i \in I_{k+1}\}$  con

$$I_{k+1} = \bar{I}^{p+1},$$

si pone  $k = k + 1$  e si torna al Passo 1.

Al Passo 1 l'algoritmo seleziona, tra tutti gli intervalli generati, quelli che sono potenzialmente ottimi. Per ognuno di questi, al Passo 2 determina l'insieme degli indici  $J$  che corrispondono ai spigoli più lunghi dell'intervallo scelto. Al Passo 3, a partire dal centro dell'intervallo, vengono generati dei nuovi punti lungo gli assi coordinati (i ver-

sori  $e_j$ ) i cui indici appartengono a  $J$  e lungo i loro opposti. Questi nuovi punti vengono generati a distanza dal centro pari ad un terzo dell'ampiezza massima dell'intervallo. Al Passo 4, l'intervallo scelto viene diviso in sottointervalli che hanno come centri i punti generati al Passo 3 ed il punto  $x^h$ , centro dell'intervallo di partenza.

Per quanto riguarda la generazione dei sottointervalli che partizionano un intervallo potenzialmente ottimo, si ha la seguente procedura.

**Procedura di Partizione.**

*Passo 0:* Dati l'intervallo  $\mathcal{D}^h$ , lo scalare  $\delta$ , l'insieme  $J$ , i punti  $x^{h_0}$ ,  $x^{h_j}$ ,  $x^{h_{j+m}}$ , con  $j \in J$ , si pone  $\tilde{\mathcal{D}}^0 = \mathcal{D}^h$ ,  $\tilde{J}^0 = J$  e  $p = 0$ ;

*Passo 1:* si determina l'indice  $\ell \in \tilde{J}^p$  tale che:

$$w^\ell = \min_{j \in \tilde{J}^p} w^j,$$

con  $w^j = \min\{f(x^{h_j}), f(x^{h_{j+m}})\}$ .

*Passo 2:* si determinano gli insiemi:

$$\begin{aligned} \mathcal{D}^{h_\ell} &= \tilde{\mathcal{D}}^p \cap \{x \in R^n : (u^h)_\ell - \frac{\delta}{3} \leq (x)_\ell \leq (u^h)_\ell\}, \\ \mathcal{D}^{h_{\ell+m}} &= \tilde{\mathcal{D}}^p \cap \{x \in R^n : (l^h)_\ell \leq (x)_\ell \leq (l^h)_\ell + \frac{\delta}{3}\}, \end{aligned}$$

e si pone:

$$\begin{aligned} \tilde{\mathcal{D}}^{p+1} &= \tilde{\mathcal{D}}^p \cap \{x \in R^n : (l^h)_\ell + \frac{\delta}{3} \leq (x)_\ell \leq (u^h)_\ell - \frac{\delta}{3}\}, \\ \tilde{J}^{p+1} &= \tilde{J}^p \setminus \{\ell\}; \end{aligned}$$

*Passo 3:* se  $\tilde{J}^{p+1} \neq \emptyset$ , si pone  $p = p + 1$  e si torna al Passo 1;

*Passo 4:* si pone:

$$\mathcal{D}^{h_0} = \tilde{\mathcal{D}}^{p+1}$$

e la procedura termina.

Nella precente procedura si determina la forma dei nuovi intervalli cercando di sfruttare i valori della funzione obiettivo ottenuti nei punti generati al Passo 3 dell'Algoritmo Direct. L'obiettivo è di associare dimensioni più grandi agli intervalli in cui al centro la funzione obiettivo assume valori più piccoli. In questo modo si aumenta la possibilità che, all'iterazione successiva dell'Algoritmo Direct, questi intervalli siano giudicati potenzialmente ottimi. Per ottenere questo obiettivo la precedente procedura determina, ad ogni passo, il versore  $e_j$  lungo il quale si ha il valore più basso della funzione obiet-

tivo. Poi il sottointervallo in cui c'è il centro  $x^h$  dell'intervallo di partenza viene diviso in tre parti uguali lungo la direzione indentificata da  $e_j$ .

In figura 3.7 è riportato un esempio di partizione dell'intervallo  $\mathcal{D}_h$ , nel caso in cui il valore più piccolo della funzione obiettivo è nel punto  $x^{h_4}$

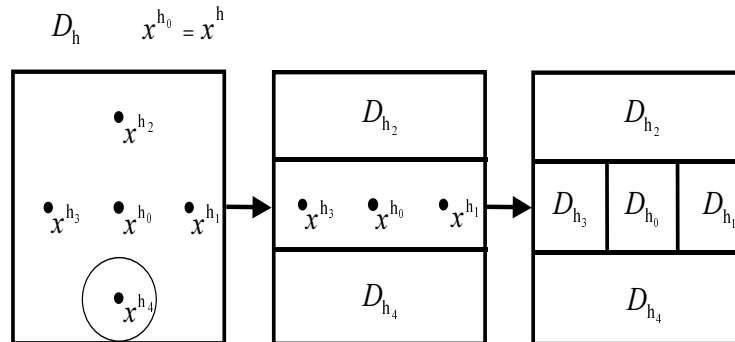


Figura 3.7: Suddivisione dell'intervallo  $\mathcal{D}_h$ .

La figura 3.8 descrive il comportamento della Procedura di Partizione nel caso in cui l'intervallo  $\mathcal{D}_h$  diventa potenzialmente ottimo.

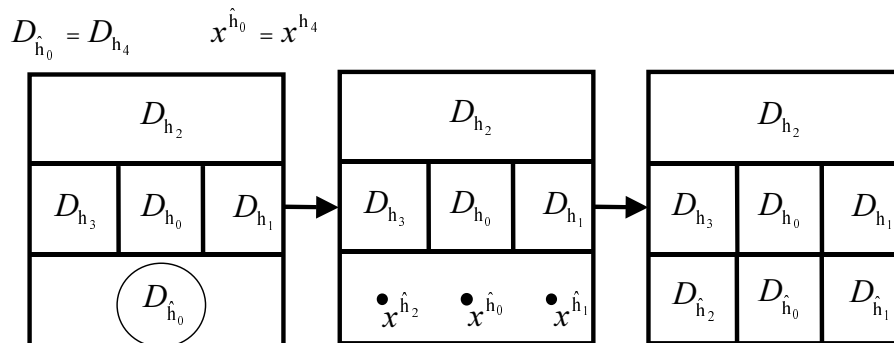


Figura 3.8: Suddivisione dell'intervallo  $\mathcal{D}_{h_4}$ .

L'analisi delle proprietà dell'Algoritmo Direct segue facilmente dalle seguenti due proposizioni. La prima evidenzia che l'Algoritmo Direct partiziona i sottoinsiemi identificati in maniera da soddisfare l'Assunzione 3.2.1.

**Proposizione 3.2.21** *Siano  $\mathcal{D}^{h_0}, \mathcal{D}^{h_j}, \mathcal{D}^{h_{j+m}}, j \in J$ , i sottointervalli generati dalla partizione dell'intervallo  $\mathcal{D}^h$  al Passo 4 dell'Algoritmo Direct, allora l'Assunzione 3.2.1 è soddisfatta.*

**Prova.** Le istruzioni della Procedura di Partizione implicano che la norma delle diagonali dei sottoinsiemi generati  $\mathcal{D}^{\tilde{h}}$ ,  $\tilde{h} \in \cup_{j \in J} \{h_j, h_{j+m}\} \cup \{h_0\}$ , soddisfano

$$\begin{aligned} \|u^{\tilde{h}} - l^{\tilde{h}}\|^2 &= \sum_{i=1}^n (u^{\tilde{h}} - l^{\tilde{h}})_i^2 \geq \frac{1}{9} \sum_{i=1}^n (u^h - l^h)_i^2 \\ &= \frac{1}{9} \|u^h - l^h\|^2 \end{aligned} \quad (3.62)$$

$$\begin{aligned} \|u^{\tilde{h}} - l^{\tilde{h}}\|^2 &\leq \sum_{i \neq j} (u^h - l^h)_i^2 + \frac{(u^h - l^h)_j^2}{9} \\ &= \sum_{i=1}^n (u^h - l^h)_i^2 - \frac{8}{9} (u^h - l^h)_j^2 \end{aligned} \quad (3.63)$$

dove  $j \in J$  (l'insieme  $J$  è definito al Passo 3 dell'Algoritmo Direct). Inoltre si ha che la norma della diagonale di  $\mathcal{D}^h$  soddisfa la seguente relazione

$$\|u^h - l^h\|^2 = \sum_{i=1}^n (u^h - l^h)_i^2 \leq n(u^h - l^h)_j^2,$$

da cui segue

$$\|u^{\tilde{h}} - l^{\tilde{h}}\|^2 \leq \left(1 - \frac{8}{9n}\right) \|u^h - l^h\|^2. \quad (3.64)$$

Dalla (3.62), dalla (3.64) e dalle istruzioni della procedura di partizione segue che l'Assunzione 3.2.1 è soddisfatta con  $\varepsilon_1 = 1/3$ ,  $\varepsilon_2 = \sqrt{1 - 8/(9n)}$  e  $\varepsilon_3 = 1/3$ .  $\square$

La seconda proposizione mette in evidenza è il fatto che, in ogni iterazione, un particolare sottoinsieme con ampiezza massima risulta sempre potenzialmente ottimo e che quindi l'Assunzione 3.2.2 è soddisfatta.

**Proposizione 3.2.22** *Se  $\{I_k^*\}$  è la sequenze degli insiemi di indici identificati dall'Algoritmo Direct allora l'Assunzione 3.2.2 è soddisfatta.*

**Prova.** Sia  $\mathcal{D}^\ell$ ,  $\ell \in I_k$ , un sottoinsieme tale che  $\ell \in I_k^{max}$  (con  $I_k^{max}$  dato dalla (3.38)) e  $f(x^\ell) \leq f(x^i)$ , per ogni  $i \in I_k^{max}$ , allora il sottoinsieme  $\mathcal{D}^\ell$  è potenzialmente ottimo. La dimostrazione segue facilmente notando che ogni costante  $\bar{L}^\ell > 0$  tale che:

$$\bar{L}^\ell > 2 \max \left\{ \frac{f(x^\ell) - f_{min} + \varepsilon |f_{min}|}{d^\ell}, \max_{j \in I_k \setminus I_k^{max}} \frac{f(x^\ell) - f(x^j)}{d^\ell - d^j} \right\},$$

soddisfa alle richieste (3.60) e (3.61).  $\square$

Utilizzando la Proposizione 3.2.21, e la Proposizione 3.2.8 si può concludere con la seguente proposizione.

**Proposizione 3.2.23** *Tutte le sequenze di insiemi  $\{\mathcal{D}^{i_k}\}$  generate dall'Algoritmo Direct sono strettamente annidate. Inoltre, per ogni  $\tilde{x} \in \mathcal{D}$ , l'algoritmo genera una sequenza di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata tale che*

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\tilde{x}\}.$$

Dalla precedente proposizione segue che l'Algoritmo Direct genera un insieme di punti che, al crescere delle iterazioni, tende a diventare un insieme denso su  $\mathcal{D}$ . Tuttavia, computazionalmente, l'Algoritmo Direct sembra mostrare una buona capacità di visitare prima le regioni più interessanti dal punto della localizzazione dei minimi globali.

### 3.2.6 Estensione dell'Algoritmo Direct che utilizza una stima della costante di Lipschitz della funzione obiettivo

In questa sezione si considera una possibile variazione dell'Algoritmo di Direct nel caso in cui si ha a disposizione una sovrastima della costante di Lipschitz. Il punto di partenza del nuovo algoritmo è la seguente definizione di sottointervallo fortemente potenzialmente ottimo.

**Definizione 3.2.24** *Data una costante  $\bar{L} > 0$  ed una partizione  $\{\mathcal{D}^i, i \in I\}$  di  $\mathcal{D}$ . Un sottointervallo  $\mathcal{D}^h$ ,  $h \in I$ , è detto fortemente potenzialmente ottimo se, scelti due parametri  $\varepsilon > 0$  e  $\tilde{\varepsilon} > 0$ , una delle seguenti condizioni è verificata:*

*i) esiste una costante a constant  $L^h \in (0, \bar{L})$  tale che:*

$$f(x^h) - \frac{L^h}{2} \|u^h - l^h\| \leq f(x^i) - \frac{L^h}{2} \|u^i - l^i\|, \text{ per tutti } i \in I_k, \quad (3.65)$$

$$f(x^h) - \frac{L^h}{2} \|u^h - l^h\| \leq f^{min} - \varepsilon \max\{|f^{min}|, \tilde{\varepsilon}\}, \quad (3.66)$$

*dove  $f^{min}$  è dato nella Definizione 3.2.20;*

*ii)*

$$f(x^h) - \frac{\bar{L}}{2} \|u^h - l^h\| \leq f(x^i) - \frac{\bar{L}}{2} \|u^i - l^i\|, \text{ per tutti } i \in I_k. \quad (3.67)$$

Utilizzando la precedente definizione si può definire un nuovo algoritmo che coincide con quello di Direct a meno della scelta dei sottointervalli da partizionare. Infatti, ad ogni iterazione, invece di selezionare quelli potenzialmente ottimi vengono scelti i sottointervalli fortemente potenzialmente ottimi. Grazie a questa variazione, il nuovo algoritmo presenta delle proprietà teoriche più forti, come è descritto dal seguente risultato.

**Proposizione 3.2.25** *Sia  $\bar{L}$  la costante che compare nella definizione di sottoinsieme strettamente potenzialmente ottimo. Si consideri la variazione dell'Algoritmo di Direct in cui l'insieme  $I_k^*$  è costituito dagli indici dei sottoinsiemi fortemente potenzialmente ottimi. Allora*

- i) *l'algoritmo produce almeno una sequenze di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata;*  
ii) *se in corrispondenza ad un minimo globale  $x^* \in X^*$  esiste un indice  $\bar{k}$  tale che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha per ogni  $k \geq \bar{k}$ ,*

$$f(x^{j_k}) - \frac{\bar{L}}{2} \|u^{j_k} - l^{j_k}\| \leq f(x^*),$$

*allora ogni sequenze di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata prodotta dall'algoritmo è tale che:*

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} \subseteq X^*;$$

- iii) *se per ogni minimo globale  $x^* \in X^*$  esistono una costante  $\delta > 0$  ed un indice  $\bar{k}$  tali che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha per ogni  $k \geq \bar{k}$ ,*

$$f(x^{j_k}) - \frac{\bar{L}}{2} \|u^{j_k} - l^{j_k}\| < f(x^*) - \delta \|u^{j_k} - l^{j_k}\|,$$

*allora, per ogni  $x^* \in X^*$ , l'algoritmo produce una sequenza di insiemi  $\{\mathcal{D}^{i_k}\}$  strettamente annidata tale che*

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{x^*\}.$$

**Prova (la prova non fa parte del programma di esame).** Il punto i) segue direttamente dalle proprietà della tecnica di partizione dell'Algoritmo Direct che continua a rimanere immutata nel nuovo algoritmo.

Punto ii). Si procede assumendo, per assurdo, che l'algoritmo produca una sequenza strettamente annidata di insiemi per cui si abbia:

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

con

$$f(\bar{x}) > f(x^*). \quad (3.68)$$

Sia  $K \subseteq \{1, 2, \dots\}$  il sottoinsieme degli indici delle iterazioni dell'algoritmo in cui l'insieme  $\mathcal{D}^{i_k}$  viene suddiviso. Dalle istruzioni dell'algoritmo si avrebbe che, per ogni  $k \in K$ , l'insieme  $\mathcal{D}^{i_k}$  avrebbe soddisfatto la Definizione 3.2.24 di fortemente potenzialmente ottimo e, quindi, una delle due condizioni della definizione sarebbe stata soddisfatta.



Se fosse soddisfatta la condizione i) della Definizione 3.2.24 dovrebbe esistere una costante  $L^{i_k} \in (0, \bar{L})$  tale che:

$$L^{i_k} \geq 2 \left( \frac{f(x^{i_k}) - f^{min} + \varepsilon \max\{|f^{min}|, \tilde{\varepsilon}\}}{\|u^{i_k} - l^{i_k}\|} \right). \quad (3.69)$$

Poichè la sequenza  $\{\mathcal{D}^{i_k}\}$  è strettamente annidata, la Proposizione 3.2.3 assicurerebbe che

$$\lim_{k \rightarrow \infty} \|u^{i_k} - l^{i_k}\| = 0. \quad (3.70)$$

Le due precedenti relazioni (3.69) e (3.70) implicherebbero che per valori sufficientemente grandi di  $k$  si avrebbe:

$$L^{i_k} \geq \bar{L}.$$

Perciò esisterebbe un  $\tilde{k}$  tale che per ogni  $k \in K$  e  $k \geq \tilde{k}$  dovrebbe essere soddisfatta la condizione ii) della Definizione 3.2.24. Quindi, per ogni  $k \in K$  e  $k \geq \tilde{k}$ , si avrebbe che:

$$f(x^{i_k}) - \frac{\bar{L}}{2} \|u^{i_k} - l^{i_k}\| \leq f(x^{j_k}) - \frac{\bar{L}}{2} \|u^{j_k} - l^{j_k}\|. \quad (3.71)$$

Da cui, utilizzando l'ipotesi fatta nel punto ii) della proposizione, si avrebbe che per ogni  $k \in K$  e  $k \geq \max\{\bar{k}, \tilde{k}\}$  si otterrebbe:

$$f(x^{i_k}) - \frac{\bar{L}}{2} \|u^{i_k} - l^{i_k}\| \leq f(x^*). \quad (3.72)$$

Ricordando che la sequenza  $\{\mathcal{D}^{i_k}\}$  è strettamente annidata si avrebbe che vale la (3.70). Quindi facendo il limite per  $k$  tendente all'infinito dei due termini della (3.72) si otterrebbe:

$$f(\bar{x}) \leq f(x^*),$$

che porterebbe ad un assurdo con la (3.68).

Punto iii). Si assuma, per assurdo, che esista un minimo globale  $x^* \in X^*$  per cui, se  $\{\mathcal{D}^{j_k}\}$  è la sequenza di insiemi tali che

$$x^* \in \mathcal{D}^{j_k},$$

per ogni  $k$ , allora la sequenza di insiemi  $\{\mathcal{D}^{j_k}\}$  non sia strettamente annidata. Quindi la Proposizione 3.2.3 assicurerebbe l'esistenza di uno scalare  $\varepsilon > 0$  e di un indice  $\bar{k}$  tali che per  $k \geq \bar{k}$  si avrebbe

$$\|u^{j_k} - l^{j_k}\| \geq \varepsilon. \quad (3.73)$$

Sia  $\{\mathcal{D}^{i_k}\}$  una sequenza strettamente annidata di insiemi prodotta dall'algoritmo con

$$\bigcap_{k=0}^{\infty} \mathcal{D}^{i_k} = \{\bar{x}\},$$

(ovviamente  $\bar{x} \neq x^*$ ) e sia  $K \subseteq \{1, 2, \dots\}$  il sottoinsieme degli indici delle iterazioni dell'algoritmo in cui l'insieme  $\mathcal{D}^{i_k}$  sarebbe suddiviso, cioè  $i_k \in I_k^*$ , per ogni  $k \in K$ .

Quindi, per ogni  $k \in K$ , si avrebbe che l'insieme  $\mathcal{D}^{j_k}$  avrebbe soddisfatto una delle condizioni della Definizione 3.2.24. Per quanto detto nella dimostrazione del punto i) della proposizione esisterebbe un  $\bar{k}$  tale che per ogni  $k \in K$  e  $k \geq \bar{k}$  l'insieme  $\mathcal{D}^{j_k}$  avrebbe soddisfatto la condizione ii) della Definizione 3.2.24. Perciò, per ogni  $k \in K$  e  $k \geq \bar{k}$ , si avrebbe che:

$$f(x^{i_k}) - \frac{\bar{L}}{2} \|u^{i_k} - l^{i_k}\| \leq f(x^{j_k}) - \frac{\bar{L}}{2} \|u^{j_k} - l^{j_k}\|. \quad (3.74)$$

Utilizzando l'assunzione fatta al punto ii) della proposizione, la (3.73), la (3.70) e facendo il limite per  $k$  tendente all'infinito, della precedente relazione si otterrebbe il seguente assurdo

$$f(\bar{x}) \leq f(x^*) - \delta\varepsilon.$$

□

Per questa nuova versione dell'Algoritmo di Direct è possibile definire un criterio di arresto come è indicato dal seguente risultato .

**Proposizione 3.2.26** *Sia  $\bar{L}$  la costante che compare nella definizione di sottoinsieme strettamente potenzialmente ottimo. Si consideri l'Algoritmo di Direct in cui l'insieme  $I_k^*$  è costituito dagli indici dei sottoinsiemi fortemente potenzialmente ottimi. Se in corrispondenza ad un minimo globale  $x^* \in X^*$  esiste un indice  $\bar{k}$  tale che, se  $\mathcal{D}^{j_k}$ ,  $j_k \in I_k$ , è l'insieme per cui  $x^* \in \mathcal{D}^{j_k}$ , si ha per ogni  $k \geq \bar{k}$ ,*

$$f(x^{j_k}) - \frac{\bar{L}}{2} \|u^{j_k} - l^{j_k}\| \leq f(x^*),$$

allora, per  $k \geq \bar{k}$ , si ha:

$$f(x^{h_k}) - f(x^*) \leq \frac{\bar{L}}{2} \|u^{h_k} - l^{h_k}\|,$$

dove l'indice  $h_k$  è dato da:

$$f(x^{h_k}) - \frac{\bar{L}}{2} \|u^{h_k} - l^{h_k}\| = \min_{i \in I_k} \left\{ f(x^i) - \frac{\bar{L}}{2} \|u^i - l^i\| \right\}.$$

**Prova (la prova non fa parte del programma di esame).** Dalla assunzione della proposizione e dalla definizione dell'indice  $h_k$  si ha che, per  $k \geq \bar{k}$ ,

$$f(x^*) \geq f(x^{j_k}) - \frac{\bar{L}}{2} \|u^{j_k} - l^{j_k}\| \geq f(x^{h_k}) - \frac{\bar{L}}{2} \|u^{h_k} - l^{h_k}\|,$$

da cui segue il risultato. □

### 3.3 Funzioni di tipo “Filled”

Le funzioni filled sono state introdotte allo scopo di cercare di uscire dalle zone di attrazione dei minimi locali e di riuscire, attraverso minimizzazioni locali, a determinare dei nuovi punti stazionari a cui corrispondono valori della funzione obiettivo più bassi. Un algoritmo che utilizza le funzioni filled può essere rappresentato dal seguente schema.

**Algoritmo che usa una funzione filled.**

*Passo 0:* sia dato un punto  $x_0 \in R^n$ ; si pone  $k = 0$ ;

*Passo 1:* partendo da  $x_k$  si applica un algoritmo di ottimizzazione locale ottenendo un punto  $x_k^* \in R^n$ ;

*Passo 2:* utilizzando  $x_k^*$  si definisce una funzione filled e si applica un algoritmo di ottimizzazione locale a questa nuova funzione  $U_k(x)$  arrestandolo appena si ottiene un punto  $x_{k+1} \neq x_k^*$  tale che

$$f(x_{k+1}) \leq f(x_k^*);$$

*Passo 3:* si pone  $k = k + 1$  e si ritorna al Passo 1.

Determinato, ad una certa iterazione  $k$ , un minimo locale o, più in generale, un punto stazionario  $x_k^*$ , la Proposizione 1.6.2 indica la possibilità teorica di perturbare localmente la funzione obiettivo in maniera da ottenere una nuova funzione in cui  $x_k^*$  sia un massimo locale stretto ed in cui, invece, gli altri punti stazionari rimangano inalterati. Purtroppo l'applicazione diretta della Proposizione 1.6.2 è particolarmente difficile per la necessità di evitare che i termini di perturbazione introducano dei nuovi punti stazionari o nascondano i minimi globali della funzione obiettivo di partenza.

Le funzioni filled, pur traendo ispirazione dalla Proposizione 1.6.2, rinunciano ad introdurre dei termini di perturbazione locale che lasciano invariata la funzione obiettivo al di fuori di un fissato intorno. Infatti esse modificano la funzione obiettivo in maniera più complessa ed accettano il fatto che la funzione obiettivo sia perturbata su tutto l'insieme di definizione.

Le prime funzioni filled risalgono alle fine degli anni ottanta. Recentemente si è avuto un crescente interesse verso queste funzioni che ha portato alla definizione di molte nuove funzioni filled. La caratteristica comune di tutte le funzioni filled proposte è il fatto di essere composte da due termini  $\eta_k(x)$  ed  $\phi_k(x)$ . A seconda di come sono combinati questi termini si hanno le seguenti due classi di funzioni filled:

- *funzioni filled moltiplicative*

$$U_k(x) = \phi_k(x)\eta_k(x),$$

- *funzioni filled additive*

$$U_k(x) = \phi_k(x) + \eta_k(x),$$

Nel seguito ci si limiterà a considerare solamente le funzioni filled additive che sono state introdotte recentemente e che hanno mostrato di essere più efficienti dal punto di vista computazionale.

I due termini  $\eta_k(x)$  ed  $\phi_k(x)$ , al di là delle particolari espressioni, giocano sempre ruoli simili:

- la funzione del termine  $\phi_k(x)$  è quello di eliminare i punti stazionari della funzione obiettivo che corrispondono a valori della funzione più grandi di  $f(x_k^*)$ ;
- la funzione del termine  $\eta_k(x)$  è quello di permettere che il punto  $x_k^*$  possa essere utilizzato come punto di partenza di una minimizzazione locale della funzione filled.

Il termine  $\phi_k(x)$  può dipendere da uno o più parametri e, come detto, cerca di eliminare i punti stazionari della funzione  $f(x)$  che hanno valore della funzione obiettivo più alto o uguale di  $f(x_k^*)$ . Questo fatto, se unito alla garanzia che la funzione filled ha un minimo globale, assicura che un algoritmo di minimizzazione locale è in grado di determinare un punto stazionario della funzione filled in cui il valore della funzione obiettivo è più basso di  $f(x_k^*)$ .

Una scelta molto semplice e, relativamente intuitiva, è la seguente:

$$\phi_k(x) = \tau \min\{0, f(x) - f(x_k^*) + \varrho\}^3, \quad (3.75)$$

dove  $\varrho > 0$  e  $\tau > 0$  sono parametri da scegliere. Tale termine è nullo in tutti i punti  $x$  in cui  $f(x) \geq f(x_k^*) - \varrho$ , mentre negli altri punti assume valori decrescenti all'aumentare del valore del parametro  $\tau$ . Nella figura (3.9) è riportato un esempio di una funzione obiettivo e della corrispondente funzione  $\phi_k(x)$ .

Un'altra scelta proposta per il termine  $\phi_k(x)$  è la seguente:

$$\phi_k(x) = 1 - \exp(-\tau(f(x) - f(x_k^*) + \varrho)), \quad (3.76)$$

dove, di nuovo,  $\varrho > 0$  e  $\tau > 0$  sono parametri da scegliere. Al crescere del valore del parametro  $\tau$  questo termine tende ad assumere valori pari ad uno in punti in cui  $f(x) \geq f(x_k^*) - \varrho$  mentre tende ad assumere valori sempre più decrescenti in punti in cui  $f(x) < f(x_k^*) - \varrho$ .

L'altro termine  $\eta_k(x)$ , che compare nella struttura di una funzione filled, ha il ruolo di assicurare che il vettore  $x_k^*$  non sia un punto stazionario della funzione filled oppure sia un massimo locale stretto in modo da poter utilizzare un algoritmo di minimizzazione locale per ottenere un punto stazionario in cui la funzione filled assume un valore più basso rispetto a quello che aveva in  $x_k^*$ .

Questa proprietà può essere garantita adottando due differenti strategie per scegliere la struttura della funzione  $\eta_k(x)$ . Queste due strategie danno luogo a due differenti classi di funzioni filled:

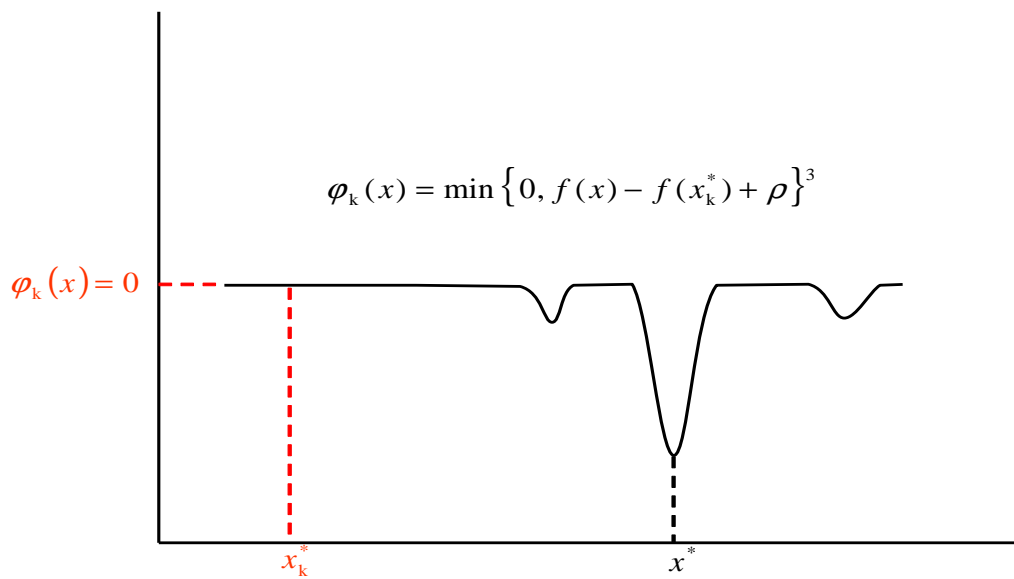
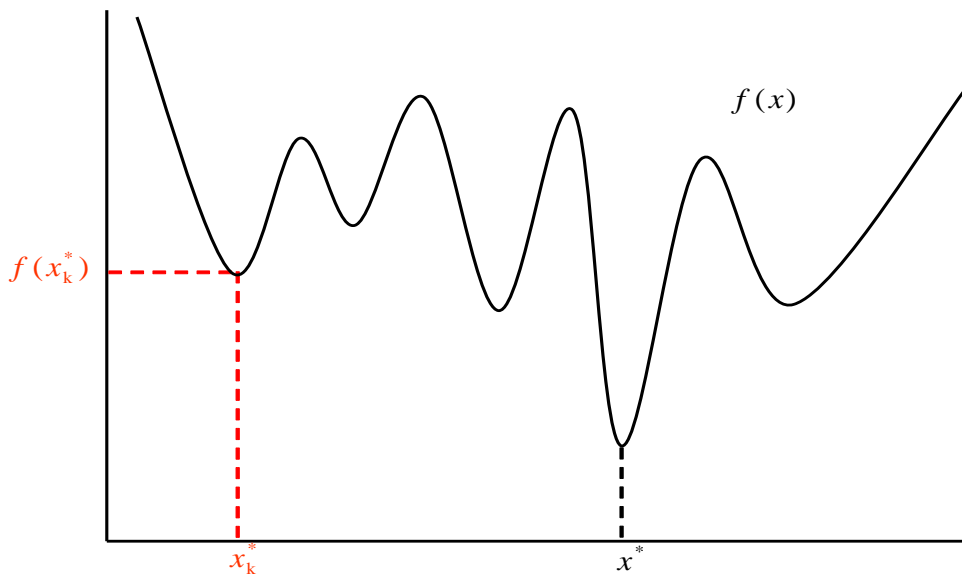


Figura 3.9: Esempio di una funzione obiettivo e della corrispondente funzione  $\phi_k(x) = \min\{0, \tau(f(x) - f(x_k^*) + \varrho)\}^3$ .

- le funzioni filled di tipo 1, in cui il termine  $\eta_k(x)$  cerca di garantire che, attraverso una minimizzazione locale, si possa ottenere un significativo spostamento dal punto  $x_k^*$ ;
- le funzioni filled di tipo 2, in cui il termine  $\eta_k(x)$  cerca di assicurare una “buona struttura” alla funzione filled.

### 3.3.1 Funzioni filled di tipo 1

La maggior parte delle funzioni filled che appartengono a questa classe utilizzano un termine  $\eta_k(x)$  che trasforma il punto stazionario  $x_k^*$  in un massimo locale della funzione filled stessa. Una scelta molto utilizzata per la funzione  $\eta_k(x)$  è la seguente:

$$\eta_k(x) = \exp\left(-\frac{\|x - x_k^*\|^2}{\gamma^2}\right), \quad (3.77)$$

dove  $\gamma > 0$  è una fissata costante.

Utilizzando come  $\phi_k(x)$  e  $\eta_k(x)$  le espressioni date da (3.75) e (da 3.77) si ottiene la seguente funzione filled additiva:

$$Q(x; \tau, \varrho) = \exp\left(-\frac{\|x - x_k^*\|^2}{\gamma^2}\right) + \min\{0, \tau(f(x) - f(x_k^*) + \varrho)\}^3. \quad (3.78)$$

In figura (3.10) è descritto un esempio di una funzione obiettivo e della corrispondente funzione  $Q_k(x)$ .

La prossima proposizione descrive le proprietà teoriche di una classe di funzioni filled di tipo 1 che hanno in comune il fatto di utilizzare come termine  $\eta(x)$  la funzione (3.77). Le particolari funzioni filled di questa classe si differenziano nella scelta del termine  $\phi_k(x)$  ed la proposizione seguente identifica le proprietà generali che questo termine deve soddisfare per garantire interessanti proprietà teoriche alla risultante funzione filled (nella proposizione si usa la seguente notazione  $\dot{\phi}(t) = d\phi(t)/dt$ ).

**Proposizione 3.3.1** *Sia la funzione  $f$  due volte continuamente differenziabile su  $R^n$  e sia dato un punto  $x_0 \in R^n$  tale che l'insieme*

$$\mathcal{L}_f(x_0) = \{x \in R^n : f(x) \leq f(x_0)\}$$

*sia compatto. Sia  $\phi : R \rightarrow R$  una funzione due volte continuamente differenziabile e tale che:*

- (a)  $\phi(t) \geq 0$ , per ogni  $t > 0$ ;
- (b)  $|\dot{\phi}(t)|$  è monotonicamente decrescente per valori positivi di  $t$  e  $\lim_{t \rightarrow \infty} t|\dot{\phi}(t)| = 0$ ;
- (c)  $\lim_{t \rightarrow -\infty} \phi(t) = -\infty$ .

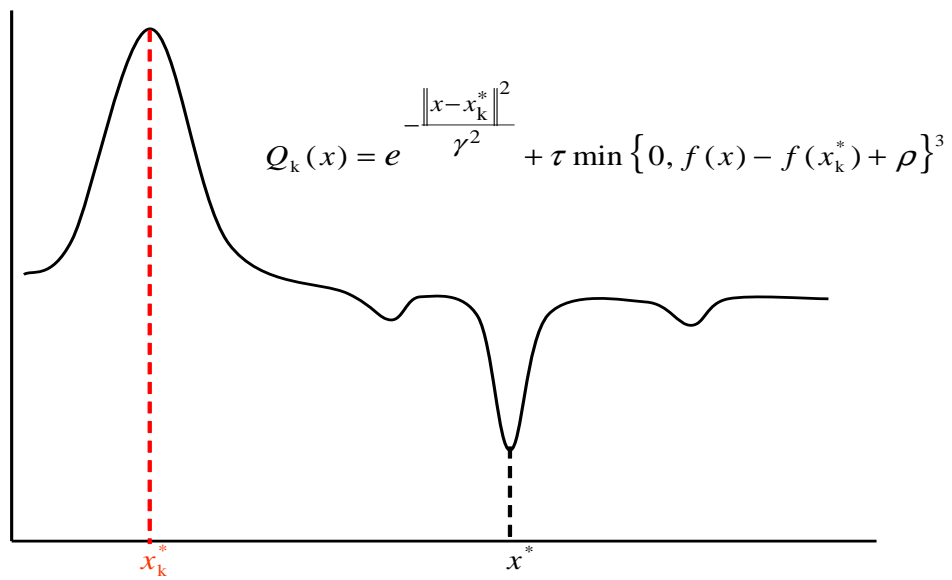
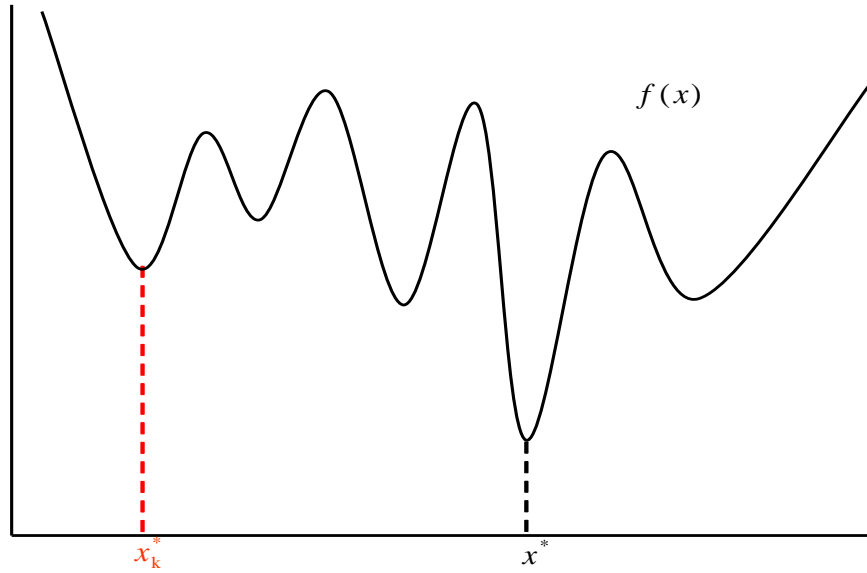


Figura 3.10: Esempio di una funzione obiettivo e della corrispondente funzione  $Q_k(x)$ .

Allora, per ogni punto stazionario  $x_k^*$  di  $f(x)$  in  $\mathcal{L}_f(x_0)$  e per ogni  $\varrho > 0$ , esiste un valore  $\bar{\tau} > 0$  tale che, per ogni  $\tau \geq \bar{\tau}$ , la funzione filled definita da

$$Q_k(x; \tau, \varrho) = \exp\left(-\frac{\|x - x_k^*\|^2}{\gamma^2}\right) + \phi(\tau(f(x) - f(x_k^*) + \varrho)), \quad (3.79)$$

dove  $\gamma > 0$  è una costante, ha le seguenti proprietà:

- (i) il punto  $x_k^*$  è un massimo locale isolato della funzione filled  $Q_k(x; \tau, \varrho)$ ;
- (ii)  $Q_k(x; \tau, \varrho)$  non ha punti stazionari non vincolati in  $\{x \in \mathcal{L}_f(x_0) : f(x) \geq f(x_k^*)\}$  eccetto  $x_k^*$ ;
- (iii) se  $x_k^*$  non è un minimo globale di  $f(x)$  e  $\varrho$  soddisfa la condizione

$$0 < \varrho < f(x_k^*) - f^*, \quad (3.80)$$

dove  $f^*$  è il valore ottimo di  $f(x)$ , allora tutti i minimi globali  $\check{x}$  della funzione filled  $Q_k(x; \tau, \varrho)$  su  $\mathcal{L}_f(x_0)$  sono punti stazionari non vincolati ed appartengono alla regione  $\{x \in \mathcal{L}_f(x_0) : f(x) < f(x_k^*)\}$ .

**Prova.** Prima di tutto si può notare che il gradiente di  $Q_k(x; \tau, \varrho)$  ha la seguente espressione:

$$\nabla Q_k(x; \tau, \varrho) = -2\frac{(x - x_k^*)}{\gamma^2} \exp\left(-\frac{\|x - x_k^*\|^2}{\gamma^2}\right) + \tau \nabla f(x) \dot{\phi}(\tau(f(x) - f(x_k^*) + \varrho)). \quad (3.81)$$

Punto (i). Poichè il punto  $x_k^*$  è un punto stazionario di  $f(x)$  la (3.81) implica che

$$\nabla Q_k(x_k^*, \tau, \varrho) = 0,$$

e quindi  $x_k^*$  è un punto stazionario di  $Q_k(x; \tau, \varrho)$ . Inoltre, si ha

$$\nabla^2 Q_k(x_k^*, \tau, \varrho) = -\frac{2}{\gamma^2} I + \tau \nabla^2 f(x_k^*) \dot{\phi}(\tau \varrho)$$

da cui segue che per ogni  $y \in \mathfrak{R}^n$

$$y^T \nabla^2 Q_k(x_k^*, \tau, \varrho) y = y^T \left( -\frac{2}{\gamma^2} I + \tau \nabla^2 f(x_k^*) \dot{\phi}(\tau \varrho) \right) y \leq \left( -\frac{2}{\gamma^2} + \tau \lambda_{\max} |\dot{\phi}(\tau \varrho)| \right) \|y\|^2,$$

dove  $\lambda_{\max} = \max_{x \in \mathcal{L}_f(x_k^*)} \lambda(\nabla^2 f(x_k^*))$ , che implica, insieme alla proprietà (b), l'esistenza di un valore  $\tau_1 > 0$  tale che, per tutti  $\tau \geq \tau_1$ , la matrice Hessiana  $\nabla^2 Q_k(x_k^*, \tau, \varrho)$  è definita negativa. Perciò il punto  $x_k^*$  è un massimo locale isolato di  $Q_k(x; \tau, \varrho)$  per tutti  $\tau \geq \tau_1$ .

Punto (ii). Si supponga, per assurdo, che esista un punto stazionario nonvincolato  $\hat{x} \in \mathcal{L}_f(x_0)$  di  $Q_k(x; \tau, \varrho)$  tale che  $\hat{x} \neq x_k^*$  e  $f(\hat{x}) \geq f(x_k^*)$ . Ricordando l'espressione (3.81) del gradiente di  $Q_k(x; \tau, \varrho)$ , in tale punto dovrebbe essere soddisfatta la seguente uguaglianza:

$$2\frac{\|\hat{x} - x_k^*\|}{\gamma^2} \exp\left(-\frac{\|\hat{x} - x_k^*\|^2}{\gamma^2}\right) = \tau \|\nabla f(\hat{x})\| |\dot{\phi}(\tau(f(\hat{x}) - f(x_k^*) + \varrho))|. \quad (3.82)$$



Il punto (i) implica l'esistenza di un  $\epsilon > 0$  tale che  $\|\hat{x} - x_k^*\| > \epsilon$ . Dalla compattezza di  $\mathcal{L}_f(x_0)$  segue che esistono due costanti positive  $\sigma_1$  ed  $\sigma_2$  tali che  $\|x - x_k^*\| \leq \sigma_1$  e  $\|\nabla f(x)\| \leq \sigma_2$  per tutti  $x \in \mathcal{L}_f(x_0)$ . Perciò (3.82) e proprietà (b) implicherebbero

$$2\frac{\epsilon}{\gamma^2} \exp\left(-\frac{\sigma_1^2}{\gamma^2}\right) \leq 2\frac{\|\hat{x} - x_k^*\|}{\gamma^2} \exp\left(-\frac{\|\hat{x} - x_k^*\|^2}{\gamma^2}\right) \quad (3.83)$$

e

$$\tau\|\nabla f(\hat{x})\| |\dot{\phi}(\tau(f(\hat{x}) - f(x_k^*) + \varrho))| \leq \tau\sigma_2|\dot{\phi}(\tau\varrho)|. \quad (3.84)$$

Quindi le (3.83),(3.84) e la proprietà (b) implicherebbero l'esistenza di un  $\tau_2 \geq \tau_1$  tale che per tutti  $\tau \geq \tau_2$  la condizione (3.82) non sarebbe vera.

Punto (iii). La compattezza di  $\mathcal{L}_f(x_0)$  implica l'esistenza di un minimo globale  $\tilde{x} \in \mathcal{L}_f(x_0)$  di  $Q_k(x; \tau, \varrho)$ . Per ogni  $\hat{x} \in \partial\mathcal{L}_f(x_0)$  si ha che  $f(\hat{x}) \geq f(x_0) \geq f(x_k^*)$ . Allora dalla proprietà (a) si ottiene che

$$Q_k(\hat{x}, \tau, \varrho) = \exp\left(-\frac{\|\hat{x} - x_k^*\|^2}{\gamma^2}\right) + \phi(\tau(f(\hat{x}) - f(x_k^*) + \varrho)) > 0.$$

Sia  $x^*$  un minimo globale di  $f(x)$ . Allora la (3.80) e la proprietà (c) implicano che esiste un  $\tau_3$  tale che, per tutti  $\tau \geq \tau_3$ , si ha:

$$Q_k(x^*, \tau, \varrho) < 0$$

da cui si ottiene:

$$Q_k(\tilde{x}, \tau, \varrho) \leq Q_k(x^*, \tau, \varrho) < 0, \quad (3.85)$$

che implica che  $\tilde{x}$  è strettamente interno a  $\mathcal{L}_f(x_0)$ .  $\square$

Come detto, nella precedente proposizione non viene specificata una struttura particolare per la funzione  $\phi(t)$ , invece si caratterizzano delle proprietà generali che una tale funzione deve soddisfare per garantire la tesi della proposizione. Si può notare che le scelte (3.75) e (3.76) (cioè  $\phi(t) = \min\{0, t\}^3$  e  $\phi(t) = 1 - e^{-t}$ ) soddisfano le proprietà (a)-(c).

Il precedente risultato mostra che, per valori opportuni dei parametri  $\tau$  e  $\varrho$ , le funzioni filled (3.79) presentano l'interessante caratteristica di avere come unico punto stazionario nell'insieme  $\{x \in \mathcal{L}_f(x_0) : f(x) \geq f(x_k^*)\}$  il punto  $x_k^*$  che è un massimo locale isolato e che, quindi, non è un "punto di attrazione" di un algoritmo di minimizzazione locale. Mentre, se  $x_k^*$  non è un minimo globale di  $f(x)$ , gli altri punti stazionari della funzione filled appartenenti a  $\mathcal{L}_f(x_0)$  hanno valore della funzione minore di  $f(x_k^*)$ .

Dal punto di vista applicativo le funzioni filled appartenenti a questa classe presentano alcuni difetti. Il primo è costituito dall'impossibilità di dimostrare che non abbiano punti stazionari al di fuori dell'insieme  $\mathcal{L}_f(x_0)$  e che, quindi, un algoritmo di minimizzazione locale applicato a queste funzioni filled non possa essere attratto da questi punti stazionari che possono avere valori della funzione obiettivo più alti di  $f(x_k^*)$  (per esempio, nella figura (3.11) si indentifica l'insieme  $\mathcal{L}_f(x_0)$  e nella figura (3.12) si indica, con le

frece, le zone in cui una minimizzazione locale potrebbe essere attratta). L'altro pesante difetto è costituito dal fatto che hanno insiemi di livello che non sono compatti (si può osservare il comportamento della funzione filled (3.78)) e questo inficia le proprietà teoriche e computazionali degli algoritmi di minimizzazione locale da utilizzare per minimizzarle.

### 3.3.2 Funzioni filled di tipo 2.

Le funzioni filled appartenenti a questa classe si differenziano dalle precedenti nella scelta della funzione  $\eta_k(x)$ . Infatti, come già accennato, le funzioni  $\eta_k(x)$  utilizzate in queste funzioni filled hanno anche il ruolo di garantire che le funzioni filled abbiano delle strutture che possono essere facilmente minimizzate attraverso algoritmi locali. La proprietà più importante da garantire è il fatto che le funzioni filled abbiano insiemi di livello compatti. Per riuscire ad ottenere questa proprietà si deve rinunciare al fatto che il punto  $x_k^*$  sia trasformato in un massimo, ma ci si deve accontentare di trasformarlo in un punto non stazionario della funzione filled.

Una scelta molto semplice utilizzata in questa classe di funzioni filled è la seguente:

$$\eta_k(x) = \|x - \tilde{x}\|^2, \quad (3.86)$$

dove  $\tilde{x}$  è un qualsiasi punto su  $R^n$ .

Il ruolo del termine  $\phi_k(x)$  per queste funzioni filled è lo stesso delle funzioni filled considerate precedentemente. In particolare una funzione filled additiva appartenente a questa classe è la seguente

$$V_k(x; \tau, \varrho) = \|x - \tilde{x}\|^2 + \min\{0, \tau(f(x) - f(x_k^*) + \varrho)\}^3. \quad (3.87)$$

Nella figura (3.13) è descritto un esempio di costruzione della precedente funzione filled a partire da una data funzione obiettivo.

La prossima proposizione descrive le proprietà teoriche di una funzioni filled di tipo 2 che utilizzano come termine  $\eta_k(x)$  la (3.86) (nuovamente si usa la notazione  $\dot{\phi}(t) = d\phi(t)/dt$ ).

**Proposizione 3.3.2** *Sia la funzione obiettivo due volte continuamente differenziabile su  $R^n$  e sia tale che*

$$\lim_{\|x\| \rightarrow \infty} f(x) = \infty. \quad (3.88)$$

*Sia  $\phi : R \rightarrow R$  una funzione una volta continuamente differenziabile e tale che:*

- (a)  $\phi(t) < 0$  per ogni  $t < 0$ ,  $\phi(0) = 0$ ,  $\phi(t) \geq 0$  per ogni  $t \geq 0$ ;
- (b)  $|\dot{\phi}(t)|$  è monotonicamente decrescente per valori positivi di  $t$  e  $\lim_{t \rightarrow \infty} t|\dot{\phi}(t)| = 0$ ;
- (c)  $\lim_{t \rightarrow \infty} \phi(t) = B \geq 0$ ;

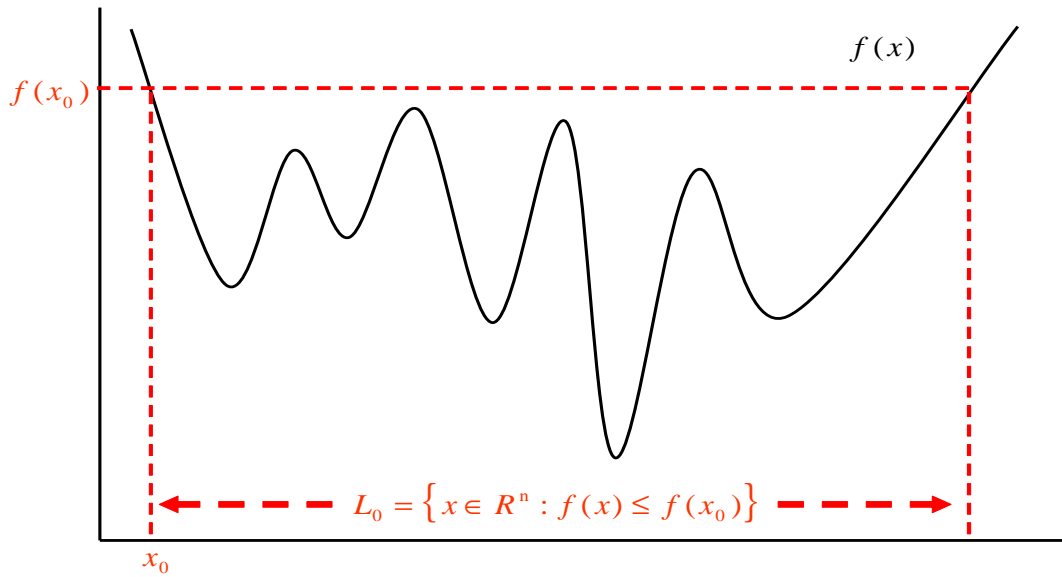


Figura 3.11: Esempio di un insieme  $\mathcal{L}_f(x_0) = \{x \in R^n : f(x) \leq f(x_0)\}$ .

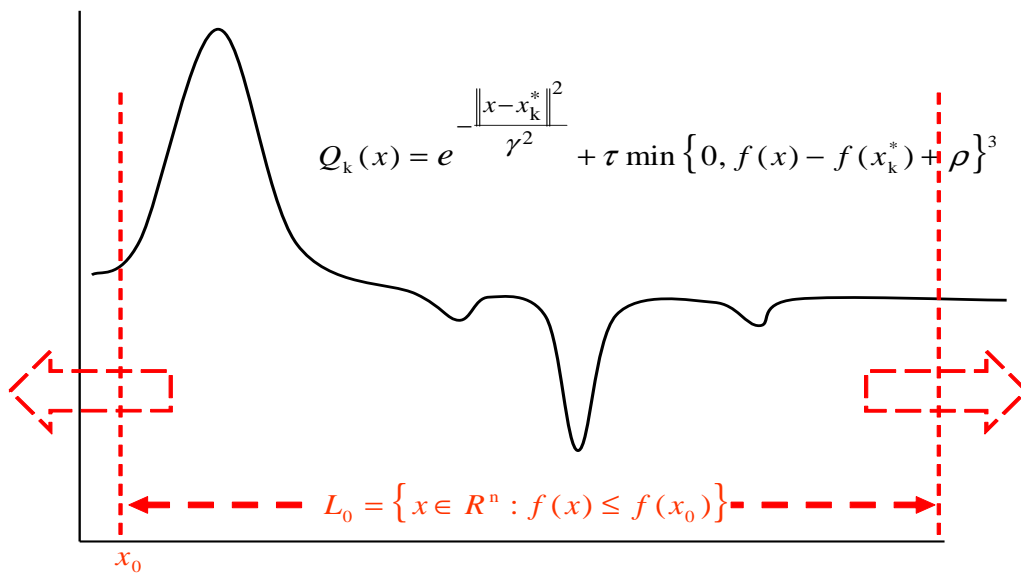


Figura 3.12: l'insieme  $\mathcal{L}_f(x_0) = \{x \in R^n : f(x) \leq f(x_0)\}$  riferito alla funzione filled  $Q_k$ .

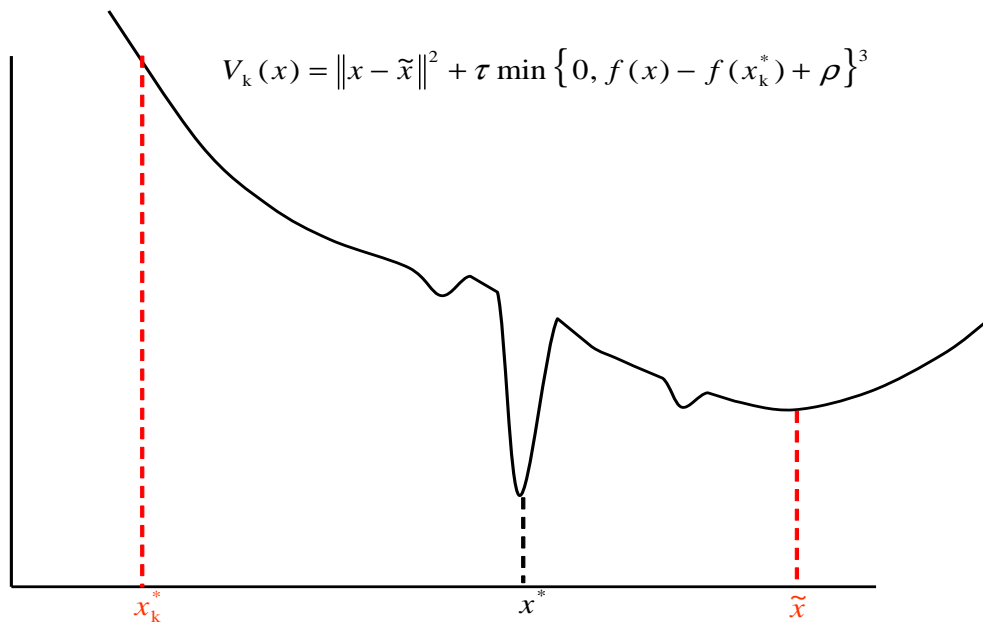
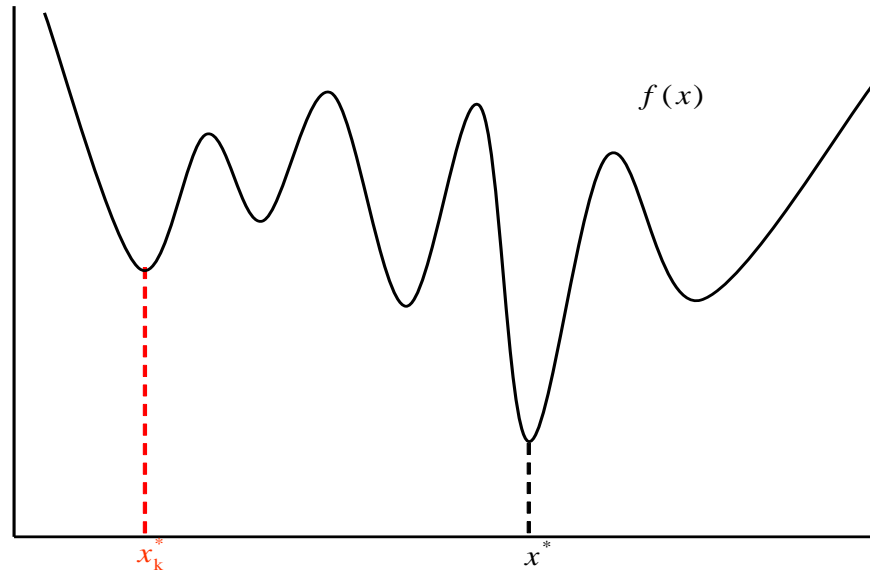


Figura 3.13: Esempio di una funzione obiettivo e della corrispondente funzione  $V_k(x)$ .

(d)  $\lim_{t \rightarrow -\infty} \phi(t) = -\infty$ .

Allora la seguente la funzione filled:

$$V_k(x; \tau, \varrho) = \|x - \tilde{x}\|^2 + \phi(\tau(f(x) - f(x_k^*) + \varrho)) \quad (3.89)$$

ha le seguenti proprietà:

(i) per ogni punto stazionario  $x_k^*$ , per ogni  $\varrho > 0$  e per ogni  $\tau > 0$  esiste un insieme compatto  $\Delta$  tale che:

$$\mathcal{L}_{V_k}(x_0, \tau, \varrho) = \{x \in R^n : V_k(x, \tau, \varrho) \leq V_k(x_0, \tau, \varrho)\} \subseteq \Delta. \quad (3.90)$$

Per ogni punto stazionario  $x_k^*$  di  $f(x)$ , per ogni  $\varrho > 0$  ed ogni  $\varepsilon > 0$ , esiste un valore  $\bar{\tau} > 0$  tale che, per ogni  $\tau \geq \bar{\tau}$ , si ha:

(ii) la funzione  $V_k(x; \tau, \varrho)$  non ha punti stazionari in  $\{x \in \mathcal{L}_{V_k}(x_0, \tau, \varrho) : f(x) \geq f(x_k^*)\}$  eccetto in un intorno  $\mathcal{B}(\tilde{x}; \varepsilon)$  di  $\tilde{x}$ , dove può esserci un minimo locale isolato.

(iii) se  $x_k^*$  non è un minimo globale di  $f(x)$  e  $\varrho$  soddisfa la condizione

$$0 < \varrho < f(x_k^*) - f^*, \quad (3.91)$$

dove  $f^*$  è il valore ottimo di  $f(x)$ , allora tutti i minimi globali  $\tilde{x}$  della funzione filled  $V_k(x; \tau, \varrho)$  appartengono alla regione  $\{x \in \mathcal{L}_{V_k}(x_0, \tau, \varrho) : f(x) < f(x_k^*)\}$ .

Anche in questa proposizione non è specificata l'espressione della funzione  $\phi(t)$ , in modo da caratterizzare scelte diverse per la funzione  $\phi_k$  (tra quelle che sono accettabili ci sono, di nuovo, sia  $\phi(t) = \min\{0, t\}^3$  e  $\phi(t) = 1 - e^{-t}$ ).

Si noti che il precedente risultato mostra che queste funzioni filled hanno insiemi di livello compatti. Inoltre i punti stazionari di interesse di  $V_k(x; \tau, \varrho)$  appartengono all'insieme  $\mathcal{L}_{V_k}(x_0, \tau, \varrho)$ . Perciò, ogni algoritmo che minimizza localmente la  $V_k(x; \tau, \varrho)$  e che usa come punto di partenza un punto appartenente all'insieme  $\mathcal{L}_{V_k}(x_0, \tau, \varrho)$ , non può essere attratto da punti stazionari che sono al di fuori di  $\mathcal{L}_{V_k}(x_0, \tau, \varrho)$ . Quindi queste nuove funzioni filled superano molti dei problemi delle precedenti funzioni filled. Tuttavia, come mostra il punto (ii) della precedente proposizione, queste funzioni filled presentano il difetto di avere un punto di minimo locale in cui il valore della funzione obiettivo può essere più grande o uguale al valore  $f(x_k^*)$ . Sebbene si sappia dove sia questo punto stazionario "spurio" ("arbitrariamente" vicino al punto  $\tilde{x}$ ) non si può evitare che "attragga" un algoritmo di minimizzazione locale. Nella figura (3.14) è descritto un esempio in cui una minimizzazione locale di  $V_k$  produce un punto stazionario in cui si ha un miglioramento della funzione obiettivo. Nella figura (3.15) è invece riportato un esempio in cui una minimizzazione locale di  $V_k$  produce un punto stazionario spurio.

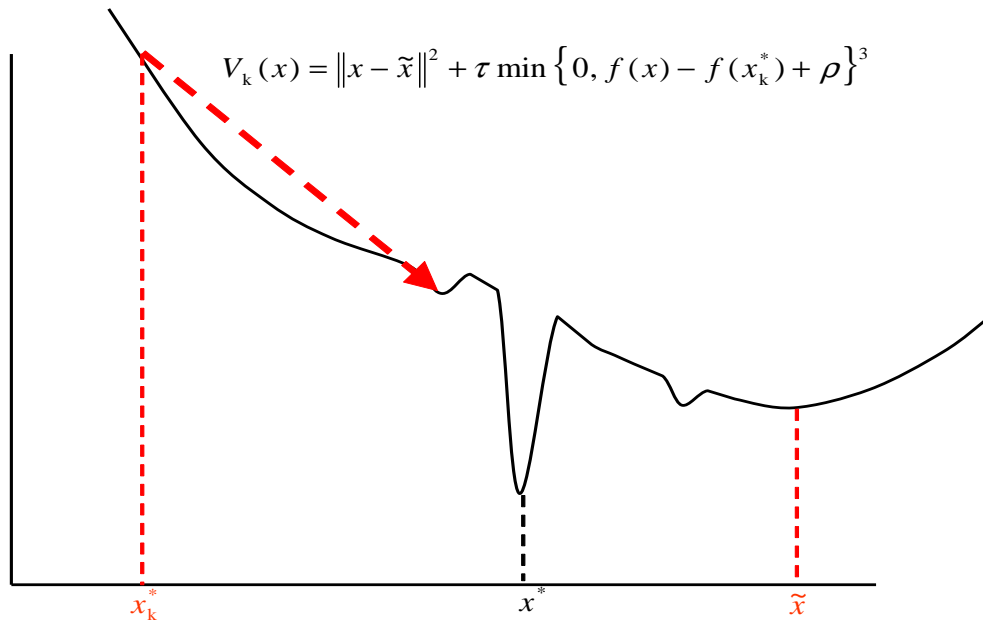


Figura 3.14: Esempio in cui una minimizzazione locale di  $V_k$  produce un punto stazionario in cui si ha un miglioramento della funzione obiettivo.

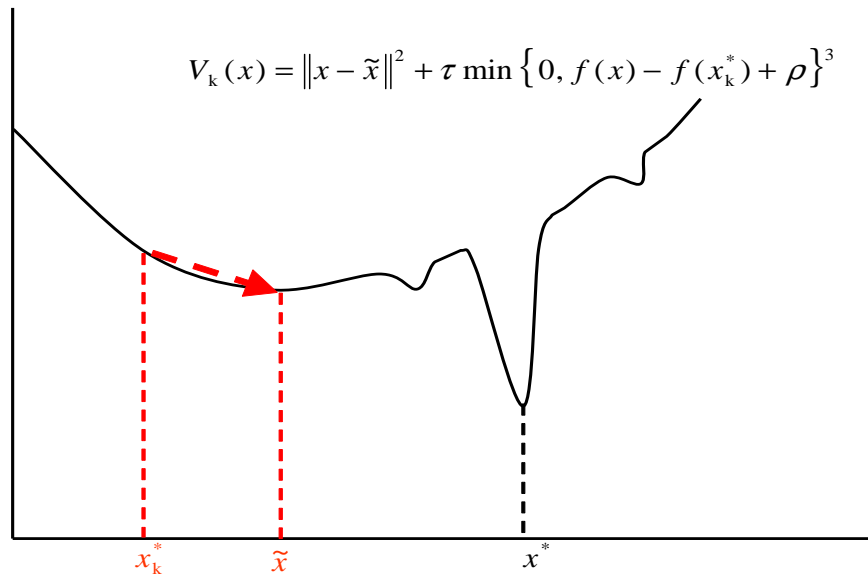


Figura 3.15: Esempio in cui una minimizzazione locale di  $V_k$  produce un punto stazionario spurio.