Ottimizzazione dei Sistemi Complessi

G. Liuzzi[1]

Venerdì 3 Marzo 2017

[1] Istituto di Analisi dei Sistemi ed Informatica IASI - CNR

Pseudo-code di "compass search" rivisto o deboleUn nuovo metodo

INPUT: x_0, Δ_0, Δ_{min}, $\mathtt{maxit}$, $D = \{\pm e_i, i = 1, \dots, n\}$
$k \leftarrow 0$, $x \leftarrow x_0$, $\Delta \leftarrow \Delta_0$
while $k \leq \mathtt{maxit}$ **and** $\Delta \geq \Delta_{min}$ **do**
  $k \leftarrow k + 1$, $y \leftarrow x$
  Let $\bar{d} \in D$ be s.t. $f(x + \Delta \bar{d}) = \min_{d_i \in D} f(x + \Delta d_i)$
  if $f(x + \Delta \bar{d}) < f(x)$ **then**

  for each $\bar{d} \in D$
  if $\exists\ \bar{d} \in D$ s.t. $f(xy + \Delta \bar{d}) < f(xy)$ **then**
    if $f(y + \Delta \bar{d}) < f(y)$ **then**
    $x \leftarrow x + \Delta \bar{d}$
    $y \leftarrow y + \Delta \bar{d}$
      $y \leftarrow y + \Delta \bar{d}$, **break**
      $y \leftarrow y + \Delta \bar{d}$
    endif
  end for

Un “nuovo” metodo

INPUT: x_0, Δ_0, Δ_{min}, $\mathtt{maxit}$, $D = \{\pm e_i, i = 1, \ldots, n\}$

$k \leftarrow 0$, $x \leftarrow x_0$, $\Delta \leftarrow \Delta_0$

while $k \leq \mathtt{maxit}$ **and** $\Delta \geq \Delta_{min}$ **do**

$k \leftarrow k + 1$, $y \leftarrow x$
for each $d \in D$
if $f(y + \Delta d) < f(y)$ **then** $y \leftarrow y + \Delta d$
end for
if $f(y) < f(x)$ **then**
$x \leftarrow y$
else $\Delta \leftarrow \Delta/2$

end while

RETURN: x (miglior punto determinato)

Domande?

Il metodo di Hooke&Jeeves

INPUT: x_0, Δ_0, Δ_{min}, $\mathtt{maxit}$, $D = \{\pm e_i, i = 1, \dots, n\}$

$k \leftarrow 0$, $x \leftarrow x_0$, $\Delta \leftarrow \Delta_0$

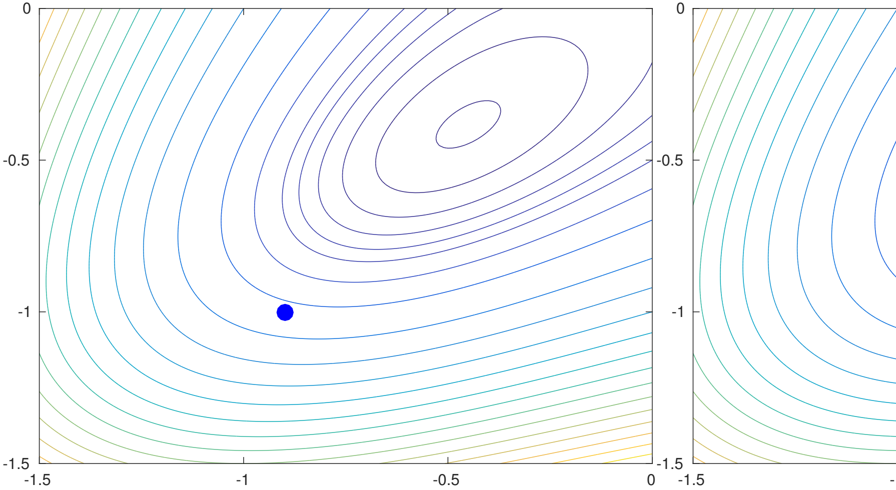
while $k \leq \mathtt{maxit}$ **and** $\Delta \geq \Delta_{min}$ **do**

> $k \leftarrow k + 1$, $y \leftarrow x$
> **for each** $d \in D$ (*exploratory moves from* x)
> > **if** $f(y + \Delta d) < f(y)$ **then** $y \leftarrow y + \Delta d$
>
> **end for**
> **if** $f(y) < f(x)$ **then** (*pattern move along* $y - x$)
> > $z \leftarrow y + (y - x)$
> > **for each** $d \in D$ (*exploratory moves from* z)
> > > **if** $f(z + \Delta d) < f(z)$ **then** $z \leftarrow z + \Delta d$
> >
> > **end for**
> > **if** $f(z) < f(y)$ **then** $x \leftarrow z$ **else** $x \leftarrow y$
>
> **else** $\Delta \leftarrow \Delta/2$

end while

RETURN: x (miglior punto determinato)

Esempio su Funzione di Broyden



Convergenza di Hooke&Jeeves

Nota bene: Come per i metodi precedenti, nel metodo di H&J il passo è ridotto solo quando il punto corrente non cambia

Non stupisce che H&J abbia le medesime proprietà teoriche e cioè

- $\lim_{k \to \infty} \Delta_k = 0$
- almeno un punto limite è stazionario
    - $\nabla f(\bar{x}) = 0$
    - $\liminf_{k \to \infty} \|\nabla f(x_k)\| = 0$

Il metodo di Nelder-Mead

Alla generica iterazione k lo stato dell'algoritmo è dato da:

$$X_k = \{x_1, x_2, \ldots, x_{n+1}\}$$

indichiamo $f_i = f(x_i)$ e supponiamo X_k ordinato in modo che

$$f_1 \le f_2 \le \cdots \le f_{n+1}.$$

Ricordiamo l'espressione del "centroide" dei primi n punti in X_k

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Idea: Sfruttare x_{n+1} e $\bar{x}$ per cercare un punto "migliore" di x_{n+1}

Il metodo di Nelder-Mead

Dato un parametro μ, definiamo

$$x(\mu) = \bar{x} + \mu(\bar{x} - x_{n+1})$$

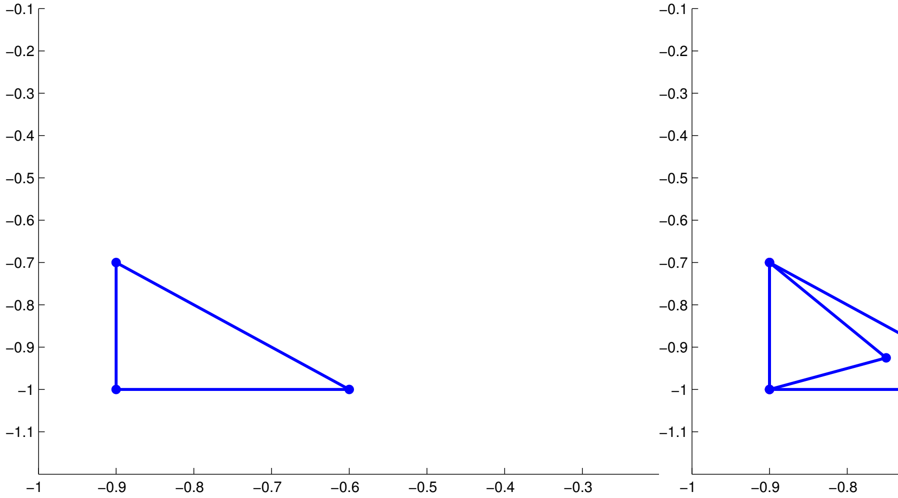
Il metodo usa: $-1 < \mu_{ic} < 0 < \mu_{oc} < \mu_r < \mu_e$ e

$$\begin{array}{lllllll}
\text{(inner contraction)} & x^{ic} & = & x(\mu_{ic}), & f^{ic} & = & f(x^{ic}) \\
\text{(outer contraction)} & x^{oc} & = & x(\mu_{oc}), & f^{oc} & = & f(x^{oc}) \\
\text{(reflect)} & x^{r} & = & x(\mu_{r}), & f^{r} & = & f(x^{r}) \\
\text{(expand)} & x^{e} & = & x(\mu_{e}), & f^{e} & = & f(x^{e})
\end{array}$$

Iterazione k

- Se $\boxed{f_1 \le f^r < f_n}$, allora $X_{k+1} = X_k \setminus \{x_{n+1}\} \cup \{x^r\}$ **FINE**
- Se $\boxed{f^r < f_1}$, allora
  - Se $f^e < f^r$, allora $X_{k+1} = X_k \setminus \{x_{n+1}\} \cup \{x^e\}$ **FINE**
  - altrimenti $X_{k+1} = X_k \setminus \{x_{n+1}\} \cup \{x^r\}$ **FINE**
- Se $\boxed{f_n \le f^r < f_{n+1}}$, allora
  - Se $f^{oc} \le f^r$, allora $X_{k+1} = X_k \setminus \{x_{n+1}\} \cup \{x^{oc}\}$ **FINE**
  - altrimenti **shrink**
- Se $\boxed{f_{n+1} \le f^r}$, allora
  - Se $f^{ic} < f_{n+1}$, allora $X_{k+1} = X_k \setminus \{x_{n+1}\} \cup \{x^{ic}\}$ **FINE**
  - altrimenti **shrink**
- **shrink**:
  - $X_{k+1} = \{x_1, \hat{x}_2, \dots, \hat{x}_{n+1}\}$ dove
  - $\hat{x}_i = x_1 + \gamma(x_i - x_1)$, $i = 2, \dots, n+1$, $\gamma \in (0,1)$

Esempio su Funzione di Broyden



È un "buon" metodo?

La risposta è:

- generalmente **Si**
- però non ha proprietà di convergenza a punti stazionari
- esistono contro-esempi sui quali converge a punti **NON** stazionari
- p.es. la funzione di McKinnon

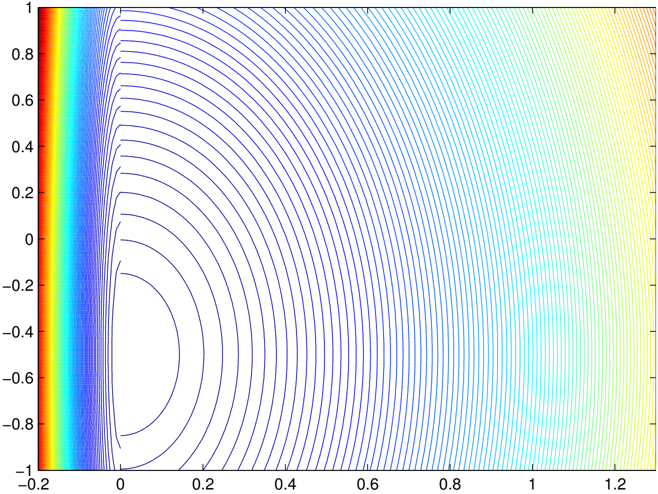
La funzione di McKinnon

$$f(x,y) = \begin{cases} \theta\phi|x|^\tau + y + y^2 & \text{se } x \le 0 \\ \theta x^\tau + y + y^2 & \text{se } x > 0 \end{cases}$$

- strettamente convessa per $\tau > 1$
- continuamente differenziabile per $\tau > 1$
- due volte cont. differenziabile per $\tau > 2$
- tre volte cont. differenziabile per $\tau > 3$

La funzione di McKinnon

Per $\tau = 2$, $\theta = 6$, $\phi = 60$, la funzione è



La funzione di McKinnon

Se inizializziamo il metodo di Nelder-Mead con il simplesso

$$X = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{1+\sqrt{33}}{88} \\ \frac{1-\sqrt{33}}{88} \end{pmatrix}, \right\}$$

La funzione di McKinnon