Metodi senza derivate per l'Ottimizzazione non vincolata

Marco Sciandrone

Dipartimento di Informatica e Sistemistica

Università degli Studi di Roma "La Sapienza"

1 Generalità

Molti problemi di ottimizzazione derivanti da applicazioni reali sono caratterizzati dal fatto che le derivate parziali della funzione obiettivo non sono analiticamente note.

Gli algoritmi che non utilizzano la conoscenza delle derivate della funzione obiettivo possono essere raggruppati in due classi: metodi che fanno uso di *approssimazioni alle differenze finite* e *metodi di ricerca diretta*.

L'approssimazione alle differenze finite delle derivate può essere utilizzata in connessione con metodi tipo-Newton, Quasi-Newton o gradiente coniugato. Un simile approccio può risultare poco vantaggioso nei casi in cui la valutazione della funzione obiettivo è molto onerosa in termini computazionali e può essere affetta da rumore (tale situazione può verificarsi, ad esempio, quando il valore della funzione obiettivo è ottenuto mediante misurazioni effettuate su sistemi fisici molto complessi).

I metodi di ricerca diretta sono basati sul confronto diretto dei valori della funzione obiettivo nei punti generati iterativamente dall'algoritmo e possono essere distinti in tre classi differenti. Nei metodi della prima classe (si veda [3],[5],[6],[7],[8]), usualmente chiamati metodi di tipo *pattern search*, la sequenza dei punti è ottenuta mediante il campionamento della funzione obiettivo lungo opportune direzioni. La caratteristica comune dei metodi della seconda classe (si veda [2],[9]), denotati come metodi di tipo *line search*, è costituita dal fatto che eseguono minimizzazioni unidimensionali lungo direzioni di ricerca. I metodi della terza classe (si veda [1],[4]) sono basati sull'idea di approssimare la funzione obiettivo con un modello che viene iterativamente aggiornato.

Risultati di convergenza globale sono noti per metodi tipo pattern search e metodi tipo line search. Nel seguito vengono descritte le versioni più semplici di alcuni metodi tipo pattern search e sono riportati i risultati di convergenza. In Appendice sono richiamate alcune proprietà topologiche dello spazio euclideo utilizzate nella prova di una proposizione basilare per dimostrare la convergenza dei metodi illustrati.

2 Metodi basati sulle direzioni coordinate

Si consideri il problema di ottimizzazione non vincolata

$$\begin{array}{ll} \min & f(x) \\ x \in R^n & \end{array}$$

dove la funzione obiettivo $f : R^n \to R$ è continuamente differenziabile e si assuma che il gradiente ∇f non sia noto.
Siano e^i, $i = 1, \dots, n$, le direzioni degli assi coordinati:

$$e^1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad e^2 = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix}, \quad e^n = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}.$$

Un metodo tipo pattern search basato sulle direzioni coordinate è descritto dallo schema seguente.

Algoritmo 1.

Dati. $x_0 \in R^n$, $\alpha_0 > 0$, $\theta \in (0,1)$.

Passo 0. Poni $k = 0$.

Passo 1. Poni $i = 1$.

Passo 2a. Se $f(x_k + \alpha_k e^i) < f(x_k)$ vai al Passo 5a;

Passo 2b. Se $f(x_k - \alpha_k e^i) < f(x_k)$ vai al Passo 5b;

Passo 3. Se $i < n$ poni $i = i + 1$ e vai al Passo 2a.

Passo 4. Poni $x_{k+1} = x_k$, $\alpha_{k+1} = \theta\alpha_k$ e vai al Passo 7.

Passo 5a. Poni $x_{k+1} = x_k + \alpha_k e^i$ e vai al Passo 6.

Passo 5b. Poni $x_{k+1} = x_k - \alpha_k e^i$.

Passo 6. Poni $\alpha_{k+1} = \alpha_k$.

Passo 7. Poni $k = k + 1$ e vai al Passo 1.

Al fine di dimostrare il risultato di convergenza relativo all'Algoritmo 1, premettiamo la seguente proposizione.

Proposizione 1 *Si assuma che* $\mathcal{L}(x_0) = \{x \in R^n : f(x) \leq f(x_0)\}$ *sia compatto; allora*

$$\lim_{k\to\infty} \alpha_k = 0. \tag{1}$$

Dim. Le istruzioni dell'algoritmo assicurano che $\alpha_{k+1} \leq \alpha_k$ per ogni $k \geq 0$. La sequenza $\{\alpha_k\}$ è quindi monotona decrescente ed è limitata inferiormente ($\alpha_k > 0$ per ogni $k \geq 0$), ciò implica che esiste il limite per k che tende all'infinito. Per dimostrare la tesi, supponiamo per assurdo che non sia vera:

$$\lim_{k\to\infty} \alpha_k = \bar{\alpha} > 0. \tag{2}$$

Per ogni $k \geq 0$ si ha

$$\alpha_{k+1} = \theta^{k_1}\alpha_0, \tag{3}$$

dove $k_1 \leq k + 1$ rappresenta il numero di volte che è stato eseguito il Passo 4 nelle prime $(k + 1)$ iterazioni dell'algoritmo. Utilizzando la (2) e la (3), considerando che $\theta \in (0,1)$, si può affermare che il Passo 4 viene eseguito un numero finito di volte, quindi esiste un $\bar{k} \geq 0$ tale che per $k \geq \bar{k}$ il Passo 4 non viene più eseguito. Ciò implica che per $k \geq \bar{k}$ risulta $f(x_{k+1}) < f(x_k)$ e

$$\alpha_k = \alpha_{\bar{k}} = \bar{\alpha}. \tag{4}$$

Si ha perciò

$$\|x_{k+1} - x_k\| = \bar{\alpha} \qquad \forall k \geq \bar{k}. \tag{5}$$

Faremo ora vedere che la sottosequenza $\{x_k\}_{k \geq \bar{k}}$ è costituita da un numero finito di punti. A tal fine, si può osservare innanzitutto che la sequenza $\{x_k\}$, e quindi anche la sottosequenza $\{x_k\}_{k \geq \bar{k}}$, appartiene all' insieme compatto $\mathcal{L}(x_0)$ (infatti risulta $f(x_{k+1}) \leq f(x_k)$ per ogni $k \geq 0$).

Si considerino ora due qualsiasi punti distinti $x_{\bar{k}+i}$, $x_{\bar{k}+j}$ ($i \neq j$ e per comodità si assuma $i, j \geq 1$) della sottosequenza $\{x_k\}_{k \geq \bar{k}}$. Utilizzando la (5) ed il fatto che i punti sono generati lungo direzioni parallele agli assi coordinati, si può scrivere

$$x_{\bar{k}+i} = x_{\bar{k}} + \bar{\alpha} \sum_{t=0}^{i-1} e^{l_t(i)} \qquad e^{l_t(i)} \in \{e^1, e^2, \ldots, e^n, -e^1, -e^2, \ldots, -e^n\}, \tag{6}$$

$$x_{\bar{k}+j} = x_{\bar{k}} + \bar{\alpha} \sum_{t=0}^{j-1} e^{l_t(j)} \qquad e^{l_t(j)} \in \{e^1, e^2, \ldots, e^n, -e^1, -e^2, \ldots, -e^n\}, \tag{7}$$

da cui

$$x_{\bar{k}+i} - x_{\bar{k}+j} = \bar{\alpha} \left(\sum_{t=0}^{i-1} e^{l_t(i)} - \sum_{t=0}^{j-1} e^{l_t(j)} \right). \tag{8}$$

Poichè $x_{\bar{k}+i}$ e $x_{\bar{k}+j}$ sono distinti si ottiene

$$\sum_{t=0}^{i-1} e^{l_t(i)} - \sum_{t=0}^{j-1} e^{l_t(j)} \neq \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

ciò implica che

$$\sum_{t=0}^{i-1} e^{l_t(i)} - \sum_{t=0}^{j-1} e^{l_t(j)} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}, \tag{9}$$

dove $a_1, \ldots, a_n$ sono interi non tutti nulli. Si ha di conseguenza

$$\| \sum_{t=0}^{i-1} e^{l_t(i)} - \sum_{t=0}^{j-1} e^{l_t(j)} \| \geq 1,$$

ed utilizzando la (8) si può concludere che risulta

$$\|x_{\bar{k}+i} - x_{\bar{k}+j}\| \geq \bar{\alpha} \qquad \forall i, j = 0, 1, \ldots, i \neq j. \tag{10}$$

I punti della sottosequenza $\{x_k\}_{\bar{k}}$ appartengono quindi ad un insieme compatto e soddisfano la (10): per la Proposizione 3 (si veda in Appendice) si può perciò affermare che la sottosequenza $\{x_k\}_{k\geq\bar{k}}$ è costituita da un numero finito di punti. Ciò implica che esistono due interi $k_2 > k_1 \geq \bar{k}$ tali che $x_{k_1} = x_{k_2}$, il che contraddice il fatto che per $k \geq \bar{k}$ deve risultare $f(x_{k+1}) < f(x_k)$. $\Box$

Le proprietà di convergenza dell'Algoritmo 1 sono riportate nel seguente teorema.

Teorema 1 *Sia $\{x_k\}$ la successione generata dall'Algoritmo 1 e si assuma che $\mathcal{L}(x_0)$ sia compatto; allora*

$$\lim_{k\to\infty} \inf \|\nabla f(x_k)\| = 0. \tag{11}$$

Dim. Poichè $f(x_{k+1}) \leq f(x_k)$ per ogni $k \geq 0$, si ha che la sequenza $\{x_k\}$ è contenuta nell'insieme compatto $\mathcal{L}(x_0)$. La Proposizione 1 e le istruzioni dell'algoritmo implicano che esiste un sottoinsieme $K_1 \subseteq \{0, 1, \ldots, \}$ tale che, $\forall k \in K_1$, α_k viene aggiornato al Passo 4. Si consideri ora la sottosequenza $\{x_k\}_{K_1}$: poichè $\{x_k\}_{K_1} \in \mathcal{L}(x_0)$, si può affermare che esiste un sottoinsieme $K_2 \subseteq K_1$ tale che

$$\lim_{k\to\infty, k\in K_2} x_k = \bar{x}. \tag{12}$$

D'altra parte, poichè per ogni $k \in K_2$ viene eseguito il Passo 4, deve necessariamente accadere che per $i = 1, \ldots, n$ e $k \in K_2$, si ha

$$f(x_k + \frac{\alpha_k}{\theta} e^i) \geq f(x_k), \tag{13}$$

$$f(x_k - \frac{\alpha_k}{\theta} e^i) \geq f(x_k). \tag{14}$$

Per $i \in \{1, \ldots, n\}$, applicando il Teorema della Media si può scrivere:

$$f(x_k + \frac{\alpha_k}{\theta} e^i) = f(x_k) + \frac{\alpha_k}{\theta} \nabla f(u_k^i)^T e^i, \tag{15}$$

$$f(x_k - \frac{\alpha_k}{\theta} e^i) = f(x_k) - \frac{\alpha_k}{\theta} \nabla f(v_k^i)^T e^i, \tag{16}$$

dove $u_k^i = x_k + \xi_k^i \frac{\alpha_k}{\theta} e^i$, con $\xi_k^i \in (0, 1)$, $v_k^i = x_k - \mu_k^i \frac{\alpha_k}{\theta} e^i$, con $\mu_k^i \in (0, 1)$.
Utilizzando la (1) e la (12), si ottiene per $i = 1, \ldots, n$

$$\lim_{k\to\infty, k\in K_2} u_k^i = \lim_{k\to\infty, k\in K_2} v_k^i = \bar{x}. \tag{17}$$

Sostituendo la (15) nella (13) e la (16) nella (14) segue

$$\frac{\alpha_k}{\theta} \nabla f(u_k^i)^T e^i \geq 0, \tag{18}$$

$$-\frac{\alpha_k}{\theta} \nabla f(v_k^i)^T e^i \geq 0. \tag{19}$$

Dalle (18) e (19), tenendo conto della (17) e della continuità di ∇f, si ha per $i = 1, \dots, n$

$$\lim_{k\to\infty, k\in K_2} \nabla f(u_k^i)^T e^i = \nabla f(\bar{x})^T e^i \geq 0, \tag{20}$$

$$\lim_{k\to\infty, k\in K_2} \nabla f(v_k^i)^T e^i = \nabla f(\bar{x})^T e^i \leq 0, \tag{21}$$

da cui

$$\nabla f(\bar{x})^T e^i = 0, \quad i = 1, \dots, n.$$

Poichè i vettori e^i sono linearmente indipendenti si può concludere che deve necessariamente risultare $\nabla f(\bar{x}) = 0$. $\square$

Un risultato di convergenza più forte può essere ottenuto modificando il precedente algoritmo nel seguente modo.

Algoritmo 2.

Dati. $x_0 \in R^n$, $\alpha_0 > 0$, $\theta \in (0, 1)$.

Passo 0. Poni $k = 0$.

Passo 1. Poni $i = 1$.

Passo 2a. Se $f(x_k + \alpha_k e^i) < f(x_k)$ vai al Passo 5;

Passo 2b. Se $f(x_k - \alpha_k e^i) < f(x_k)$ vai al Passo 5;

Passo 3. Se $i < n$ poni $i = i + 1$ e vai al Passo 2.

Passo 4. Poni $x_{k+1} = x_k$, $\alpha_{k+1} = \theta\alpha_k$ e vai al Passo 7.

Passo 5. Determina $j^+ = argmin_{i=1,\dots,n} f(x_k + \alpha_k e^i)$, $j^- = argmin_{i=1,\dots,n} f(x_k - \alpha_k e^i)$;
se $f(x_k + \alpha_k e^{j^+}) \leq f(x_k - \alpha_k e^{j^-})$ poni $x_{k+1} = x_k + \alpha_k e^{j^+}$,
altrimenti poni $x_{k+1} = x_k - \alpha_k e^{j^-}$;

Passo 6. Poni $\alpha_{k+1} = \alpha_k$.

Passo 7. Poni $k = k + 1$ e vai al Passo 1.

Osservazione 1. Sia nell'Algoritmo 1 che nell'Algoritmo 2 si ha che l'aggiornamento del punto corrente x_k ($x_{k+1} \neq x_k$) avviene qualora si sia individuato lungo una delle direzioni di ricerca un punto di stretta decrescita della funzione obiettivo. La differenza tra i due algoritmi consiste nella scelta del punto x_{k+1}: nell'Algoritmo 1 viene scelto il primo punto che determina una stretta decrescita della funzione obiettivo; nell'Algoritmo 2 occorre esaminare comunque i $2n$ *candidati* e viene scelto quello cui corrisponde il valore più piccolo della funzione obiettivo.

Le proprietà di convergenza dell'Algoritmo 2 sono riportate nel seguente teorema.

Teorema 2 *Sia $\{x_k\}$ la successione generata dall'Algoritmo 2 e si assuma che $\mathcal{L}(x_0)$ sia compatto; allora*

$$\lim_{k\to\infty} \|\nabla f(x_k)\| = 0. \tag{22}$$

Dim. Si veda [6]. □

3 Varianti dei metodi basati sulle direzioni coordinate

È immediato rendersi conto che, sia nell'Algoritmo 1 che nell'Algoritmo 2, alla generica iterazione occorre effettuare nel caso "peggiore" $2n$ valutazioni della funzione obiettivo. È possibile ridurre tale numero utilizzando, anzichè le direzioni positive e negative degli assi coordinati, le direzioni positive degli assi coordinati ed una opportuna direzione di ricerca che goda di determinate proprietà. Tale approccio è connesso al concetto di combinazione conica di vettori.

Dati i vettori $a^i \in R^n$, $i = 1, \dots, r$ e un insieme $S \subseteq R^n$, si dice che $S \subseteq cono\{a^1, \dots, a^r\}$ se ogni elemento di S è esprimibile come combinazione conica dei vettori $a^1, \dots, a^r$:

$$\forall x \in S, \exists c^i \geq 0, i = 1, \dots, r : x = \sum_{i=1}^{r} c^i a^i.$$

Proposizione 2 *Dati i vettori $a^i \in R^n$, $i = 1, \dots, r$, se $R^n = cono\{a^1, \dots, a^r\}$ allora*

$$r \geq n + 1.$$

Dim. Dati i vettori $a^i \in R^n$, $i = 1, \dots, r$, si supponga che $R^n = cono\{a^1, \dots, a^r\}$; ciò implica che a^1 è esprimibile come combinazione conica di $a^1, \dots, a^r$, cioè:

$$a^1 = \sum_{i=1}^{r} \bar{c}^i a^i, \quad \bar{c}^i \geq 0, \tag{23}$$

da cui segue

$$a^1 = \sum_{i=2}^{r} \tilde{c}^i a^i, \quad \tilde{c}^i = \frac{\bar{c}^i}{1 - \bar{c}^1}. \tag{24}$$

Un qualsiasi vettore $x \in R^n$ è esprimibile come combinazione conica dei vettori a^i, $i = 1, \dots, r$. Utilizzando la (24) si può perciò scrivere

$$x = \sum_{i=1}^{r} C^i a^i = \sum_{i=2}^{r} c^i a^i,$$

dove $c^i = C^1 \tilde{c}^i + C^i$, $i = 2, \dots, r$.

Si può così affermare che un qualsiasi vettore $x \in R^n$ è esprimibile come combinazione lineare di $a^2, \ldots, a^r$, ciò significa che

$$R^n = lin\{a^2, \ldots, a^r\}.$$

L'insieme $\{a^2, \ldots, a^r\}$ deve perciò contenere almeno una base di R^n, quindi deve essere di cardinalità maggiore o uguale a n, da cui segue $r \geq n+1$. $\Box$

Osservazione 2. Posto $d = -\sum_{i=1}^{n} \beta^i e^i$, con $\beta^i > 0$ per $i = 1, \ldots, n$, si può facilmente verificare che $R^n = cono\{e^1, \ldots, e^n, d\}$. $\Box$

Sia $R^n = cono\{e^1, \ldots, e^n, d\}$: come variante dell'Algoritmo 1 si può considerare lo schema seguente, in cui si può verificare che nel caso peggiore occorre effettuare $(n+1)$ valutazioni della funzione obiettivo.

Algoritmo 3.

Dati. $x_0 \in R^n$, $\alpha_0 > 0$, $\theta \in (0, 1)$.

Passo 0. Poni $k = 0$.

Passo 1. Poni $i = 1$.

Passo 2. Se $f(x_k + \alpha_k e^i) < f(x_k)$ vai al Passo 6;

Passo 3. Se $i < n$ poni $i = i + 1$ e vai al Passo 2.

Passo 4. Se $f(x_k + \alpha_k d) < f(x_k)$ vai al Passo 7;

Passo 5. Poni $x_{k+1} = x_k$, $\alpha_{k+1} = \theta\alpha_k$ e vai al Passo 9.

Passo 6. Poni $x_{k+1} = x_k + \alpha_k e^i$ e vai al Passo 8.

Passo 7. Poni $x_{k+1} = x_k + \alpha_k d$.

Passo 8. Poni $\alpha_{k+1} = \alpha_k$.

Passo 9. Poni $k = k + 1$ e vai al Passo 1.

Analogamente al caso dell'Algoritmo 1 vale il seguente risultato.

Teorema 3 *Sia $\{x_k\}$ la successione generata dall'Algoritmo 3 e si assuma che $\mathcal{L}(x_0)$ sia compatto; allora*

$$\lim_{k\to\infty} \inf \|\nabla f(x_k)\| = 0. \tag{25}$$

Dim. Per semplicità di esposizione si assumerà che

$$d = -\sum_{i=1}^{n} \beta^i e^i \quad \beta^i > 0 \quad i = 1, \dots, n. \tag{26}$$

In maniera analoga a quanto fatto nella prova della Proposizione 1 è possibile dimostrare che

$$\lim_{k\to\infty} \alpha_k = 0. \tag{27}$$

Ripetendo i ragionamenti utilizzati nella dimostrazione del Teorema 1, si ha che è possibile individuare un sottoinsieme $K_2 \subseteq \{0, 1, \dots\}$ tale che, per $i = 1, \dots, n$

$$\lim_{k\to\infty, k\in K_2} u_k^i = \lim_{k\to\infty, k\in K_2} v_k = \lim_{k\to\infty, k\in K_2} x_k = \bar{x}, \tag{28}$$

$$\nabla f(u_k^i)^T e^i \geq 0, \tag{29}$$

$$\nabla f(v_k)^T d \geq 0. \tag{30}$$

Tenendo conto della (28) e della continuità di ∇f, si ha per $i = 1, \dots, n$

$$\lim_{k\to\infty, k\in K_2} \nabla f(u_k^i)^T e^i = \nabla f(\bar{x})^T e^i \geq 0, \tag{31}$$

$$\lim_{k\to\infty, k\in K_2} \nabla f(v_k)^T d = \nabla f(\bar{x})^T d = -\beta^i \sum_{i=1}^{n} \nabla f(\bar{x})^T e^i \geq 0, \tag{32}$$

da cui

$$\nabla f(\bar{x})^T e^i = 0, \quad i = 1, \dots, n.$$

Poichè i vettori e^i sono linearmente indipendenti si può concludere che deve necessariamente risultare $\nabla f(\bar{x}) = 0$. □

In maniera analoga a quanto fatto nell'Algoritmo 3 si può ottenere una variante dell'Algoritmo 2 che presenta le stesse proprietà di convergenza di quest'ultimo (si veda [7]).

4 Appendice

Vengono riportate in questa appendice alcune proprietà topologiche dello spazio R^n. Gli insiemi cui si farà riferimento devono quindi intendersi come sottoinsiemi di R^n.

Definizione 1 *Una famiglia $G = \{G_1, G_2, \ldots, \}$ di insiemi si definisce copertura di un insieme S se S è contenuto nell' unione degli insiemi di G, cioè*

$$S \subseteq \bigcup_{G_i \in G} G_i.$$

Teorema 4 (Heine-Borel) *Se G è una famiglia di insiemi aperti copertura di un insieme compatto S, allora è possibile estrarre da G una famiglia costituita da un numero finito di insiemi che rappresenta una copertura di S.*

Proposizione 3 *Sia $I = \{x_1, x_2, \ldots, \}$ un insieme contenuto in un insieme compatto S e si assuma che*

$$\|x_i - x_j\| \geq \sigma > 0 \qquad \forall x_i, x_j \in I, x_i \neq x_j. \tag{33}$$

Allora, l'insieme I è di cardinalità finita.

Dim. Per ogni $x \in S$ si consideri la sfera aperta $B(x; \sigma/2)$ di centro x e raggio $\sigma/2$. La famiglia costituita da tali sfere rappresenta una copertura di S e per il Teorema di Heine-Borel è possibile quindi estrarre un numero finito r di sfere aperte tali che

$$S \subseteq \bigcup_{l=1,\ldots,r} B(\tilde{x}_l; \sigma/2). \tag{34}$$

Se due punti y, z appartengono ad una di tali sfere $B(\tilde{x}_l; \sigma/2)$ risulta

$$\|y - z\| = \|y - \tilde{x}_l + \tilde{x}_l - z\| \leq \|y - \tilde{x}_l\| + \|\tilde{x}_l - z\| < \sigma/2 + \sigma/2 = \sigma.$$

Quindi, utilizzando la (33) si ha che in nessuna delle sfere $B(\tilde{x}_l; \sigma/2)$ possono esistere due punti distinti x_i, x_j dell'insieme I. Dalla (34) segue che ogni punto dell'insieme I appartiene ad una sfera $B(\tilde{x}_l; \sigma/2)$ e poichè punti diversi di I appartengono a sfere diverse si ottiene che $|I| \leq r < +\infty$. $\square$

Riferimenti bibliografici

[1] A. R. Conn and P. L. Toint. An algorithm using quadratic interpolation for unconstrained derivative free optimization. In G. Di Pillo and F. Giannessi, editors, *Nonlinear Optimization and Applications*, Erice, 1995. Plenum.

[2] L. Grippo, F. Lampariello, and S. Lucidi. Global convergence and stabilization of unconstrained minimization methods without derivatives. *Journal of Optimization Theory and Applications*, 56(3):385–406, 1988.

[3] J. Nelder and R. Mead. A simplex method for function minimization. *Comput. J.*, 7:308–313, 1965.

[4] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. In K. A. Publishers, editor, *Advances in Optimization nad Numerical Analysis, Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, pages 51–67, Dordrecht, NL, 1994.

[5] V. Torczon. On the convergence of the multidirectional search algorithm. *SIAM J. Optimization*, 1:123–145, 1991.

[6] V. Torczon. On the convergence of pattern search algorithm. Technical Report 93/10, Department of Mathematical Sciences, Rice University, Houston, Texas, 1993. *Apparirà su SIAM J. Optimization.*

[7] R.M. Lewis and V. Torczon. Rank ordering and positive bases in pattern search algorithms. 1996.

[8] E. Polak. *Computational Methods in Optimization.* Academic Press, New York, 1971.

[9] W. Zangwill. Minimizing a function without calculating derivatives. *Comput. J.*, 10:293–296, 1967.