Algoritmi numerici pentru optimizare

II - Algoritmi determinişti de ordin zero

Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică,
Departamentul de Electrotehnică

Suport didactic pentru disciplina *Algoritmi Numerici*, 2017-2018

Notes

Cuprins

1. Introducere
2. Optimizare unidimensională (1D)
   - Metoda căutării simultane
   - Metoda căutării dihotomice
   - Metoda Fibonacci
   - Metoda secţiunii de aur
3. Optimizare multidimensională (nD)
   - Metoda simplexului descendent (Nelder-Mead)
   - Metoda Powell

Notes

Formularea problemei

Să se găsească n parametri independenţi, notaţi x_1^*, x_2^*, ..., x_n^*, pentru care expresia E este minimă, unde

$$E = f(x_1, x_2, \ldots, x_n), \tag{1}$$

şi $f : \Omega \to \mathbb{R}$, $\Omega \subset \mathbb{R}^n$, este dată.
Pe scurt:

$$(x_1^*, x_2^*, \ldots, x_n^*) = \arg\min f(x_1, x_2, \ldots, x_n). \tag{2}$$

Notaţii

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T \in \Omega. \tag{3}$$

$$\mathbf{x}_{\min} = [x_1^*, x_2^*, \ldots, x_n^*]^T \in \Omega \tag{4}$$

$$\mathbf{x}_{\min} = \arg\min f(\mathbf{x}), \quad \mathbf{x} \in \Omega$$

$$E_{\min} = f(\mathbf{x}_{\min}).$$

Notes

Formularea problemei

$$\mathbf{x}_{\min} = \arg\min f(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{5}$$

$$E_{\min} = f(\mathbf{x}_{\min}). \tag{6}$$

Optimizarea "scalară" - <u>un singur număr</u> înglobează criterii

- de proiectare (‖ performanţa cerută − cea obţinută ‖);
- de economie (preţul).

⇒ f este numită *funcţie obiectiv*, *funcţie de cost*, *funcţie de merit*, *criteriu de performanţă*.

Notes

Minime globale/locale

$$\mathbf{x}_{\min} = \arg\min f(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{7}$$

$$E_{\min} = \min\{\, f(\mathbf{x}) | \, \mathbf{x} \in \Omega\} \tag{8}$$

$\mathbf{x}_{\min}$ este *minim global* dacă

$$E_{\min} \leq f(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \tag{9}$$

- dacă $E_{\min} \leq f(\mathbf{x})$ doar într-o vecinătate a lui $\mathbf{x}_{\min}$ atunci minimul este *local*.
- în practică este dificil de stabilit dacă un minim găsit este local sau global;
- minimul global s-ar putea să nu fie unic.

Notes

Metode de optimizare

I. Deterministe - conduc la aceeaşi soluţie pentru rulări diferite ale programului, dacă pornesc din aceleaşi condiţii iniţiale şi au aceiaşi parametri.

- $-$: găsesc întotdeauna un minim local, dependent de iniţializare;
- $+$: efort de calcul mic.
  În problemele de optimizare din efortul de calcul se exprimă în număr de evaluări de funcţii obiectiv.

Pot fi

1. de ordin zero - necesită doar evaluări de funcţii obiectiv;
   Ex: metoda căutării simultane; metoda căutării dihotomice; metoda Fibonacci; metoda secţiunii de aur; metoda simplexului descendent (Nelder-Mead); metoda Powell, etc.
2. de ordin superior (1,2) - necesită şi evaluări ale derivatelor funcţiei obiectiv.

II. Stocastice

Notes

Metode de optimizare

$$(x_1^*, x_2^*, \ldots, x_n^*) = \arg\min f(x_1, x_2, \ldots, x_n). \qquad (10)$$

"Optimizare 1D": $n = 1$
$\rightarrow$ În contextul optimizării 1D: f se numeşte **unimodală** atunci când are un singur minim în domeniul ei de definiţie.

"Optimizare nD": $n > 1$

Metode

- *de căutare* = intervalul care conţine minimul este micşorat prin evaluarea lui f în anumite puncte;
- *de aproximare* = funcţia de optimizat este aproximată printr-o funcţie cunoscută care poate fi analizată uşor.

Notes

Metoda căutării simultane

Ideea:

- se împarte $[a, b]$ în n subintervale $x_i = a + ih$, $i = 0, \ldots, n$, $h = (b - a)/n$.
- se selectează valoarea minimă dintre $f(x_i)$.

Obs: h - suficient de mic

```
funcţie met_căutării_simultane(real a, b, funcţie f, întreg n)
h = (b − a)/n
minim = f(b)
pentru i = 0 : n − 1
        x = a + ih
        val = f(x)
        dacă (val < minim)
                minim = val;
întoarce minim
```

Complexitate[1] $T = O(n)$

[1] Operaţia de referinţă este evaluarea funcţiei f.

Notes

Metoda căutării simultane

Acurateţe (pp f unimodală)

Dacă x_k este punctul de minim obţinut $\Rightarrow x_{\min} \in [x_{k-1}, x_{k+1}]$.

Obs:

- $[x_{k-1}, x_{k+1}]$ - *interval de incertitudine*
  $x_{\min} \in [x_k - h, x_k + h] \Leftrightarrow$ "$x_{\min} = x_k \pm h$".
- Eroarea absolută
$$|x_{\min} - x_k| \leq h$$
- Eroarea relativă
$$\left|\frac{x_{\min} - x_k}{x_{\min}}\right| \leq \frac{h}{\min(|x_{k-1}|, |x_{k+1}|)}$$
  $\rightarrow$ dificultate dacă $x_{k-1} = 0$ sau $x_{k+1} = 0$

Notes

Metoda căutării simultane

Acurateţe (pp f unimodală)

Se preferă
Estimator al erorii relative = raportul dintre intervalul de incertitudine final şi cel iniţial

$$\mathrm{er}_{\mathrm{rel}} = \frac{2\mathrm{h}}{\mathrm{b} - \mathrm{a}} = \frac{2}{\mathrm{n}}$$

Pentru ca $\mathrm{er}_{\mathrm{rel}} \leq \varepsilon \Rightarrow \frac{2}{n} \leq \varepsilon \Rightarrow n = \left[\frac{2}{\varepsilon}\right] + 1$ evaluări.
Dacă $\varepsilon = 10^{-m} \Rightarrow n = 2 \cdot 10^m + 1$ (ex: $\varepsilon = 1\% \Rightarrow n = 201$).

Generalizare

Ideea metodei s-ar putea generaliza în spaţiul n-dimensional, dar efortul de calcul ar creşte enorm de mult.

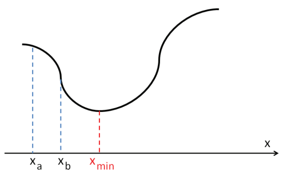
Notes

Metoda căutării dihotomice

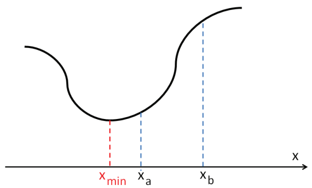
Pp. f unimodală.

Ideea căutării dihotomice[2]

- dacă x_a şi x_b sunt două puncte situate de aceeaşi parte a punctului de minim $x_{\min}$ atunci cel care se găseşte mai aproape de $x_{\min}$ furnizează o aproximaţie mai bună.



$x_a < x_b < x_{\min} \Rightarrow$
$f(x_b) < f(x_a)$



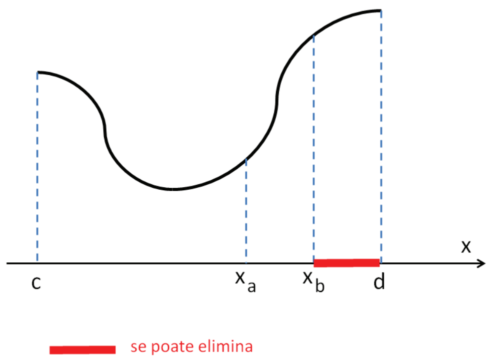
$x_{\min} < x_a < x_b \Rightarrow$
$f(x_a) < f(x_b)$

[2] Dihotomie = diviziune în două părţi; bifurcare; împărţirea unei noţiuni în alte două noţiuni care epuizează întreaga sferă a noţiunii împărţite.

Metoda căutării dihotomice

Ideea algoritmului

Fie $I = (c, d)$ intervalul de incertitudine şi $x_a < x_b$ două puncte în acest interval.



dacă $f(x_a) < f(x_b)$
atunci $I_{\text{nou}} = (c, x_b)$

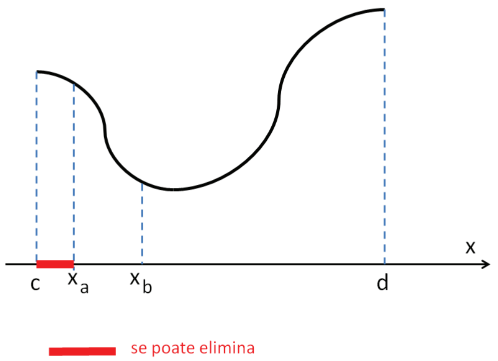
Notes

Notes

Metoda căutării dihotomice

Ideea algoritmului

Fie $I = (c, d)$ intervalul de incertitudine şi $x_a < x_b$ două puncte în acest interval.



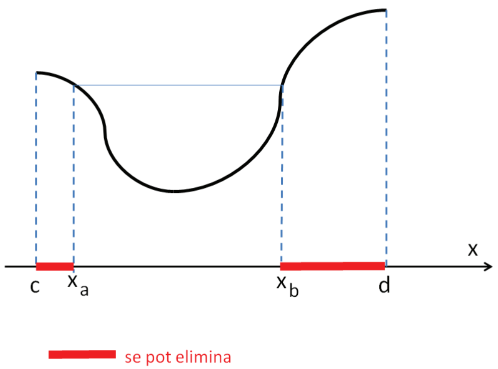
dacă $f(x_a) > f(x_b)$
atunci $I_{\text{nou}} = (x_a, d)$

Notes

Metoda căutării dihotomice

Ideea algoritmului

Fie $I = (c, d)$ intervalul de incertitudine şi $x_a < x_b$ două puncte în acest interval.



dacă $f(x_a) = f(x_b)$
atunci $I_{\text{nou}} = (x_a, x_b)$

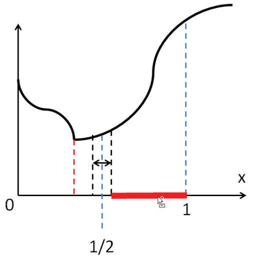
Notes

Metoda căutării dihotomice

Ipoteza de unimodalitate permite micşorarea succesivă a intervalului de incertitudine.

Algoritm: se înjumătăţeşte intervalul de incertitudine plasând perechi de puncte test foarte aproape una de cealaltă (Δx - distanţa dintre ele) în interiorul fiecărui interval.

Exemplu - dacă $I = [0, 1]$



După primul pas

$$I_{\text{nou}} = [0, \frac{1}{2} + \frac{\Delta x}{2})$$

Notes

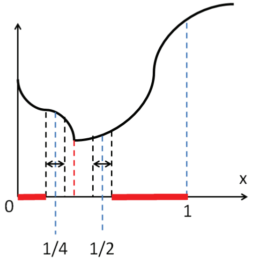
Introducere
Optimizare unidimensională (1D)
Optimizare multidimensională (nD)
Metoda căutării simultane
Metoda căutării dihotomice
Metoda Fibonacci
Metoda secţiunii de aur

Metoda căutării dihotomice

Ipoteza de unimodalitate permite micşorarea succesivă a intervalului de incertitudine.

Algoritm: se înjumătăţeşte intervalul de incertitudine plasând perechi de puncte test foarte aproape una de cealaltă (Δx - distanţa dintre ele) în interiorul fiecărui interval.

Exemplu - dacă $I = [0, 1]$



După al doilea pas

$$I_{\text{nou}} = [\frac{1}{4} - \frac{\Delta x}{2}, \frac{1}{2} + \frac{\Delta x}{2})$$

Notes

Metoda căutării dihotomice

Acurateţe

- m paşi: $2m$ evaluări şi un interval de incertitudine de $l/2^m$.
- După n evaluări intervalul de incertitudine este $l/2^{n/2}$.
- Eroarea relativă

$$\mathrm{er}_{\mathrm{rel}} = \frac{l/2^{n/2}}{l} = \frac{1}{2^{n/2}}$$

Pentru ca $\mathrm{er}_{\mathrm{rel}} \leq \varepsilon \Rightarrow \frac{1}{2^{n/2}} \leq \varepsilon \Rightarrow n = 2\log_2\left(\frac{1}{\varepsilon}\right) + 1$ evaluări.
Dacă $\varepsilon = 1\% \Rightarrow n = 14$, mai eficient decât la căutarea simultană.

Dezavantaje

- Δx nu poate fi mai mic decât zeroul maşinii
- Este ineficientă evaluarea funcţiei pentru valori atât de apropiate. Erorile de rotunjire ar putea duce la decizii greşite.
- Metoda nu se poate generaliza în cazul n-dimensional.

Notes

Metoda Fibonacci

Ideea: eliminarea intervalelor similar ca la metoda căutarii dihotomice, dar punctele intermediare x_a şi x_b nu sunt apropiate.
Reamintire: şirul lui Fibonacci

$$a_0 = a_1 = 1 \quad a_n = a_{n-1} + a_{n-2}, \quad n = 2, 3, \ldots \tag{11}$$

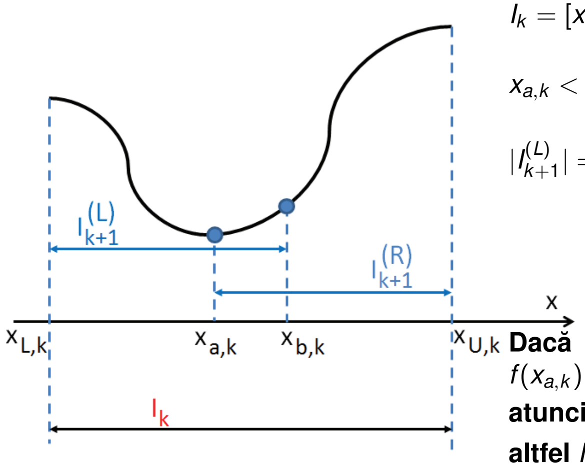
$1, 1, 2, 3, 5, 8, 13, 21, \ldots$
Obs:

$$\lim_{n\to\infty} \frac{a_{n-1}}{a_n} = \frac{-1+\sqrt{5}}{2} \approx 0.615 \tag{12}$$

Dem:
$a_n/a_{n-1} = 1 + a_{n-2}/a_{n-1} \Rightarrow 1/x = 1 + x \Rightarrow x^2 + x - 1 = 0$, etc.

Notes

Metoda Fibonacci



$I_k = [x_{L,k}, x_{U,k}]$

$x_{a,k} < x_{b,k}$

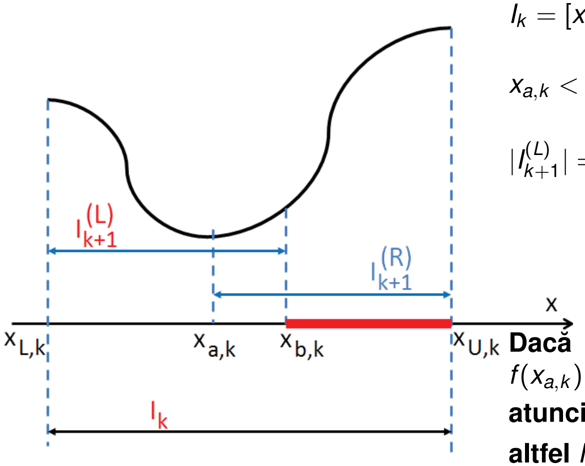
$|I_{k+1}^{(L)}| = |I_{k+1}^{(R)}|$

Dacă $f(x_{a,k}) < f(x_{b,k})$
atunci $I_{k+1} = I_{k+1}^{(L)}$
altfel $I_{k+1} = I_{k+1}^{(R)}$

Notes

Introducere
Optimizare unidimensională (1D)
Optimizare multidimensională (nD)
Metoda căutării simultane
Metoda căutării dihotomice
Metoda Fibonacci
Metoda secțiunii de aur

Metoda Fibonacci



$I_k = [x_{L,k}, x_{U,k}]$

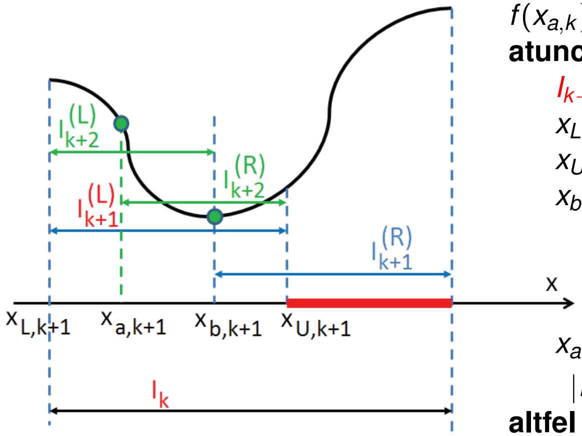
$x_{a,k} < x_{b,k}$

$|I_{k+1}^{(L)}| = |I_{k+1}^{(R)}|$

Dacă $f(x_{a,k}) < f(x_{b,k})$
atunci $I_{k+1} = I_{k+1}^{(L)}$
altfel $I_{k+1} = I_{k+1}^{(R)}$

Notes

Metoda Fibonacci



Dacă
$f(x_{a,k}) < f(x_{b,k})$
atunci

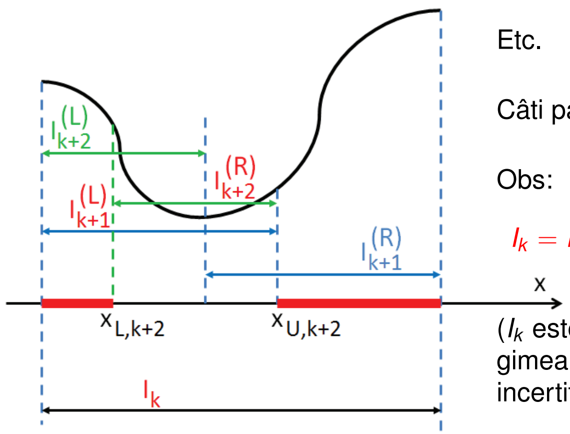
$$I_{k+1} = I_{k+1}^{(L)}$$
$$x_{L,k+1} = x_{L,k}$$
$$x_{U,k+1} = x_{b,k}$$
$$x_{b,k+1} = x_{a,k}$$

$x_{a,k+1}$ ales a.î
$$|I_{k+2}^{(L)}| = |I_{k+2}^{(R)}|$$
altfel
...

Notes

Introducere
Optimizare unidimensională (1D)
Optimizare multidimensională (nD)
Metoda căutării simultane
Metoda căutării dihotomice
Metoda Fibonacci
Metoda secţiunii de aur

Metoda Fibonacci



Etc.

Câţi paşi ?

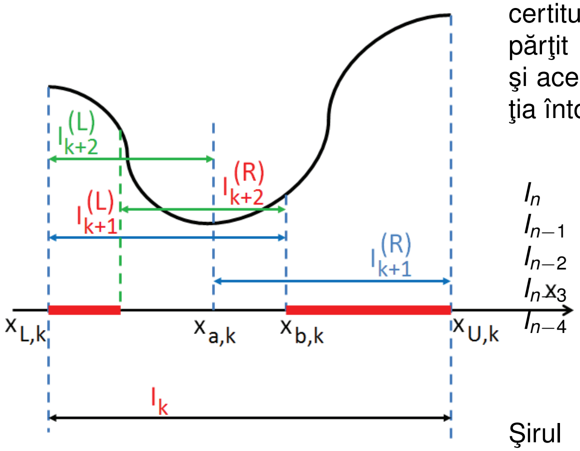
Obs:

$$I_k = I_{k+1} + I_{k+2} \quad (13)$$

(I_k este de acum lungimea intervalului de incertitudine).

Notes

Metoda Fibonacci



Ultimul interval de incertitudine va fi împărţit in 2 şi nu în 3 şi aceasta va fi soluţia întoarsă.

$$\begin{aligned} I_n &= I_{n+1} = 1 I_n \\ I_{n-1} &= I_n + I_{n+1} = 2 I_n \\ I_{n-2} &= I_{n-1} + I_n = 3 I_n \\ I_{n-3} &= I_{n-2} + I_{n-1} = 5 I_n \\ I_{n-4} &= I_{n-3} + I_{n-2} = 8 I_n \\ &\vdots \end{aligned} \tag{14}$$

Şirul Fibonacci: 1, 1, 2, 3, 5, 8, . . .

Notes

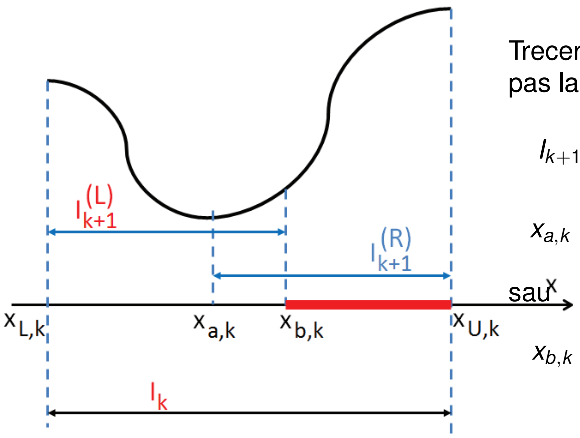
Metoda Fibonacci

$$\begin{aligned} I_{n-1} &= a_2 I_n \\ I_{n-2} &= a_3 I_n \\ I_{n-3} &= a_4 I_n \\ I_{n-4} &= a_5 I_n \\ &\vdots \\ I_k &= a_{n-k+1} I_n \\ &\vdots \\ I_1 &= a_n I_n \end{aligned} \tag{15}$$

n se stabileşte de la început!

Notes

Metoda Fibonacci



Trecerea de la un pas la altul:

$$I_{k+1} = \frac{a_{n-k}}{a_{n-k+1}} I_k \quad (16)$$

$$x_{a,k} = x_{U,k} - I_{k+1} \quad (17)$$

sau

$$x_{b,k} = x_{L,k} + I_{k+1} \quad (18)$$

Notes

Metoda Fibonacci

funcţie [**real** *x_min*,tol_x,*f_min*] = met_fibonacci(**real** *la*, *lb*, **funcţie** *f*, **întreg** *n*)
a = numere_fibonacci(n) ; calculează numerele Fibonacci între 1 şi n
$xL_1 = la$; capătul din stânga al intervalului iniţial
$xU_1 = lb$; capătul din dreapta al intervalului iniţial
$I_1 = lb - la$; lungimea intervalului iniţial
$I_2 = (a_{n-1}/a_n)I_1$
$xa_1 = xU_1 - I_2$
$xb_1 = xL_1 + I_2$
$fa = f(xa_1)$
$fb = f(xb_1)$
pentru $k = 2, n-1$
    $I_{k+1} = (a_{n-k}/a_{n-k+1})I_k$
    dacă ($fa <= fb$) **atunci**
        $xL_k = xL_{k-1}$
        $xU_k = xb_{k-1}$
        $xa_k = xU_k - I_{k+1}$
        $xb_k = xa_{k-1}$
        $fb = fa$
        $fa = f(xa_k)$
    altfel
        $xL_k = xa_{k-1}$
        $xU_k = xU_{k-1}$
        $xb_k = xL_k + I_{k+1}$
        $xa_k = xb_{k-1}$
        $fa = fb$
        $fb = f(xb_k)$
*x*_min = $(xL_k + xU_k)/2$
tol_x= I_{n-1}
*f*_min = *f*(_min)

Notes

Metoda Fibonacci

Acurateţe şi efort de calcul

Eroarea relativă

$$\frac{l_n}{l_1} = \frac{1}{a_n} \tag{19}$$

se obţine cu un efort de $n+1$ evaluări de funcţii.

a_n: $1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, \ldots$

$\frac{l_{11}}{l_1} = \frac{1}{144}$ după 12 evaluări

Căutare simultană: 21 de evaluări pentru a scădea intervalul de incertitudine de 10 ori.

Fibonacci: $\frac{l_6}{l_1} = \frac{1}{13} \Rightarrow$ 7 evaluări.

Notes

Metoda Fibonacci

Numărul de evaluări de funcţii trebuie impus ca parametru de intrare.

- Dacă se cere ca valoarea funcţiei să scadă de un anumit număr de ori faţă de valoarea iniţială, atunci numărul de evaluări nu poate fi estimat.
- Dacă se cere ca intervalul de incertitudine să scadă de un anumit număr de ori faţă de valoarea iniţială, atunci se poate determina cea mai mică valoare N pentru care $1/a_N < \varepsilon$, şi folosit acest N ca parametru pentru procedura Fibonacci.
- Se poate demonstra că dintre metodele care cer un număr fixat de evaluări de funcţii, metoda Fibonacci realizează cea mai mare micşorare a intervalului de incertitudine.

Metoda nu se poate generaliza în cazul multidimensional

Notes

Metoda secţiunii de aur

Fibonacci: primele două puncte x_a şi x_b depind de numărul de evaluări de funcţii. Odată specificat acest număr şi pusă condiţia ca la sfârşit $x_a = x_b$ rezultă lungimea intervalelor.

Metoda secţiunii de aur - presupune tot că

$$I_k = I_{k+1} + I_{k+2}, \qquad (20)$$

dar, impune ca

$$\frac{I_k}{I_{k+1}} = \frac{I_{k+1}}{I_{k+2}} = \varphi \qquad (21)$$

Notes

Metoda secţiunii de aur

$$I_k = I_{k+1} + I_{k+2}$$

$$\frac{I_k}{I_{k+2}} = \frac{I_{k+1}}{I_{k+2}} + 1,$$

$$\varphi^2 = \varphi + 1,$$

$$\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$$

Secţiunea de aur = împărţirea unui interval în două subintervale a.î. raportul dintre intervalul întreg şi subintervalul mai mare este egal cu raportul dintre subintervalul mai mare şi cel mai mic.
Vedeţi şi https://en.wikipedia.org/wiki/Golden_ratio

Notes

Metoda secţiunii de aur

Algoritm - seamană cu cel al metodei Fibonacci, se fac următoarele modificări:

- a_{n-1}/a_n se înlocuieşte cu φ
- a_{n-k}/a_{n-k+1} se înlocuieşte cu φ
- în loc de ciclu cu contor se foloseşte ciclu cu test
- criteriul de oprire este mai natural - bazat pe reducerea lungimii intervalului de incertitudine de un număr impus de ori.

Temă - scrieţi pseudocodul algoritmului, implementaţi-l, testaţi-l şi comparaţi rezultatele cu cele obţinute cu metoda Fibonacci.

Notes

Metoda secţiunii de aur

Comparaţie cu metoda Fibonacci

Se poate arăta că

$$a_n \approx \frac{\varphi^{n+1}}{\sqrt{5}} \tag{22}$$

Fibonacci:

$$\varepsilon_F = \frac{l_n}{l_1} = \frac{1}{a_n} = \frac{\sqrt{5}}{\varphi^{n+1}} \tag{23}$$

Secţiunea de aur (pentru acelaşi număr de evaluări):

$$\varepsilon_g = \frac{l_n}{l_1} = \frac{1}{\varphi^{n-1}} \tag{24}$$

Notes

Metoda secţiunii de aur

Comparaţie cu metoda Fibonacci

Rezultă că

$$\frac{\varepsilon_g}{\varepsilon_F} = \frac{\varphi^2}{\sqrt{5}} \approx 1.17 \tag{25}$$

Pentru acelaşi număr de evaluări de funcţii, reducerea intervalului la Fibonacci este mai mic cu 17% decât la secţiunea de aur.

Acest avantaj se obţine însă pe baza specificării în avans a numărului de evaluări de funcţii.

Notes

Metoda simplexului descendent (Nelder-Mead)

Formularea problemei:

$$(x_1^*, x_2^*, \ldots, x_n^*) = \arg\min f(x_1, x_2, \ldots, x_n). \tag{26}$$

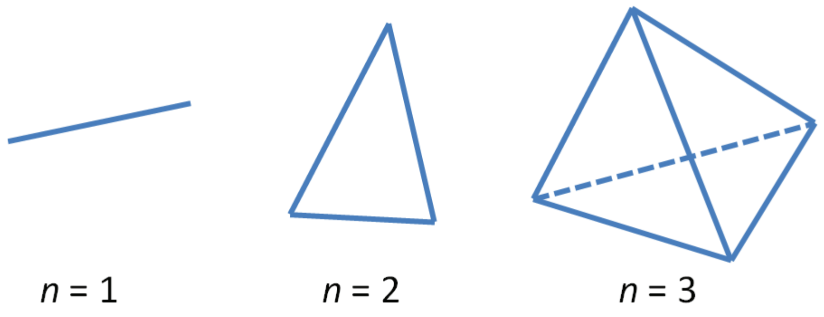
$n > 1$
Nelder şi Mead (1965) - metoda simplexului descendent:

- E simplă conceptual şi are o naturaleţe geometrică;
- Nu are nevoie de un algoritm de minimizare 1D;
- Nu este foarte eficientă dpdv al efortului de calcul;
- E frecvent utilizată dacă efortul evaluării funcţiei este mic;
- Nu trebuie confundată cu metoda simplex a programării liniare;

Notes

Metoda simplexului descendent (Nelder-Mead)

Simplex = poliedru cu $n+1$ vârfuri



- nu este neaparat regulat;
- ne interesează cele nedegenerate (cu măsura nenulă).

Notes

Metoda simplexului descendent (Nelder-Mead)

Ideea: simplexul se "mişcă" în spaţiul n-dimensional, până când se întâlneşte un minim.

Iniţializarea: este nevoie de $n+1$ puncte de start.

Mişcările:

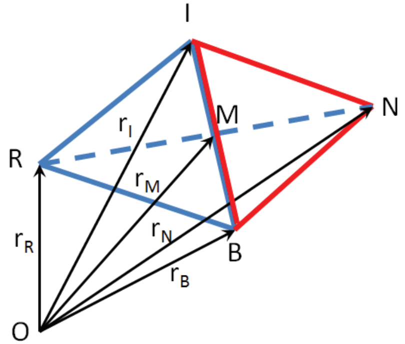
- reflectare
- expansiune
- contracţie parţială
- contracţie totală

mută (în general) vârful căreia îi corespunde cea mai proastă valoare a funcţiei obiectiv.

Notes

Metoda simplexului descendent (Nelder-Mead)

Reflectarea - mişcarea punctului cel mai prost se face prin simetrie faţă de centrul feţei opuse a.î măsura simplexului să se conserve.



Simplex IBR, unde
$f(B) < f(I) < f(R)$
(B = bun, I = intermediar, R = rău)
M - mijlocul segmentului IB:
$\mathbf{r}_M = (\mathbf{r}_I + \mathbf{r}_B)/2$
$\|\mathrm{MN}\| = \|\mathrm{RM}\|$

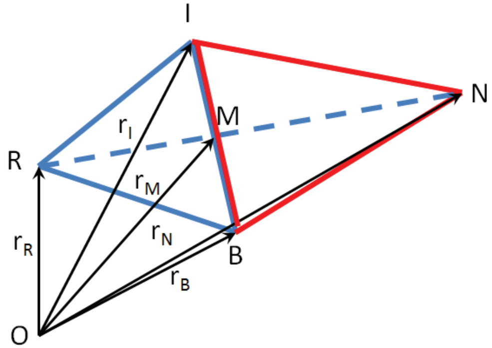
$$\mathbf{r}_N = \mathbf{r}_M + \mathbf{MN} = \mathbf{r}_M + \|\mathrm{MN}\| \frac{\mathbf{RM}}{\|\mathrm{RM}\|} = \mathbf{r}_M + \mathbf{r}_M - \mathbf{r}_R = 2\mathbf{r}_M - \mathbf{r}_R. \quad (27)$$

Dacă $f(N) < f(R)$ **atunci** reflectarea este reuşită, noul simplex este

Notes

Metoda simplexului descendent (Nelder-Mead)

Expansiunea - e o mişcare făcută pentru a accelera căutarea.



Simplex IBR, unde
$f(B) < f(I) < f(R)$
$\mathbf{r}_M = (\mathbf{r}_I + \mathbf{r}_B)/2$
$\|\mathrm{MN}\| > \|\mathrm{RM}\|$

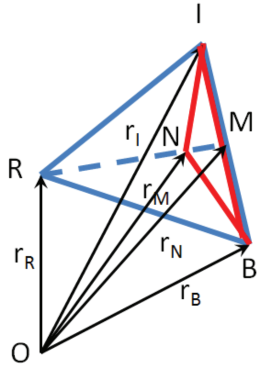
$$\mathbf{r}_N = \mathbf{r}_M + \mathbf{MN} = \mathbf{r}_M + \|\mathrm{MN}\| \frac{\mathbf{RM}}{\|\mathrm{RM}\|} = \mathbf{r}_M + g(\mathbf{r}_M - \mathbf{r}_R). \quad (28)$$

$g > 1$. Uzual $g = 2$. După expansiune măsura simplexului creşte.
"Expansiunea" cu $g < 1$ se numeşte tot reflectare.

Notes

Metoda simplexului descendent (Nelder-Mead)

Contractarea parţială - se face atunci când simplexul ajunge într-o vale şi el încearcă să pătrundă mai adânc în vale.



Simplex IBR
$f(B) < f(I) < f(R)$
$\mathbf{r}_M = (\mathbf{r}_I + \mathbf{r}_B)/2$

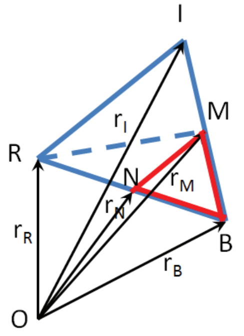
$$\mathbf{r}_N = \mathbf{r}_M + \mathbf{MN} = \mathbf{r}_M + b\mathbf{MR} = \mathbf{r}_M + b(\mathbf{r}_R - \mathbf{r}_M). \quad (29)$$

$b \in (0, 1)$, în mod uzual $b = 0.5$

Notes

Metoda simplexului descendent (Nelder-Mead)

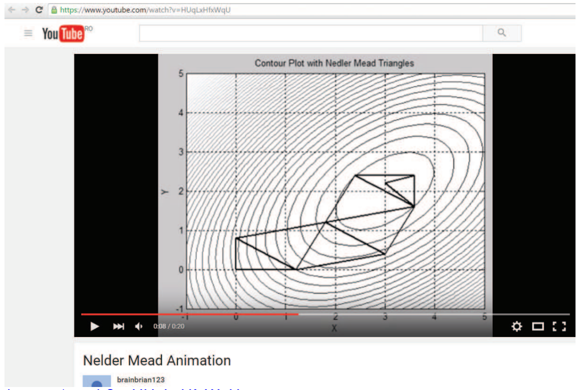
Contractarea totală - se face atunci când simplexul încearcă să treacă printr-un relief ca o ureche de ac de cusut şi atunci se contractă în toate direcţiile, orientându-se după punctul cel mai bun.



Simplex IBR
$f(B) < f(I) < f(R)$
$\mathbf{r}_M = (\mathbf{r}_I + \mathbf{r}_B)/2$
$\mathbf{r}_N = (\mathbf{r}_R + \mathbf{r}_B)/2$

Notes

Metoda simplexului descendent (Nelder-Mead)



https://www.youtube.com/watch?v=HUqLxHfxWqU

Notes

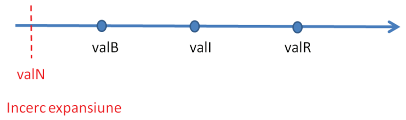
Metoda simplexului descendent (Nelder-Mead)

Algoritm - se bazează pe reflectare şi în funcţie de rezultat se încearcă alte mişcari

```
; inițalizare simplex
P1 = ...
P2 = ...
P3 = ...
g = 2   ; factor de expansiune
b = 0.5; factor de contratare parţială
; sortare [B, I, R, valB, valI, valR] = sortează(P1,P2,P3)
repetă
        ; încearcă reflectare
        M = (B + I)/2
        N = 2 M - R
        valN = f(N)
```


Notes

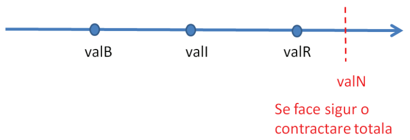
Metoda simplexului descendent (Nelder-Mead)




```
; cazul 1
dacă valN < valB ; sunt în direcţia bună, încerc accelerare
        ; încerc expansiune
        Nexp = M + g(M-R)
        valNexp = f(Nexp)
        dacă valNexp < valN
                ; expansiune reuşită
                R = I
                I = B
                B = Nexp
                valR = valI
                valI = valB
                valB = valNexp
        altfel
        ; reflectare reuşită
                R = I
                I = B
                B = N
                valR = valI
                valI = valB
                valB = valN
```


Notes

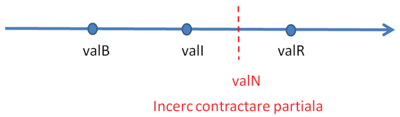
Metoda simplexului descendent (Nelder-Mead)




```
; cazul 2
dacă valN > valR ; sunt într-o direcţie total greşită
        ; se face contracţie totală păstrându-l pe cel mai bun
        N = (B+R)/2
        [B, I, R, valB, valI, valR] = sortează(B,M,N)
```


Notes

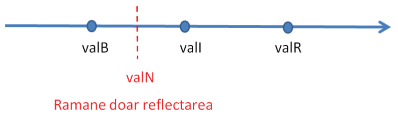
Metoda simplexului descendent (Nelder-Mead)



; cazul 3
dacă valN ∈ (valR,valI) ;
    ; încerc contractare parţială
    Nc = M + b(R-M)
    valNc = f(Nc)
    dacă valNc < valN
        ; contractare parţială reuşită
        [B, I, R, valB, valI, valR] = sortează(B,I,Nc)
    altfel
        ; reflectare reuşită, dar acelaşi B
        R = N
        valR = valN

Notes

Metoda simplexului descendent (Nelder-Mead)



; cazul 4
dacă valN ∈ (valI,valB) ;
    ; reflectare reuşită, acelaşi B
    R = I
    I = N
    valR = valI
    valI = valN
până când (criteriu)

Notes

Metoda simplexului descendent (Nelder-Mead)

Criteriul de oprire

- Este delicat în orice optimizare multidimensională deoarece nu există posibilitatea de a aplica tehnici de încadrare, deci nu se poate cere o toleranţă pentru fiecare variabilă independentă;
- Exemple de criterii de oprire:
  1. $$\|\mathrm{valBnou} - \mathrm{valB}\| \leq \varepsilon \|\mathrm{valB}\| + \mathrm{eps} \tag{30}$$
  2. $$\|\mathrm{Bnou} - \mathrm{B}\| \leq \varepsilon \|\mathrm{B}\| + \mathrm{eps} \tag{31}$$
  3. Numărul de evaluări de funcţii $\geq$ o valoare impusă.
- Oricare din criterii poate conduce la o soluţie proastă, de aceea se recomandă restartarea algoritmilor din punctul în care se pretinde că s-a găsit minimul.

Notes

Metoda Powell

Formularea problemei:

$$(x_1^*, x_2^*, \ldots, x_n^*) = \arg\min f(x_1, x_2, \ldots, x_n). \tag{32}$$

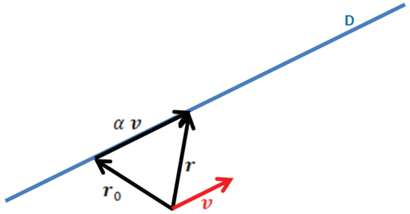
$n > 1$

- Metoda Powell are nevoie de un algoritm de minimizare 1D ca parte a strategiei de calcul.

Notes

Metoda Powell

Reamintire: ecuaţia vectorială a unei drepte în spaţiul nD.



Dreapta care conţine punctul $\mathbf{r}_0$ şi are direcţia $\mathbf{v}$

$$\mathbf{r} = \mathbf{r}_0 + \alpha \mathbf{v} \qquad (33)$$

- $\mathbf{r}$ - vector de poziţie;
- $\mathbf{r}_0$ - vector de poziţie al unui punct fix pe dreaptă;
- α - coordonata de-a lungul dreptei;
- $\mathbf{v}$ - vector (fix) ce orientează dreapta.

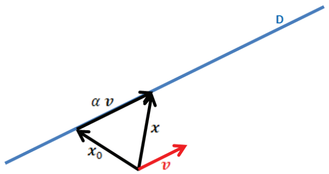
Notes

Introducere
Optimizare unidimensională (1D)
Optimizare multidimensională (nD)
Metoda simplexului descendent (Nelder-Mead)
Metoda Powell

Metoda Powell

Minimizarea după o direcţie a unei funcţii de mai multe variabile

$f : \mathbb{R}^n \to \mathbb{R} \qquad f(\mathbf{x}) = f(x_1, x_2, \ldots, x_n)$



$$f(\mathbf{x})|_{\mathbf{x} \in D} = f(\mathbf{x}_0 + \alpha \mathbf{v})$$

Se defineşte $t : \mathbb{R} \to \mathbb{R}$

$$t(\alpha) = f(\mathbf{x}_0 + \alpha \mathbf{v})$$

unde $\mathbf{x}_0$ şi $\mathbf{v}$ sunt date.

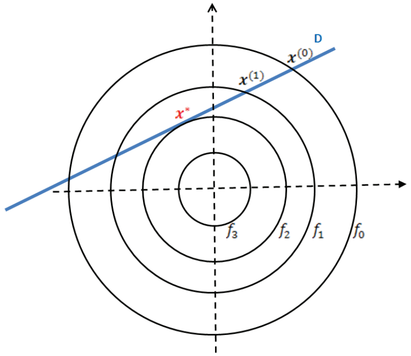
Problema minimizării nD funcţiei f după direcţia $\mathbf{v}$ se reduce la problema minimizării 1D a funcţiei t.

Notes

Metoda Powell

Minimizarea după o direcţie a unei funcţii de mai multe variabile

Semnificaţie geometrică:



$f : \mathbb{R}^2 \to \mathbb{R}$
$f(x_1, x_2) = x_1^2 + x_2^2$
$f_3 < f_2 < f_1 < f_0$

- Minimul dupa direcţia D se află pe cercul cu centrul în origine, tangent la D.
- Minimizarea nu se face exact.

procedură [**x**,fmin] = linmin(**x**,**v**)
[α,tol,fmin] = secţiunea_aur(...,f1D,...)
$\mathbf{x} = \mathbf{x} + \alpha\mathbf{v}$
retur

funcţie [t] = f1D(α)
$t = f(\mathbf{x} + \alpha\mathbf{v})$
întoarce t

Notes

Introducere
Optimizare unidimensională (1D)
Optimizare multidimensională (nD)
Metoda simplexului descendent (Nelder-Mead)
Metoda Powell

Metoda Powell

Ideea: căutări succesive pe n direcţii liniar independente $\mathbf{v}^{(i)}$, $i = 1, n$.

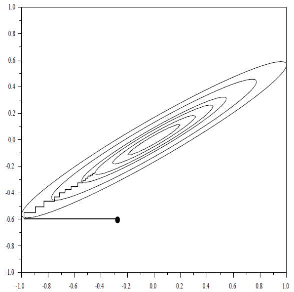
- Iniţializare $\mathbf{x}^{(0)}$
- Direcţia $\mathbf{v}^{(1)} \Rightarrow D_1$: $\mathbf{x} = \mathbf{x}^{(0)} + \alpha\mathbf{v}^{(1)} \Rightarrow \alpha_1$ prin minimizare;
  Minimul după acestă direcţie: $\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_1\mathbf{v}^{(1)}$
- Direcţia $\mathbf{v}^{(2)} \Rightarrow D_2$: $\mathbf{x} = \mathbf{x}^{(1)} + \alpha\mathbf{v}^{(2)} \Rightarrow \alpha_2$
  Minimul: $\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \alpha_2\mathbf{v}^{(2)}$
- În general $\mathbf{x}^{(i)} = \mathbf{x}^{(i-1)} + \alpha_i\mathbf{v}^{(i)}$ unde α_i se determină prin minimizare după direcţia $\mathbf{v}^{(i)}$, $i = 1, n$.

Dacă $|f(x_n) - f(x_0)|$ e mare, atunci se reia căutarea cu o nouă iniţializare, e.g. $\mathbf{x}^{(0)}_{\text{nou}} = \mathbf{x}^{(n)}$.
"Iteraţie" în metoda Powell = calculul unui minim aproximativ pornind dintr-o iniţializare şi făcând n minimizări consecutive după direcţiile $\mathbf{v}^{(i)}$.

Notes

Metoda Powell

Ideea: căutări succesive pe n direcţii .
Căutarea după direcţiile axelor s-ar putea să ducă la o convergenţă foarte lentă.



Ex - vale îngustă, iar axa văii nu coincide cu nicio direcţie de căutare.
$\Rightarrow$
Trebuie ca setul de direcţii de căutare să fie adaptat funcţiei, a.î. avansul către minim să fie cât mai rapid.

Notes

Metoda Powell

Setul de direcţii se ajustează după fiecare iteraţie Powell, pe baza informaţiilor obţinute la acea iteraţie.
Ideea:

1. se elimină o direcţie din cele n;
2. se adaugă direcţia $\mathbf{v}^{(m)}$ a deplasării medii

$$\mathbf{v}^{(m)} = \mathbf{x}^{(n)} - \mathbf{x}^{(0)} \quad (34)$$

3. noua iniţializare se determină făcând încă o minimizare după direcţia $\mathbf{v}^{(m)}$

$$\mathbf{x}_{\mathrm{nou}}^{(0)} = \mathbf{x}^{(n)} + \alpha_{n+1}\mathbf{v}^{(m)} \quad (35)$$

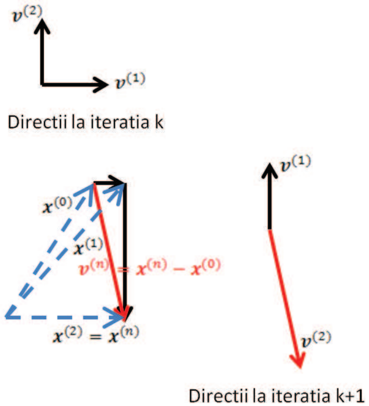
Elementul cheie este alegerea direcţiei eliminate.
Există două tehnici posibile.

Notes

Metoda Powell

Metoda Powell de bază:

- se elimină întotdeauna $\mathbf{v}^{(1)}$;
- se renumerotează direcţiile ($\mathbf{v}^{(2)}$ devine $\mathbf{v}^{(1)}_{\mathrm{nou}}$, etc.);
- se adaugă $\mathbf{v}^{(m)}$ ca ultimă direcţie.



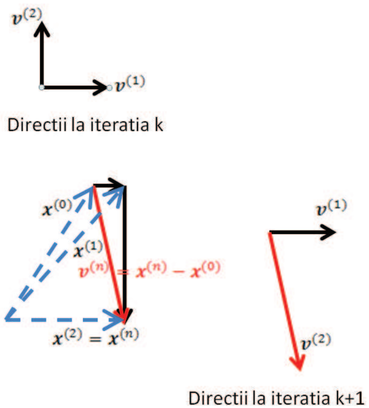
Poate da rezultate proaste dacă direcţiile adăugate tind să devină aproape coliniare cu unele din direcţiile existente.

Notes

Metoda Powell

Metoda Powell modificată:

- se elimină direcţia cea mai bună $\mathbf{v}^{(j)}$ (după care funcţia a avut cea mai mare scădere la iteraţia curentă);
- se pune $\mathbf{v}^{(n)}$ în locul lui $\mathbf{v}^{(j)}$;
- se adaua $\mathbf{v}^{(m)}$ ca ultimă direcţie.



Notes

Metoda Powell modificată

Date: iniţializarea $\mathbf{x}_0$, direcţiile iniţiale $\mathbf{v}_i$, $i = 1, \ldots, n$

```
f0 = f(x0)
repetă
    dfmax = 0                       ; cea mai mare scădere a funcţiei
    j = 0                           ; direcţia cu cea mai mare scădere
    xv = x0,  fv = f0
    pentru i = 1, n                             ; parcurge direcţiile
        [xn, fn] = linmin(xv, v_i)              ; minimizează pe directia v_i
        dacă | fn - fv | > dfmax atunci         ; am găsit o scădere mai mare
            dfmax = | fn - fv |
            j = i
        xv = xn                                 ; actualizează x şi f(x)
        fv = fn
    vm = xn - x0                    ; direcţia medie
    deltaf = | fn - f0 |            ; variaţia funcţiei la iteraţia curentă
    v_j = v_n                       ; pune direcţia v_n pe poziţia j
    v_n = vm                        ; pune direcţia v_m pe ultima poziţie
    [x0,f0] = linmin(xn, vm)        ; determină noua iniţializare
până când deltaf < eps
```

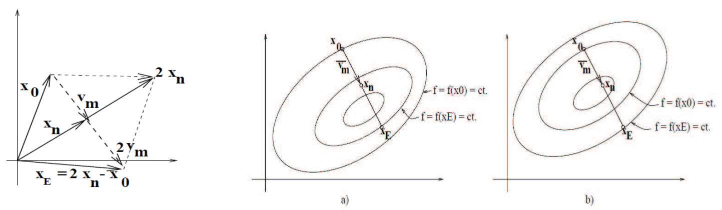
Condiţia de oprire ar putea fi de tip relativ:

$$|f(\mathbf{x}^{(n)}) - f(\mathbf{x}^{(0)})| \leq \mathrm{ftol} \cdot ((f(\mathbf{x}^{(n)}) + f(\mathbf{x}^{(0)}))/2 + \mathrm{eps})$$

Notes

Metoda Powell

Îmbunătăţiri - uneori este mai bine să nu se modifice direcţiile de căutare.



$$\mathbf{x}_E = \mathbf{x}_n + \mathbf{v}_m = \\ = 2\mathbf{x}_n - \mathbf{x}_0$$

b): $f(\mathbf{x}_E) >= f(\mathbf{x}_0) \Rightarrow$ nu se modifică direcţiile.

Notes

Metoda Powell

Îmbunătăţiri - uneori este mai bine să nu se modifice direcţiile de căutare.

Dacă la iteraţia curentă scăderea nu se datorează cu precădere unei anumite direcţii, adică dacă

$$f_0 - f_n \gg \Delta f_{\max}$$

unde $\Delta f_{\max}$ este scăderea cea mai mare (după direcţia j)
$\Rightarrow$ nu se modifică direcţiile.

Notes

Metoda Powell

Ordin de complexitate

- Se dem. că metoda Powell de bază aplicată unei funcţii pătratice de n variabile conduce la minim dupa n iteraţii $\Rightarrow$ $n(n+1)$ minimizări 1D $\Rightarrow$ $O(n^2)$ dacă se consideră ca operaţie elementară minimizarea unei funcţii 1D
- Dacă funcţia nu e pătratică sunt necesare mai multe iteraţii.
- În metoda modificată se pierde ordinul de complexitate pătratic, dar algoritmul devine mai robust.

Notes

Referinţe

- [Ciuprina02] G.Ciuprina, D.Ioan, I.Munteanu, M.Rebican, R.Popa, Optimizarea numerica a dispozitivelor electromagnetice, Editura Printech, 2002.

  disponibilă la http://www.lmn.pub.ro/∼gabriela/books/opt2002.pdf

- [Cheney08] Ward Cheney and David Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole publishing Company,2008. (Capitolul 16 - Minimization of functions)
- [Press02] W.H.Press, S.A.Teukolsky, W.T. Wetterling, B.P. Flannery, *Numerical Recipies in C*, 2002. (Capitolul 10)

  Disponibilă la https://www2.units.it/ipl/students_area/imm2/files/Numerical_Recipes.pdf

Notes

Notes