Capitolul 5

METODA PASULUI DESCENDENT

În acest capitol vom prezenta metoda clasică de optimizare fără restricții cunoscută ca metoda pasului descendent a lui Cauchy. Aceasta este metoda fundamentală de minimizare fără restricții. Virtual, orice metodă de minimizare a funcțiilor suficient de netede îşi are originea în metoda pasului descendent. Metoda are caracteristici care sunt de dorit în cadrul algoritmilor de optimizare, şi în acelaşi timp este tarată de caracteristici care trebuie evitate în situațiile practice. De aceea prezentarea şi studiul acestei metode constituie o cale ideală de ilustrare a metodelor moderne de minimizare fără restricții.

Metoda pasului descendent, sau încă a gradientului descendent a fost proiectată de Cauchy, fiind una dintre multele contribuții ale acestuia în domeniul matematicii. Esența metodei constă în utilizarea unui model liniar al funcției de minimizat, model dat de dezvoltarea Taylor de prim ordin. Pentru cele mai multe situații practice, metoda nu este o schemă computațională de utilizat. Convergența ei este liniară şi câteodată mizerabil de lentă.

Metoda pasului descendent prezentată în acest capitol este o metodă iterativă de primul ordin. Într-adevăr, estimația soluției de la o iterație se calculează pe baza estimației soluției de la iterația anterioară. Întotdeauna valoarea unei scheme computaționale iterative de rezolvare a unei probleme de minimizare, şi în general a unei probleme oarecare, este măsurată în termenii efortului de calcul pentru obținerea şirului de estimații ale soluției, rata de convergență a acestui şir, şi complexitatea calculului necesar obținerii soluției.

Scopul capitolului este de a prezenta cu detalii şi riguros matematic metoda pasului descendent a lui Cauchy împreună cu rezultatele teoretice fundamentale privind convergența şi complexitatea computațională a acestei metode.

Capitolul se structurează astfel. În prima parte prezentăm algoritmul pasului descendent, după care prezentăm metoda pasului descendent pentru funcții pătratice. Importanța funcțiilor pătratice rezidă în faptul că pentru acestea lungimea pasului de deplasare se poate calcula analitic şi, ca atare, se pot prezenta rezultate de complexitate. În continuare se prezintă cu detalii şi riguros matematic teoria convergenței metodei pasului descendent, în care se arată că metoda este liniar convergentă. În finalul acestui capitol prezentăm o serie de variante ale acestei metode, care chiar dacă păstrează caracterul convergenței liniare, totuşi îmbunătățesc procesul de căutare a optimului.

Să considerăm problema

$$min\, f(x), \tag{5.0.1}$$

unde $f : R^n \to R$ este o funcție de două ori continuu diferențiabilă pe R^n cu domeniul $dom\, f$. Valoarea optimă a acestei probleme o notăm cu f^*, adică $\inf_x f(x) = f^*$. Deoarece f este diferențiabilă, atunci o condiție necesară pentru ca punctul x^* să fie soluția optimă a lui (5.0.1) este:

$$\nabla f(x^*) = 0\,. \tag{5.0.2}$$

Condiția (5.0.2) se numește *condiția necesară de optimalitate de primul ordin*. Punctele care satisfac (5.0.2) se numesc *puncte staționare* sau *critice.* Ca atare, rezolvarea problemei (5.0.1) a fost redusă la determinarea unei soluții pentru (5.0.2), care, după cum se vede, este un sistem algebric de n ecuații în n variabile. De obicei pentru rezolvarea problemei de optimizare (5.0.1) vom folosi o metodă iterativă, care generează un şir de puncte $x_0, x_1, \cdots, x_k, \cdots \in dom\, f$ cu proprietatea că $f(x^k) \to f^*$ când $k \to \infty$. Un astfel de şir cu această proprietate se numeşte *şir minimizant* pentru (5.0.1). Astfel conceput, un algoritm care generează un şir minimizant se opreşte de îndată ce un anumit criteriu de oprire a iterațiilor este îndeplinit. În analiza care urmează vom folosi criteriul referitor la apropierea valorii funcției de minimizat față de valoarea sa optimă:

$$\left| f(x_k) - f^* \right| \le \varepsilon\,, \tag{5.0.3}$$

unde $\varepsilon > 0$ este o toleranță specificată. Menționăm faptul că în implementările curente se utilizează şi alte criterii de oprire a iterațiilor, de exemplu, distanța dintre estimațiile variabilelor $\|x_{k+1} - x_k\|_2$ sau norma gradientului funcției de minimizat $\|\nabla f(x_k)\|_2$ sau $\|\nabla f(x_k)\|_\infty$, de-a lungul iterațiilor.

Orice metodă de optimizare neliniară cere specificarea unui punct inițial x_0 cu care să se înceapă calculele. Pentru consistența calculelor, punctul de plecare trebuie să se afle în $dom\, f$ şi în plus mulțimea de nivel constant:

$$S = \left\{ x \in dom\, f : f(x) \le f(x_0) \right\}$$

trebuie să fie închisă. Impunerea condiției de închidere a mulțimii S asigură eliminarea posibilității convergenței algoritmului într-un punct de pe frontiera lui $dom\, f$. Dacă $dom\, f = R^n$, atunci condiția este satisfăcută pentru orice x_0.

Metoda pasului descendent este una din metodele fundamentale de minimizare a funcțiilor diferențiabile. Ne reamintim că un vector d este o direcție descendentă pentru o funcție f dacă există un $\delta > 0$ astfel încât $f(x + \alpha d) < f(x)$ pentru orice $\alpha \in (0, \delta)$. În particular, dacă $\lim_{\alpha \to 0} (f(x + \alpha d) - f(x)) / \alpha < 0$, atunci d este o direcție descendentă. Metoda pasului descendent se deplasează de-a lungul direcției d cu $\|d\| = 1$, care

minimizează limita de mai sus. Dacă funcția f este diferențiabilă, atunci $-\nabla f(x)/\|\nabla f(x)\|$ este într-adevăr *direcția pasului descendent,* sau încă a *gradientului descendent.*

5.1. ALGORITMUL PASULUI DESCENDENT

Augustin Cauchy (1789-1857)

Metoda pasului descendent a fost proiectată de Cauchy încă din 1847 fiind repede admisă ca una dintre cele mai frecvent utilizate metode de optimizare datorită mai ales ușurinței de utilizare pentru probleme de mici dimensiuni. Totuşi odată cu dezvoltarea tehnicii de calcul metoda a fost eclipsată de alte metode iterative mult mai complicate din punct de vedere algoritmic, dar mai eficiente computațional.
Importanța metodei rezidă în faptul că aceasta furnizează conceptele şi ideile fundamentale ale metodelor de optimizare fără restricții. Proprietatea pe care se bazează metoda este următoarea. Presupunem că funcția $f(x)$ este continuu diferențiabilă într-o vecinătate a punctului curent x_k şi gradientul este nenul, adică $g_k \triangleq \nabla f(x_k) \neq 0$.
Din dezvoltarea în serie Taylor:

$$f(x) = f(x_k) + (x - x_k)^T g_k + o\left(\|x - x_k\|\right)$$

observăm că dacă scriem $x - x_k = \alpha d_k$, atunci direcția d_k care satisface condiția $d_k^T g_k < 0$ realizează cerința fundamentală de reducere a valorilor funcției, adică $f(x) < f(x_k)$, se numeşte *direcție descendentă*. Fixând α, atunci cu cât valoarea $d_k^T g_k$ este mai mică (adică cu cât valoarea $\left|d_k^T g_k\right|$ este mai mare), cu atât reducerea valorilor funcției de minimizat va fi mai pronunțată. Din inegalitatea Cauchy-Schwartz $\left|d_k^T g_k\right| \leq \|d_k\|\|g_k\|$, rezultă că $d_k^T g_k$ îşi atinge valoarea minimă *dacă şi numai dacă* $d_k = -g_k$. Deci $-g_k$ este direcția metodei pasului descendent. Schema iterativă a metodei pasului descendent este:

$$x_{k+1} = x_k - \alpha_k g_k, \tag{5.1.1}$$

unde, după cum ştim, α_k este lungimea pasului. Deoarece metoda utilizează ca direcție de deplasare gradientul cu semn schimbat, atunci aceasta se mai numeşte şi

metoda gradientul descendent. Cu acestea algoritmul metodei pasului descendent este următorul.

Algoritmul 5.1.1. (Algoritmul pasului descendent)

Pasul 1.	*Iniţializare*. Se consideră un punct iniţial $x_0 \in R^n$, precum şi toleranţa de terminare a iteraţiilor algoritmului de minimizare $0 < \varepsilon \ll 1$. Se pune $k = 0$.
Pasul 2.	Se calculează $g_k = \nabla f(x_k)$. Dacă $\|g_k\| \le \varepsilon$, atunci stop.
Pasul 3.	Se calculează lungimea pasului α_k utilizând căutarea liniară exactă $$f(x_k - \alpha_k g_k) = \min_{\alpha \ge 0} f(x_k - \alpha g_k),$$ sau aproximativă.
Pasul 4.	Se pune $x_{k+1} = x_k - \alpha_k g_k$, se consideră $k = k+1$ şi se continuă cu pasul 2. ♦

Observăm că algoritmul este extrem de simplu, implementarea lui fiind imediată. În pasul 3 se poate utiliza fie căutarea liniară exactă de-a lungul razei $\{x_k - \alpha g_k,\ \alpha \ge 0\}$, fie căutarea liniară aproximativă (pe care le-am discutat în capitolul 4). De cele mai multe ori în acest pas se utilizează căutarea liniară cu backtracking. După cum vom vedea algoritmul este liniar convergent. De aceea în pasul 2 se utilizează norma gradientului ca criteriu de oprire a iteraţiilor. Dacă problema este de mari dimensiuni, atunci se poate utiliza norma infinit. Următorul exemplu ilustrează funcţionarea şi caracteristicile algoritmului pasului descendent.

Exemplul 5.1.1. Fie funcţia

$$f(x) = -(0.5 + 0.5x_1)^4 x_2^4 \exp(2 - (0.5 + 0.5x_1)^4 - x_2^4)$$

a cărei reprezentare grafică este ilustrată în figura 5.1.1a şi 5.1.1b.

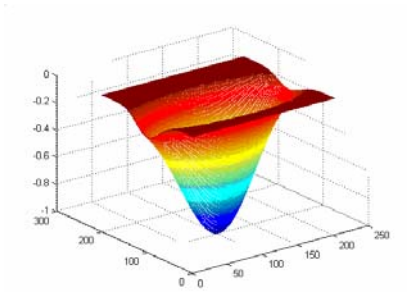

Fig. 5.1.1a. Reprezentarea funcţiei f.

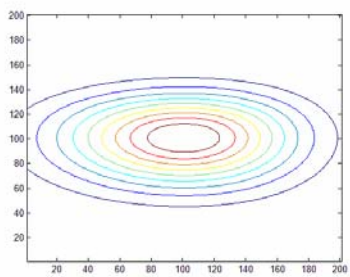

Fig. 5.1.1b. Conturul funcţiei f.

Observăm că deşi funcţia f este foarte neliniară, totuşi în domeniul $[0,2]\times[0,2]$ curbele de nivel constant ale acesteia sunt elipse alungite în direcţia axei x_1.

Minimul local al funcției considerate se realizează în punctul $x^* = [1,1]^T$, în care funcția are valoarea $f^* = -1$. Să considerăm acum punctul inițial $x_0 = [0.1, 0.8]^T$, atunci algoritmul pasului descendent cu backtracking furnizează o soluție în 48 de iterații. Evoluția erorii $\left| f(x_k) - f^* \right|$, precum şi a normei $\|g_k\|_2$ sunt ilustrate în figurile 5.1.1c şi respectiv 5.1.1d.

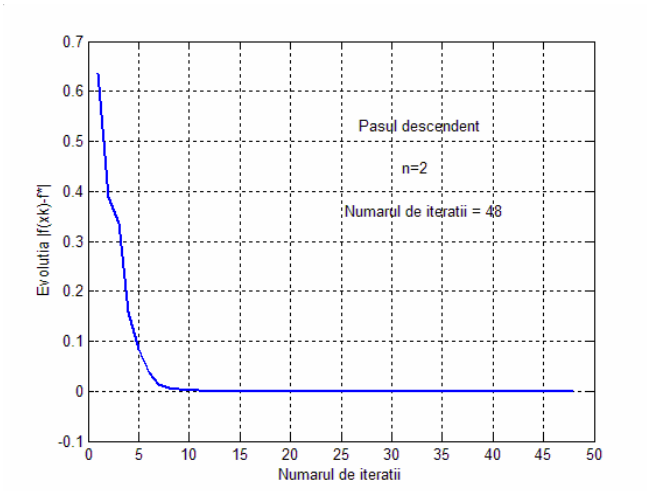

Fig. 5.1.1c. Evoluția erorii $\left| f(x_k) - f^* \right|$.

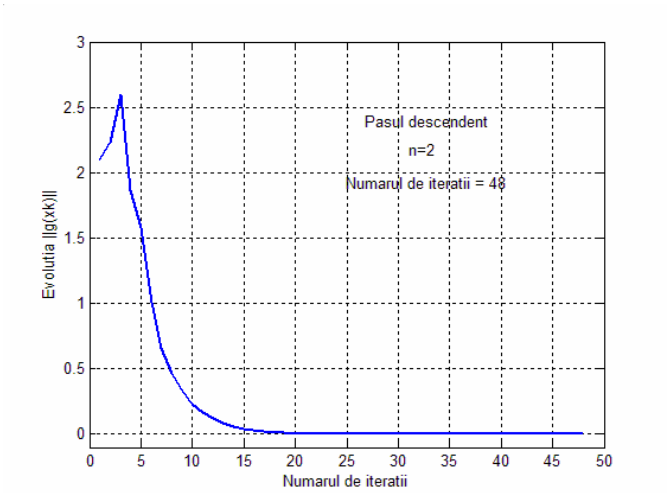

Fig. 5.1.1d. Evoluția normei $\|g_k\|_2$.

Observăm că eroarea $\left| f(x_k) - f^* \right|$ converge la zero aproximativ ca o serie geometrică, adică convergența este aproximativ liniară. În acest exemplu, eroarea

este redusă de la 0.6348 la 10^{-6} în 20 de iterații. Din figura 5.1.1c vedem că eroarea $\left|f(x_k)-f^*\right|$ se reduce foarte pronunțat în primele iterații, după care algoritmul devine din ce în ce mai lent convergent. Aceasta este o caracteristică definitorie pentru metoda pasului descendent.

Dacă în cadrul algoritmului se utilizează căutarea liniară Wolfe, atunci numărul de iterații necesare obținerii soluției este egal cu 13. În figurile 5.1.1e şi 5.1.1f se arată iterațiile generate de algoritmul 5.1.1 echipat cu backtracking şi respectiv cu căutarea liniară Wolfe.

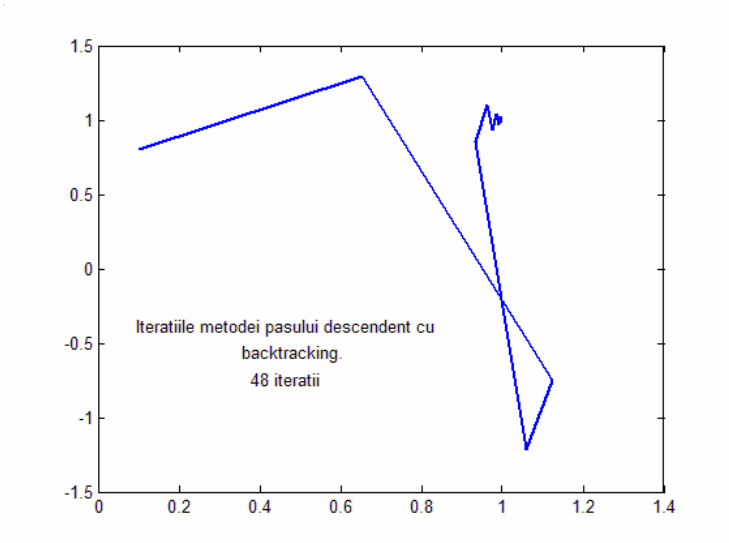

Fig. 5.1.1e. Iterațiile metodei pasului descendent cu backtracking.

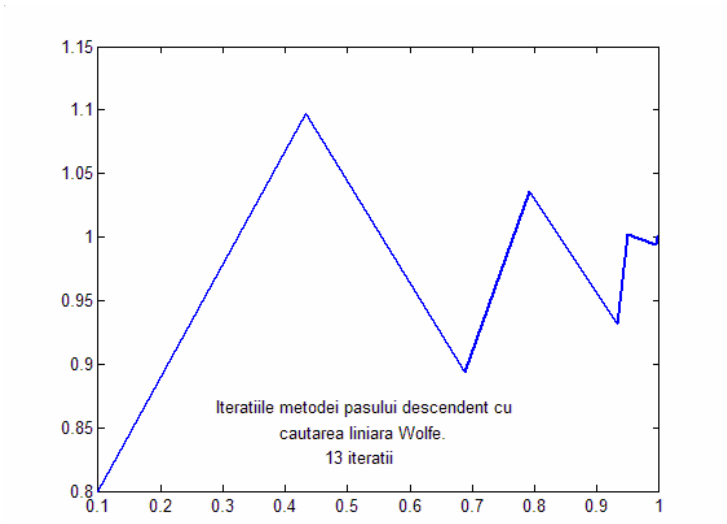

Fig. 5.1.1f. Iterațiile metodei pasului descendent cu căutarea liniară Wolfe.

Vedem că o căutare liniară mai pretențioasă este benefică. Totuşi, direcția pasului descendent este numai local descendentă. Pentru foarte multe probleme metoda pasului descendent nu este de loc a pasului descendent, fiind foarte lentă. Deşi, aşa după cum am spus metoda lucrează bine în prima parte a procesului iterativ, aceasta merge către soluție în zigzag. Cu cât graficul funcției de minimizat este mai aproape de o „vale îngustă", cu atât fenomenul de zigzag este mai pronunțat. Explicația acestui fenomen este imediată. Deoarece în căutarea liniară exactă se realizează condiția $g_{k+1}^T d_k = 0$, atunci aceasta în metoda pasului descendent devine $g_{k+1}^T g_k = d_{k+1}^T d_k = 0$. Aceasta arată că *gradienții succesivi sunt ortogonali*, adică *direcțiile succesive sunt ortogonale*, ceea ce conduce la fenomenul de zigzag. Deoarece căutarea liniară este inexactă, vedem că în figurile 5.1.1e şi 5.1.1f direcțiile nu sunt ortogonale. Din acest punct de vedere vedem că căutarea liniară cu backtracking este ceva mai „departe" decât cea dată de condițiile Wolfe. Când punctul staționar este atins, $\|g_k\|$ devine din ce în ce mai mic. Atunci din dezvoltarea Taylor

$$f(x_k - \alpha g_k) = f(x_k) - \alpha g_k^T g_k + o(\|\alpha g_k)\|,$$

se vede imediat că termenul de ordinul unu $\alpha g_k^T g_k = \alpha \|g_k\|^2$ este foarte mic. Deci reducerea funcției va fi mică. Acesta este principalul defect al metodei pasului descendent. Clar, metoda nu se recomandă pentru rezolvarea problemelor concrete. Totuşi aceasta este importantă mai ales prin conceptele pe care le promovează, tehnicile de demonstrație şi mai ales prin posibilitățile pe care le oferă pentru proiectarea de metode eficiente de optimizare fără restricții. Într-adevăr, orice modificare prin intermediul unei matrice pozitiv definite a direcției pasului descendent conduce la direcții descendente. Aceasta arată faptul că avem o multitudine de metode de minimizare, toate generate din metoda pasului descendent. În continuare să vedem cum funcționează această metodă pentru cazul funcțiilor pătratice.

5.2. METODA PASULUI DESCENDENT PENTRU FUNCȚII PĂTRATICE

În cele ce urmează vom prezenta o particularizare a acestor rezultate foarte generale pentru cazul funcțiilor pătratice. Funcțiile pătratice joacă un rol foarte important în teoria şi practica optimizării. În primul rând acestea deseori furnizează o soluție analitică. În al doilea rând acestea constituie foarte bune aproximații locale ale funcțiilor generale, care se pot utiliza în procesul de optimizare.

Fie funcția pătratică

$$f(x) = \frac{1}{2} x^T Q x - x^T b, \tag{5.2.1}$$

unde $Q \in R^{n\times n}$ este o matrice simetrică şi pozitiv definită. Fie $\lambda_{\min}$ şi $\lambda_{\max}$ valorile proprii minimă şi maximă a matricei Q. După cum ştim metoda direcțiilor descendente defineşte un şir minimizant:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{5.2.2}$$

unde d_k este o direcție descendentă care satisface

$$\nabla f(x_k)^T d_k < 0, \tag{5.2.3}$$

astfel încât $f(x_k + \alpha d_k) < f(x_k)$ pentru valori mici ale lui α. Pentru funcția pătratică f de mai sus, valoarea lui α care minimizează f de-a lungul liniei prin x_k în direcția d_k se calculează analitic foarte simplu ca:

$$\alpha = \frac{d_k^T r_k}{d_k^T Q d_k}, \tag{5.2.4}$$

unde $e_k = x^* - x_k$ este eroarea iterației x_k față de soluția optimă x^*, şi

$$r_k = Qe_k = b - Qx_k \tag{5.2.5}$$

este reziduul de la iterația k.

În metoda pasului descendent

$$d_k = -\nabla f(x_k) = r_k. \tag{5.2.6}$$

Introducând $Q-$ norma ca: $\|x\|_Q^2 = x^T Qx$, atunci $Q-$norma erorii este:

$$\begin{aligned}
\|e_{k+1}\|_Q^2 &= \left\|x^* - (x_k + \alpha_k d_k)\right\|_Q^2 = \|e_k - \alpha_k d_k\|_Q^2 = \\
&= \|e_k\|_Q^2 - 2\alpha_k e_k^T Q d_k + \alpha_k^2 \|d_k\|_Q^2 \\
&= \|e_k\|_Q^2 - 2\frac{r_k^T r_k}{r_k^T Q r_k} e_k^T Q r_k + \frac{(r_k^T r_k)^2}{(r_k^T Q r_k)^2} r_k^T Q r_k \\
&= \|e_k\|_Q^2 \left(1 - \frac{(r_k^T r_k)^2}{r_k^T Q r_k r_k^T Q^{-1} r_k}\right)
\end{aligned} \tag{5.2.7}$$

Pentru a demonstra convergența metodei pasului descendent avem nevoie de un rezultat tehnic cunoscut ca inegalitatea lui Kantorovich:

Inegalitatea Kantorovich. *Fie A o matrice simetrică şi pozitiv definită şi fie $0 < \lambda_{\min} < \lambda_{\max}$ valorile proprii minimă şi respectiv maximă a acestei matrice. Atunci:*

$$\min_{y\neq 0} \frac{(y^T y)^2}{y^T A y y^T A^{-1} y} = \frac{4\lambda_{\min}\lambda_{\max}}{(\lambda_{\min} + \lambda_{\max})^2}.$$

Demonstrație. Observăm că inegalitatea lui Kantorovici este invariantă la scalarea lui y. Deci este suficient să o demonstrăm pentru vectori unitari. Fie deci

$v_1, \ldots, v_n$ o bază ortogonală din R^n formată din vectorii proprii ai matricei A corespunzători valorilor proprii $\lambda_1, \ldots, \lambda_n$. Atunci

$$y = \sum_{i=1}^{n} y_i v_i,$$

astfel încât

$$\|y\|_2^2 = \sum_{i=1}^{n} y_i^2 = 1, \quad y^T A y = \sum_{i=1}^{n} y_i^2 \lambda_i, \quad y^T A^{-1} y = \sum_{i=1}^{n} y_i^2 \lambda_i^{-1}.$$

Deci demonstrația inegalității lui Kantorovich se reduce la rezolvarea următoarei probleme de optimizare cu restricții:

$$max\, g(y) = y^T A y y^T A^{-1} y$$

referitor la: (5.2.8)

$$h(y) = y^T y - 1 = 0.$$

Maximul acestei probleme cu necesitate trebuie să satisfacă condiția:

$$\nabla g(y) = \mu \nabla h(y),$$

unde μ este multiplicatorul Lagrange. Prin calcul direct obținem:

$$\frac{\partial g}{\partial y_i} = 2 y_i \lambda_i y^T A^{-1} y + 2 y_i \lambda_i^{-1} y^T A y,$$

$$\frac{\partial h}{\partial y_i} = 2 y_i.$$

Deci din condiția necesară de mai sus obținem:

$$y_i \lambda_i y^T A^{-1} y + y_i \lambda_i^{-1} y^T A y = \mu y_i,$$

sau

$$(\lambda_i^2 y^T A^{-1} y - \lambda_i \mu + y^T A y) \frac{y_i}{\lambda_i} = 0.$$

Expresia din paranteză este o ecuație pătratică în λ_i, şi poate fi zero numai pentru cel mult două valori proprii distincte. Deci, pot fi cel mult două componente nenule y_i şi y_j. În acest moment împărțind relația de mai sus prin y_i şi egalând expresiile pentru indicii i şi j obținem:

$$\lambda_i \left(\frac{y_i^2}{\lambda_i} + \frac{y_j^2}{\lambda_j} \right) + \frac{y_i^2 \lambda_i + y_j^2 \lambda_j}{\lambda_i} = \lambda_j \left(\frac{y_i^2}{\lambda_i} + \frac{y_j^2}{\lambda_j} \right) + \frac{y_i^2 \lambda_i + y_j^2 \lambda_j}{\lambda_j},$$

sau

$$(y_j^2 - y_i^2) \left(\frac{\lambda_i}{\lambda_j} + \frac{\lambda_j}{\lambda_i} - 2 \right) = (y_j^2 - y_i^2) \frac{(\lambda_i - \lambda_j)^2}{\lambda_i \lambda_j}.$$

Deci $y_i^2 = y_j^2 = 1/2,$ cu excepția cazului când $\lambda_i = \lambda_j$. Substituind aceste valori în expresia lui $g(y)$ din (5.2.8) obținem:

$$\frac{1}{4}(\lambda_i+\lambda_j)\left(\frac{1}{\lambda_i}+\frac{1}{\lambda_j}\right)=\frac{1}{4}\left(\frac{\lambda_i}{\lambda_j}+1\right)\left(\frac{\lambda_j}{\lambda_i}+1\right)=\frac{(\lambda_i+\lambda_j)^2}{4\lambda_i\lambda_j}.$$

Dar expresia din mijlocul lanțului de egalități de mai sus arată că aceasta este crescătoare în funcție de raportul λ_i/λ_j, când $\lambda_i>\lambda_j$. Deci considerând $\lambda_i=\lambda_{\max}$ şi $\lambda_j=\lambda_{\min}$ obținem inegalitatea Kantorovich. ■

Aplicând acum inegalitatea lui Kantorovich membrului drept din (5.2.7) şi observând că

$$1-\frac{4\lambda_{\min}\lambda_{\max}}{(\lambda_{\min}+\lambda_{\max})^2}=\frac{(\lambda_{\max}-\lambda_{\min})^2}{(\lambda_{\max}+\lambda_{\min})^2},$$

obținem:

$$\|e_{k+1}\|_Q\le\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\|e_k\|_Q\equiv\frac{\kappa-1}{\kappa+1}\|e_k\|_Q,$$

unde $\kappa=\lambda_{\max}/\lambda_{\min}$ este numărul de condiționare al matricei Q. Cu alte cuvinte, *convergența metodei pasului descendent pentru funcții pătratice este liniară*.

Exemplul 5.2.1. Să vedem funcționarea algoritmului pasului descendent pentru funcția pătratică:

$$f(x)=\frac{1}{2}x^TQx$$

unde $Q=diag(1,2,\ldots,n)$. Observăm că Q este pozitiv definită şi are valorile proprii numerele întregi pozitive de la 1 la n, deci numărul de condiționare în norma euclidiană este egal cu n, care este foarte mare. Pentru această funcție $\nabla f(x)=Qx$ şi deci lungimea pasului de deplasare se calculează ca:

$$\alpha_k=\frac{g_k^Tg_k}{g_k^TQg_k},\quad\text{unde}\quad g_k=Qx_k.$$

După cum am văzut, pentru această alegere *optimă* a lui α_k algoritmul pasului descendent are rata de convergență liniară:

$$\|x_k-x^*\|_Q\le\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\|x_{k-1}-x^*\|_Q,$$

unde $\|z\|_Q=z^TQz$, iar $\lambda_{\min}$ şi $\lambda_{\max}$ sunt valorile proprii minimă şi respectiv maximă a matricei Q. Soluția acestei probleme de minimizare este vectorul zero, pentru care valoarea funcției este nulă. Considerând $n=500$, $\varepsilon=10^{-6}$ şi punctul inițial $x_0=[0.5,0.5,\cdots,0.5]$, atunci algoritmul pasului descendent furnizează următoarea evoluție a erorii $|f(x_k)-f(x^*)|$ de-a lungul celor 3342 de iterații necesare obținerii unei soluții cu acuratețea impusă, figura 5.2.1.

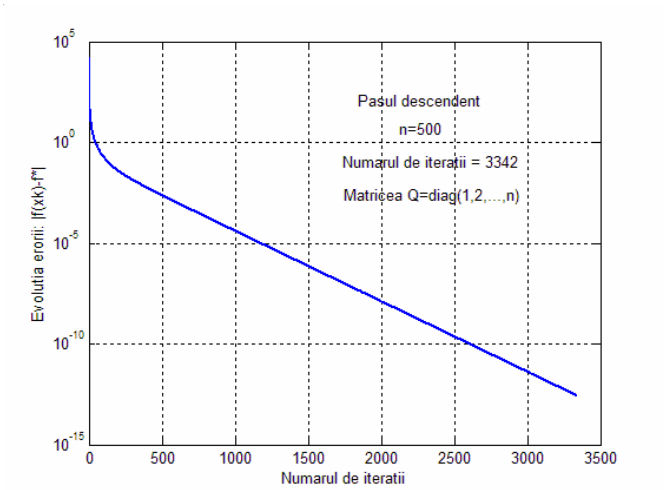

Fig. 5.2.1. Evoluția erorii $\left|f(x_k)-f(x^*)\right|$ pentru algoritmul pasului descendent cu lungimea optimă a pasului.

Vedem că acesta are un comportament liniar, mai ales în partea finală a iterațiilor. Într-adevăr, în acest caz, pentru valori mari ale lui n raportul

$$\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}$$

tinde la 1, ceea ce justifică convergența foarte lentă a algoritmului.

Considerând acum matricea $Q=diag(1,10)$, atunci algoritmul pasului descendent are comportarea ca în figurile 5.2.1a şi 5.2.1b. În figura 5.2.1a se vede evoluția pur liniară a algoritmului. Căutarea liniară fiind exactă, traiectoria către soluție este în *zigzag în unghiuri drepte*, ca în figura 5.2.1b.

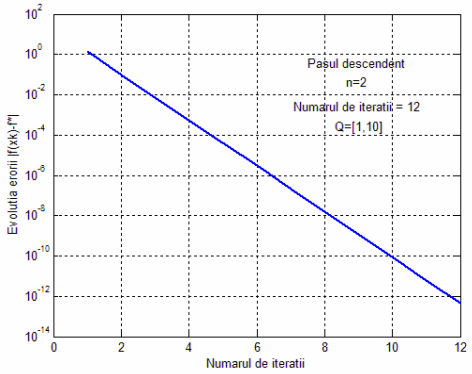

Fig. 5.2.1a. Evoluția erorii $\left|f(x_k)-f(x^*)\right|$.

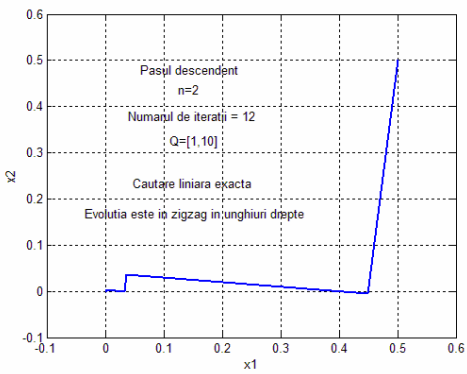

Fig. 5.2.1b. Iterațiile algoritmului.

Vedem că rata de convergență a algoritmului pasului descendent depinde de raportul dintre cea mai lungă axă şi cea mai scurtă axă a elipsoidului corespunzător

matricei Q. Cu cât acest raport este mai mare cu atât convergența este mai lentă. Detalii asupra convergenței metodei pasului descendent pentru funcții pătratice sunt date de următoarea teoremă.

Teorema 5.2.1. *Fie problema*

$$\min_{x\in R^n} f(x) \equiv \frac{1}{2}x^T Qx, \tag{5.2.9}$$

unde Q este o matrice simetrică, pozitiv definită cu $\lambda_{\min}$ şi $\lambda_{\max}$ valorile proprii minimă şi respectiv maximă. Fie x^* soluția problemei (5.2.9). Atunci şirul generat de algoritmul pasului descendent converge la x^*, rata de convergență este cel puțin liniară şi următoarele relații au loc:

$$\frac{f(x_{k+1})-f(x^*)}{f(x_k)-f(x^*)} \le \frac{(\kappa-1)^2}{(\kappa+1)^2} = \frac{(\lambda_{\max}-\lambda_{\min})^2}{(\lambda_{\max}+\lambda_{\min})^2}, \tag{5.2.10}$$

$$\frac{\left\|x_{k+1}-x^*\right\|_Q}{\left\|x_k-x^*\right\|_Q} \le \frac{\kappa-1}{\kappa+1} = \left(\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\right), \tag{5.2.11}$$

$$\frac{\left\|x_{k+1}-x^*\right\|}{\left\|x_k-x^*\right\|} \le \sqrt{\kappa}\,\frac{\kappa-1}{\kappa+1} = \sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}\left(\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\right), \tag{5.2.12}$$

unde $\kappa = \lambda_{\max} / \lambda_{\min}$.

Demonstrație. Considerăm algoritmul pasului descendent $x_{k+1} = x_k - \alpha_k g_k$, unde $\alpha_k = \dfrac{g_k^T g_k}{g_k^T Q g_k}$, şi $g_k = Qx_k$. Atunci

$$\frac{f(x_k)-f(x_{k+1})}{f(x_k)} = \frac{\frac{1}{2}x_k^T Qx_k - \frac{1}{2}(x_k-\alpha_k g_k)^T Q(x_k-\alpha_k g_k)}{\frac{1}{2}x_k^T Qx_k}$$

$$= \frac{\alpha_k g_k^T Qx_k - \frac{1}{2}\alpha_k^2 g_k^T Qg_k}{\frac{1}{2}x_k^T Qx_k} = \frac{\left(g_k^T g_k\right)^2}{\left(g_k^T Qg_k\right)\left(g_k^T Q^{-1} g_k\right)}.$$

Utilizând inegalitatea Kantorovich obținem imediat

$$\frac{f(x_{k+1})}{f(x_k)} = 1 - \frac{\left(g_k^T g_k\right)^2}{\left(g_k^T Qg_k\right)\left(g_k^T Q^{-1} g_k\right)} = 1 - \frac{4\lambda_{\max}\lambda_{\min}}{(\lambda_{\max}+\lambda_{\min})^2} = \left(\frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}}\right)^2,$$

care este chiar (5.2.10).

Fie acum $e_k = x_k - x^*$. Deoarece Q este simetrică şi pozitiv definită, atunci

$$\lambda_{\min} e_k^T e_k \le e_k^T Q e_k \le \lambda_{\max} e_k^T e_k.$$

Deoarece $x^* = 0$, rezultă că

$$\left\| x_k - x^* \right\|_Q^2 = e_k^T Q e_k = x_k^T Q x_k = 2 f(x_k) .$$

Acum utilizând şirul de inegalităţi de mai sus rezultă că pentru orice $k \geq 0$

$$\lambda_{\min} \left\| x_k - x^* \right\|^2 \leq 2 f(x_k) \leq \lambda_{\max} \left\| x_k - x^* \right\|^2 .$$

Din relaţiile de mai sus obţinem

$$\frac{\lambda_{\min} \left\| x_{k+1} - x^* \right\|^2}{\lambda_{\max} \left\| x_k - x^* \right\|^2} \leq \frac{\left\| x_{k+1} - x^* \right\|_Q^2}{\left\| x_k - x^* \right\|_Q^2} \leq \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \right)^2 ,$$

care furnizează (5.2.11) şi (5.2.12). ■

5.3. TEORIA CONVERGENŢEI METODEI PASULUI DESCENDENT

În cele ce urmează prezentăm convergenţa globală şi rata de convergenţă locală a metodei pasului descendent. În finalul acestei secţiuni ne ocupăm de cazul funcţiilor tare convexe.

Teorema 5.3.1. *Fie funcţia $f : R^n \to R$ continuu diferenţiabilă. Atunci orice punct de acumulare al şirului $\{x_k\}$ generat de algoritmului pasului descendent 5.1.1 cu căutare liniară exactă este un punct staţionar al funcţiei f.*

Demonstraţie. Fie $\bar{x}$ un punct de acumulare al şirului $\{x_k\}$ şi K o mulţime infinită de indici astfel încât $\lim_{k \in K} x_k = \bar{x}$. Considerăm $d_k = -\nabla f(x_k)$. Deoarece f este continuu diferenţiabilă, şirul $\{d_k : k \in K\}$ este uniform mărginit şi $\|d_k\| = \|\nabla f(x_k)\|$. Deoarece ipotezele teoremei 4.2.2 sunt îndeplinite, rezultă că $\|\nabla f(\bar{x})\|^2 = 0$, adică $\nabla f(\bar{x}) = 0$. ■

Teorema 5.3.2. *Fie funcţia $f : R^n \to R$ de două ori continuu diferenţiabilă pe R^n şi $\|\nabla^2 f(x)\| \leq M$, unde M este o constantă pozitivă. Fie punctul iniţial x_0 şi toleranţa $\varepsilon > 0$ date. Atunci şirul generat de algoritmului pasului descendent 5.1.1 satisface condiţia de terminare după un număr finit de iteraţii, sau $\lim_{k \to \infty} f(x_k) = -\infty$, sau $\lim_{k \to \infty} \nabla f(x_k) = 0$.*

Demonstrație. Considerăm cazul infinit. Din algoritmul 5.1.1 şi teorema 4.2.1 rezultă că

$$f(x_k) - f(x_{k+1}) \geq \frac{1}{2M}\|\nabla f(x_k)\|^2 .$$

Atunci

$$f(x_0) - f(x_k) = \sum_{i=0}^{k-1}[f(x_i) - f(x_{i+1})] \geq \frac{1}{2M}\sum_{i=0}^{k-1}\|\nabla f(x_i)\|^2 .$$

La limită obținem fie $\lim_{k\to\infty} f(x_k) = -\infty$, fie $\lim_{k\to\infty} \nabla f(x_k) = 0$. ■

Următoarea teoremă stabileşte rata de convergența a metodei pasului descendent cu căutare liniară exactă.

Teorema 5.3.3. *Fie funcția $f : R^n \to R$ care satisface ipotezele teoremei 4.2.4. Dacă şirul generat de algoritmul pasului descendent 5.1.1 converge la x^*, atunci rata de convergența este liniară.*

Demonstrația este o consecință directă a teoremei 4.2.4.

Rata de convergență a metodei pasului descendent pentru cazul funcțiilor generale este stabilită de următoarea teoremă.

Teorema 5.3.4. *Fie funcția $f : R^n \to R$ de două ori continuu diferențiabilă într-o vecinătate a lui x^* cu $\nabla f(x^*) = 0$ şi Hessianul $\nabla^2 f(x^*)$ o matrice pozitiv definită. Fie $\lambda_{\max}$ şi $\lambda_{\min}$ valoarea proprie maximă, respectiv minimă a lui $\nabla^2 f(x^*)$ care satisfac $0 < m \leq \lambda_{\min} \leq \lambda_{\max} \leq M$. Fie $\{x_k\}$ şirul generat de algoritmul pasului descendent 5.1.1 convergent la x^*. Fie*

$$\frac{f(x_{k+1}) - f(x^*)}{f(x_k) - f(x^*)} = \beta_k . \tag{5.3.1}$$

Atunci, pentru orice k, $\beta_k < 1$ şi

$$\limsup_{k\to+\infty} \beta_k \leq \frac{M-m}{M} < 1. \tag{5.3.2}$$

Demonstrație. Din teorema 4.2.1 avem imediat

$$[f(x_k) - f(x^*)] - [f(x_{k+1}) - f(x^*)] = f(x_k) - f(x_{k+1})$$
$$\geq \frac{1}{2M}\|\nabla f(x_k)\|^2 .$$

Acum, utilizând definiția lui β_k rezultă că

$$(1-\beta_k)[f(x_k)-f(x^*)] \geq \frac{1}{2M}\|\nabla f(x_k)\|^2 .$$

Deci, din ipotezele teoremei obținem

$$\beta_k \leq 1-\frac{\|\nabla f(x_k)\|^2}{2M[f(x_k)-f(x^*)]} < 1 . \tag{5.3.3}$$

Presupunem că

$$\frac{x_k - x^*}{\|x_k - x^*\|} \to \bar{d} .$$

Este clar că

$$\|\nabla f(x_k)\|^2 = \|x_k - x^*\|^2 \left(\|\nabla^2 f(x^*)\bar{d}\|^2 + o(1)\right)$$

şi

$$f(x_k)-f(x^*) = \frac{1}{2}\|x_k - x^*\|^2 \left(\bar{d}^T \nabla^2 f(x^*)\bar{d} + o(1)\right).$$

Utilizând inegalitățile de mai sus obținem

$$\lim_{k\to\infty} \frac{\|\nabla f(x_k)\|^2}{f(x_k)-f(x^*)} = \frac{2\|\nabla^2 f(x^*)\bar{d}\|^2}{\bar{d}^T \nabla^2 f(x^*)\bar{d}} \geq 2m . \tag{5.3.4}$$

Deci, din (5.3.3) şi (5.3.4) rezultă că

$$\limsup_{k\to\infty} \beta_k \leq 1 - \liminf_{k\to\infty} \frac{\|\nabla f(x_k)\|^2}{2M[f(x_k)-f(x^*)]} \leq 1 - \frac{m}{M} < 1 ,$$

ceea ce completează demonstrația. ■

Margini asupra funcțiilor tare convexe. În cele ce urmează vom presupune că funcția din (5.0.1) este tare convexă pe S, ceea ce înseamnă că există constanta $m > 0$ astfel încât pentru toți $x \in S$:

$$\nabla^2 f(x) \geq mI . \tag{5.3.5}$$

Impunerea acestei condiții are anumite consecințe pe care le vom preciza imediat. Observăm că pentru $x, y \in S$ are loc relația:

$$f(y) = f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(z)(y-x) ,$$

unde z este un punct din segmentul $[x, y]$. Ipoteza de convexitate tare (5.3.5) determină ca ultimul termen din membrul drept al relației de mai sus să fie mărginit de $(m/2)\|y-x\|_2^2$, astfel încât are loc inegalitatea:

$$f(y) \geq f(x) + \nabla f(x)^T (y-x) + \frac{m}{2}\|y-x\|_2^2 , \tag{5.3.6}$$

valabilă pentru toți x şi y din S. Notăm că dacă $m = 0$, atunci redescoperim inegalitatea fundamentală care caracterizează convexitatea. Pe de altă parte, dacă $m > 0$, atunci obținem o margine inferioară mult mai bună asupra lui $f(y)$ decât cea dată numai de simpla convexitate.

Inegalitatea (5.3.6), consecință directă a convexității tari, se poate utiliza pentru a obține margini atât pentru valorile funcției de optimizat, $f(x) - f^*$, cât şi pentru valorile variabilelor problemei, $\|x - x^*\|_2$.

Pentru început să arătăm cum inegalitatea (5.3.6) se poate utiliza pentru a mărgini $f(x) - f^*$ în termenii normei $\|\nabla f(x)\|_2$. Într-adevăr, pentru x fixat, membrul drept din (5.3.6) este o funcție pătratică convexă. Anulând gradientul în raport cu y, atunci minimul acestei funcții este $\tilde{y} = x - (1/m)\nabla f(x)$. Deci

$$\begin{aligned} f(y) &\geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|_2^2 \\ &\geq f(x) + \nabla f(x)^T (\tilde{y} - x) + \frac{m}{2}\|\tilde{y} - x\|_2^2 \\ &= f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 . \end{aligned}$$

Deoarece această inegalitate are loc pentru orice $y \in S$, rezultă că:

$$f^* \geq f(x) - \frac{1}{2m}\|\nabla f(x)\|_2^2 . \tag{5.3.7}$$

Această inegalitate arată că dacă gradientul într-un punct este mic, atunci acel punct este lângă soluția optimă a problemei. Deci în virtutea lui (5.3.2) observăm că (5.3.7) se poate interpreta ca o condiție de *suboptimalitate*, care generalizează condiția foarte tare de optimalitate (5.0.2), adică:

$$\|\nabla f(x)\|_2 \leq (2m\varepsilon)^{1/2} \quad \Rightarrow \quad f(x) - f^* \leq \varepsilon . \tag{5.3.8}$$

Acum să vedem cum inegalitatea (5.3.6) se poate utiliza pentru a obține o margine asupra lui $\|x - x^*\|_2$, adică asupra distanței între x şi punctul de optim x^*, în funcție de norma $\|\nabla f(x)\|_2$. Pentru aceasta din (5.3.6) cu $y = x^*$ obținem:

$$\begin{aligned} f^* = f(x^*) &\geq f(x) + \nabla f(x)^T (x^* - x) + \frac{m}{2}\|x^* - x\|_2^2 \\ &\geq f(x) - \|\nabla f(x)\|_2 \|x^* - x\|_2 + \frac{m}{2}\|x^* - x\|_2^2 , \end{aligned}$$

în care am aplicat inegalitatea Cauchy-Schwarz. Dar, deoarece $f^* \leq f(x)$, rezultă că:

$$-\|\nabla f(x)\|_2 \|x^* - x\|_2 + \frac{m}{2}\|x^* - x\|_2^2 \leq 0 ,$$

de unde obținem imediat:

$$\left\|x - x^*\right\|_2 \le \frac{2}{m}\left\|\nabla f(x)\right\|_2 . \tag{5.3.9}$$

Consecința directă a lui (5.3.9) este că punctul optimal x^* este unic, ceea ce era şi cazul deoarece funcția de optimizat este presupusă tare convexă.

Convexitatea tare a lui f pe mulțimea S determină că mulțimile de nivel constant conținute în S sunt mărginite, în particular S este mărginită. Deci valoarea proprie maximă a Hessianului $\nabla^2 f(x)$, care este o funcție continuă de x pe S, este mărginită superior pe S, adică există o constantă M astfel încât pentru toți $x \in S$:

$$\nabla^2 f(x) \le MI . \tag{5.3.10}$$

Această margine superioară asupra Hessianului implică faptul că pentru orice x şi y din S are loc inegalitatea:

$$f(y) \le f(x) + \nabla f(x)^T (y - x) + \frac{M}{2}\left\|y - x\right\|_2^2 , \tag{5.3.11}$$

care este analogă cu (5.3.6). Procedând în aceeaşi manieră ca mai sus, adică minimizând în raport cu y ambii membri din (5.3.11) obținem:

$$f^* \le f(x) - \frac{1}{2M}\left\|\nabla f(x)\right\|_2^2 , \tag{5.3.12}$$

replica relației (5.3.7).

Cuplând relațiile (5.3.7) şi (5.3.12) obținem următoarea mărginire a valorii funcției de optimizat într-un punct x:

$$\frac{1}{2M}\left\|\nabla f(x)\right\|_2^2 \le f(x) - f^* \le \frac{1}{2m}\left\|\nabla f(x)\right\|_2^2 ,$$

care precizează suboptimalitatea lui x în termenii normei gradientului funcției în punctul x.

Din inegalitatea de convexitate tare (5.3.5) şi inegalitatea (5.3.10) obținem că pentru toți $x \in S$:

$$mI \le \nabla^2 f(x) \le MI . \tag{5.3.13}$$

Raportul $\kappa = M / m$ este astfel o margine superioară a numărului de condiționare a Hessianului $\nabla^2 f(x)$, adică raportul dintre cea mai mare valoare proprie şi respectiv cea mai mică valoare proprie a acestuia.

Numărul de condiționare al mulțimilor de nivel constant. În acest moment vom introduce un concept care se dovedeşte a avea o importanță foarte mare asupra eficienței metodelor de optimizare fără restricții. Este vorba de numărul de condiționare al unei mulțimi (convexe) care are atât o interpretare geometrică cât şi una algebrică foarte naturală. În acest context putem da o interpretare geometrică a lui (5.3.13) în termenii mulțimilor de nivel constant.

Pentru aceasta fie o mulțime convexă $C \subseteq R^n$. Definim *lățimea în direcția* q a acesteia, unde $\|q\|_2 = 1$, ca:

$$W(C,q) = \sup_{z \in C} q^T z - \inf_{z \in C} q^T z .$$

Cu aceasta se poate defini *lățimea minimă*, respectiv *lățimea maximă* a unei mulțimi convexe C ca:

$$W_{\min} = \inf_{\|q\|_2 = 1} W(C,q), \quad W_{\max} = \sup_{\|q\|_2 = 1} W(C,q) .$$

Cu aceste elemente se poate defini *numărul de condiționare* al unei mulțimi convexe C ca:

$$cond(C) = \frac{W_{\max}^2}{W_{\min}^2},$$

adică pătratul raportului dintre lățimea maximă şi cea minimă. Numărul de condiționare al unei mulțimi convexe dă o măsură a excentricității, sau a deformării, acestei mulțimi, adică a devierii acesteia de la o sferă. Dacă numărul de condiționare a unei mulțimi convexe C este mic (apropiat de unu) atunci aceasta înseamnă că mulțimea are aproximativ aceeaşi lățime în toate direcțiile, adică este aproape sferică. Dacă, pe de altă parte, acest număr este mare, atunci aceasta înseamnă că mulțimea este mai lată în anumite direcții decât în altele. Vom considera imediat un exemplu semnificativ [Boyd şi Vandenberghe, 2003].

Numărul de condiționare al unui elipsoid. Fie elipsoidul:

$$E = \left\{ x : (x - x_0)^T A^{-1} (x - x_0) \le 1 \right\},$$

unde matricea A este simetrică şi pozitiv definită. Lățimea elipsoidului E în direcția q este dată de:

$$\sup_{z \in E} q^T z - \inf_{z \in E} q^T z = \left(\left\| A^{1/2} q \right\|_2 + q^T x_0 \right) - \left(- \left\| A^{1/2} q \right\|_2 + q^T x_0 \right) = 2 \left\| A^{1/2} q \right\|_2 .$$

Deci lățimile minimă şi maximă sunt:

$$W_{\min} = 2\lambda_{\min}(A)^{1/2}, \quad \text{şi} \quad W_{\max} = 2\lambda_{\max}(A)^{1/2} .$$

Ca atare, numărul de condiționare al elipsoidului E este

$$cond(E) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \kappa(A),$$

adică este egal cu numărul de condiționare al matricei A care defineşte elipsoidul.

Să presupunem că pentru toți x funcția f satisface condiția de mărginire a Hessianului $mI \le \nabla^2 f(x) \le MI$. Cu aceasta să calculăm o margine a numărului de condiționare a mulțimilor de nivel constant egal cu α :

$$C_\alpha = \left\{ x : f(x) \le \alpha \right\}$$

unde $\alpha > f^*$. Considerând în inegalitățile (5.3.6) şi (5.3.11) $x = x^*$ obținem:

$$f^* + \frac{M}{2}\left\|y - x^*\right\|_2^2 \geq f(y) \geq f^* + \frac{m}{2}\left\|y - x^*\right\|_2^2.$$

Aceasta implică următoarele incluziuni $B_{in} \subseteq C_\alpha \subseteq B_{ou}$, unde:

$$B_{in} = \left\{ y : \left\|y - x^*\right\|_2 \leq \left(\frac{2(\alpha - f^*)}{M} \right)^{1/2} \right\},$$

$$B_{ou} = \left\{ y : \left\|y - x^*\right\|_2 \leq \left(\frac{2(\alpha - f^*)}{m} \right)^{1/2} \right\}.$$

Cu alte cuvinte, mulțimea de nivel constant egal cu α conține sfera B_{in} şi este conținută în sfera B_{ou}. Ca atare, raportul razelor acestor sfere furnizează o margine superioară a numărului de condiționare a lui C_α:

$$cond(C_\alpha) \leq \frac{M}{m}.$$

De aici rezultă următoarea interpretare geometrică a numărului de condiționare $\kappa(\nabla^2 f(x^*))$ a Hessianului în punctul de optim. Într-adevăr, în jurul lui x^* putem scrie:

$$f(y) \cong f^* + \frac{1}{2}(y - x^*)^T \nabla^2 f(x^*)(y - x^*)$$

de unde rezultă că pentru valori α suficient de apropiate de f^*,

$$C_\alpha \approx \left\{ y : (y - x^*)^T \nabla^2 f(x^*)(y - x^*) \leq 2(\alpha - f^*) \right\},$$

adică această mulțime de nivel constant este bine aproximată printr-un elipsoid centrat în x^*. Deci

$$\lim_{\alpha \to f^*} cond(C_\alpha) = \kappa(\nabla^2 f(x^*)),$$

care arată cum mulțimile de nivel constant de valori $\alpha > f^*$ caracterizează numărul de condiționare al Hessianului în punctul de optim.

În secțiunile următoare ale acestui capitol vom prezenta o analiză a algoritmilor de optimizare fără restricții care precizează margini asupra numărului de iterații necesare pentru a realiza inegalitatea $f(x_k) - f^* \leq \varepsilon$, unde ε este o toleranță în raport cu care se face optimizarea. *Toate aceste margini utilizează constantele necunoscute m şi M, ceea ce implică imediat lipsa de valoare practică a acestor rezultate.* Totuşi, *conceptual* aceste rezultate sunt valoroase deoarece ele stabilesc convergența algoritmului, chiar dacă marginile asupra numărului de iterații necesar obținerii unei acuratețe date depind de constante necunoscute. Există o excepție de la această situație definită de *funcțiile convexe auto-concordante.* Pentru aceste funcții se poate da o analiză completă a convergenței care nu depinde de nici o constantă necunoscută, aşa după cum vom vedea în capitolul 6 când considerăm metoda Newton.

În acest moment să facem o mică recapitulare a dezvoltărilor făcute până acum. Observăm că utilizând constantele m şi M cu care putem mărgini Hessianul unei funcții tare convexe am determinat margini ale suboptimalității punctului x în termenii gradientului acestei funcții în acest punct. *Critica ce se poate aduce acestui mod de abordare este că aceste constante m şi M sunt cunoscute numai în cazuri foarte particulare (adică pentru funcții foarte speciale), aşa încât inegalitatea (5.3.8) nu are nici o valoare practică pentru a fi implementată ca un criteriu de oprire a iterațiilor. Implicația (5.3.8) are numai o valoare calitativă, şi poate fi considerată numai ca un criteriu conceptual de oprire a iterațiilor în sensul că dacă gradientul funcției f în punctul x este suficient de mic, atunci diferența dintre $f(x)$ şi f^* este mică.* Altfel spus, dacă oprim un algoritm de optimizare de îndată ce $\|\nabla f(x_k)\|_2 \le \eta$, unde η este ales suficient de mic, în fapt mai mic decât $\sqrt{m\varepsilon}$, atunci sperăm că $f(x_k) - f^* \le \varepsilon$. *Remarcăm imediat că toate aceste considerații sunt valabile numai pentru funcții tare convexe. Pentru funcțiile neconvexe nu putem spune nimic.*

În cele ce urmează să prezentăm analiza convergenței algoritmului de gradient descendent pentru situația în care lungimea pasului se determină prin căutare liniară exactă şi apoi prin backtracking. *Importanța acestei analize constă în faptul că aceasta ilustrează o metodologie de studiu a convergenței, care se poate aplica şi altor metode de optimizare.* În particular o vom aplica metodei Newton.

Presupunem că funcția f este tare convexă pe S, aşa încât, după cum am văzut în secțiunile anterioare, rezultă că există constantele pozitive m şi M, astfel încât $mI \le \nabla^2 f(x) \le MI$. Definim funcția $\varphi : R \to R$ prin:

$$\varphi(\alpha) = f(x_k - \alpha \nabla f(x_k)).$$

În analiza care urmează vom presupune că $x_k - \alpha \nabla f(x_k) \in S$. Din inegalitatea (5.3.11), în care punem $y = x_k - \alpha \nabla f(x_k)$, obținem:

$$\varphi(\alpha) \le f(x_k) - \alpha \|\nabla f(x_k)\|_2^2 + \frac{M\alpha^2}{2} \|\nabla f(x_k)\|_2^2. \qquad (5.3.14)$$

Căutare liniară exactă. Presupunem că în pasul 3 al algoritmului 5.1.1, al pasului descendent, pentru determinarea lungimii pasului, utilizăm o căutare liniară exactă. Să minimizăm în raport cu α ambii membrii ai inegalității (5.3.14). Membrul stâng furnizează valoarea $\varphi(\alpha_{exact})$, unde α_{exact} este lungimea pasului care minimizează φ. Membrul drept este o funcție pătratică, care este minimizată de $\alpha = 1/M$ şi are valoarea minimă egală cu $f(x_k) - (1/2M)\|\nabla f(x_k)\|_2^2$. Deci:

$$f(x_{k+1}) = \varphi(\alpha_{exact}) \le f(x_k) - \frac{1}{2M}\|\nabla f(x_k)\|_2^2 .$$

Scăzând f^* din ambii membrii obținem:

$$f(x_{k+1}) - f^* \le f(x_k) - f^* - \frac{1}{2M}\|\nabla f(x_k)\|_2^2 .$$

Dar din (5.3.7) observăm că $\|\nabla f(x_k)\|_2^2 \ge 2m(f(x_k) - f^*)$ ceea ce implică:

$$f(x_{k+1}) - f^* \le (1 - m/M)(f(x_k) - f^*) .$$

Aplicând recursiv această inegalitate, obținem:

$$f(x_k) - f^* \le c^k\left(f(x_0) - f^*\right), \tag{5.3.15}$$

unde $c = 1 - \frac{m}{M}$<1, ceea ce arată că $f(x_k)$ converge la f^* când $k \to \infty$. În particular, din (5.3.15) rezultă că condiția $f(x_k) - f^* \le \varepsilon$ se realizează după cel mult:

$$\frac{\log\left((f(x_0) - f^*)/\varepsilon\right)}{\log(1/c)} \tag{5.3.16}$$

iterații. Această margine asupra numărului de iterații cerut de algoritmul gradientului descendent furnizează o anumită înțelegere a comportării algoritmului. Observăm că numărătorul din (5.3.16) este chiar logaritmul raportului dintre suboptimalitatea inițială (diferența dintre $f(x_0)$ şi f^*) şi suboptimalitatea finală (care trebuie să fie mai mică sau egală cu ε). *Ca atare, numărul de iterații depinde de cât de bună este aproximația inițială* x_0 *a punctului de optim.* Pe de altă parte, numitorul din (5.3.16) este o funcție de M/m, care după cum am văzut este o margine a numărului de condiționare a Hessianului $\nabla^2 f(x)$ pe S, sau altfel spus, numărul de condiționare a mulțimilor de nivel constant $\{z : f(z) \le \alpha\}$. Pentru o margine a numărului de condiționare M/m mare, avem:

$$\log(1/c) = \log(1/(1 - m/M)) \cong m/M ,$$

astfel încât marginea asupra numărului de iterații creşte aproximativ liniar cu creşterea lui M/m. *Deci, când Hessianul lui* f *lângă* x^* *are un număr de condiționare mare, atunci algoritmul pasului descendent cere un număr mare de iterații. Invers, dacă mulțimile de nivel constant ale lui* f *sunt relativ isotrope, astfel încât marginea* M/m *a numărului de condiționare este relativ mică, atunci din (5.3.15) se vede că algoritmul este rapid convergent la soluție, deoarece* c *este mic, sau cel puțin nu este apropiat de 1.* Relația (5.3.15) arată că eroarea $f(x_k) - f^*$ tinde la zero ca o serie geometrică, adică *convergența algoritmului de gradient descendent este liniară, şi aceasta este tot ce se poate spera de la acesta.*

Căutare cu backtracking. În continuare să considerăm cazul în care în pasul 3 al algoritmului pasului descendent 5.1.1 utilizăm o căutare liniară cu backtracking. Vom arăta că condiția de terminare a căutării liniare cu backtracking:

$$\varphi(\alpha) \le f(x_k) - \rho\alpha \left\|\nabla f(x_k)\right\|_2^2$$

este satisfăcută ori de câte ori $0 \le \alpha \le 1/M$. Pentru aceasta, notăm că din convexitatea lui $-\alpha + M\alpha^2/2$ rezultă că:

$$0 \le \alpha \le 1/M \quad \Rightarrow \quad -\alpha + \frac{M\alpha^2}{2} \le -\frac{\alpha}{2}.$$

Cu acest rezultat şi utilizând (5.3.14) obținem că pentru $0 \le \alpha \le 1/M$:

$$\begin{aligned}\varphi(\alpha) &\le f(x_k) - \alpha \left\|\nabla f(x_k)\right\|_2^2 + \frac{M\alpha^2}{2}\left\|\nabla f(x_k)\right\|_2^2 \\ &\le f(x_k) - (\alpha/2)\left\|\nabla f(x_k)\right\|_2^2 \\ &\le f(x_k) - \rho\alpha \left\|\nabla f(x_k)\right\|_2^2,\end{aligned}$$

deoarece $\rho < 1/2$. Deci căutarea liniară cu backtracking se termină fie cu $\alpha = 1$, fie cu o valoare $\alpha \ge \beta/M$. Aceasta furnizează o margine inferioară asupra reducerii funcției de minimizat. În primul caz avem:

$$f(x_{k+1}) \le f(x_k) - \rho \left\|\nabla f(x_k)\right\|_2^2,$$

iar în al doilea caz:

$$f(x_{k+1}) \le f(x_k) - \left(\beta\rho/M\right)\left\|\nabla f(x_k)\right\|_2^2.$$

Împreună aceste inegalități dau:

$$f(x_{k+1}) \le f(x_k) - min\left\{\rho, \beta\rho/M\right\}\left\|\nabla f(x_k)\right\|_2^2.$$

Procedând analog ca în cazul căutării liniare exacte, adică scăzând din ambii membri f^* şi ținând seama că $\left\|\nabla f(x_k)\right\|_2^2 \ge 2m(f(x_k) - f^*)$ obținem:

$$f(x_{k+1}) - f^* \le \left(1 - min\left\{2m\rho, \frac{2\beta\rho m}{M}\right\}\right)\left(f(x_k) - f^*\right).$$

Deci:

$$f(x_k) - f^* \le c^k\left(f(x_0) - f^*\right),$$

unde

$$c = 1 - min\left\{2m\rho, \frac{2\beta\rho m}{M}\right\}. \tag{5.3.17}$$

Ca atare, la fel ca în cazul căutării liniare exacte, $f(x_k)$ converge la f^* cel puțin la fel de repede ca o serie geometrică cu o rată care depinde de marginea numărului de condiționare M/m a Hessianului. Aceasta înseamnă că algoritmul converge liniar.

În finalul acestei secţiuni menţionăm că alegerea normei utilizate pentru definirea direcţiei pasului descendent este importantă în stabilirea convergenţei algoritmului. Dacă considerăm norma euclidiană $\|.\|$, atunci, în punctul curent, direcţia pasului descendent este chiar direcţia negativă a gradientului, adică $d_k = -\nabla f(x_k)$. Cu alte cuvinte, în norma euclidiană metoda pasului descendent coincide cu metoda gradientului descendent. Acum, dacă considerăm norma pătratică $\|z\|_P = \left(z^T P z\right)^{1/2}$, unde P este o matrice simetrică şi pozitiv definită, atunci în raport cu această normă direcţia pasului descendent este dată de $d_k = -P^{-1}\nabla f(x_k)$. Interpretarea acestei situaţii este simplă: direcţia de căutare este chiar direcţia negativă a gradientului după schimbarea de coordonate $\overline{x} = P^{1/2}x$. Într-adevăr, utilizând această schimbare de coordonate atunci problema originală de minimizare a funcţiei f se poate rezolva prin rezolvarea problemei echivalente de minimizare a funcţiei $\overline{f} : R^n \to R$, unde $\overline{f}(\overline{x}) = f(P^{-1/2}\overline{x}) = f(x)$. Dacă aplicăm metoda pasului descendent funcţiei $\overline{f}$, atunci direcţia de căutare în punctul $\overline{x}$ (care corespunde punctului $x = P^{-1/2}\overline{x}$ pentru problema originală) este $\overline{d} = -\nabla\overline{f}(\overline{x}) = -P^{-1/2}\nabla f(x)$. Dar, această direcţie corespunde direcţiei $d = P^{-1/2}\left(-P^{-1/2}\nabla f(x)\right) = -P^{-1}\nabla f(x)$ pentru variabilele originale, adică metoda pasului descendent în norma pătratică dată de matricea P este aceeaşi ca metoda gradientului aplicată problemei după o schimbare de coordonate definită de matricea $P : \overline{x} = P^{1/2}x$. Acum, cunoaştem că metoda gradientului lucrează bine când numerele de condiţionare ale mulţimilor de nivel constant nu sunt mari şi lucrează prost când aceste numere sunt mari. Deci, când după o schimbare de coordonate mulţimile de nivel constant devin bine condiţionate, atunci metoda pasului descendent este eficientă. Această observaţie furnizează o idee de alegere a matricei P. Aceasta se alege astfel încât mulţimile de nivel constant ale lui f, transformate prin $P^{-1/2}$, să fie bine condiţionate. De exemplu, dacă în punctul de optim x^* se cunoaşte o aproximaţie $\hat{B}$ a Hessianului $\nabla^2 f(x^*)$, atunci o foarte bună alegere a lui P este $P = \hat{B}$, deoarece Hessianul lui $\overline{f}$ în punctul de optim este $\hat{B}^{-1/2}\nabla^2 f(x^*)\hat{B}^{-1/2} \cong I$, care are un număr de condiţionare mic.

În ***concluzie***, putem spune că:

- *Algoritmul pasului descendent are o convergenţă liniară, adică eroarea $f(x_k) - f^*$ tinde la zero ca o serie geometrică.*
- *Rata de convergenţă depinde foarte mult de numărul de condiţionare al Hessianului funcţiei de minimizat, care este necunoscut, sau de mulţimile de nivel constant ale acesteia. Convergenţa poate fi extrem de slabă chiar pentru probleme relativ bine condiţionate. Când numărul de condiţionare este mare,*

atunci algoritmul este aşa de lent convergent încât nu are nici o valoare practică.

- *Parametrii ρ şi β utilizaţi în căutarea liniară cu backtracking nu au un efect foarte mare asupra convergenţei algoritmului. Căutarea liniară exactă îmbunătăţeşte convergenţa, dar fără un efect semnificativ.*

5.4. METODA PASULUI DESCENDENT RELAXAT

Este foarte interesant să vedem ce se întâmplă dacă *modificăm aleator lungimea pasului* în intervalul $[0,1]$, adică considerăm iteraţii de tipul:

$$x_{k+1} = x_k + \theta_k \alpha_k d_k, \tag{5.4.1}$$

unde $d_k = -\nabla f(x_k)$, iar θ_k este o variabilă aleatoare uniform distribuită în intervalul $[0,1]$. Obţinem astfel un algoritm de *gradient descendent relaxat*. În acest caz, pentru exemplul 5.2.1, evoluţia erorii $\left| f(x_k) - f(x^*) \right|$ este ca în figura 5.4.1. Observăm că în acest caz, al funcţiilor pătratice pozitiv definite, algoritmul pasului descendent relaxat este superior celui clasic. O evoluţie ceva mai bună din punctul de vedere al numărului de iteraţii se obţine când θ_k este selectat ca o variabilă aleatoare uniform distribuită în intervalul $[0,2]$.

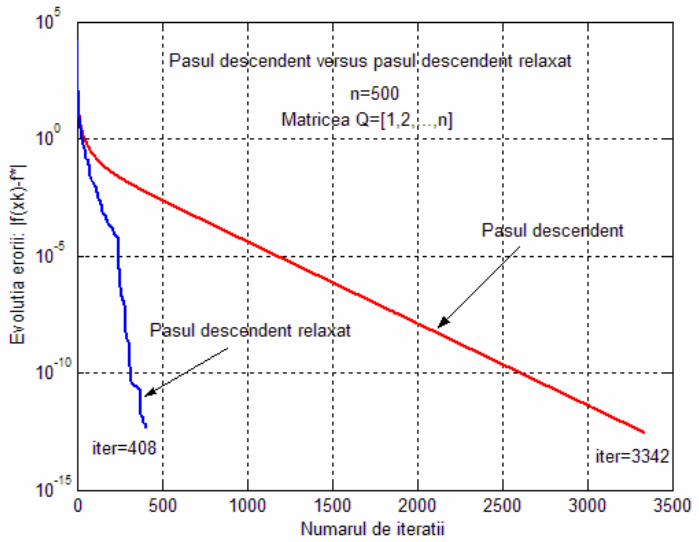

Fig. 5.4.1. Evoluţia erorii $\left| f(x_k) - f(x^*) \right|$ pentru algoritmul pasului descendent relaxat în comparaţie cu algoritmul pasului descendent.

Considerând diferite valori pentru n observăm comportarea acestor algoritmi. Tabelul 5.4.1 arată numărul de iterații corespunzător algoritmului pasului descendent şi a variantei sale relaxate, pentru situația în care $\theta_k \in [0,1]$ şi respectiv $\theta_k \in [0,2]$.

Tabelul 5.4.1. Numărul de iterații corespunzător algoritmului pasului descendent şi a versiunii sale relaxate.

n	**Pasul descendent**	**Pasul descendent relaxat** $\theta_k \in [0,1]$	**Pasul descendent relaxat** $\theta_k \in [0,2]$
1000	6682	812	860
2000	13358	950	1676
3000	20034	1627	1684
4000	26710	1407	1780
5000	33386	1615	1962
6000	40062	2198	1992
7000	46738	2684	1850
8000	534414	2440	2348
9000	60092	1765	3028
10000	66768	3005	3231

Aceasta ilustrează o foarte serioasă *limitare a selectării lungimii optime* a pasului de deplasare în algoritmul pasului descendent şi în acelaşi timp o anumită derută deoarece o modificare foarte simplă a lungimii pasului provoacă schimbări semnificative în comportarea acestuia. În fapt, pentru funcții pătratice pozitiv definite, în condiții foarte comode asupra lui $\theta_k \in [0,2]$ **,** putem stabili următorul rezultat de convergență a algoritmului pasului descendent relaxat [Raydan şi Svaiter, 2002].

Teorema 5.4.1. *Pentru problema*

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

unde $Q \in R^{n \times n}$ *este o matrice simetrică şi pozitiv definită, dacă şirul* $\theta_k \in [0,2]$ *are un punct de acumulare* $\theta^* \in [0,2]$, *atunci şirul* x_k *:*

$$x_{k+1} = x_k - \theta_k \alpha_k g_k \ ; \qquad \alpha_k = \frac{g_k^T g_k}{g_k^T Q g_k}, \qquad g_k = Qx_k,$$

generat de algoritmul pasului descendent relaxat, este convergent liniar la x^*.

Demonstrație. Observăm că pentru orice k funcția

$$\Phi_k(\theta) = f(x_k - \theta \alpha_k g_k)$$

este un polinom convex de ordinul doi care îşi atinge minimul în punctul $\theta = 1$. Mai mult, vedem că $\Phi_k(0) = \Phi_k(2)$ şi pentru orice $\theta \in [0,2]$, $\Phi_k(\theta) \le \Phi_k(0)$. Deci pentru toți k, $f(x_{k+1}) \le f(x_k)$. Deoarece f este mărginită inferior, rezultă că

$$\lim_{k\to\infty}(f(x_k) - f(x_{k+1})) = 0.$$

Un calcul simplu arată că:

$$f(x_k - \theta\alpha_k g_k) = f(x_k) - \left(\theta - \frac{1}{2}\theta^2\right)\frac{\left(g_k^T g_k\right)^2}{g_k^T Q g_k}.$$

Observăm că pentru $\theta \in [0,2]$, $\theta - \theta^2/2 \ge 0$. Deci $f(x_k)$ este un şir necrescător convergent la $f(x^*)$.

Există un $\xi \in (0,1)$ astfel încât $\xi < \theta^* < 2 - \xi$. Deoarece $\Phi_k(\theta)$ este convexă pentru orice $\theta \in (\xi, 2-\xi)$ rezultă că $\Phi_k(\theta) < \Phi_k(\xi)$. Dar,

$$\Phi_k(\xi) < \Phi_k(0) - \xi\left(\Phi_k(0) - \Phi_k(1)\right).$$

Cum

$$\Phi_k(0) - \Phi_k(1) = \frac{1}{2}\frac{\left(g_k^T g_k\right)^2}{g_k^T Q g_k},$$

rezultă că

$$\Phi_k(\xi) < \Phi_k(0) - \frac{\xi}{2}\frac{\left(g_k^T g_k\right)^2}{g_k^T Q g_k}.$$

Deci

$$\Phi_k(0) - \Phi_k(\xi) > \frac{\xi}{2}\frac{\left(g_k^T g_k\right)^2}{g_k^T Q g_k} \ge \frac{\xi}{2}\frac{g_k^T g_k}{\lambda_{\max}},$$

adică

$$f(x_k) - f(x_k - \xi\alpha_k g_k) > \frac{\xi}{2\lambda_{\max}}\|g_k\|_2.$$

Dar, $f(x_k) - f(x_k - \xi\alpha_k g_k) \to 0$, ceea ce arată că $g_k \to 0$ şi deci $x \to x^*$, soluția problemei. Cum $f(x_k)$ este un şir necrescător, acesta tinde la $f(x^*)$ ceea ce implică $x \to x^*$. ■

Teorema 5.4.1 arată că metoda gradientului descendent cu relaxare merită a fi considerată ca o îmbunătățire a metodei gradientului descendent clasice. Totuşi, în anumite cazuri particulare selecția lungimii optime este cea mai bună alternativă. De exemplu, dacă direcția de căutare este chiar un vector propriu atunci metoda clasică furnizează soluția într-un singur pas. În acest caz relaxarea nu este de nici un folos. Dar această situație este foarte rară, aşa încât relaxarea poate fi o tehnică foarte bună de accelerare a metodei gradientului descendent.

Următorul exemplu arată comportarea algoritmului pasului descendent echipat cu căutarea liniară cu backtracking.

Exemplul 5.4.1. Fie funcția:

$$f(x) = \sum_{i=1}^{n} i x_i^2 + \frac{1}{100}\left(\sum_{i=1}^{n} x_i\right)^2 .$$

Considerând $x_0 = [0.5, 0.5, \ldots, 0.5]$, $\rho = 0.0001$ şi $\beta = 0.8$ în procedura de căutare liniară cu backtracking (Algoritmul 4.3.2) şi cu criteriul de oprire a iterațiilor

$$\|\nabla f(x_k)\| \le \varepsilon_g \quad \text{cu} \quad \varepsilon_g = 10^{-6} ,$$

atunci pentru $n = 100$, evoluția erorii $|f(x_k) - f^*|$ dată de algoritmul pasului descendent în comparație cu a celei corespunzătoare versiunii relaxate este dată în figura 5.4.2. Pasul descendent necesită 721 de iterații, față de numai 227 de iterații cât necesită varianta relaxată a algoritmului pasului descendent. Din figura 5.4.2 intuim imediat că şi versiunea relaxată a algoritmului pasului descendent are o convergență liniară.

Tabelul 5.4.2 arată numărul de iterații şi lungimea medie a pasului corespunzător algoritmului pasului descendent (PD) şi a versiunii relaxate (PDR) cu $\theta_k \in [0,1]$, pentru diferite valori ale lui n.

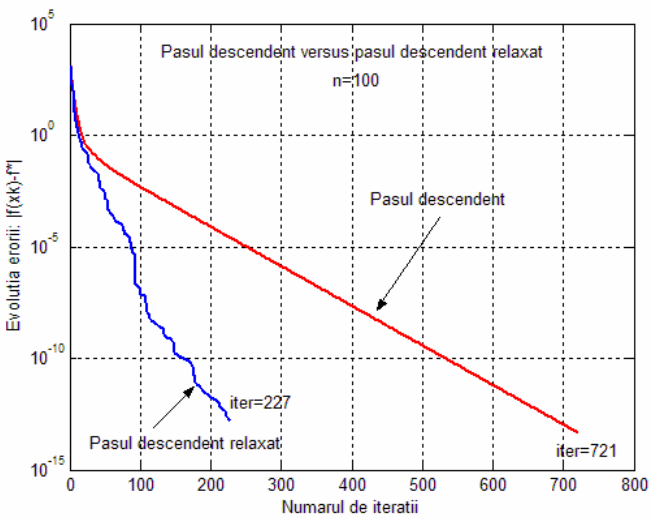

Fig. 5.4.2. Pasul descendent versus pasul descendent relaxat.

Din tabelul 5.4.2 se vede că în cazul algoritmului pasului descendent relaxat, lungimea medie a pasului de-a lungul direcției negative a gradientului este cu un

ordin de mărime mai mare decât lungimea medie generată în cazul algoritmului pasului descendent.

Tabelul 5.4.2. Numărul de iterații şi lungimea medie a pasului de deplasare pentru algoritmul PD şi PDR echipate cu backtracking.

	PD		PDR	
n	**# iter**	**pasul mediu**	**# iter**	**pasul mediu**
500	3601	0.00200632	584	0.0215758
1000	7207	0.00100033	613	0.0169619
2000	15258	0.00050113	1171	0.0102804
3000	21542	0.00033496	2045	0.0069395
4000	29540	0.00025162	1910	0.0056208
5000	36948	0.00020132	1513	0.0067056

Observăm imediat că şi în cazul funcțiilor convexe versiunea relaxată a algoritmului pasului descendent este superioară celei clasice.

Extensia acestui algoritm de pas descendent relaxat la cazul funcțiilor generale este imediată [Andrei, 2004b]. Într-adevăr, următorul exemplu arată comportarea variantei relaxate a algoritmului pasului descendent: $x_{k+1} = x_k + \theta_k \alpha_k d_k$, unde $d_k = -\nabla f(x_k)$, θ_k este o variabilă aleatoare uniform distribuită în intervalul $[0,1]$, iar α_k este determinat prin backtracking.

Exemplul 5.4.2. Fie funcția:

$$f(x) = (x_1 - 3)^2 + \sum_{i=2}^{n} \left(x_1 - 3 - 2(x_1 + x_2 + \cdots + x_i)^2 \right)^2 .$$

Considerând punctul inițial $x_0 = [0.001, \ldots, 0.001]$, şi criteriul de oprire a iterațiilor $\|\nabla f(x_k)\| \le \varepsilon_g$, unde $\varepsilon_g = 10^{-6}$, atunci numărul de iterații corespunzător celor doi algoritmi echipați cu backtracking ($\rho = 0.0001$ şi $\beta = 0.8$), pentru diferite valori ale lui n, este ca în tabelul 5.4.3.

Tabelul 5.4.3. Numărul de iterații şi lungimea medie a pasului de deplasare pentru algoritmul PD şi PDR echipate cu backtracking.

	PD		PDR	
n	**# iter**	**pasul mediu**	**# iter**	**pasul mediu**
10	186997	0.103413	40462	0.327121
20	253806	0.047808	101595	0.217972
30	410108	0.031707	105885	0.172842
40	780362	0.024489	122293	0.146184
50	829749	0.020759	144316	0.126295

Aceste exemple numerice arată *limitele algoritmului pasului descendent*. Observăm că o modificare multiplicativă, foarte simplă, a lungimii pasului prin intermediul unei variabile aleatoare, uniform distribuită în intervalul $[0,1]$,

provoacă schimbări majore în comportarea algoritmului gradientului descendent. Aceasta ilustrează o robustețe foarte scăzută a algoritmul pasului descendent la variația lungimii pasului. Procedura de backtracking, precum şi cea bazată pe căutarea liniară exactă, limitează foarte mult performanța algoritmului gradientului descendent. În fapt, pentru cazul funcțiilor tare convexe se poate demonstra următoarea teoremă.

Teorema 5.4.2. [Raydan şi Svaiter, 2002] *Dacă şirul* θ_k *are un punct de acumulare* $\theta^* \in (0,1)$, *atunci şirul* x_k *generat de algoritmul pasului descendent relaxat converge la* x^*.

Demonstrație. Să considerăm funcția $\varphi(\theta) = f(x_k - \theta\alpha_k g_k)$, unde $g_k = \nabla f(x_k)$. Putem scrie:

$$f(x_k - \theta\alpha_k g_k) = f(x_k) - \theta\alpha_k g_k^T g_k + \frac{1}{2}\theta^2\alpha_k^2 g_k^T \nabla^2 f(x_k) g_k.$$

Deoarece funcția f este tare convexă, urmează că $\varphi(\theta)$ este o funcție convexă şi $\varphi(0) = f(x_k)$. Din tare convexitatea lui f urmează că:

$$f(x_k - \theta\alpha_k g_k) \le f(x_k) - \left(\theta - \frac{M\alpha_k}{2}\theta^2\right)\alpha_k \|g_k\|_2^2.$$

Dar, $\theta - \frac{M\alpha_k}{2}\theta^2$ este o funcție concavă şi nenegativă pe $\left(0, 2/M\alpha_k\right)$ şi are valoarea maximă $1/2M\alpha_k$ în $1/M\alpha_k$. Deci $f(x_{k+1}) \le f(x_k)$ pentru toți k. Deoarece f este mărginită inferior, rezultă că:

$$\lim_{k\to\infty}\left(f(x_k) - f(x_{k+1})\right) = 0. \ \blacksquare$$

În continuare să presupunem că f este tare convexă şi că mulțimea de nivel constant $S = \left\{x \in dom\, f : f(x) \le f(x_0)\right\}$ este închisă. Atunci putem demonstra următoarea teoremă care arată convergența liniară a algoritmului pasului descendent relaxat echipat cu căutarea liniară cu backtracking.

Teorema 5.4.3. *Pentru funcții tare convexe algoritmul pasului descendent relaxat prin şirul* θ_k, *unde* θ_k *este o variabilă aleatoare uniform distribuită în intervalul* $[0,1]$, *care utilizează căutarea liniară cu backtracking (4.3.6) este liniar convergent şi*

$$f(x_k) - f^* \le \left(\prod_{i=0}^{k-1} c_i\right)\left(f(x_0) - f^*\right), \tag{5.4.2}$$

unde

$$c_i = 1 - min\{2m\rho\theta_i \ , 2m\rho\theta_i\beta / M\} < 1. \qquad (5.4.3)$$

Demonstrație. Considerăm $0 < \theta < 1$, atunci

$$f(x_k - \theta\alpha_k g_k) \le f(x_k) - \left(\theta - \frac{M\alpha_k}{2}\theta^2\right)\alpha_k \|g_k\|_2^2.$$

Notăm că $\theta - \frac{M\alpha_k}{2}\theta^2$ este o funcție concavă şi că pentru toți $0 \le \theta \le \frac{1}{M\alpha_k}$,

$$\theta - \frac{M\alpha_k}{2}\theta^2 \ge \frac{\theta}{2}.$$

Deci

$$f(x_k - \theta\alpha_k g_k) \le f(x_k) - \frac{\theta}{2}\alpha_k \|g_k\|_2^2 \le f(x_k) - \rho\theta\alpha_k \|g_k\|_2^2,$$

deoarece $\rho \le 1/2$. Ca atare, căutarea liniară cu backtracking se termină cu $\alpha_k = 1$ sau cu o valoare $\alpha_k \ge \beta / M$. Cu aceasta, la pasul k se obține o margine inferioară a descreşterii funcției. În primul caz avem:

$$f(x_{k+1}) \le f(x_k) - \rho\theta_k \|g_k\|_2^2,$$

iar în al doilea:

$$f(x_{k+1}) \le f(x_k) - \rho\theta_k \frac{\beta}{M}\|g_k\|_2^2.$$

Deci

$$f(x_{k+1}) \le f(x_k) - min\{\rho\theta_k \ , \rho\theta_k\beta / M\}\|g_k\|_2^2.$$

De unde obținem:

$$f(x_{k+1}) - f^* \le f(x_k) - f^* - min\{\rho\theta_k \ , \rho\theta_k\beta / M\}\|g_k\|_2^2.$$

Dar $\|g_k\|_2^2 \ge 2m\left(f(x_k) - f^*\right)$. Combinând aceste relații găsim:

$$f(x_{k+1}) - f^* \le \left(1 - min\{2m\rho\theta_k \ , 2m\rho\theta_k\beta / M\}\right)\left(f(x_k) - f^*\right).$$

Notând: $c_k = 1 - min\{2m\rho\theta_k \ , 2m\rho\theta_k\beta / M\}$ urmează că pentru orice $k = 0,1,\ldots$

$$f(x_{k+1}) - f^* \le c_k\left(f(x_k) - f^*\right),$$

care demonstrează formula corespunzătoare suboptimalității de la pasul k. Deoarece $c_k < 1$, şirul $f(x_k)$ converge la f^* ca o serie geometrică cu un factor care parțial depinde de marginea asupra numărului de condiționare M / m, parametrii de backtracking ρ şi β (vezi algoritmii 4.3.1 sau 4.3.2), şirul θ_k al numerelor aleatoare uniform distribuite în intervalul [0,1], precum şi de suboptimalitatea inițială. Deci algoritmul pasului descendent relaxat este liniar convergent. ■

Studiu Numeric (PDR versus PD)[1]

Următorul set de experimente numerice ilustrează comportarea algoritmului PDR în comparaţie cu algoritmul clasic al pasului descendent PD. În acest sens am considerat un set de 32 de probleme de optimizare fără restricţii şi pentru fiecare problemă am considerat 10 instanţieri în care numărul de variabile n ia valorile: 100,200,...,1000. Deci în total am rezolvat 320 de probleme. Atât algoritmul PD cât şi PDR utilizează acelaşi test de oprire a iteraţiilor dat de:

$$\|\nabla f(x_k)\|_2 \le \varepsilon_g \quad \text{sau} \quad \alpha_k \left| g_k^T d_k \right| \le \varepsilon_f \left| f(x_{k+1}) \right|,$$

unde $\varepsilon_g = 10^{-6}$ şi $\varepsilon_f = 10^{-16}$. În procedura de căutare liniară prin backtracking se utilizează aceleaşi valori pentru parametrii ρ şi β: $\rho = 0.0001$, $\beta = 0.8$. Problemele de test sunt din colecţia CUTE [Bongartz, Conn, Gould şi Toint, 1995], împreună cu alte probleme de optimizare fără restricţii construite în acest scop.

Studiul numeric a fost făcut în următorul context. Fie f_i^{PD} şi f_i^{PDR} valorile funcţiei de minimizat i în punctul de optim calculate de algoritmul PD, respectiv de PDR, pentru $i = 1,...,320$. Zicem că în cazul particular al problemei i algoritmul PDR este mai performant decât algoritmul PD dacă

$$\left| f_i^{PDR} - f_i^{PD} \right| < 10^{-3}$$

şi numărul de iteraţii, sau numărul de evaluări ale funcţiei şi gradientului sau timpul de calcul al lui PDR este mai mic decât numărul de iteraţii sau numărul de evaluări ale funcţiei şi gradientului sau timpul de calcul corespunzător lui PD. Din cele 320 de probleme rezolvate de PDR şi PD, numai 280 satisfac criteriul de mai sus.

Tabelul 5.4.4 conţine numărul de probleme din cele 280 pentru care PDR versus PD a realizat numărul minim de iteraţii (#iter), numărul minim de evaluări ale funcţiei şi gradientului (#fg), respectiv timpul de calcul CPU minim (CPU).

Tabelul 5.4.4. Performanţa lui PDR versus PD. 280 de probleme

	PDR	**PD**	**=**
#iter	224	56	0
#fg	229	51	0
CPU	229	51	0

De exemplu, referitor la numărul de iteraţii, PDR a fost mai bun în 224 de probleme (adică a realizat numărul minim de iteraţii în 224 de probleme) faţă de PD care a realizat numărul minim de iteraţii doar în 56 de probleme. PDR şi PD nu au rezolvat nici o problemă în acelaşi număr de iteraţii. Din acest tabel vedem clar că PDR este mult mai performant decât PD. Performanţele acestor algoritmi au fost vizualizate utilizând profilul Dolan-Moré [2002]. Pentru fiecare algoritm reprezentăm grafic fracţia din numărul de probleme pentru care algoritmul a fost cu un anumit factor mai bun în privinţa numărului de iteraţii sau a timpului de calcul.

[1] Programul STEPDES.FOR din CD-ul anexat lucrării implementează algoritmii de pas descendent şi de pas descendent relaxat.

Partea din stânga din figurile 5.4.3a şi 5.4.3b arată procentajul din cele 280 de probleme pentru care un algoritm este mai bun. Partea din dreapta arată procentajul problemelor care au fost rezolvate cu succes de fiecare algoritm. Vedem că ambii algoritmi au rezolvat cu succes toate cele 280 de probleme.

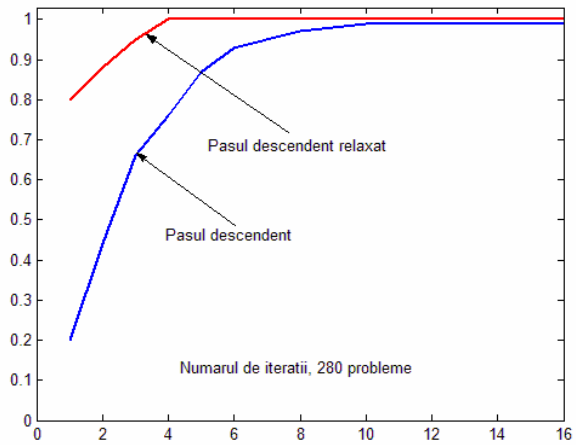

Fig. 5.4.3a. PDR versus PD. Numărul de iterații.

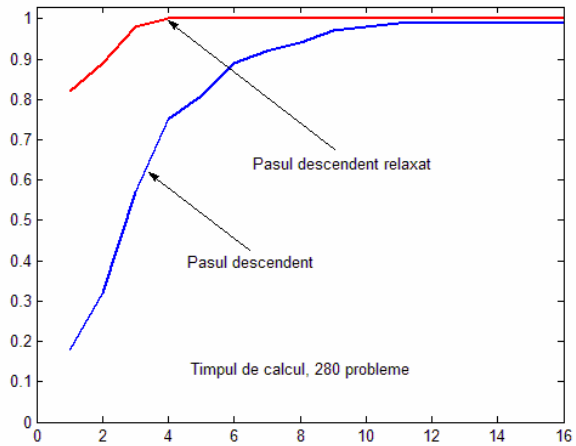

Fig. 5.4.3b. PDR versus PD. Timpul de calcul.

Curba superioară corespunde algoritmului care a rezolvat cele mai multe probleme într-un număr de iterații (figura 5.4.3a) sau într-un timp de calcul (figura 5.4.3b) care este un multiplu τ al celui mai bun număr de iterații sau respectiv al celui mai bun timp de calcul. Deoarece această curbă corespunde algoritmului pasului descendent relaxat, rezultă clar că acesta este de departe superior algoritmul pasului descendent clasic.

În continuare să vedem o altă analiză dată de *performanța relativă a numărului de iterații* (iter), sau a *numărului de evaluări ale funcției* de minimizat şi a gradientului acesteia (fg), sau a *timpului de calcul* (time) pentru algoritmul PDR sau PD. Performanța relativă corespunzătoare numărului de iterații este definită sub forma:

$$iter^i_{PDR-PD} = -\log_2\left(\frac{iter^i_{PDR}}{iter^i_{PD}}\right),$$

unde $iter^i_{PDR}$ este numărul de iterații corespunzător algoritmului PDR pentru a rezolva a $i-$a problemă. $iter^i_{PD}$ are o interpretare similară, pentru algoritmul PD. Vedem imediat că semnul lui $iter^i_{PDR-PD}$ indică algoritmul câştigător. Într-adevăr, *în toate cazurile în care* $iter^i_{PDR-PD}$ *este pozitiv rezultă că algoritmul PDR este superior lui PD din punctul de vedere al numărului de iterații.* Pentru numărul de evaluări ale funcției şi gradientului sau în ceea ce priveşte timpul de calcul, performanța relativă se defineşte analog. Observăm că acest indicator de performanță relativă este exact profilul de performanță definit de Dolan şi Moré, pentru $\tau = 1$. În figurile 5.4.4a şi 5.4.4b prezentăm valorile lui $iter^i_{PDR-PD}$, respectiv $time^i_{PDR-PD}$, pentru acest set de 280 de probleme, unde problemele au fost plasate în ordinea descrescătoare lui $\left|iter^i_{PDR-PD}\right|$, şi respectiv ale lui $\left|time^i_{PDR-PD}\right|$.

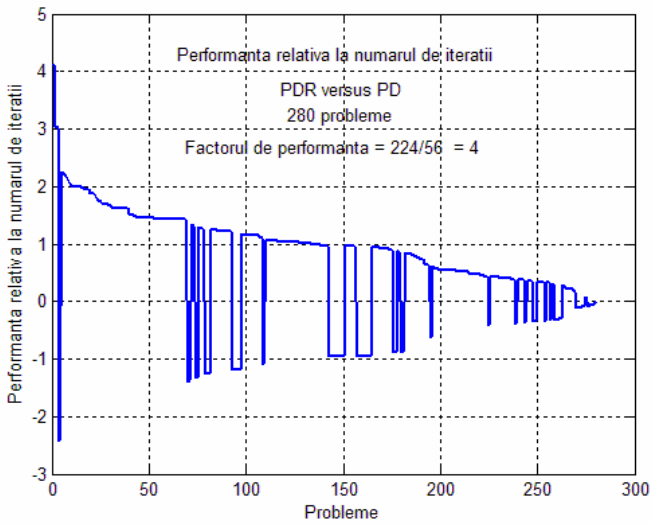

Fig. 5.4.4a. Performanța relativă la numărul de iterații: $iter_{PDR-PD}$.

Numărul de probleme pentru care $iter^i_{PDR-PD}$ este pozitiv este 224, iar numărul de probleme pentru care $iter^i_{PDR-PD}$ este negativ este 56. Vedem imediat că *factorul de*

performanţă relativ la numărul de iteraţii, care se defineşte ca raportul dintre numărul de probleme pentru care $iter^i_{PDR-PD}$ este pozitiv şi numărul de probleme pentru care $iter^i_{PDR-PD}$ este negativ este egal cu 4. Numărul de probleme pentru care $time^i_{PDR-PD}$ este pozitiv este 229, iar numărul de probleme pentru care $time^i_{PDR-PD}$ este negativ este 51.

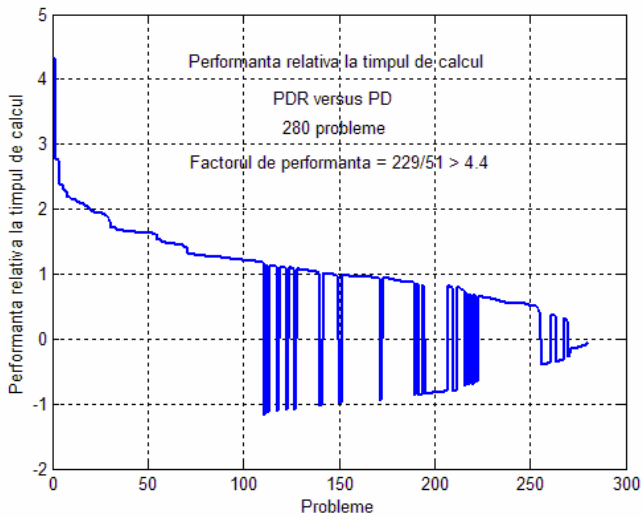

Fig. 5.4.4b. Performanţa relativă la timpul de calcul $time^i_{PDR-PD}$.

Vedem că *factorul de performanţă relativ la timpul de calcul* este mai mare decât 4.49, adică numărul de probleme rezolvare de algoritmul PDR într-un timp mai bun este de 4.49 ori mai mare decât cel rezolvat de algoritmul PD.

5.5. ALGORITMUL PASULUI DESCENDENT ACCELERAT CU BACKTRACKING

Algoritmul pasului descendent se dovedeşte a fi eficient pentru funcţii bine condiţionate, dar pentru funcţii prost condiţionate acesta este excesiv de lent, neavând nici o valoare practică. Chiar pentru funcţii pătratice algoritmul pasului descendent cu căutare liniară exactă se comportă prost în condiţiile în care numărul de condiţionare al matricei se deteriorează (creşte). După cum cunoaştem, în punctul curent x_k direcţia negativă a gradientului este cea mai bună direcţie de căutare a minimului funcţiei f , şi aceasta este direcţia pasului descendent. Totuşi, de îndată ce ne deplasăm în această direcţie, ea încetează de a mai fi cea mai bună şi continuă să se deterioreze până când devine ortogonală pe $-\nabla f(x_k)$. Adică, algoritmul începe să facă paşi din ce în ce mai mici neînregistrând nici un progres către minim. Acesta este principalul defect al metodei, lungimea paşilor de

deplasare este prea mică, adică pe segmentul de linie care unește x_k și x_{k+1} se găsesc puncte z_k în care $-\nabla f(z_k)$ furnizează o direcție de căutare mai bună decât $-\nabla f(x_{k+1})$.

În această secțiune prezentăm algoritmul *pasului descendent accelerat cu backtracking* pentru rezolvarea problemei (5.0.1) [Andrei 2006]. Considerând punctul inițial x_0 putem imediat calcula $f_0 = f(x_0)$, $g_0 = \nabla f(x_0)$ și prin backtracking α_0. Cu acestea, la următoarea iterație se calculează $x_1 = x_0 - \alpha_0 g_0$ unde din nou f_1 și g_1 pot fi calculate. Acum, la iterația $k = 1,2,\ldots$ cunoaștem x_k, f_k și g_k. Utilizând procedura de backtracking se determină lungimea pasului α_k cu care se calculează punctul $z = x_k - \alpha_k g_k$. Prin backtracking obținem un $\alpha_k \in (0,1]$ astfel încât:

$$f(z) = f(x_k - \alpha_k g_k) \le f(x_k) - \rho \alpha_k g_k^T g_k.$$

Cu acestea să introducem algoritmul pasului descendent accelerat prin intermediul următoarei relații de recurență:

$$x_{k+1} = x_k - \theta_k \alpha_k g_k, \tag{5.5.1}$$

unde $\theta_k > 0$ este un parametru care urmează a fi determinat cu scopul de a îmbunătăți comportarea algoritmului pasului descendent. Avem:

$$f(x_k - \alpha_k g_k) = f(x_k) - \alpha_k g_k^T g_k + \frac{1}{2}\alpha_k^2 g_k^T \nabla^2 f(x_k) g_k + o\left(\|\alpha_k g_k\|^2\right).$$

Pe de altă parte, pentru $\theta > 0$, avem:

$$f(x_k - \theta\alpha_k g_k) = f(x_k) - \theta\alpha_k g_k^T g_k + \frac{1}{2}\theta^2\alpha_k^2 g_k^T \nabla^2 f(x_k) g_k + o\left(\|\theta\alpha_k g_k\|^2\right).$$

Putem scrie:

$$f(x_k - \theta\alpha_k g_k) = f(x_k - \alpha_k g_k) + \Psi_k(\theta), \tag{5.5.2}$$

unde

$$\begin{aligned}\Psi_k(\theta) = (1-\theta)\alpha_k g_k^T g_k - \frac{1}{2}(1-\theta^2)\alpha_k^2 g_k^T \nabla^2 f(x_k) g_k \\ +\theta^2\alpha_k o\left(\alpha_k \|g_k\|^2\right) - \alpha_k o\left(\alpha_k \|g_k\|^2\right).\end{aligned} \tag{5.5.3}$$

Notăm:

$$\begin{aligned} a_k &= \alpha_k g_k^T g_k \ge 0, \\ b_k &= \alpha_k^2 g_k^T \nabla^2 f(x_k) g_k, \\ \varepsilon &= o\left(\alpha_k \|g_k\|^2\right). \end{aligned}$$

Observăm că pentru funcții convexe $b_k \ge 0$.

Deci

$$\Psi_k(\theta) = (1-\theta)a_k - \frac{1}{2}(1-\theta^2)b_k + \theta^2\alpha_k\varepsilon - \alpha_k\varepsilon. \tag{5.5.4}$$

Acum, vedem că $\Psi'_k(\theta) = (b_k + 2\alpha_k\varepsilon)\theta - a_k$ şi $\Psi'_k(\theta_m) = 0$, unde

$$\theta_m = \frac{a_k}{b_k + 2\alpha_k\varepsilon}.$$

Observăm că $\Psi'_k(0) = -a_k < 0$. Deci, $\Psi_k(\theta)$ este o funcție pătratică convexă cu valoarea minimă în punctul θ_m şi

$$\Psi_k(\theta_m) = -\frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)} \le 0.$$

Considerând $\theta = \theta_m$ în (5.5.2), şi $b_k \ge 0$, vedem că pentru orice k

$$f(x_k - \theta_m\alpha_k g_k) = f(x_k - \alpha_k g_k) - \frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)} \le f(x_k - \alpha_k g_k),$$

care este o posibilă îmbunătățire a valorilor funcției f, (când $a_k - (b_k + 2\alpha_k\varepsilon) \ne 0$).

Deci, utilizând această simplă modificare multiplicativă a lungimii pasului α_k sub forma $\theta_k\alpha_k$, unde $\theta_k = \theta_m = a_k/(b_k + 2\alpha_k\varepsilon)$, şi ținând seama de condiția de backtracking $f(x_k - \alpha_k g_k) \le f(x_k) - \rho\alpha_k g_k^T g_k$ (vezi algoritmul 4.3.2) obținem:

$$f(x_{k+1}) = f(x_k - \theta_k\alpha_k g_k) \le f(x_k) - \rho\alpha_k g_k^T g_k - \frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)}$$

$$= f(x_k) - \left[\rho a_k + \frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)}\right] \le f(x_k), \qquad (5.5.5)$$

deoarece

$$\rho a_k + \frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)} \ge 0,$$

unde $\rho \in (0, 1/2)$. Observăm că dacă $a_k < b_k$ atunci

$$\frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)} > \frac{(a_k - b_k)^2}{2b_k}$$

şi din (5.5.5) obținem:

$$f(x_{k+1}) \le f(x_k) - \left[\rho a_k + \frac{(a_k - (b_k + 2\alpha_k\varepsilon))^2}{2(b_k + 2\alpha_k\varepsilon)}\right]$$

$$< f(x_k) - \left[\rho a_k + \frac{(a_k - b_k)^2}{2b_k}\right] \le f(x_k). \qquad (5.5.6)$$

Deci, neglijând contribuția termenilor de ordin superior ε încă obținem o îmbunătățire a valorilor funcției de minimizat.

Pentru a stabili algoritmul trebuie să determinăm o cale comodă de calcul a lui b_k . Pentru aceasta, în punctul $z = x_k - \alpha_k g_k$ avem:

$$f(z) = f(x_k - \alpha_k g_k) = f(x_k) - \alpha_k g_k^T g_k + \frac{1}{2}\alpha_k^2 g_k^T \nabla^2 f(\tilde{x}_k) g_k,$$

unde $\tilde{x}_k$ este un punct de pe segmentul de linie care uneşte x_k şi z. Pe de altă parte, în punctul $x_k = z + \alpha_k g_k$ avem:

$$f(x_k) = f(z + \alpha_k g_k) = f(z) + \alpha_k g_z^T g_k + \frac{1}{2}\alpha_k^2 g_k^T \nabla^2 f(\bar{x}_k) g_k,$$

unde $g_z = \nabla f(z)$ şi $\bar{x}_k$ este un punct de pe segmentul de linie care uneşte x_k şi z. Având în vedere caracterul local al căutării şi că distanţa dintre x_k şi z este mică, putem considera $\tilde{x}_k = \bar{x}_k = x_k$. Cu acestea, adunând egalităţile de mai sus obţinem:

$$b_k = \alpha_k^2 g_k^T \nabla^2 f(x_k) g_k = -\alpha_k y_k^T g_k, \tag{5.5.7}$$

unde $y_k = g_z - g_k$.

Deoarece de-a lungul iteraţiilor algoritmului pasului descendent $g_k \to 0$ putem neglija contribuţia lui ε şi să considerăm $\theta_k = \theta_m = a_k / b_k$. Cu acestea se poate prezenta următorul algoritm de pas descendent accelerat.

Algoritmul 5.5.1. (Algoritmul pasului descendent accelerat)

Pasul 1.	Se consideră un punct iniţial $x_0 \in dom f$ şi se calculează: $f_0 = f(x_0)$ şi $g_0 = \nabla f(x_0)$. Se pune $k = 0$.
Pasul 2.	Utilizând căutarea liniară cu backtracking se determină lungimea pasului α_k.
Pasul 3.	Se calculează: $z = x_k - \alpha_k g_k$, $g_z = \nabla f(z)$ şi $y_k = g_z - g_k$.
Pasul 4.	Se calculează: $a_k = \alpha_k g_k^T g_k$, $b_k = -\alpha_k y_k^T g_k$ şi $\theta_k = a_k / b_k$.
Pasul 5.	Se actualizează variabilele: $x_{k+1} = x_k - \theta_k \alpha_k g_k$ şi se calculează f_{k+1} şi g_{k+1}.
Pasul 6.	Se testează un criteriu de oprire a iteraţiilor. De exemplu, dacă $\lVert g_{k+1} \rVert \le \varepsilon$, unde $0 < \varepsilon \ll 1$ este o toleranţă de terminare a iteraţiilor, atunci stop; altfel se pune $k = k + 1$ şi se continuă cu pasul 2. ♦

Algoritmul pasului descendent (PD) se poate obţine imediat din algoritmul pasului descendent accelerat (PDA) prin evitarea paşilor 3 şi 4 şi considerarea valorii $\theta_k = 1$ în pasul 5 unde variabilele sunt actualizate.

Observăm că dacă $a_k > b_k$, atunci $\theta_k > 1$. În acest caz $\theta_k \alpha_k > \alpha_k$ şi este posibil ca $\theta_k \alpha_k \le 1$ sau $\theta_k \alpha_k > 1$, adică este posibil ca lungimea pasului $\theta_k \alpha_k$ să

fie mai mare decât 1. Pe de altă parte, dacă $a_k \le b_k$, atunci $\theta_k \le 1$. În acest caz $\theta_k \alpha_k \le \alpha_k \le 1$, adică lungimea pasului $\theta_k \alpha_k$ este redusă. Deci, dacă $a_k \ne b_k$, atunci $\theta_k \ne 1$ şi lungimea pasului α_k calculată prin backtracking va fi modificată, prin creşterea ei sau reducerea ei cu factorul θ_k, astfel evitând ca algoritmul să facă paşi ortogonali de-a lungul iterațiilor.

Neglijând ε, din (5.5.4), vedem că dacă $a_k \le b_k / 2$, atunci $\Psi_k(0) = a_k - b_k / 2 \le 0$ şi $\theta_k < 1$. Pentru orice $\theta \in [0,1]$, $\Psi_k(\theta) \le 0$. În consecință pentru orice $\theta \in (0,1)$, $f(x_k - \theta\alpha_k g_k) < f(x_k)$. În acest caz, pentru orice $\theta \in [0,1]$, $\theta\alpha_k \le \alpha_k$. Totuşi în algoritmul 5.5.1 am selectat $\theta_k = \theta_m$ ca punctul care realizează valoarea minimă a lui $\Psi_k(\theta)$. Remarcăm imediat că în cazul algoritmului de pas descendent relaxat am selectat parametrul θ_k în mod aleatoriu din intervalul $(0,1)$. Esența accelerării pasului descendent constă tocmai în selectarea acelei valori a lui θ_k care realizează minimul valorii funcției $\Psi_k(\theta)$.

Teorema 5.5.1. *Presupunem că f este o funcție tare convexă pe mulțimea de nivel constant $S = \{x : f(x) \le f(x_0)\}$. Atunci, şirul x_k generat de algoritmul PDA converge liniar la x^*.*

Demonstrație. Din (5.5.5) avem că $f(x_{k+1}) \le f(x_k)$, pentru toți k. Deoarece f este mărginită inferior, rezultă că

$$\lim_{k\to\infty} \left(f(x_k) - f(x_{k+1}) \right) = 0.$$

Deoarece f este tare convexă, atunci există constantele pozitive m şi M astfel încât $mI \le \nabla^2 f(x) \le MI$ pe mulțimea de nivel constant S. Presupunem că $x_k - \alpha g_k \in S$ şi $x_k - \theta_m \alpha g_k \in S$, pentru $0 < \alpha \le 1$.

Presupunând că $a_k < b_k$, atunci

$$f(x_k - \theta_m \alpha g_k) \le f(x_k - \alpha g_k) - \frac{(a_k - b_k)^2}{2b_k}.$$

Dar, din proprietatea de tare convexitate avem următoarea margine superioară pătratică asupra lui $f(x_k - \alpha g_k)$:

$$f(x_k - \alpha g_k) \le f(x_k) - \alpha \|g_k\|_2^2 + \frac{M\alpha^2}{2} \|g_k\|_2^2.$$

Observăm că pentru $0 \le \alpha \le 1/M$, $-\alpha + M\alpha^2/2 \le -\alpha/2$ care urmează din convexitatea lui $-\alpha + M\alpha^2/2$. Cu acest rezultat obținem:

$$f(x_k - \alpha g_k) \le f(x_k) - \alpha \|g_k\|_2^2 + \frac{M\alpha^2}{2} \|g_k\|_2^2$$

$$\leq f(x_k) - \frac{\alpha}{2}\|g_k\|_2^2 \leq f(x_k) - \rho\alpha\|g_k\|_2^2,$$

deoarece $\rho \leq 1/2$.

Căutarea liniară cu backtracking se termină fie cu $\alpha = 1$, fie cu o valoare $\alpha \geq \beta / M$. Aceasta furnizează o margine inferioară asupra reducerii valorilor funcției f. Pentru $\alpha = 1$ avem:

$$f(x_k - \alpha g_k) \leq f(x_k) - \rho\|g_k\|_2^2$$

şi pentru $\alpha \geq \beta / M$

$$f(x_k - \alpha g_k) \leq f(x_k) - \frac{\rho\beta}{M}\|g_k\|_2^2 .$$

Deci, pentru $0 \leq \alpha \leq 1/M$, întotdeauna

$$f(x_k - \alpha g_k) \leq f(x_k) - min\left\{\rho, \frac{\rho\beta}{M}\right\}\|g_k\|_2^2. \qquad (5.5.8)$$

Pe de altă parte,

$$\frac{(a_k - b_k)^2}{2b_k} \geq \frac{\left(\alpha\|g_k\|_2^2 - \alpha^2 M\|g_k\|_2^2\right)^2}{2\alpha^2 M\|g_k\|_2^2} = \frac{(1-\alpha M)^2}{2M}\|g_k\|_2^2.$$

Acum, ca mai sus, pentru $\alpha = 1$:

$$\frac{(a_k - b_k)^2}{2b_k} \geq \frac{(1-M)^2}{2M}\|g_k\|_2^2.$$

Pentru $\alpha \geq \beta / M$:

$$\frac{(a_k - b_k)^2}{2b_k} \geq \frac{(1-\beta)^2}{2M}\|g_k\|_2^2.$$

Deci,

$$\frac{(a_k - b_k)^2}{2b_k} \geq min\left\{\frac{(1-M)^2}{2M}, \frac{(1-\beta)^2}{2M}\right\}\|g_k\|_2^2. \qquad (5.5.9)$$

Considerând (5.5.8) împreună cu (5.5.9) obținem:

$$f(x_k - \theta_m \alpha g_k) \leq f(x_k) - min\left\{\rho, \frac{\rho\beta}{M}\right\}\|g_k\|_2^2$$
$$-min\left\{\frac{(1-M)^2}{2M}, \frac{(1-\beta)^2}{2M}\right\}\|g_k\|_2^2. \qquad (5.5.10)$$

Deci

$$f(x_k) - f(x_{k+1}) \geq \left[min\left\{\rho, \frac{\rho\beta}{M}\right\} + min\left\{\frac{(1-M)^2}{2M}, \frac{(1-\beta)^2}{2M}\right\}\right]\|g_k\|_2^2.$$

Dar, $f(x_k) - f(x_{k+1}) \to 0$ şi în consecinţă g_k tinde la zero, adică x_k converge la x^*. Având în vedere că $f(x_k)$ este un şir necrescător, urmează că $f(x_k)$ converge la $f(x^*)$. Din (5.5.10) vedem că

$$f(x_{k+1}) \le f(x_k) - \left[min\left\{ \rho, \frac{\rho\beta}{M} \right\} + min\left\{ \frac{(1-M)^2}{2M}, \frac{(1-\beta)^2}{2M} \right\} \right] \|g_k\|_2^2 .$$

Combinând aceasta cu

$$\|g_k\|_2^2 \ge 2m\left(f(x_k) - f^* \right)$$

şi scăzând f^* din ambii membrii ai inegalităţii de mai sus concluzionăm că:

$$f(x_{k+1}) - f^* \le c\left(f(x_k) - f^* \right), \tag{5.5.11}$$

unde

$$c = 1 - min\left\{ 2m\rho, \frac{2m\rho\beta}{M} \right\} - min\left\{ \frac{(1-M)^2 m}{M}, \frac{(1-\beta)^2 m}{M} \right\} < 1. \tag{5.5.12}$$

Deci, $f(x_k)$ converge la f^* cel puţin ca o serie geometrică cu un factor care depinde de parametrii de backtracking şi marginea asupra numărului de condiţionare M / m. Deci convergenţa algoritmului este cel puţin liniară. ■

La fiecare iteraţie k, selectând $\theta_k = \theta_m$ in (5.5.1), algoritmul PDA reduce valorile funcţiei ca în (5.5.11) unde c este dat de (5.5.12). Deoarece algoritmul PD realizează (5.5.11) cu

$$c = 1 - min\left\{ 2m\rho, \frac{2m\rho\beta}{M} \right\} < 1,$$

ca în (5.3.17), rezultă că, dacă $\theta_m \ne 1$, atunci algoritmul PDA constituie o îmbunătăţire, adică o accelerare a algoritmului PD [Andrei, 2006].

Exemplul 5.5.1. Să ilustrăm funcţionarea algoritmului de pas descendent accelerat cu backtracking prin minimizarea următoarei funcţii (funcţia trigonometrică extinsă [Andrei, 2003]).

$$f(x) = \sum_{i=1}^{n} \left(\left(n - \sum_{j=1}^{n} \cos x_j \right) + i(1 - \cos x_i) - \sin x_i \right)^2 .$$

Considerăm $n = 100$, punctul inițial $x_0 = [0.2, \ldots, 0.2]$, şi parametrii $\rho = 0.0001$ şi $\beta = 0.8$ în procedura de backtracking. Criteriul de oprire a iteraţiilor utilizat în acest experiment numeric este:

$$\|\nabla f(x_k)\|_2 \le \varepsilon_g \quad \text{sau} \quad \alpha_k \left| g_k^T d_k \right| \le \varepsilon_f \left| f(x_{k+1}) \right|,$$

unde $\varepsilon_g = 10^{-6}$ şi $\varepsilon_f = 10^{-16}$. Atunci, evoluția erorii în valorile funcției $\left|f(x_k) - f^*\right|$, a contribuțiilor ρa_k corespunzătoare algoritmului pasului descendent şi $(a_k - b_k)^2 / 2b_k$ corespunzătoare algoritmului pasului descendent accelerat la reducerea valorilor funcției de minimizat, la fiecare iterație, (vezi (5.5.6)) sunt ilustrate în figura 5.5.1.

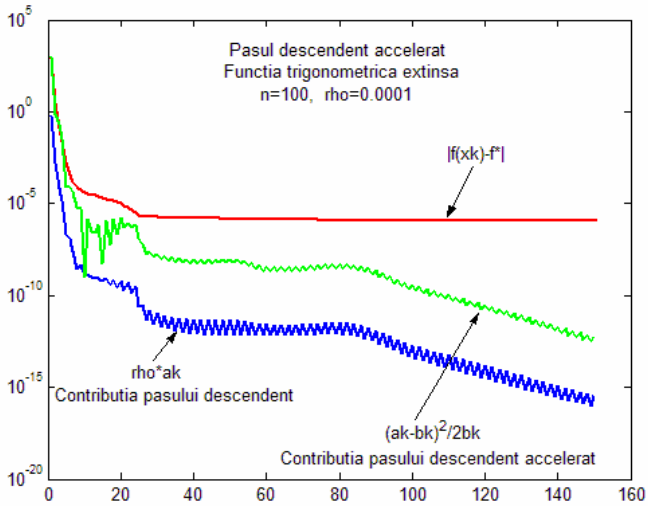

Fig. 5.5.1. Pasul descendent accelerat. ρ=0.0001

Pasul descendent accelerat are aceeaşi comportare ca pasul descendent. În primele iterații eroarea $\left|f(x_k) - f^*\right|$ este puternic redusă, după care algoritmul devine din ce în ce mai lent. Remarcăm faptul că contribuția $(a_k - b_k)^2 / 2b_k$ corespunzătoare algoritmului pasului descendent accelerat la reducerea valorilor funcției de minimizat este mai mare decât contribuția ρa_k corespunzătoare algoritmului pasului descendent. Vedem că această caracteristică se menține de-a lungul tuturor iterațiilor. Aceasta face ca algoritmul pasului descendent accelerat să fie superior de pas descendent.

Contribuția ρa_k la reducerea valorilor funcției din cadrul algoritmului pasului descendent depinde în mod direct de parametrul ρ din căutarea liniară cu backtracking. Este foarte interesant să vedem cum se modifică comportarea algoritmului pasului descendent accelerat la modificarea parametrului ρ din căutarea liniară cu backtracking. În figurile 5.5.1a şi 5.5.1b se arată evoluția erorii şi a contribuțiilor ρa_k şi $(a_k - b_k)^2 / 2b_k$ pentru $\rho = 0.01$ şi respectiv pentru $\rho = 0.1$.

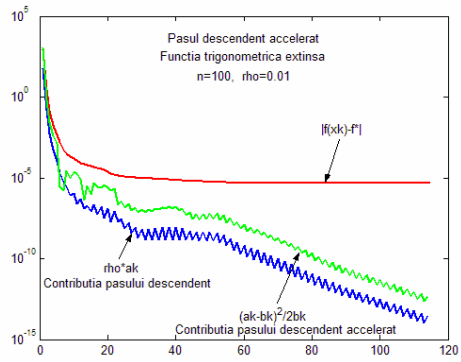

Fig. 5.5.1a. Pasul descendent accelerat. ρ=0.01

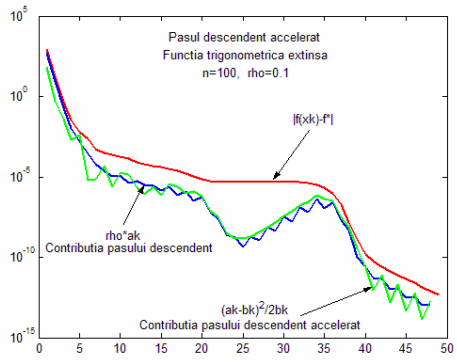

Fig. 5.5.1b. Pasul descendent accelerat. ρ = 0.1

Observăm că în acest caz parametrul ρ din căutarea liniară cu backtracking are o importanţă semnificativă în comportarea algoritmului de pas descendent şi a variantei sale accelerate. Valori mici ale lui ρ oferă o anumită uniformitate în comportarea algoritmilor. În aceste situaţii vedem că $(a_k - b_k)^2 / 2b_k \geq \rho a_k$ pentru majoritatea iteraţiilor. Dacă ρ este ceva mai mare, atunci din (5.5.6) vedem că algoritmul pasului descendent accelerat este încă o îmbunătăţire a algoritmului clasic de pas descendent. Interesant este faptul că cele două contribuţii ρa_k şi $(a_k - b_k)^2 / 2b_k$ se urmăresc de-a lungul iteraţiilor. Mai mult, acest experiment numeric arată că selecţia $\rho = 0.0001$ din căutarea liniară cu backtracking este foarte convenabilă, aceasta ilustrând încă odată că pretenţii nu prea mari în reducerea valorilor funcţiei de minimizat prin backtracking sunt de preferat (vezi figura 5.5.1). Parametrul β din căutarea liniară cu backtracking nu influenţează accelerarea pasului descendent.

Studiu numeric (PDA versus PDR şi PD)[2]

În cele ce urmează prezentăm rezultatele unui studiu numeric privind comportarea algoritmului pasului descendent accelerat (PDA) în comparaţie cu algoritmul pasului descendent (PD) şi algoritmul pasului descendent relaxat (PDR). În acest sens am considerat un număr de 31 probleme de optimizare fără restricţii şi pentru fiecare problemă am considerat 10 instanţieri în care numărul de variabile este $n = 100, ..., 1000$. Ca şi în studiul numeric anterior problemele sunt din colecţia CUTE [Bongartz, Conn, Gould şi Toint, 1995], împreună cu alte probleme de optimizare fără restricţii construite în acest scop. Toţi algoritmii consideraţi sunt implementaţi în Fortran utilizând acelaşi stil de programare, precum şi acelaşi criteriu de oprire a iteraţiilor:

[2] Programul PDACC.FOR din CD-ul anexat lucrării implementează algoritmul de pas descendent accelerat.

$$\left\|\nabla f(x_k)\right\|_2 \le \varepsilon_g \quad \text{sau} \quad \alpha_k \left|g_k^T d_k\right| \le \varepsilon_f \left|f(x_{k+1})\right|,$$

unde $\varepsilon_g = 10^{-6}$ şi $\varepsilon_f = 10^{-16}$. În procedura de căutare liniară prin backtracking se utilizează aceleaşi valori pentru parametrii ρ şi β: $\rho = 0.0001$, $\beta = 0.8$. Din cele 310 probleme rezolvate de aceşti algoritmi am selectat 270 pentru care diferenţa dintre valorile funcţiilor de minimizat în punctul de minim este mai mică decât 10^{-3}.

Tabelul 5.5.1 conţine numărul de probleme, din cele 270, pentru care PDA, PDR şi PD realizează numărul minim de iteraţii (#iter), numărul minim de evaluări ale funcţiei şi gradientului (#fg), precum şi timpul minim de calcul (CPU).

Tabelul 5.5.1. Performanţa lui PDA versus PDR şi PD. 270 de probleme.

	#iter	**#fg**	**CPU**
PDA	193	154	156
PDR	81	85	85
PD	4	31	29

Referindu-ne, de exemplu, la timpul de calcul, observăm că PDA a realizat un timp de calcul mai bun în 156 de probleme, faţă de PDR care a fost mai bun doar în 85 de probleme, sau faţă de PD care a fost mai bun numai în 29 de probleme din cele 270 rezolvate. Aceeaşi comportare o găsim şi în ceea ce priveşte numărul de iteraţii sau numărul de evaluări ale funcţiei şi gradientului.

În figurile 5.5.2a şi 5.5.2b prezentăm profilul Dolan-Moré pentru aceşti algoritmi, adică reprezentăm grafic fracţia din numărul de probleme pentru care un algoritm a fost cu un anumit factor mai bun în privinţa numărului de iteraţii sau a timpului de calcul.

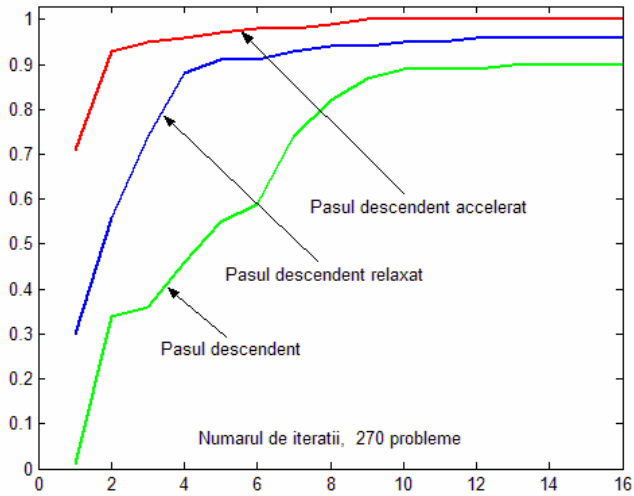

Fig. 5.5.2a. PDA versus PDR şi PD.

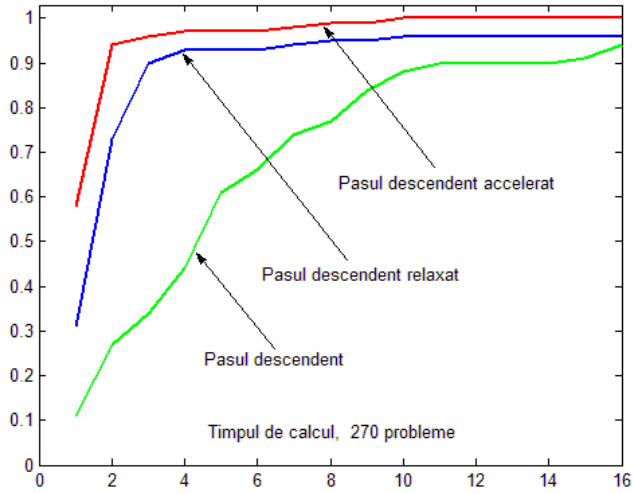

Fig. 5.5.2b. PDA versus PDR şi PD.

Observăm că algoritmul pasului descendent accelerat este net superior algoritmului pasului descendent relaxat şi celui de pas descendent clasic. Pe lângă acestea, din partea finală a acestor grafice, vedem că algoritmul pasului descendent accelerat este mai robust decât algoritmul pasului descendent clasic sau relaxat. Studiul numeric relevă faptul că tehnica de accelerare prezentată mai sus constituie o îmbunătăţire a backtracking-ului în contextul pasului descendent [Andrei, 2006].

5.6. UN NOU ALGORITM DE PAS DESCENDENT

În 1988 Barzilai şi Borwein au propus un algoritm de gradient descendent care utilizează o strategie diferită de selecţie a lungimii pasului de deplasare. În principiu, aceasta utilizează o aproximaţie în două puncte a ecuaţiei secantei din cadrul metodelor quasi-Newton. Schema iterativă propusă de Barzilai şi Borwein (BB) este următoarea:

$$x_{k+1} = x_k - \frac{1}{\gamma_k}\nabla f(x_k), \tag{5.6.1}$$

unde scalarul γ_k este calculat sub forma:

$$\gamma_k^{BB} = \frac{s^T y}{s^T s} \quad \text{sau} \quad \gamma_k^{IBB} = \frac{y^T y}{s^T y},$$

în care $s = x_k - x_{k-1}$ şi $y = \nabla f(x_k) - \nabla f(x_{k-1})$. Observăm că această schemă utilizează o lungime a pasului care nu foloseşte conceptul de căutare liniară, ci determină lungimea pasului ca inversa unei aproximaţii scalare a Hessianului funcţiei de minimizat. Aceste formule sunt obţinute din ecuaţia quasi-Newton $B_k s = y$, unde B_k este o aproximaţie simetrică şi pozitiv definită a lui $\nabla^2 f(x_k)$.

Prima valoare γ_k^{BB} se obține prin minimizarea lui $\|\gamma s - y\|$ care măsoară discrepanța dintre cei doi membrii ai ecuației secantei când $B_k = \gamma I$. A doua valoare se obține prin minimizarea lui $\|s - (1/\gamma)y\|$. Raydan [1993] a arătat că pentru funcții pătratice strict convexe algoritmul BB este global convergent. Cu alte cuvinte, lungimea pasului sugerată de Barzilai şi Borwein este chiar inversa unei aproximații scalare a matricei Hessian în punctul curent, calculată din ecuația secantei. Algoritmul de minimizare propus de Barzilai şi Borwein utilizează două puncte inițiale cu care se demarează calculele.

În cele ce urmează vom propune un algoritm de gradient descendent în care spre deosebire de modul de abordare al lui Barzilai şi Borwein, lungimea pasului se prezintă ca inversa unei aproximații scalare a matricei Hessian calculată prin intermediul valorilor funcției şi a gradientului acesteia în două puncte succesive. Astfel obținem un nou algoritm de gradient descendent [Andrei, 2004f].

Să considerăm un punct inițial x_0 unde $f(x_0)$ şi $g_0 = \nabla f(x_0)$ se pot imediat calcula. Utilizând o procedură de backtracking (inițializată cu $\alpha = 1$) se poate calcula lungimea α_0 a pasului, cu care o nouă estimație a minimului, $x_1 = x_0 - \alpha_0 g_0$, este calculată, în care din nou $f(x_1)$ şi $g_1 = \nabla f(x_1)$ se pot calcula. Astfel, primul pas este calculat prin backtracking utilizând direcția negativă a gradientului. Acum, în punctul $x_{k+1} = x_k - \alpha_k g_k$, $k = 0,1,\ldots$ avem: $f(x_{k+1}) = f(x_k) - \alpha_k g_k^T g_k + \frac{1}{2}\alpha_k^2 g_k^T \nabla^2 f(z) g_k$, unde $z \in [x_k, x_{k+1}]$. Având în vedere caracterul local al procedurii de căutare a minimului, precum şi faptul că distanța dintre x_k şi x_{k+1} este destul de mică (mai ales în partea finală a procesului iterativ), rezultă că putem alege $z = x_{k+1}$ şi considera $\gamma(x_{k+1})I$ ca o aproximare a lui $\nabla^2 f(x_{k+1})$, unde $\gamma(x_{k+1}) \in R$. Cu alte cuvinte, am considerat un *punct de vedere anticipativ* în care o aproximație scalară a Hessianului în punctul x_{k+1} este calculată utilizând informațiile locale din punctul x_k. Ca atare, putem scrie:

$$\gamma(x_{k+1}) = \frac{2}{g_k^T g_k}\frac{1}{\alpha_k^2}\left[f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k\right]. \qquad (5.6.2)$$

Acum, pentru a calcula următoarea estimație $x_{k+2} = x_{k+1} - \alpha_{k+1} g_{k+1}$ a minimului trebuie să precizăm o procedură de calcul a lungimii α_{k+1} a pasului. Pentru aceasta să considerăm funcția:

$$\Phi_{k+1}(\alpha) = f(x_{k+1}) - \alpha g_{k+1}^T g_{k+1} + \frac{1}{2}\alpha^2 \gamma(x_{k+1}) g_{k+1}^T g_{k+1}.$$

Observăm că $\Phi_{k+1}(0) = f(x_{k+1})$ şi $\Phi'_{k+1}(0) = -g_{k+1}^T g_{k+1} < 0$. Deci $\Phi_{k+1}(\alpha)$ este o funcție convexă pentru toți $\alpha \geq 0$. Pentru ca $\Phi_{k+1}(\alpha)$ să aibă un minim trebuie ca $\gamma(x_{k+1}) > 0$. Considerând pe moment că $\gamma(x_{k+1}) > 0$, atunci din $\Phi'_{k+1}(\alpha) = 0$ obținem:

$$\bar{\alpha}_{k+1} = \frac{1}{\gamma(x_{k+1})}, \tag{5.6.3}$$

ca minimul lui $\Phi_{k+1}(\alpha)$. Acum,

$$\Phi_{k+1}(\bar{\alpha}_{k+1}) = f(x_{k+1}) - \frac{1}{2\gamma(x_{k+1})}\|g_{k+1}\|_2^2,$$

care arată că dacă $\gamma(x_{k+1}) > 0$ atunci valoarea funcției f este redusă. Aceasta sugerează să determinăm lungimea α_{k+1} prin backtracking, sub forma:

$$\alpha_{k+1} = \underset{\alpha \le \bar{\alpha}_{k+1}}{arg\,min}\, f(x_{k+1} - \alpha g_{k+1}). \tag{5.6.4}$$

Pentru a completa algoritmul, trebuie să considerăm situația în care $\gamma(x_{k+1}) < 0$. Dacă $f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k < 0$, atunci reducerea $f(x_k) - f(x_{k+1})$ este mai mare decât $\alpha_k g_k^T g_k$. În acest caz trebuie să modificăm mărimea pasului α_k sub forma $\alpha_k + \eta_k$ astfel încât $f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k) g_k^T g_k > 0$.

Pentru a obține o valoare pentru η_k, selectăm un $\delta > 0$, şi considerăm:

$$\eta_k = \frac{1}{g_k^T g_k}\left[f(x_k) - f(x_{k+1}) - \alpha_k g_k^T g_k\right] + \delta \tag{5.6.5}$$

cu care o nouă valoare pentru $\gamma(x_{k+1})$ se poate imediat calcula:

$$\gamma(x_{k+1}) = \frac{2}{g_k^T g_k}\frac{1}{(\alpha_k + \eta_k)^2}\left[f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k) g_k^T g_k\right]. \tag{5.6.6}$$

care este strict pozitivă. Noul Algoritm (NA) corespunzător este următorul:

Algoritmul 5.6.1. (Un Nou Algoritm de pas descendent)

Pasul 1.	Selectăm $x_0 \in dom\, f$. Calculăm $f(x_0)$, $g_0 = \nabla f(x_0)$ precum şi $\alpha_0 = \underset{\alpha<1}{argmin}\, f(x_0 - \alpha g_0)$. Calculăm $x_1 = x_0 - \alpha_0 g_0$, $f(x_1)$ şi $g_1 = \nabla f(x_1)$. Punem $k = 0$.
Pasul 2.	Test pentru continuarea calculelor. Dacă un criteriu (sau mai multe) pentru oprirea calculelor este îndeplinit, atunci stop, altfel se continuă cu pasul 3.
Pasul 3.	Calculăm aproximarea scalară a Hessianului funcției f în x_{k+1} ca în (5.6.2)
Pasul 4.	Dacă $\gamma(x_{k+1}) < 0$, atunci selectăm $\delta > 0$ şi calculăm o nouă valoare pentru $\gamma(x_{k+1})$ sub forma (5.6.6), unde η_k este dat de (5.6.5).
Pasul 5.	Se calculează lungimea inițială a pasului: $\bar{\alpha}_{k+1}$ ca în (5.6.3).
Pasul 6.	Utilizând o procedură de backtracking se determină α_{k+1} sub forma: $\alpha_{k+1} = \underset{\alpha<\bar{\alpha}_{k+1}}{argmin}\, f(x_{k+1} - \alpha g_{k+1})$.

Pasul 7.	Se actualizează variabilele: $x_{k+2} = x_{k+1} - \alpha_{k+1} g_{k+1}$, se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Proprietăţile privind convergenţa şi buna definire a acestui algoritm sunt date de următoarele teoreme.

Teorema 5.6.1. *Pentru funcţii tare convexe algoritmul de gradient descendent de mai sus cu căutare liniară cu backtracking este liniar convergent şi:*

$$f(x_k) - f^* \le \left(\prod_{i=0}^{k-1} c_i\right)\left(f(x_0) - f^*\right),$$

unde

$$c_i = 1 - min\left\{2m\rho, 2m\rho\beta^{p_i}\right\} < 1$$

şi $p_i \ge 1$ *este un întreg, (* $p_i = 1,2,\ldots$ *dat de procedura de backtracking).*

Demonstraţie. Putem scrie:

$$f(x_{k+1}) = f(x_k) - \left(\alpha - \frac{1}{2}\alpha^2\gamma(x_{k+1})\right)\|g_k\|_2^2.$$

Dar după cum se vede imediat, $\alpha - \alpha^2\gamma(x_{k+1})/2$ este o funcţie concavă, şi pentru toţi $0 \le \alpha \le 1/\gamma(x_{k+1})$, $\alpha - \gamma(x_{k+1})\alpha^2/2 \ge \alpha/2$. Deci

$$f(x_{k+1}) \le f(x_k) - \frac{\alpha}{2}\|g_k\|_2^2 \le f(x_k) - \rho\alpha\|g_k\|_2^2.$$

Procedura de backtracking se termină fie cu $\alpha = 1$, fie cu $\alpha = \beta^{p_k}$, unde p_k este un întreg. Deci

$$f(x_{k+1}) \le f(x_k) - min\left\{\rho, \rho\beta^{p_k}\right\}\|g_k\|_2^2.$$

Având în vedere tare convexitatea, rezultă că $\|g_k\|_2^2 \ge 2m\left(f(x_k) - f^*\right)$, adică

$$f(x_{k+1}) - f^* \le c_k\left(f(x_k) - f^*\right),$$

unde $c_k = 1 - min\left\{2m\rho, 2m\rho\beta^{p_k}\right\}$. Deoarece $c_k < 1$ rezultă că şirul $f(x_k)$ este liniar convergent, ca o serie geometrică, la f^*. ■

Teorema 5.6.2. *Pentru toţi* $k = 0,1,\ldots$, $\gamma(x_{k+1})$, *generat de algoritmul de mai sus, este mărginit faţă de zero.*

Demonstraţie. Pentru toţi $k = 0,1,\ldots$ algoritmul ne asigură că $f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k > 0$. Deci, $f(x_k) - f(x_{k+1}) < \alpha_k g_k^T g_k$. Cu aceasta avem:

$$\gamma(x_{k+1}) = \frac{2}{\alpha_k} - \frac{2\left(f(x_k) - f(x_{k+1})\right)}{\alpha_k^2\left(g_k^T g_k\right)} > \frac{2}{\alpha_k} - \frac{2\alpha_k\left(g_k^T g_k\right)}{\alpha_k^2\left(g_k^T g_k\right)} = 0. \quad \blacksquare$$

În continuare să ilustrăm funcţionarea algoritmilor de gradient descendent în formula clasică, cea relaxată, precum şi noua variantă propusă mai sus.

Exemplul 5.6.1. Să considerăm funcţia cu punctul iniţial:

$$f(x) = \sum_{i=1}^{n-1}\left(x_{i+1} - x_i^3\right)^2 + \left(1 - x_i\right)^2, \quad x_0 = [-1.2, 1, -1.2, 1, \ldots].$$

În tabelul 5.6.1 se arată numărul de iteraţii şi lungimea medie a pasului pentru cei trei algoritmi: PD-pasul descendent clasic, PDR-pasul descendent relaxat şi pentru NA-noul algoritm.

Tabelul 5.6.1. Numărul de iteraţii şi lungimea medie a pasului.

	PD		**PDR**		**NA**	
n	**# iter**	**pasul mediu**	**# iter**	**pasul mediu**	**# iter**	**pasul mediu**
1000	1061	0.0598325	418	0.258687	185	0.265502
2000	1069	0.0595312	451	0.246441	217	0.242480
3000	1069	0.0595137	438	0.257312	203	0.251972
4000	1070	0.0595230	449	0.252725	218	0.237006
5000	1063	0.0598308	485	0.231459	198	0.257031

Tabelul 5.6.2 arată numărul de iteraţii ale algoritmului NA corespunzător diferitelor valori ale parametrului δ. În coloana de sub γ se prezintă numărul de iteraţii în care $\gamma(x_{k+1}) < 0$.

Tabelul 5.6.2. Algoritmul NA. Numărul de iteraţii pentru diferite valori ale lui δ.

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 1$		$\delta = 10$		$\delta = 100$	
n	**# iter**	γ	**# iter**	γ	**# iter**	γ	**# iter**	γ	**# iter**	γ
10000	224	1	219	1	204	1	239	1	208	1
20000	207	1	200	1	230	1	210	1	190	1
30000	194	1	226	1	234	1	224	1	204	1
40000	195	1	207	1	215	1	222	1	226	1
50000	212	1	222	1	222	1	224	1	205	1

Vedem că toţi aceşti algoritmi sunt liniar convergenţi. Din punctul de vedere al iteraţiilor, algoritmul NA este superior. Totuşi, acesta îşi păstrează caracterul de convergenţă liniară. Mai mult, teorema 5.4.3 ne arată convergenţa liniară a algoritmului de gradient descendent relaxat, dar pentru acesta $c_k > c$, (vezi (5.3.15)) ceea ce nu ne asigură că versiunea relaxată realizează întotdeauna o reducere mai pronunţată a valorilor funcţiei de minimizat faţă de versiunea clasică

a algoritmului de gradient descendent. Aceeaşi concluzie rezultă şi din interpretarea teoremei 5.6.1.

Este foarte interesantă o comparaţie dintre aproximaţiile scalare ale matricei Hessian date de Barzilai-Borwein şi cele considerate în algoritmul NA. Într-adevăr, pentru acest exemplu, evoluţia diferenţei $\gamma(x_{k+1}) - \gamma_k^{BB}$ este arătată în figura 5.6.1.

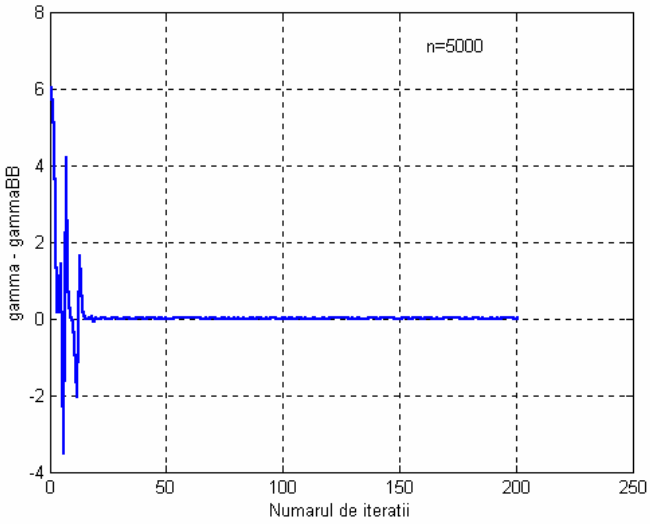

Fig. 5.6.1. Evoluţia diferenţei $\gamma(x_{k+1}) - \gamma_k^{BB}$.

Vedem că aproximaţiile matricei Hessian devin din ce în ce mai apropiate. Aceasta arată o anumită unitate în ceea ce priveşte ecuaţia secantei şi formula de calcul a lui $\gamma(x_{k+1})$. În general, aproximaţiile matricei Hessian date de algoritmul NA sunt mai mari decât cele corespunzătoare algoritmului Barzilai-Borwein. Aceasta arată că iniţializarea în căutarea liniară cu backtracking din algoritmul NA este mai mică.

Exemplul 5.6.2. Să considerăm funcţia cu punctul iniţial:

$$f(x) = \sum_{i=1}^{n} \frac{i}{10}\left(\exp(x_i) - x_i\right), \qquad x_0 = [1,1,\ldots,1].$$

Obţinem următoarele rezultate.

Tabelul 5.6.3. Numărul de iteraţii şi lungimea medie a pasului.

	PD		**PDR**		**NA**	
n	**# iter**	**pasul mediu**	**# iter**	**pasul mediu**	**# iter**	**pasul mediu**
1000	2696	0.020215	1168	0.117336	588	0.1199889
2000	4380	0.010106	1159	0.080865	758	0.0498313
3000	5514	0.006744	1340	0.063615	1465	0.0429736
4000	5788	0.005044	1177	0.057248	1401	0.0348057
5000	6108	0.004036	1523	0.045275	1316	0.0189150

Tabelul 5.6.4. Algoritmul NA. Numărul de iteraţii pentru diferite valori ale lui δ.

	$\delta = 0.01$		$\delta = 0.1$		$\delta = 1$		$\delta = 10$		$\delta = 100$	
n	**# iter**	γ	**# iter**	γ	**# iter**	γ	**# iter**	γ	**# iter**	γ
10000	2413	0	2413	0	2413	0	2413	0	2413	0
20000	2586	3	3091	19	1896	4	1702	4	1918	2
30000	2806	3	2405	2	2858	16	2173	2	2076	8
40000	2449	23	3099	4	2865	4	2504	8	2407	3
50000	3847	11	3931	11	3368	19	2915	1	3427	14

Observăm că şi pentru acest exemplu convergenţa este liniară. Valorile lui δ nu au un impact deosebit asupra numărului de iteraţii.

În această secţiune am prezentat o serie de variante de algoritmi de pas descendent care se referă la calculul lungimii pasului de deplasare de-a lungul direcţiei negative a gradientului în punctul curent. Deci, toate acestea modifică lungimea pasului, lăsând direcţia de deplasare ca: $-\nabla f(x_k)$. Drept urmare, se obţin algoritmi pentru care convergenţa este liniară. Totuşi, anumite variante de algoritmi pot determina o pantă mai pronunţată a reducerii valorilor funcţiei de minimizat, în funcţie de informaţiile utilizate în punctul curent.

Aceasta arată limitele metodelor de pas (gradient) descendent, care se bazează numai pe informaţiile locale date de gradientul funcţiei în punctul curent. Menţinerea direcţiei de deplasare ca direcţia negativă a gradientului cu modificarea numai a procedurilor de determinare a lungimii pasului nu schimbă clasa algoritmului. Importanţa acestor abordări este simplitatea şi uşurinţa implementării acestor algoritmi în programe de calcul. Mai mult, metoda gradientului descendent se află la originea tuturor celorlalte metode de optimizare.

Algoritmul pasului descendent, sau încă de gradient descendent, nu este un algoritm de încredere în sensul de a fi preferat pentru rezolvarea aplicaţiilor în care este vorba de minimizarea unei funcţii netede. În esenţa lui acesta ţine seama numai de direcţia de variaţie maximă a funcţiei, dată de gradientul acesteia în punctul curent, şi nu ia în considerare curbura funcţiei în direcţia negativă a gradientului.

Pentru a obţine algoritmi care să fie cel puţin superliniar convergenţi, algoritmi care au o valoare practică şi care pot fi utilizaţi pentru rezolvarea problemelor concrete, trebuie definită o altă direcţie de deplasare, care să ţină seama de proprietăţile locale ale funcţiei în punctul curent. Ca atare, modificările de substanţă ale metodei pasului (gradientului) descendent se referă la direcţia de deplasare şi nu la lungimea pasului.