Capitolul 4

FUNDAMENTELE METODEI DE DIRECȚII DESCENDENTE PENTRU OPTIMIZARE FĂRĂ RESTRICȚII

În acest capitol vom prezenta aspectele teoretice fundamentale ale metodelor descendente de optimizare fără restricții. Rezultatele pe care le vom demonstra sunt foarte generale şi se aplică oricărui algoritm care utilizează principiul de optimizare dat de conceptul de direcție descendentă. În acest context, orice algoritm de optimizare este compunerea a două aplicații: prima determină direcția de deplasare către minim, o direcție descendentă. A doua determină lungimea pasului de deplasare de-a lungul direcției calculate. În acest capitol nu vom discuta modul cum se calculează direcția de deplasare. Acesta este subiectul celorlalte capitole din această monografie. Aici suntem interesați de definirea unei metode de direcții descendente şi modul cum se calculează lungimea pasului de deplasare.

Ideea algoritmului este foarte simplă. Se consideră un *punct inițial* din care se demarează calculele. În acest punct se determină o direcție care este direcționată către minimul funcției de minimizat, numită *direcție de descendență*. De-a lungul acestei direcții se execută o deplasare convenabilă, numită *căutare liniară*, astfel obținându-se un nou punct din care se pot relua calculele. Toate aceste elemente care definesc o metodă descendentă de optimizare au propriile lor principii şi subtilități care ilustrează bogăția şi maturitatea domeniului optimizării fără restricții.

Scopul acestui capitol este de a *prezenta în mod riguros matematic şi cu detalii algoritmul general de direcții descendente, precum şi rezultatele teoretice fundamentale privind convergența acestuia în funcție de tehnicile şi metodele de calcul ale lungimii pasului de deplasare.* Menționăm faptul că algoritmii de optimizare fără restricții se individualizează după modul cum se calculează direcția de deplasare. Un element comun al acestora este metoda de calcul a lungimii pasului pe care o vom detalia în acest capitol.

Structura capitolului este următoarea. Pentru început vom prezenta algoritmul general de direcții descendente pentru optimizare fără restricții. În acest context vom preciza conceptul de căutare liniară, şi vom identifica tipurile de căutare liniară: *căutare liniară exactă* şi *căutare liniară inexactă* sau *aproximativă.* În continuare vom prezenta teoria convergenței algoritmului general pentru căutare

liniară exactă şi apoi pentru căutare liniară inexactă. Studiile referitoare la analiza convergenței se pot clasifica în următoarele patru categorii.

1. *Convergența globală*. În acest caz, problema este de a vedea dacă iterațiile converg la soluție în condiția în care algoritmul este inițializat în puncte plasate departe de punctul de optim.
2. *Convergența locală.* Aici, obiectivul este de a arăta că există o vecinătate a soluției şi o alegere convenabilă a parametrilor metodei pentru care convergența la punctul de optim este garantată.
3. *Rata asimptotică de convergență.* Aceasta reprezintă viteza algoritmului, când acesta converge la soluție.
4. *Rata globală de convergență* sau *eficiența globală*. Obiectivul constă în a determina reducerea funcției la fiecare iterație. În acelaşi timp se are în vedere studiul convergenței algoritmului în cele mai defavorabile situații (numerice sau de structură).

Rezultatele de convergență sunt greu de obținut. Menționăm faptul că literatura de specialitate acoperă problematica punctelor 1-3 de mai sus. Studiile de eficiență globală sunt rare şi de obicei sunt obținute doar pentru probleme speciale ca cele pătratice, convexe sau strict convexe.

4.1. ALGORITMUL GENERAL DE DIRECȚII DESCENDENTE

Să considerăm deci problema

$$min\ f(x) \tag{4.1.1}$$

unde $f : R^n \to R$ este o funcție continuu diferențiabilă cu domeniul $dom\, f$. Valoarea optimă a acestei probleme o notăm cu f^*, adică $inf_x f(x) = f^*$. După cum am văzut, date punctul x_k şi direcția descendentă de deplasare d_k, atunci schema iterativă după care se calculează punctul de optim x^* pentru (4.1.1) este

$$x_{k+1} = x_k + \alpha_k d_k\,, \tag{4.1.2}$$

unde α_k este *lungimea pasului de deplasare* de-a lungul direcției d_k. Menționăm că o direcție descendentă în punctul x_k satisface condiția $\nabla f(x_k)^T d_k < 0$, numită *condiția de descendență*, precum şi condiția de a fi mărginită față de zero, $\|d_k\| \neq 0$. Prima condiție înseamnă că aceasta trebuie să facă un unghi ascuțit cu gradientul cu semn schimbat al funcției în punctul curent x_k. Dacă $\alpha_k \neq 0$, vedem că a doua condiție face ca (4.1.2) să se încadreze în principiul fundamental de construcție a algoritmilor de analiză numerică, care recomandă ca o estimație a

unei soluții a unei probleme să se calculeze ca o mică modificare a estimației anterioare.

Pentru demararea calculelor, metoda cere specificarea unui punct inițial x_0. Punctul de plecare x_0 trebuie să se afle în $dom\, f$ şi în plus mulțimea de nivel constant

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

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.

În acest capitol vom prezenta deci metode şi proprietățile acestora pentru calculul eficient a scalarului α_k. În celelalte capitole ne vom ocupa de calculul direcției d_k. Fie

$$\varphi(\alpha) = f(x_k + \alpha d_k). \tag{4.1.3}$$

Deci, problema este acum ca plecând din punctul x_k să se determine lungimea pasului α_k, în direcția d_k, astfel încât $\varphi(\alpha_k) < \varphi(0)$.

Dacă α_k se determină astfel încât funcția f este minimizată în direcția d_k, adică

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k) \tag{4.1.4}$$

sau

$$\varphi(\alpha_k) = \min_{\alpha > 0} \varphi(\alpha), \tag{4.1.5}$$

atunci avem o *căutare liniară exactă* sau încă o *căutare liniară optimă*, iar α_k este *lungimea optimă* a pasului de deplasare.

Pe de altă parte, dacă α_k se alege astfel încât se obține o reducere acceptabilă a funcției obiectiv, adică reducerea $f(x_k) - f(x_k + \alpha_k d_k) > 0$ este acceptabilă, atunci avem o *căutare liniară inexactă, acceptabilă* sau *aproximativă.*

Menționăm că deoarece în situațiile practice concrete, atât din considerente teoretice cât şi datorită costului de calcul prohibitiv – deci din considerente practice, mărimea optimă a lungimii pasului, cu foarte mici excepții (cazul funcțiilor pătratice de exemplu), nu poate fi determinată, rezultă că de obicei se utilizează o căutare liniară inexactă, care implementează algoritmi ce determină o reducere acceptabilă a valorilor funcției de minimizat. Este clar că din punctul de vedere al costului computațional este mult mai bine să ne mulțumim cu o reducere acceptabilă a valorilor funcției de-a lungul direcției de căutare, decât cu reducerea optimă (maximală).

În principiu, structura unui algoritm general de direcții descendente este următoarea.

Algoritmul 4.1.1. (Algoritm de optimizare fără restricții cu căutare liniară exactă)

Pasul 1.	*Inițializare*. Se consideră un punct inițial $x_0 \in R^n$, precum şi toleranța de terminare a iterațiilor algoritmului de minimizare $\varepsilon \ll 1$. Se pune $k = 0$.
Pasul 2.	Se calculează direcția de descendență d_k.
Pasul 3.	Se calculează lungimea pasului α_k astfel încât $$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k).$$
Pasul 4.	Se pune $$x_{k+1} = x_k + \alpha_k d_k.$$
Pasul 5.	Dacă $\lVert \nabla f(x_{k+1}) \rVert \leq \varepsilon$, atunci stop; altfel se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Observăm că algoritmul este foarte general. Acesta nu precizează modul cum se determină direcția de descendență în pasul 2. De fapt, multitudinea de algoritmi de minimizare fără restricții se diferențiază tocmai prin procedura de calcul a direcției de deplasare din acest pas. În capitolele următoare ale acestei monografii ne vom concentra asupra determinării direcțiilor descendente. În pasul 5 este posibil să se implementeze şi alte criterii care certifică obținerea unei soluții a problemei (4.1.1), aşa după cum am arătat în capitolul 2.

4.2. TEORIA CONVERGENȚEI PENTRU CĂUTARE LINIARĂ EXACTĂ

În această secțiune suntem interesați de funcționarea algoritmului în condițiile căutării liniare exacte, pe care o definim imediat. În finalul acestui capitol vom trata convergența algoritmului de mai sus în condițiile căutării liniare inexacte.

Fie deci

$$\varphi(\alpha) = f(x_k + \alpha d_k). \tag{4.2.1}$$

Observăm că $\varphi(0) = f(x_k)$ şi algoritmul realizează $\varphi(\alpha) \leq \varphi(0)$. Condiția (4.1.4) este foarte pretențioasă şi constă în a determina *minimul global* al funcției $\varphi(\alpha)$ ceea ce, mai ales din punct de vedere practic, este foarte dificil. De aceea, ne vom limita la a căuta primul punct de staționaritate al funcției, adică să determinăm α_k sub forma:

$$\alpha_k = \min_{\alpha \geq 0} \left\{ \alpha \geq 0 : \nabla f(x_k + \alpha d_k)^T d_k = 0 \right\}. \tag{4.2.2}$$

Deoarece prin (4.1.4) obținem minimul exact al funcției f şi prin (4.2.2) obținem un punct staționar al funcției $\varphi(\alpha)$, atunci condițiile (4.1.4) şi (4.2.2) determină căutarea liniară exactă.

Fie acum $\langle d_k, -\nabla f(x_k)\rangle$ unghiul dintre vectorii d_k şi $-\nabla f(x_k)$, atunci

$$\cos\langle d_k, -\nabla f(x_k)\rangle = -\frac{d_k^T \nabla f(x_k)}{\|d_k\|\|\nabla f(x_k)\|}. \tag{4.2.3}$$

Următoarea teoremă precizează o margine a reducerii valorilor funcției f la fiecare iterație a algoritmului de mai sus în condițiile căutării liniare exacte.

Teorema 4.2.1. *Fie $\alpha_k > 0$ soluția problemei (4.1.4). Fie $\|\nabla^2 f(x_k + \alpha d_k)\| \le M$ pentru orice $\alpha > 0$, unde M este o constantă pozitivă. Atunci*

$$f(x_k) - f(x_k + \alpha_k d_k) \ge \frac{1}{2M}\|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle. \tag{4.2.4}$$

Demonstrație. Din ipoteza teoremei, pentru orice $\alpha > 0$, putem scrie:

$$f(x_k + \alpha d_k) \le f(x_k) + \alpha d_k^T \nabla f(x_k) + \frac{\alpha^2}{2} M \|d_k\|^2. \tag{4.2.5}$$

Fie acum

$$\bar{\alpha} = -\frac{d_k^T \nabla f(x_k)}{M\|d_k\|^2}.$$

Atunci, utilizând (4.2.5) şi (4.2.3) obținem:

$$\begin{aligned} f(x_k) - f(x_k + \alpha_k d_k) &\ge f(x_k) - f(x_k + \bar{\alpha} d_k) \ge -\bar{\alpha} d_k^T \nabla f(x_k) - \frac{\bar{\alpha}^2}{2} M \|d_k\|^2 \\ &= \frac{1}{2}\frac{\left(d_k^T \nabla f(x_k)\right)^2}{M\|d_k\|^2} = \frac{1}{2M}\|\nabla f(x_k)\|^2 \frac{\left(d_k^T \nabla f(x_k)\right)^2}{\|d_k\|\|\nabla f(x_k)\|^2} \\ &= \frac{1}{2M}\|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle. \ \blacksquare \end{aligned}$$

Teorema arată că reducerea valorilor funcției de minimizat este cu atât mai pronunțată cu cât unghiul dintre direcția de deplasare d_k şi direcția negativă a gradientului $-\nabla f(x_k)$ este mai mic.

În acest moment suntem pregătiți să prezentăm proprietatea de convergență a algoritmului general 4.1.1 de optimizare fără restricții cu căutare liniară exactă. Următoarele două teoreme reprezintă două ipostaze ale aceluiaşi rezultat.

Teorema 4.2.2. *Fie* $f(x)$ *o funcție continuu diferențiabilă pe mulțimea deschisă* $D \subset R^n$ *şi presupunem că direcția* d_k *este descendentă, adică* $d_k^T \nabla f(x_k) \le 0$ *şi şirul* $\{x_k\}$ *generat de algoritmul 4.1.1 satisface condiția de monotonie* $f(x_{k+1}) \le f(x_k)$. *Fie* $\bar{x} \in D$ *un punct de acumulare al şirului* $\{x_k\}$ *şi o mulțime* K_1 *de indici cu* $K_1 = \{k : \lim x_k = \bar{x}\}$. *Presupunem de asemenea că există o constantă* $M > 0$ *astfel încât pentru orice* $k \in K_1$, $\|d_k\| < M$. *Dacă* $\bar{d}$ *este un punct de acumulare oarecare al şirului* $\{d_k\}$, *atunci*

$$\bar{d}^T \nabla f(\bar{x}) = 0. \tag{4.2.6}$$

În plus, dacă $f(x)$ *este de două ori continuu diferențiabilă, atunci*

$$\bar{d}^T \nabla^2 f(\bar{x}) \bar{d} \ge 0. \tag{4.2.7}$$

Demonstrație. Vom demonstra numai (4.2.6), deoarece (4.2.7) se demonstrează în mod similar. Să considerăm $K_2 \subset K_1$ o mulțime de indici pentru care $\bar{d} = \lim_{k \in K_2} d_k$. Dacă $\bar{d} = 0$, atunci (4.2.6) este trivial satisfăcută, altfel vom considera următoarele două cazuri.

(i) Există o mulțime de indici $K_3 \subset K_2$ astfel încât $\lim_{k \in K_3} \alpha_k = 0$. Deoarece α_k este lungimea exactă a pasului, rezultă că $d_k^T \nabla f(x_k + \alpha_k d_k) = 0$. Mai mult, deoarece $\|d_k\|$ este uniform mărginită superior şi întru-cât $\alpha_k \to 0$, la limită avem că $\bar{d}^T \nabla f(\bar{x}) = 0$.

(ii) Considerăm acum cazul $\liminf_{k \in K_2} \alpha_k = \bar{\alpha} > 0$. Fie $K_4 \subset K_2$ o mulțime de indici pentru care $\alpha_k \ge \bar{\alpha}/2$. Presupunem că concluzia (4.2.6) a teoremei nu este adevărată, atunci $\bar{d}^T \nabla f(\bar{x}) < -\delta < 0$. Deci există o vecinătate $N(\bar{x})$ a lui $\bar{x}$ şi o mulțime de indici $K_5 \subset K_4$ astfel încât de îndată ce $x \in N(\bar{x})$ şi $k \in K_5$, $d_k^T \nabla f(x) \le -\delta/2 < 0$. Fie $\hat{\alpha}$ un număr pozitiv suficient de mic astfel încât pentru toți $0 \le \alpha \le \hat{\alpha}$ şi toți $k \in K_5$, $x_k + \alpha d_k \in N(\bar{x})$. Considerăm $\alpha^* = min\{\bar{\alpha}/2, \hat{\alpha}\}$. Atunci utilizând proprietatea de monotonie necrescătoare a algoritmului, căutarea liniară exactă şi dezvoltarea în serie Taylor, avem:

$$\begin{aligned} f(\bar{x}) - f(x_0) &= \sum_{k=0}^{\infty} [f(x_{k+1}) - f(x_k)] \le \sum_{k \in K_5} [f(x_{k+1}) - f(x_k)] \\ &\le \sum_{k \in K_5} [f(x_k + \alpha^* d_k) - f(x_k)] \\ &= \sum_{k \in K_5} \alpha^* \nabla f(x_k + \tau_k d_k)^T d_k \end{aligned} \tag{4.2.8}$$

$$\leq \sum_{k\in K_5} -\left(\frac{\delta}{2}\right)\alpha^* = -\infty,$$

unde $0 \leq \tau_k \leq \alpha^*$. Contradicția de mai sus arată că (4.2.6) are loc şi în cazul *(ii)*.

Demonstrația lui (4.2.7) este similară. Este suficient să utilizăm dezvoltarea în serie Taylor de ordinul doi în locul lui (4.2.8). Într-adevăr:

$$\begin{aligned} f(\bar{x}) - f(x_0) &\leq \sum_{k\in K_5} [f(x_k + \alpha^* d_k) - f(x_k)] \\ &= \sum_{k\in K_5} \left[\alpha^* \nabla f(x_k)^T d_k + \frac{(\alpha^*)^2}{2} d_k^T \nabla^2 f(x_k + \tau_k d_k) d_k \right] \\ &\leq \sum_{k\in K_5} \frac{(\alpha^*)^2}{2} d_k^T \nabla^2 f(x_k + \tau_k d_k) d_k \\ &\leq \sum_{k\in K_5} \left[-\frac{1}{2}\left(\frac{\delta}{2}\right)(\alpha^*)^2 \right] = -\infty. \end{aligned}$$

Astfel, am obținut o contradicție care demonstrează (4.2.7). ■

Teorema 4.2.3. *Fie* $\nabla f(x)$ *uniform continuă pe mulțimea de nivel constant* $L = \{x \in R^n : f(x) \leq f(x_0)\}$. *Fie de asemenea* $\theta_k = \langle d_k, -\nabla f(x_k)\rangle$, *unde* d_k *este direcția generată de algoritmul 4.1.1. Dacă*

$$\theta_k \leq \frac{\pi}{2} - \mu, \text{ pentru un anumit } \mu > 0, \tag{4.2.9}$$

atunci $\nabla f(x_k) = 0$ *pentru un anumit* k *; sau* $f(x_k) \to -\infty$ *; sau* $\nabla f(x_k) \to 0$.

Demonstrație. Presupunem că pentru toți k, $\nabla f(x_k) \neq 0$ şi că $f(x_k)$ este mărginită inferior. Deoarece $\{f(x_k)\}$ este monoton descrescător, atunci limita lui există. Deci

$$f(x_k) - f(x_{k+1}) \to 0. \tag{4.2.10}$$

Presupunem, prin contradicție, că $\nabla f(x_k) \to 0$ nu este adevărată. Atunci, există un $\varepsilon > 0$ astfel încât $\|\nabla f(x_k)\| \geq \varepsilon$. Deci

$$-\frac{\nabla f(x_k)^T d_k}{\|d_k\|} = \|\nabla f(x_k)\| \cos\theta_k \geq \varepsilon \sin\mu \triangleq \varepsilon_1 \tag{4.2.11}$$

Observăm că

$$\begin{aligned} f(x_k + \alpha d_k) &= f(x_k) + \alpha \nabla f(\xi_k)^T d_k \\ &= f(x_k) + \alpha \nabla f(x_k)^T d_k + \alpha[\nabla f(\xi_k) - \nabla f(x_k)]^T d_k \end{aligned}$$

$$\leq f(x_k) + \alpha \|d_k\| \left(\frac{\nabla f(x_k)^T d_k}{\|d_k\|} + \|\nabla f(\xi_k) - \nabla f(x_k)\| \right), \quad (4.2.12)$$

unde ξ_k se află pe segmentul de dreaptă determinat de x_k şi $x_k + \alpha d_k$. Deoarece $\nabla f(x)$ este uniform continuă pe mulțimea de nivel constant L, există $\bar{\alpha}$ astfel încât când $0 \leq \alpha \|d_k\| \leq \bar{\alpha}$, atunci

$$\|\nabla f(\xi_k) - \nabla f(x_k)\| \leq \frac{1}{2}\varepsilon_1 . \quad (4.2.13)$$

Din (4.2.10)-(4.2.13) obținem:

$$f\left(x_k + \bar{\alpha} \frac{d_k}{\|d_k\|} \right) \leq f(x_k) + \bar{\alpha} \left(\frac{\nabla f(x_k)^T d_k}{\|d_k\|} + \frac{1}{2}\varepsilon_1 \right) \leq f(x_k) - \frac{1}{2}\bar{\alpha}\varepsilon_1$$

Deci

$$f(x_{k+1}) \leq f\left(x_k + \bar{\alpha} \frac{d_k}{\|d_k\|} \right) \leq f(x_k) - \frac{1}{2}\bar{\alpha}\varepsilon_1 ,$$

care contrazice (4.2.10). Deci $\nabla f(x_k) \to 0$, ceea ce completează demonstrația. ■

În cele ce urmează să prezentăm rata de convergență a algoritmului de minimizare cu căutare liniară exactă. Pentru început vom demonstra câteva rezultate tehnice.

Propoziția 4.2.1. *Fie $\varphi(\alpha)$ cel puțin de două ori continuu diferențiabilă pe intervalul închis $[0,b]$ şi $\varphi'(0) < 0$. Dacă minimul $\alpha^* \in (0,b)$ al lui $\varphi(\alpha)$ pe $[0,b]$, atunci*

$$\alpha^* \geq \tilde{\alpha} = -\varphi'(0)/M , \quad (4.2.14)$$

unde M este un număr pozitiv astfel încât $\varphi''(\alpha) \leq M$, pentru orice $\alpha \in [0,b]$.

Demonstrație. Considerăm funcția auxiliară $\phi(\alpha) = \varphi'(0) + M\alpha$ care are un zero unic $\tilde{\alpha} = -\varphi'(0)/M$. Dar pe $[0,b]$, $\varphi''(\alpha) \leq M$, deci

$$\varphi'(\alpha) = \varphi'(0) + \int_0^\alpha \varphi''(\sigma)\mathrm{d}\sigma \leq \varphi'(0) + \int_0^\alpha M\mathrm{d}\sigma = \phi(\alpha) .$$

Considerând $\alpha = \alpha^*$ şi observând că $\varphi'(\alpha^*) = 0$, din inegalitatea de mai sus obținem $0 \leq \phi(\alpha^*) = \varphi'(0) + M\alpha^*$, ceea ce demonstrează propoziția. ■

Propoziția 4.2.2. *Fie $f(x)$ de două ori continuu diferențiabilă pe R^n. Atunci, pentru orice $x, d \in R^n$ şi orice scalar α are loc:*

$$f(x+\alpha d)=f(x)+\alpha\nabla f(x)^T d+\alpha^2\int_0^1(1-t)[d^T\nabla^2 f(x+t\alpha d)d]\mathrm{d}t\ . \quad (4.2.15)$$

Demonstrație. Putem scrie:

$$f(x+\alpha d)-f(x)=\int_0^1 \mathrm{d}f(x+t\alpha d)=-\int_0^1[\alpha\nabla f(x+t\alpha d)^T d]\mathrm{d}(1-t)$$

$$=-[(1-t)\alpha\nabla f(x+t\alpha d)^T d]_0^1+\int_0^1(1-t)\mathrm{d}[\alpha\nabla f(x+t\alpha d)^T d]$$

$$=\alpha\nabla f(x)^T d+\alpha^2\int_0^1[(1-t)d^T\nabla^2 f(x+t\alpha d)d]\mathrm{d}t\ . \blacksquare$$

Propoziția 4.2.3. *Fie* $f(x)$ *o funcție de două ori continuu diferențiabilă într-o vecinătate a minimului* x^*. *Presupunem că există constantele* $\varepsilon>0$ *şi* $M>m>0$, *astfel încât pentru orice* $y\in R^n$, *de îndată ce* $\|x-x^*\|<\varepsilon$,

$$m\|y\|^2\le y^T\nabla^2 f(x)y\le M\|y\|^2. \quad (4.2.16)$$

Atunci

$$\frac{1}{2}m\|x-x^*\|^2\le f(x)-f(x^*)\le\frac{1}{2}M\|x-x^*\|^2 \quad (4.2.17)$$

şi

$$\|\nabla f(x)\|\ge m\|x-x^*\|. \quad (4.2.18)$$

Demonstrație. Din propoziția 4.2.2 rezultă că

$$f(x)-f(x^*)=\nabla f(x^*)^T(x-x^*)+\int_0^1(1-t)(x-x^*)^T\nabla^2 f(tx+(1-t)x^*)(x-x^*)\mathrm{d}t$$

$$=\int_0^1(1-t)(x-x^*)^T\nabla^2 f(tx+(1-t)x^*)(x-x^*)\mathrm{d}t\ . \quad (4.2.19)$$

Acum din (4.2.16) şi teorema de medie obținem

$$m\|x-x^*\|^2\int_0^1(1-t)\mathrm{d}t\le\int_0^1(1-t)(x-x^*)^T\nabla^2 f(tx+(1-t)x^*)(x-x^*)\mathrm{d}t$$

$$\le M\|x-x^*\|^2\int_0^1(1-t)\mathrm{d}t\ . \quad (4.2.20)$$

Din (4.2.19) şi (4.2.20) obținem imediat (4.2.17). Pentru a demonstra (4.2.18) observăm că

$$\nabla f(x) = \nabla f(x) - \nabla f(x^*) = \int_0^1 \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t .$$

Deci

$$\|\nabla f(x)\|\|x - x^*\| \geq (x - x^*)^T \nabla f(x) = \int_0^1 (x - x^*)^T \nabla^2 f(tx + (1-t)x^*)(x - x^*)\mathrm{d}t$$

$$\geq m\|x - x^*\|^2 ,$$

care demonstrează (4.2.18). ■

Următoarea teoremă este centrală în stabilirea ratei de convergenţă a algoritmului 4.1.1, algoritmul general de optimizare fără restricţii cu căutare liniară exactă. Se demonstrează că dacă funcţia $f(x)$ este de două ori continuu diferenţiabilă şi are Hessianul mărginit, atunci acesta este cel puţin liniar convergent.

Teorema 4.2.4. *Fie $\{x_k\}$ şirul generat de algoritmul general de optimizare fără restricţii cu căutare liniară exactă 4.1.1, convergent la minimul x^* al funcţiei $f(x)$. Fie $f(x)$ de două ori continuu diferenţiabilă într-o vecinătate a lui x^* şi presupunem că există constantele $\varepsilon > 0$ şi $M > m > 0$, astfel încât pentru orice $y \in R^n$, de îndată ce $\|x - x^*\| < \varepsilon$,*

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2 .$$

Atunci şirul $\{x_k\}$ este cel puţin liniar convergent la x^*.

Demonstraţie. Fie $\lim_{k\to\infty} x_k = x^*$. Aceasta înseamnă că pentru k suficient de mare putem presupune că $\|x_k - x^*\| \leq \varepsilon$. Deoarece $\|x_{k+1} - x^*\| < \varepsilon$, atunci există un $\delta > 0$ astfel încât

$$\|x_k + (\alpha_k + \delta)d_k - x^*\| = \|x_{k+1} - x^* + \delta d_k\| < \varepsilon . \qquad (4.2.21)$$

Observăm că $\varphi(\alpha) = f(x_k + \alpha d_k)$ şi $\varphi'(\alpha) = \nabla f(x_k + \alpha d_k)^T d_k$. Dar d_k este o direcţie descendentă, atunci $\varphi'(0) = \nabla f(x_k)^T d_k < 0$ şi $|\varphi'(0)| \leq \|\nabla f(x_k)\|\|d_k\|$. Deci, pentru un anumit $\rho \in (0,1)$ putem scrie

$$\rho\|\nabla f(x_k)\|\|d_k\| \leq -\varphi'(0) \leq \|\nabla f(x_k)\|\|d_k\|$$

şi

$$\varphi''(\alpha) = d_k^T \nabla^2 f(x_k + \alpha d_k) d_k \leq M\|d_k\|^2 .$$

Atunci, din propoziţia 4.2.1 cunoaştem că minimul α_k al lui $\varphi(\alpha)$ pe $[0, \alpha_k + \delta]$ satisface

$$\alpha_k \geq \tilde{\alpha}_k = \frac{-\varphi'(0)}{M\|d_k\|^2} \geq \frac{\rho\|\nabla f(x_k)\|}{M\|d_k\|} \triangleq \bar{\alpha}_k . \qquad (4.2.22)$$

Considerăm $\bar{x}_k = x_k + \bar{\alpha}_k d_k$. Din (4.2.21) rezultă că $\|\bar{x}_k - x^*\| < \varepsilon$. Deci

$$f(x_k + \alpha_k d_k) - f(x_k) \leq f(x_k + \bar{\alpha}_k d_k) - f(x_k)$$

$$= \bar{\alpha}_k \nabla f(x_k)^T d_k + \bar{\alpha}_k^2 \int_0^1 (1-t)[d_k^T \nabla^2 f(x_k + t\bar{\alpha}_k d_k) d_k]\mathrm{d}t \quad \text{(din propoziția 4.2.2)}$$

$$\leq \bar{\alpha}_k(-\rho)\|\nabla f(x_k)\|\|d_k\| + \frac{1}{2} M \bar{\alpha}_k^2 \|d_k\|^2 \quad \text{(din ipoteza teoremei şi (4.2.22))}$$

$$\leq -\frac{\rho^2}{2M}\|\nabla f(x_k)\|^2 \leq -\frac{\rho^2}{2M} m^2 \|x_k - x^*\|^2 \quad \text{(din (4.2.22) şi (4.2.18))}$$

$$\leq -\left(\frac{\rho m}{M}\right)^2 \left(f(x_k) - f(x^*)\right). \quad \text{(din (4.2.17))}$$

Deci

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

$$\leq \left[1 - \left(\frac{\rho m}{M}\right)^2\right][f(x_k) - f(x^*)]. \qquad (4.2.23)$$

Fie acum

$$\omega = \left[1 - \left(\frac{\rho m}{M}\right)^2\right]^{\frac{1}{2}}.$$

Evident $\omega \in (0,1)$. Ca atare (4.2.23) se poate rescrie ca

$$f(x_k) - f(x^*) \leq \omega^2[f(x_{k-1}) - f(x^*)] \leq \cdots \leq \omega^{2k}[f(x_0) - f(x^*)].$$

Din (4.2.17) rezultă că

$$\|x_k - x^*\|^2 \leq \frac{2}{m}[f(x_k) - f(x^*)] \leq \frac{2}{m}\omega^2[f(x_{k-1}) - f(x^*)] \leq \frac{2}{m}\omega^2 \frac{M}{2}\|x_{k-1} - x^*\|^2,$$

de unde obținem

$$\|x_k - x^*\| \leq \sqrt{\frac{M}{m}}\omega\|x_{k-1} - x^*\|,$$

care arată că şirul $\{x_k\}$ este cel puțin liniar convergent la x^*. ■

Următoarea teoremă precizează o margine a reducerii valorilor funcției de minimizat realizată de algoritmul general de optimizare fără restricții cu căutare liniară exactă.

Teorema 4.2.5. *Presupunem că gradientul funcției de minimizat* $f(x)$ *satisface condiția*

$$(x-z)^T[\nabla f(x)-\nabla f(z)] \geq \eta \|x-z\|^2,$$

pentru orice $x, z \in R^n$ *şi fie* α_k *lungimea pasului obținută prin căutare liniară exactă. Atunci*

$$f(x_k)-f(x_k+\alpha_k d_k) \geq \frac{1}{2}\eta \|\alpha_k d_k\|^2.$$

Demonstrație. Deoarece α_k este obținută prin căutare liniară exactă, atunci

$$\nabla f(x_k+\alpha_k d_k)^T d_k = 0.$$

Deci, utilizând teorema de medie şi ipoteza teoremei, obținem

$$f(x_k)-f(x_k+\alpha_k d_k) = \int_0^{\alpha_k} -d_k^T \nabla f(x_k+td_k)\mathrm{d}t$$

$$= \int_0^{\alpha_k} d_k^T[\nabla f(x_k+\alpha_k d_k)-\nabla f(x_k+td_k)]\mathrm{d}t \geq \|d_k\|^2 \int_0^{\alpha_k} \eta(\alpha_k-t)\mathrm{d}t = \frac{1}{2}\eta\|\alpha_k d_2\|^2,$$

care demonstrează teorema. ■

4.3. METODE DE CĂUTARE LINIARĂ INEXACTĂ

În paragraful anterior am discutat proprietățile de convergență ale algoritmului general de direcții descendente echipat cu căutarea liniară exactă, adică lungimea pasului se calculează ca soluție a problemei de minimizare mono-dimensională:

$$f(x_k+\alpha_k d_k) = \min_{\alpha>0} f(x_k+\alpha d_k) \tag{4.3.1}$$

sau

$$\alpha_k = \min_{\alpha\geq 0}\left\{\alpha \geq 0 : \nabla f(x_k+\alpha d_k)^T d_k = 0\right\}. \tag{4.3.2}$$

După cum se vede căutarea liniară exactă, care cere rezolvarea problemei (4.3.1), este costisitoare, mai ales în situațiile în care punctul curent este departe de punctul de optim al problemei (4.1.1). Pe de altă parte, pentru multe metode de optimizare, în special pentru metodele Newton şi quasi-Newton, rata lor de convergență nu depinde de căutarea liniară exactă. De obicei aceste metode acceptă o lungime unitară a lungimii pasului şi de aceea căutarea liniară în aceste metode nu este foarte pretențioasă. Deci, în cadrul algoritmului general de direcții descendente 4.1.1 atât timp cât se obține o lungime a pasului care să realizeze o reducere acceptabilă a valorilor funcției de minimizat, atunci căutarea liniară exactă se poate foarte bine evita, ceea ce conduce la o reducere semnificativă a efortului de calcul.

În acest paragraf vom prezenta mai multe metode de căutare liniară inexactă care s-au dovedit eficiente din punct de vedere computațional, după care

vom detalia teoria convergenței algoritmului de direcții descendente cu acest tip de căutare liniară.

4.3.1. Căutarea liniară Armijo

Armijo [1966] a prezentat următoarea metodă, cunoscută ca *regula lui Armijo*. Fie scalarii $\beta \in (0,1)$, $\tau_k = -(\nabla f(x_k)^T d_k)/\|d_k\|^2$ şi $\rho \in (0,1/2)$. Considerăm $\alpha_k = \beta^{m_k}\tau_k$, unde m_k este primul întreg nenegativ m pentru care

$$f(x_k) - f(x_k + \beta^m \tau_k d_k) \geq -\rho\beta^m \tau_k \nabla f(x_k)^T d_k, \tag{4.3.3}$$

adică se încearcă succesiv $m = 0,1,\ldots$ până când inegalitatea de mai sus este satisfăcută pentru un $m = m_k$. Inegalitatea (4.3.3) se numeşte *condiția lui Armijo*. Cu alte cuvinte se încearcă succesiv $\alpha_k = \tau_k, \beta\tau_k, \beta^2\tau_k, \ldots$ până când (4.3.3) este îndeplinită. Următoarea teoremă arată că într-adevăr există un interval de lungimi α care satisfac condiția (4.3.3) a lui Armijo.

Teorema 4.3.1. *Fie* $x_k, d_k \in R^n$ *astfel încât* $d_k \neq 0$, $\nabla f(x_k)^T d_k < 0$ *şi* $\rho \in (0,1)$. *Atunci există* $\varepsilon = \varepsilon(\rho)$ *astfel încât*

$$f(x_k) - f(x_k + \alpha d_k) \geq -\rho\alpha \nabla f(x_k)^T d_k, \tag{4.3.4}$$

pentru orice $\alpha \in (0, \varepsilon]$.

Demonstrație. Avem:

$$0 \neq \nabla f(x_k)^T d_k = \lim_{\alpha \to 0} \frac{f(x_k + \alpha d_k) - f(x_k)}{\alpha}.$$

Deci

$$\lim_{\alpha \to 0} \frac{f(x_k + \alpha d_k) - f(x_k)}{\alpha \nabla f(x_k)^T d_k} = 1.$$

Ca atare, există un $\varepsilon > 0$ astfel încât pentru orice $\alpha \in (0, \varepsilon]$,

$$0 < \frac{f(x_k + \alpha d_k) - f(x_k)}{\alpha \nabla f(x_k)^T d_k} \leq \rho,$$

de unde rezultă concluzia teoremei. ■

Figura 4.3.1 arată intervalul de lungimi acceptate de regula lui Armijo, intervalul OA.

Presupunem acum că procedura de căutare Armijo este inițializată cu valoarea: $\bar{\alpha}_k = -\nabla f(x_k)^T d_k / \|d_k\|^2$. Atunci, următoarea teoremă ne asigură că în condiții destul de comode lungimea pasului dată de regula lui Armijo este mărginită inferior.

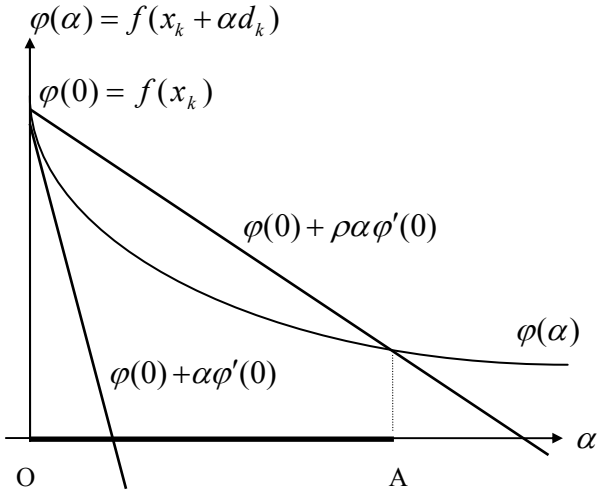


Fig. 4.3.1. Intervalul [O,A] acceptat de regula lui Armijo.

Teorema 4.3.2. *Presupunem că* d_k *este o direcție descendentă şi că* $\nabla f(x)$ *satisface condiția Lipschitz* $\|\nabla f(y) - \nabla f(x)\| \le L\|y - x\|$ *pentru orice* $x, y \in \{x : f(x) \le f(x_0)\}$, *unde* L *este o constantă pozitivă. Dacă căutarea liniară satisface condiția (4.3.3) a lui Armijo, atunci*

$$\alpha_k \ge min\left\{1, \frac{\beta(1-\rho)}{L}\frac{-g_k^T d_k}{\|d_k\|^2}\right\}.$$

Demonstrație. Fie $K_1 = \{k : \alpha_k = \tau_k\}$ şi $K_2 = \{k : \alpha_k < \tau_k\}$. Atunci, pentru orice $k \in K_1$ avem

$$f_k - f_{k+1} \ge -\rho\tau_k \nabla f(x_k)^T d_k$$

şi pentru orice $k \in K_2$

$$f_k - f_{k+1} \ge -\rho\alpha_k \nabla f(x_k)^T d_k .$$

Utilizând regula lui Armijo şi deoarece pentru orice $k \in K_2$, $\alpha_k / \beta \le \tau_k$, rezultă că pentru orice $k \in K_2$ are loc:

$$f_k - f(x_k + \frac{\alpha_k}{\beta} d_k) < -\rho\frac{\alpha_k}{\beta}\nabla f(x_k)^T d_k .$$

Acum, utilizând teorema de medie în membrul stâng al inegalității de mai sus, rezultă că există un $\xi_k \in [0,1]$ astfel încât

$$\nabla f(x_k + \frac{\alpha_k}{\beta}\xi_k d_k)^T d_k > \frac{\rho}{\beta}\nabla f(x_k)^T d_k$$

pentru orice $k \in K_2$.

Având în vedere condiția Lipschitz şi aplicând inegalitatea Cauchy-Schwartz, inegalitatea de mai sus furnizează:

$$\begin{aligned}\frac{\alpha_k}{\beta}L\|d_k\|^2 &\geq \left\|\nabla f(x_k + \frac{\alpha_k}{\beta}\xi_k d_k) - \nabla f(x_k)\right\|\|d_k\| \\ &\geq \left(\nabla f(x_k + \frac{\alpha_k}{\beta}\xi_k d_k) - \nabla f(x_k)\right)^T d_k \\ &\geq \rho\nabla f(x_k)^T d_k - \nabla f(x_k)^T d_k = -(1-\rho)\nabla f(x_k)^T d_k\end{aligned}$$

pentru toți $k \in K_2$. Combinând aceasta cu inegalitatea corespunzătoare pentru $k \in K_1$, obținem concluzia teoremei. ■

4.3.2. Căutarea liniară Goldstein

Goldstein [1965] a prezentat următoarea regulă. Fie intervalul

$$J = \{\alpha > 0 : f(x_k + \alpha d_k) < f(x_k)\}. \tag{4.3.5}$$

Pentru a garanta că funcția de minimizat se reduce suficient de mult, atunci ar trebui să alegem α care să se afle departe de capetele intervalului. Atunci, următoarele două condiții par rezonabile pentru a modela această situație:

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho\alpha\nabla f(x_k)^T d_k \tag{4.3.6}$$

şi

$$f(x_k + \alpha d_k) \geq f(x_k) + (1-\rho)\alpha\nabla f(x_k)^T d_k. \tag{4.3.7}$$

Evident, aceste condiții exclud punctele care se află lângă capătul din dreapta şi cele care se află lângă capătul din stânga al intervalului J, unde $0 < \rho < 1/2$. Relațiile (4.3.6) şi (4.3.7) constituie *căutarea liniară inexactă a lui Goldstein*, sau încă *regula lui Goldstein*, sau *condițiile Goldstein*.

Fie deci $\varphi(\alpha) = f(x_k + \alpha d_k)$. Atunci (4.3.6) şi (4.3.7) se pot exprima sub forma

$$\varphi(\alpha_k) \leq \varphi(0) + \rho\alpha_k\varphi'(0), \tag{4.3.8}$$

$$\varphi(\alpha_k) \geq \varphi(0) + (1-\rho)\alpha_k\varphi'(0). \tag{4.3.9}$$

Observăm că $\rho < 1/2$ este o condiție necesară. Fiindcă, dacă de exemplu $\varphi(\alpha)$ este o funcție pătratică care satisface condițiile $\varphi'(0) < 0$ şi $\varphi''(0) > 0$, atunci minimul global α^* al lui φ verifică relația $\varphi(\alpha^*) = \varphi(0) + \frac{1}{2}\alpha^*\varphi'(0)$. Deci α^* satisface (4.3.8) dacă şi numai dacă $\rho < 1/2$. Această condiție $\rho < 1/2$ are

repercusiuni asupra metodelor Newton şi quasi-Newton, permiţând $\alpha = 1$ în aceste metode şi asigurând deci convergenţa superliniară a acestora.

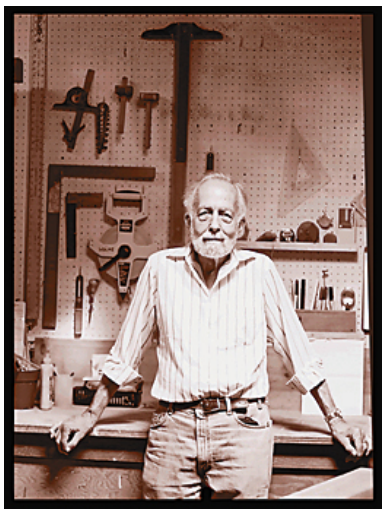
4.3.3. Căutarea liniară Wolfe

Este posibil ca relaţiile Goldstein (4.3.6)-(4.3.7) să fie prea tari şi să excludă punctele de minim α ale funcţiei $\varphi(\alpha)$. De aceea în locul condiţiei (4.3.7) Wolfe a introdus condiţia

$$\nabla f(x_{k+1})^T d_k \geq \sigma \nabla f(x_k)^T d_k, \tag{4.3.10}$$

unde $\sigma \in (\rho, 1)$. Refrazat în termenii funcţiei φ putem scrie

$$\varphi'(\alpha_k) = \nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma \nabla f(x_k)^T d_k = \sigma \varphi'(0) > \varphi'(0).$$



Philip Wolfe

Interpretarea geometrică a condiţiei (4.3.10) este că panta $\varphi'(\alpha_k)$ într-un punct acceptabil trebuie să fie mai mare decât sau egală cu un multiplu $\sigma \in (0,1)$ al pantei iniţiale $\varphi'(0)$. Relaţiile (4.3.6) şi (4.3.10) constituie *căutarea liniară inexactă a lui Wolfe*, sau încă *regula lui Wolfe*, sau *condiţiile Wolfe*. Condiţiile Wolfe sunt o extensie foarte fină a condiţiilor Goldstein în care este vorba nu numai de reducerea valorilor funcţiei, dată de (4.3.6), ci şi de panta graficului funcţiei de minimizat.

Observăm că (4.3.10) se poate obţine din teorema de medie şi (4.3.7). Într-adevăr, fie α_k care verifică (4.3.7), atunci:

$$\alpha_k \nabla f(x_k + \theta_k \alpha_k d_k)^T d_k = f(x_k + \alpha_k d_k) - f(x_k) \geq (1-\rho)\alpha_k \nabla f(x_k)^T d_k,$$

unde $\theta_k \in (0,1)$, care este chiar (4.3.10). Să arătăm că există α_k care satisface condiţiile Wolfe (4.3.6) şi (4.3.10).

Teorema 4.3.3. *Presupunem că f este continuu diferenţiabilă şi d_k este o direcţie descendentă pentru f în punctul x_k. Presupunem că f este mărginită inferior de-a lungul razei $\{x_k + \alpha d_k : \alpha \geq 0\}$. Dacă $0 < \rho < \sigma < 1$, atunci există un interval $[a,b]$, $0 < a < b$, astfel încât condiţiile Wolfe:*

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho \alpha \nabla f(x_k)^T d_k,$$
$$\nabla f(x_k + \alpha d_k)^T d_k \geq \sigma \nabla f(x_k)^T d_k,$$

sunt satisfăcute pentru toţi $\alpha \in [a,b]$.

Demonstrație. Fie $\varphi(\alpha) = f(x_k + \alpha d_k)$, $\alpha \geq 0$. Deoarece $\varphi'(0) < 0$ şi $0 < \rho < \sigma < 1$, rezultă că pentru α suficient de mic $\varphi(\alpha) < \varphi(0) + \rho\alpha\varphi'(0)$. Totuşi, deoarece φ este mărginită inferior această relație încetează de a mai fi adevărată pentru α suficient de mare. Deci, dacă definim $\bar{\alpha} = \sup\{\hat{\alpha} > 0 : \varphi(\alpha) < \varphi(0) + \rho\alpha\varphi'(0), \forall \alpha \in (0, \hat{\alpha})\}$, atunci $\bar{\alpha}$ există şi este finit. Mai mult, $\bar{\alpha}$ satisface (4.3.6) cu egalitate, adică $\varphi(\bar{\alpha}) = \varphi(0) + \rho\bar{\alpha}\varphi'(0)$. Din teorema de medie rezultă că există $\beta \in (0, \bar{\alpha})$ astfel încât

$$\varphi'(\beta) = \frac{\varphi(\bar{\alpha}) - \varphi(0)}{\bar{\alpha}} = \frac{\varphi(0) + \rho\bar{\alpha}\varphi'(0) - \varphi(0)}{\bar{\alpha}} = \rho\varphi'(0) > \sigma\varphi'(0),$$

deoarece $\sigma > \rho$ şi $\varphi'(0) < 0$. Ca atare, din continuitatea lui φ', există un interval $(a,b) \subset (0, \bar{\alpha})$ centrat în β astfel încât $\varphi'(\alpha) \geq \sigma\varphi'(0)$ pentru orice $\alpha \in (a,b)$. Deci, prin construcție pentru orice $\alpha \in (a,b)$ au loc simultan relațiile: $\varphi(\alpha) < \varphi(0) + \rho\alpha\varphi'(0)$ şi $\varphi'(\alpha) \geq \sigma\varphi'(0)$, adică condițiile Wolfe. ■

Condiția Wolfe (4.3.6), $f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k$, cere ca lungimea pasului α_k să fie astfel încât să se realizeze o reducere suficientă a lui f. De aceea (4.3.6) se mai numeşte *condiția de reducere suficientă*. Cu alte cuvinte, aceasta forțează ca rata de reducere a funcției f în direcția d_k să fie cel puțin $\rho\nabla f(x_k)^T d_k$. În practică $\rho = 10^{-4}$. Orice pas suficient de mic poate satisface condiția de reducere suficientă. De aceea pentru a evita paşi foarte mici, total ineficienți în căutarea liniară, se introduce a doua condiție (4.3.10), $\nabla f(x_{k+1})^T d_k \geq \sigma\nabla f(x_k)^T d_k$, numită *condiția de curbură*. Aceasta asigură o mărginire a lui $\nabla f(x_k + \alpha d_k)^T d_k$. Valoarea tipică a lui σ este 0.9. În general cu cât valoarea lui σ este mai mică, cu atât căutarea liniară este mai exactă. Totuşi, cu cât σ este mai mic cu atât efortul de calcul este mai mare.

Observăm că inegalitatea (4.3.10), a doua condiție Wolfe, este o aproximare a condiției de ortogonalitate $g_{k+1}^T d_k = 0$, care este satisfăcută de căutarea liniară exactă. Un dezavantaj (minor) al condiției (4.3.10) este că aceasta nu se reduce la căutarea liniară exactă atunci când $\sigma \to 0$. Totuşi dacă în locul lui (4.3.10) punem condiția

$$\left|\nabla f(x_{k+1})^T d_k\right| \leq -\sigma\nabla f(x_k)^T d_k, \tag{4.3.11}$$

atunci condițiile Wolfe se reduc la căutarea liniară exactă când $\sigma \to 0$. Relațiile (4.3.6) şi (4.3.11) constituie un alt set de condiții care permit determinarea lungimii pasului, cunoscute drept *condițiile Wolfe tari*. Deseori acestea sunt utilizate mai ales în demonstrarea proprietăților de convergență ale algoritmilor de optimizare fără restricții. Este important să notăm faptul că condițiile Wolfe tari sunt satisfăcute pentru α din intervalul $[a,b]$ construit în demonstrația teoremei 4.3.3.

Aceasta rezultă din faptul că deoarece $\varphi'(\beta) < 0$, adică $\varphi'(\beta) > \sigma\varphi'(0)$ implică $|\varphi'(\beta)| < \sigma|\varphi'(0)|$.

În algoritmii de optimizare fără restricții, şi în general în algoritmii de optimizare, deseori se utilizează condițiile Wolfe. Algoritmi pentru rezolvarea condițiilor Wolfe au fost elaborați, între alții, de Dennis şi Schnabel [1989], Fletcher [1987], Lemaréchal [1981], Moré şi Thuente [1990] şi Hager şi Zhang [2005]. Este important să notăm faptul că eficiența unui algoritm de optimizare depinde în mod crucial de acuratețea cu care este calculată lungimea pasului α_k la fiecare iterație. De aceea aceşti algoritmi sunt foarte sofisticați, constituind interpretări foarte subtile ale condițiilor Wolfe. De exemplu, algoritmul de rezolvare a condițiilor Wolfe dat de Moré şi Thuente [1990] este utilizat în pachetul L-BFGS (BFGS cu memorie limitată) elaborat de Nocedal [], precum şi în pachetul de gradient conjugat PRP+ al lui Liu, Nocedal şi Waltz [].

Hager şi Zhang [2005] prezintă o abordare diferită a acestor condiții de mare subtilitate şi eficiență prin introducerea *condițiilor Wolfe aproximative*. Pentru a vedea motivația introducerii acestora să observăm că în esența lor condițiile Wolfe, mai ales prima condiție Wolfe, implică o problemă numerică fundamentală. În figura 4.3.2 prezentăm graficul numeric al funcției $f(x) = 1 - 2x + x^2$ în vecinătatea punctului 1, mai precis în intervalul $x \in [0.999999999, 1.000000001]$.

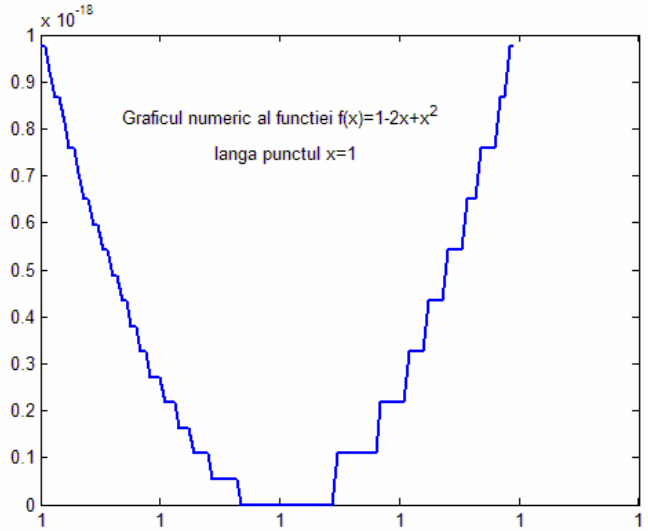


Fig. 4.3.2. Graficul numeric al funcției $f(x)=1-2x+x^2$ lângă punctul $x=1$.

Graficul, generat cu Matlab, este obținut prin evaluarea funcției în 100 de puncte din acest interval şi unirea acestor puncte prin segmente de linie. Evident, graficul real al funcției este o parabolă, în timp ce graficul calculat este constant pe porțiuni.

Întotdeauna când minimizăm o funcție netedă avem imaginea că graficul acestei funcții este neted. Dar, în realitate, reprezentarea în calculator a funcției de minimizat este constantă pe porțiuni. Din figura 4.3.2 observăm că există un interval

$$I = [0.999999999840, 1.000000000220]$$

în jurul lui $x = 1$ în care funcția f ia valoarea zero. Fiecare punct din acest interval este de fapt un minim al funcției f din calculator. Pe de altă parte, funcția f (adevărata funcție f, nu cea din calculator) are un unic punct de minim în $x = 1$. Observăm că lungimea acestui interval este 0.38×10^{-9} care este mult mai mare decât epsilon maşină, care în cazul nostru este 0.54×10^{-19}. Existența intervalului I este rezultatul scăderii unor numere de valori foarte apropiate când se evaluează funcția f. Într-adevăr, lângă $x = 1$, observăm că $1 - 2x$ este aproape -1 în timp ce x^2 este aproape $+1$. Deci, când calculatorul adună $1 - 2x$ cu x^2, în esență el scade două numere foarte apropiate. Aceasta confirmă încă odată regula cunoscută din analiza numerică că *erorile de rotunjire apar când se scad numere aproape egale*. Observăm că lungimea intervalului I este de ordinul lui radical din epsilon maşină. Cu aceste preparative să ne întoarcem la condițiile Wolfe.

După cum ştim condițiile Wolfe sunt date de (4.3.6) şi (4.3.10). În termenii funcției φ acestea se exprimă sub forma:

$$\rho\varphi'(0) \geq \frac{\varphi(\alpha_k) - \varphi(0)}{\alpha_k}, \tag{4.3.12}$$

$$\varphi'(\alpha_k) \geq \sigma\varphi'(0). \tag{4.3.13}$$

Observăm imediat că într-o vecinătate a minimului local al funcției f deoarece $\varphi(\alpha) \cong \varphi(0)$, rezultă că din punct de vedere numeric condiția (4.3.12) este cel mai greu de satisfăcut. Scăderea $\varphi(\alpha_k) - \varphi(0)$ este total lipsită de acuratețe. De fapt, dacă $\varphi(0)$ corespunde la un punct din intervalul I, atunci prima condiție Wolfe (4.3.12) nu va fi niciodată satisfăcută deoarece membrul drept din (4.3.12) întotdeauna este negativ, iar membrul stâng poate fi numai nenegativ (presupunând că d_k este descendentă). De aceea, o soluție pentru a evita această dificultate este de a construi un set de relații echivalente condițiilor de căutare liniară Wolfe în care să nu apară scăderi. Pentru aceasta, dacă înlocuim funcția $\varphi(\alpha)$ cu un polinom de interpolare de gradul doi $q(\alpha) = a\alpha^2 + b\alpha + c$, care realizează condițiile de interpolare $q(0) = \varphi(0)$, $q'(0) = \varphi'(0)$ şi $q'(\alpha_k) = \varphi'(\alpha_k)$, atunci obținem

$$a = \frac{\varphi'(\alpha_k) - \varphi'(0)}{2\alpha_k}, \quad b = \varphi'(0) \quad \text{şi} \quad c = \varphi(0).$$

Acum evaluând membrul drept din (4.3.12) utilizând q în locul lui φ, obținem:

$$\frac{\varphi(\alpha_k) - \varphi(0)}{\alpha_k} \cong \frac{q(\alpha_k) - q(0)}{\alpha_k} = \frac{\varphi'(\alpha_k) + \varphi'(0)}{2}, \tag{4.3.14}$$

în care, după cum se vede, esențial estre prezența semnului plus. Ca atare, prima condiție Wolfe (4.3.12) devine $(2\rho-1)\varphi'(0) \geq \varphi'(\alpha_k)$, care împreună cu (4.3.13) furnizează *condițiile Wolfe aproximative:*

$$(2\rho-1)\varphi'(0) \geq \varphi'(\alpha_k) \geq \sigma\varphi'(0). \qquad (4.3.15)$$

Dacă funcția f este pătratică, atunci aproximarea (4.3.14) este exactă. Accentuăm faptul că prima inegalitate din (4.3.15) este o aproximație a primei condiții Wolfe. Dar, ceea ce este foarte important, această aproximație se poate evalua cu o acuratețe mult mai mare decât condiția Wolfe originală, mai ales când iterațiile sunt lângă minimul local al funcției f, deoarece condițiile Wolfe aproximative sunt exprimate în termenii derivatei şi nu a unei diferențe de funcții [Hager şi Zhang, 2005].

4.3.4. Backtracking

Deseori, în practică, mai ales când se cunoaşte că direcția de căutare este „bună", pentru căutarea liniară se utilizează numai condiția (4.3.6), adică numai prima condiție Goldstein. Ideea de bază a acestei proceduri de căutare este de a pleca cu valoarea $\alpha_k = 1$. Dacă punctul $x_k + \alpha_k d_k$ nu este acceptabil, atunci reducem valoarea lui α_k până când prima condiție a lui Goldstein (4.3.6) $f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k$ este satisfăcută. Această metodă se numeşte *căutarea liniară inexactă cu backtracking*. Un algoritm simplu de backtracking este următorul.

Algoritmul 4.3.1. (Căutare liniară cu backtracking)

Pasul 1.	*Se consideră scalarii* $\rho \in (0,1/2)$, $0 < l < u < 1$ *şi se pune* $\alpha_k = 1$.
Pasul 2.	*Se testează* $f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k$.
Pasul 3.	*Dacă* $f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k$ *nu este satisfăcută, atunci se pune* $\alpha_k = \beta\alpha_k$, *unde* $\beta \in [l,u]$ *şi se continuă cu pasul 2; altfel, stop cu* α_k *obținut la pasul anterior.* ♦

Observăm că backtracking-ul este foarte asemănător cu regula lui Armijo. Backtracking-ul este deseori utilizat în cadrul metodelor Newton sau quasi-Newton, unde se cunoaşte că marea majoritate a paşilor de deplasare sunt de lungime unitară, ceea ce înseamnă că o singură testare a lui (4.3.6) este suficientă pentru a obține o lungime convenabilă a pasului de deplasare.

O variantă ceva mai sofisticată a căutării liniare inexacte cu backtracking constă în a obține α_k nu printr-o simplă reducere a acestuia printr-un factor luat dintr-un interval din $(0,1)$ dat, ca în algoritmul de mai sus, ci utilizând interpolarea pătratică. Într-adevăr, fie funcția $\varphi(\alpha) = f(x_k + \alpha d_k)$. La început cunoaştem $\varphi(0) = f(x_k)$ şi $\varphi'(0) = \nabla f(x_k)^T d_k$. Luând un pas de lungime unitară cunoaştem

$\varphi(1) = f(x_k + d_k)$. Dacă $f(x_k + d_k)$ nu satisface (4.3.6), atunci pentru a aproxima $\varphi(\alpha)$ se poate utiliza următorul model pătratic

$$p(\alpha) = \left(\varphi(1) - \varphi(0) - \varphi'(0)\right)\alpha^2 + \varphi'(0)\alpha + \varphi(0),$$

care ține seama de condițiile de interpolare de mai sus. Din condiția de minim $p'(\alpha) = 0$ obținem o nouă valoare pentru α ca:

$$\alpha = -\frac{\varphi'(0)}{2\left(\varphi(1) - \varphi(0) - \varphi'(0)\right)},$$

cu care se poate continua căutarea lui α_k. Pentru a preveni ca α să nu fie prea mic se pot introduce anumite protecții. De exemplu, dacă $\|\alpha_k d_k\| < \eta$, atunci căutarea liniară se oprește, unde η este dat.

O variantă simplificată a algoritmului de căutare liniară cu backtracking este următoarea.

Algoritmul 4.3.2. (Căutare liniară cu backtracking, variantă)

Pasul 1.	*Se consideră scalarii* $\rho \in (0,1/2)$, $\beta \in (0,1)$ *şi se pune* $\alpha_k = 1$.
Pasul 2.	*Dacă* $f(x_k + \alpha_k d_k) \le f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k$, *atunci stop; altfel se continuă cu pasul 3.*
Pasul 3.	*Se pune* $\alpha_k = \beta\alpha_k$ *şi se continuă cu pasul 2.* ♦

Deoarece d_k este o direcție descendentă, atunci $\nabla f(x_k)^T d_k < 0$, astfel încât pentru valori suficient de mici ale lui α_k are loc

$$f(x_k + \alpha_k d_k) \cong f(x_k) + \alpha_k \nabla f(x_k)^T d_k < f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k,$$

care arată finitudinea algoritmului. Constanta ρ se interpretează ca fracția de reducere a valorilor lui f, predicționată prin extrapolare liniară. De obicei ρ se alege între 0.001 şi 0.5, ceea ce înseamnă că acceptăm o reducere a lui f între 1% şi 50% din predicția bazată pe extrapolare liniară. Parametrul β se alege între 0.1 şi 0.9.

Următoarea teoremă arată proprietățile căutării liniare cu backtracking.

Teorema 4.3.4. *Fie funcția* $f : R^n \to R$ *continuu diferențiabilă şi mulțimea de nivel constant* $\left\{x \in R^n : f(x) \le f(x_0)\right\}$ *un compact. Presupunem că pentru orice* k, $\nabla f(x_k)^T d_k < 0$. *Atunci algoritmul de backtracking 4.3.2 determină într-un număr finit de iterații o valoare* $\alpha_k > 0$ *care verifică condiția din pasul 2. Şirul* $x_{k+1} = x_k + \alpha_k d_k$ *satisface* $f(x_{k+1}) < f(x_k)$, *pentru orice* k *şi*

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

Demonstraţie. Vom demonstra că algoritmul de backtracking nu poate cicla nedefinit între paşii 2 şi 3. Într-adevăr, dacă se întâmplă acest lucru, atunci pasul 2 nu va fi niciodată îndeplinit şi deci

$$\frac{f(x_k+\beta^j d_k)-f(x_k)}{\beta^j}>\rho\nabla f(x_k)^T d_k.$$

Dar $\beta\in(0,1)$, deci $\beta^j\to 0$ când $j\to\infty$, şi deci pentru $j\to\infty$ inegalitatea de mai sus devine $\nabla f(x_k)^T d_k>\rho\nabla f(x_k)^T d_k$, care nu este posibilă, deoarece $\rho\in(0,1/2)$ şi $\nabla f(x_k)^T d_k\neq 0$.

Deci există un j_0 astfel încât

$$f(x_k+\beta^{j_0}d_k)-f(x_k)\le\rho\beta^{j_0}\nabla f(x_k)^T d_k\neq 0.$$

Cum $\nabla f(x_k)^T d_k\neq 0$, rezultă că pentru orice k, $f(x_{k+1})<f(x_k)$. Acum din inegalitatea de mai sus obţinem

$$\frac{f(x_k+\beta^{j_0}d_k)-f(x_k)}{\beta^{j_0}\|d_k\|}\le\rho\frac{\nabla f(x_k)^T d_k}{\|d_k\|}.$$

Aplicând teorema de medie, obţinem

$$\frac{\nabla f(x_k+\beta^{j_0}d_k)^T d_k}{\|d_k\|}\le\rho\frac{\nabla f(x_k)^T d_k}{\|d_k\|},$$

de unde rezultă concluzia teoremei. ■

În finalul acestei secţiuni menţionăm faptul că utilizând aceeaşi tehnică ca în cazul căutării liniare Wolfe se poate obţine *căutarea liniară cu backtracking aproximativ*. Aceasta este chiar prima inegalitate din (4.3.15).

Condiţia din căutarea liniară cu backtracking este suficientă pentru a realiza o îmbunătăţire a valorilor funcţiei de minimizat, totuşi aceasta nu este încă destul de tare pentru a garanta că orice punct de acumulare al algoritmului general de direcţii descendente va fi un punct staţionar al funcţiei f. În fapt, dacă $n\ge 2$ este foarte posibil ca direcţia de descendenţă d_k să fie selectată astfel încât unghiul dintre d_k şi $-\nabla f(x_k)$ să fie foarte aproape de 90^0. Atunci cosinusul acestui unghi tinde la zero, caz în care algoritmul de direcţii descendente poate converge la un punct nestaţionar. Pentru a evita această situaţie vom impune ca cosinusul unghiului de mai sus să fie uniform mărginit faţă de zero. Aşadar, după cum vom vedea în paragraful următor, vom impune ca direcţiile de căutare să formeze un con ascuţit centrat în $-\nabla f(x_k)$, adică să satisfacă aşa numita condiţie de unghi:

$$\nabla f(x_k)^T d_k \le -\tau \|\nabla f(x_k)\| \|d_k\|,$$

unde $\tau \in (0,1)$.

4.4. TEORIA CONVERGENȚEI PENTRU CĂUTARE LINIARĂ INEXACTĂ

În această secțiune prezentăm teoremele de convergență ale algoritmului general de direcții descendente echipat cu metode de căutare liniară inexactă. Proprietățile de convergență ale algoritmului de direcții descendente pot fi studiate prin stabilirea unei măsuri a cât de bună este direcția de căutare. Calitatea direcției de căutare este definită de unghiul dintre direcția de căutare şi gradientul cu semn schimbat. Pentru a demonstra convergența algoritmului 4.1.1 trebuie să evităm situația în care direcția de căutare $s_k = \alpha_k d_k$ este ortogonală pe direcția negativă a gradientului. Altfel spus, unghiul θ_k dintre s_k şi $-\nabla f(x_k)$ este uniform mărginit față de 90^0, adică pentru orice k

$$\theta_k \le \frac{\pi}{2} - \mu, \tag{4.4.1}$$

unde $\mu > 0$. Inegalitatea (4.4.1) se numeşte *condiția de unghi*. Fie $g_k = \nabla f(x_k)$, atunci unghiul $\theta_k \in [0, \pi/2]$, este definit ca:

$$\cos\theta_k = -\frac{g_k^T s_k}{\|g_k\| \|s_k\|}. \tag{4.4.2}$$

În continuare să prezentăm un rezultat foarte util, datorat lui Zoutendijk [1970]. Din demonstrația teoremei care urmează ne putem face o idee clară asupra modului în care proprietățile funcției de minimizat se leagă de condițiile Wolfe.

Teorema 4.4.1. *Presupunem că f este mărginită inferior pe R^n şi că este continuu diferențiabilă într-o vecinătate N a mulțimii de nivel constant $S = \{x : f(x) \le f(x_0)\}$. Presupunem că gradientul este o funcție Lipschitz continuă, adică există constanta $L > 0$, astfel încât*

$$\|g(x) - g(y)\| \le L\|x - y\|, \tag{4.4.3}$$

pentru orice $x, y \in N$. Considerăm o iterație de forma (4.1.2), unde d_k este o direcție descendentă şi α_k satisface condițiile Wolfe (4.3.6) şi (4.3.10), atunci

$$\sum_{k \ge 1} \cos^2\theta_k \|g_k\|^2 < \infty. \tag{4.4.4}$$

Demonstrație. Din (4.3.10) obținem

$$(g_{k+1} - g_k)^T d_k \geq (\sigma - 1) g_k^T d_k .$$

Pe de altă parte, condiția Lipschitz (4.4.3) arată că

$$(g_{k+1} - g_k)^T d_k \geq \alpha_k L \|d_k\|^2 .$$

Combinând aceste două relații, obținem

$$\alpha_k \geq \left(\frac{\sigma - 1}{L} \right) \frac{g_k^T d_k}{\|d_k\|^2} . \tag{4.4.5}$$

Utilizând prima condiție Wolfe (4.3.6) şi (4.4.5) obținem

$$f(x_{k+1}) \leq f(x_k) + \rho \left(\frac{\sigma - 1}{L} \right) \frac{(g_k^T d_k)^2}{\|d_k\|^2} .$$

Ținând seama de (4.4.2), relația de mai sus se poate scrie sub forma

$$f(x_{k+1}) \leq f(x_k) + c \cos^2 \theta_k \|g_k\|^2 ,$$

unde $c = \rho(\sigma - 1)/L$. Adunând aceste expresii pentru $k \geq 1$ şi ținând seama de faptul că f este mărginită inferior, obținem concluzia teoremei. ■

Relația (4.4.4) se numeşte *condiția Zoutendijk*. Deseori aceasta intervine în mod direct în obținerea de rezultate privind convergența globală a algoritmului general de direcții descendente. Într-adevăr, dacă pentru orice k iterațiile $x_{k+1} = x_k + \alpha_k d_k$ sunt astfel încât $\cos\theta_k \geq \delta > 0$, atunci din (4.4.4) obținem $\lim_{k\to\infty} \|g_k\| = 0$. Cu alte cuvinte, dacă direcția de căutare nu tinde să fie ortogonală gradientului, atunci şirul gradienților este convergent la zero. Observăm că condiția Zoutendijk este o condiție asupra direcției de căutare. Dacă, de exemplu, direcția de căutare este chiar $-g_k$, adică este vorba de algoritmul de gradient descendent al lui Cauchy [1847], atunci $\cos\theta_k = 1$. Aceasta arată că pentru a face algoritmul lui Cauchy global convergent este necesar doar să executăm o căutare liniară adecvată. Aceasta arată încă odată importanța căutării liniare în cadrul algoritmului de direcție descendentă.

În acest context se poate imediat demonstra următorul corolar al condiției Zoutendijk.

Corolar. *Fie $f : R^n \to R$ mărginită inferior pe R^n şi continuu diferențiabilă într-o vecinătate N a mulțimii de nivel constant $S = \{x : f(x) \leq f(x_0)\}$. Dacă $\{x_k\}$ este şirul generat de algoritmul 4.1.1 în care sunt utilizate condițiile Wolfe, atunci*

$$\cos\theta_k \|\nabla f(x_k)\| \to 0 .$$

Dacă $\cos\theta_k \geq \delta > 0$ pentru toți k, atunci din condiția lui Zoutendijk avem că $\|\nabla f(x_k)\| \to 0$, adică algoritmul general 4.1.1 echipat cu căutarea liniară Wolfe converge la un punct staționar al funcției f.

Limita $\lim_{k\to\infty}\|g_k\| = 0$ este cel mai bun tip de rezultat de convergență globală care se poate obține în cadrul acestor algoritmi. Clar, nu se poate garanta că algoritmul converge la minim, ci numai că acesta intră în zona de atracție a punctelor staționare.

Algoritmul general de direcții descendente cu căutare liniară inexactă este foarte simplu şi se prezintă sub forma următoare.

Algoritmul 4.4.1. (Algoritm de optimizare fără restricții cu căutare liniară inexactă)

Pasul 1.	*Inițializare*. Se consideră un punct inițial $x_0 \in R^n$, precum şi toleranța de terminare a iterațiilor algoritmului de minimizare $\varepsilon \ll 1$. Se pune $k = 0$.
Pasul 2.	Dacă $\|g_k\| \leq \varepsilon$, atunci stop; altfel se calculează o direcție de descendență d_k astfel încât $g_k^T d_k < 0$.
Pasul 3.	Se calculează lungimea pasului α_k utilizând condițiile Goldstein (4.3.6) şi (4.3.7) sau condițiile Wolfe (4.3.6) şi (4.3.10).
Pasul 4.	Se pune $x_{k+1} = x_k + \alpha_k d_k$; $k = k+1$, şi se continuă cu pasul 2. ♦

Vedem că în acest algoritm d_k este o direcție descendentă generală care trebuie să satisfacă condiția $g_k^T d_k < 0$. Pe de altă parte α_k se determină fie din condițiile Goldstein, fie din cele ale lui Wolfe. Ca atare, algoritmul 4.4.1 este unul foarte general, incluzând o clasă foarte bogată de metode de optimizare fără restricții cu căutare liniară inexactă.

În cele ce urmează vom demonstra o teoremă privind convergența globală a algoritmului general de direcții descendente cu căutare liniară inexactă.

Teorema 4.4.2. *Presupunem că în algoritmul 4.4.1 direcția d_k satisface condiția de unghi (4.4.1) şi α_k este definit de condițiile Goldstein (4.3.6) şi (4.3.7) sau de condițiile Wolfe (4.3.6) şi (4.3.10). Dacă $\nabla f(x)$ există şi este uniform continuu pe mulțimea de nivel constant $S = \{x : f(x) \leq f(x_0)\}$, atunci fie $\nabla f(x_k) = 0$ pentru un anumit k, fie $f(x_k) \to -\infty$, fie $\nabla f(x_k) \to 0$.*

Demonstrație. Pentru început să considerăm α_k definit de condițiile Goldstein (4.3.6) şi (4.3.7). Presupunem că pentru toți k, $g_k = \nabla f(x_k) \neq 0$ şi că $f(x_k)$ este

mărginit inferior. Rezultă că $f(x_k) - f(x_{k+1}) \to 0$, deci din (4.3.6) $-g_k^T s_k \to 0$. Acum presupunem că $g_k \to 0$ nu este adevărată. Atunci, există un subşir astfel încât $\|g_k\| \geq \varepsilon$ şi $\|s_k\| \to 0$. Deoarece $\theta_k \leq \pi/2 - \mu$, obținem

$$\cos\theta_k \geq \cos\left(\frac{\pi}{2} - \mu\right) = \sin\mu,$$

deci

$$-g_k^T s_k \geq \|g_k\|\|s_k\|\sin\mu \geq \varepsilon\|s_k\|\sin\mu.$$

Dar, din dezvoltarea Taylor, putem scrie

$$f(x_{k+1}) = f(x_k) + \nabla f(\xi_k)^T s_k,$$

unde ξ_k se află pe segmentul de linie care uneşte x_k cu x_{k+1}. Din continuitatea uniformă rezultă că $\nabla f(\xi_k) \to g_k$ când $s_k \to 0$. Deci

$$f(x_{k+1}) = f(x_k) + g_k^T s_k + o(\|s_k\|).$$

Ca atare,

$$\frac{f(x_k) - f(x_{k+1})}{-g_k^T s_k} \to 1,$$

ceea ce contrazice a doua condiție Goldstein (4.3.7). Deci $g_k \to 0$, ceea ce demonstrează convergența globală când α_k este determinat din condițiile Goldstein.

În mod similar se demonstrează convergența pentru cazul condițiilor Wolfe. Observăm că din continuitatea uniformă a lui $\nabla f(x)$ rezultă că

$$g_{k+1}^T d_k = g_k^T d_k + o(\|d_k\|),$$

astfel încât

$$\frac{g_{k+1}^T d_k}{g_k^T d_k} \to 1.$$

Dar aceasta contrazice faptul că $g_{k+1}^T d_k / g_k^T d_k \leq \sigma < 1$ din (4.3.10). Deci $g_k \to 0$, ceea ce demonstrează convergența globală când α_k este determinat din condițiile Wolfe. ■

Următoarea teoremă demonstrează convergența algoritmului general de direcții descendente cu căutarea liniară inexactă dată de regula lui Wolfe. În esență vom demonstra că şirul generat de algoritmul 4.4.1 cu regula Wolfe satisface condiția Zoutendijk.

Teorema 4.4.3. *Fie $f : R^n \to R$ o funcție continuu diferențiabilă şi mărginită inferior. Presupunem că $\nabla f(x)$ este uniform continuă pe mulțimea de nivel*

constant $S = \{x : f(x) \le f(x_0)\}$ *şi că* α_k *este determinat din condițiile Wolfe (4.3.6) şi (4.3.10). Atunci şirul generat de algoritmul 4.4.1 satisface*

$$\lim_{k \to +\infty} \frac{\nabla f(x_k)^T s_k}{\|s_k\|} = 0, \tag{4.4.6}$$

ceea ce înseamnă că

$$\|\nabla f(x_k)\| \cos\theta_k \to 0. \tag{4.4.7}$$

Demonstrație. Deoarece $\nabla f(x_k)^T s_k < 0$ şi f este mărginită inferior, rezultă că şirul $\{x_k\}$ este bine definit şi $\{x_k\} \subset S$. Mai mult, deoarece $\{f(x_k)\}$ este un şir descendent, atunci el este convergent.

Vom demonstra (4.4.6) prin contradicție. Presupunem deci că (4.4.6) nu are loc. Atunci există $\varepsilon > 0$ şi un subşir indexat după mulțimea de indici K, astfel încât pentru orice $k \in K$

$$-\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \ge \varepsilon .$$

Dar, din prima condiție Wolfe (4.3.6) avem

$$f(x_k) - f(x_{k+1}) \ge \rho \|s_k\| \left(-\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \right) \ge \rho \|s_k\| \varepsilon .$$

Deoarece $\{f(x_k)\}$ este un şir convergent, rezultă că şirul $\{s_k : k \in K\}$ este convergent la zero. Pe de altă parte din a doua condiție Wolfe (4.3.10), pentru orice $k \ge 0$ avem

$$(1-\sigma)(-\nabla f(x_k)^T s_k) \le \left(\nabla f(x_k + s_k) - \nabla f(x_k)\right)^T s_k .$$

Deci, pentru orice $k \in K$

$$\varepsilon \le -\frac{\nabla f(x_k)^T s_k}{\|s_k\|} \le \frac{1}{1-\sigma} \|\nabla f(x_k + s_k) - \nabla f(x_k)\| . \tag{4.4.8}$$

Dar, întru-cât am demonstrat că $\{s_k : k \in K\} \to 0$, rezultă că membrul drept din (4.4.8) tinde la zero prin continuitatea uniformă a lui ∇f pe mulțimea S. Aceasta constituie o contradicție care demonstrează teorema. ■

Observăm că (4.4.6) implică $\|\nabla f(x_k)\| \cos\theta_k \to 0$, adică chiar condiția Zoutendijk, exprimată în (4.4.4) în altă formă. Importanța condiției Zoutendijk constă în faptul că dacă $\cos\theta_k \ge \delta > 0$, atunci $\lim_{k\to\infty} \|\nabla f(x_k)\| = 0$. Dacă ipoteza continuității uniforme este înlocuită cu condiția Lipschitz asupra gradientului, atunci obținem un alt rezultat asupra convergenței algoritmului 4.4.1 cu căutarea inexactă dată de condițiile Wolfe. Înainte de a demonstra o astfel de

teoremă să prezentăm un rezultat tehnic sub forma următoarei propoziții, care furnizează o margine a reducerii funcției f.

Propoziția 4.4.1. *Fie $f : D \to R$ o funcție continuu diferențiabilă pe mulțimea deschisă $D \subset R^n$ cu gradientul $\nabla f(x)$ care satisface condiția Lipschitz*

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

unde $L > 0$ este o constantă. Dacă $f(x_k + \alpha d_k)$ este mărginită inferior şi $\alpha > 0$, atunci pentru orice $\alpha_k > 0$ care satisface condițiile Wolfe (4.3.6) şi (4.3.10) avem

$$f(x_k) - f(x_k + \alpha_k d_k) \ge \gamma \|\nabla f(x_k)\|^2 \cos^2 \langle d_k, \nabla f(x_k)\rangle, \qquad (4.4.9)$$

unde $\gamma > 0$ este o constantă.

Demonstrație. Din condiția Lipschitz a lui ∇f şi (4.3.10) obținem

$$\alpha_k L \|d_k\|^2 \ge d_k^T \left(\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\right) \ge -(1-\sigma) d_k^T \nabla f(x_k),$$

adică

$$\begin{aligned}\alpha_k \|d_k\| &\ge \frac{1-\sigma}{L\|d_k\|}\|d_k\|\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle \\ &= \frac{1-\sigma}{L}\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle.\end{aligned}$$

Acum utilizând (4.3.6) obținem

$$\begin{aligned}f(x_k) - f(x_k + \alpha_k d_k) &\ge -\rho\alpha_k d_k^T \nabla f(x_k) \\ &= \rho\alpha_k \|d_k\|\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle \\ &\ge \rho\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle \frac{1-\sigma}{L}\|\nabla f(x_k)\|\cos\langle d_k, -\nabla f(x_k)\rangle \\ &= \frac{\rho(1-\sigma)}{L}\|\nabla f(x_k)\|^2 \cos^2\langle d_k, -\nabla f(x_k)\rangle,\end{aligned}$$

care, după cum se poate vedea este chiar (4.4.9) cu $\gamma = \rho(1-\sigma)/L$. ■

Teorema 4.4.4. *Fie $f(x)$ o funcție continuu diferențiabilă pe R^n pentru care $\nabla f(x)$ satisface condiția Lipschitz*

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

unde $L > 0$ este o constantă. De asemenea fie lungimea pasului α_k definit de condițiile Wolfe (4.3.6) şi (4.3.10). Dacă condiția de unghi (4.4.1) este satisfăcută, atunci pentru şirul $\{x_k\}$ generat de algoritmul general de direcții descendente 4.4.1, fie $\nabla f(x_k) = 0$ pentru un anumit k, fie $f(x_k) \to -\infty$, fie $\nabla f(x_k) \to 0$.

Demonstrație. Presupunem că pentru orice k, $\nabla f(x_k) \neq 0$. Din propoziția 4.4.1 rezultă că

$$f(x_k) - f(x_{k+1}) \geq \gamma \left\|\nabla f(x_k)\right\|^2 \cos^2 \theta_k ,$$

unde $\gamma = \rho(1-\sigma)/L$ este o constantă pozitivă independentă de k. Atunci, pentru orice $k > 0$ putem scrie

$$f(x_0) - f(x_k) = \sum_{i=0}^{k-1} \left(f(x_i) - f(x_{i+1}) \right) \geq \gamma \min_{0 \leq i \leq k} \left\|\nabla f(x_i)\right\|^2 \sum_{i=0}^{k-1} \cos^2 \theta_i . \quad (4.4.10)$$

Deoarece θ_k satisface condiția de unghi (4.4.1), rezultă că

$$\sum_{k=0}^{\infty} \cos^2 \theta_k = +\infty .$$

Ca atare, din (4.4.10) rezultă că fie $\nabla f(x_k) \to 0$ sau $f(x_k) \to -\infty$, ceea ce completează demonstrația teoremei. ■

Observăm imediat că teorema de mai sus este o consecință directă a condiției Zoutendijk şi a celei de unghi. Mai mult, toate aceste rezultate au fost obținute în condiții destul de comode asupra funcției de minimizat. Ceea ce se impune sunt doar continuu diferențiabilitatea şi condiția Lipschitz asupra gradientului.

În cele ce urmează prezentăm un rezultat privind reducerea funcției de minimizat în condițiile în care aceasta este uniform convexă. Într-adevăr, în propoziția 4.4.1 am prezentat o margine a reducerii la fiecare iterație a unei funcții foarte generale, în condițiile în care gradientul acesteia este o funcție Lipschitz. Estimația (4.4.9) este exprimată în funcție de condiția Zoutendijk. Uniform convexitatea ne conduce la o nouă estimație dependentă de marginile Hessianului funcției de minimizat. Am văzut că o funcție $f(x)$ este uniform convexă (sau tare convexă), dacă există o constantă $\eta > 0$ astfel încât

$$(y-z)^T \left(\nabla f(y) - \nabla f(z) \right) \geq \eta \left\| y - z \right\|^2 , \quad (4.4.11)$$

sau există constantele pozitive m şi M $(m < M)$ astfel încât

$$m \left\| y \right\|^2 \leq y^T \nabla^2 f(x) y \leq M \left\| y \right\|^2 . \quad (4.4.12)$$

Teorema 4.4.5. *Fie funcția $f(x)$ uniform convexă şi algoritmul 4.4.1 în care α_k satisface (4.3.6). Atunci*

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{\rho\eta}{1 + \sqrt{M/m}} \left\| \alpha_k d_k \right\|^2 . \quad (4.4.13)$$

Demonstrație. Considerăm două cazuri.

(i) Presupunem că $d_k^T \nabla f(x_k + \alpha_k d_k) \leq 0$. În acest caz putem scrie

$$
\begin{aligned}
f(x_k)-f(x_k+\alpha_k d_k) &= \int_0^{\alpha_k} -d_k^T \nabla f(x_k+td_k)\mathrm{d}t \\
&\geq \int_0^{\alpha_k} d_k^T \left(\nabla f(x_k+\alpha_k d_k)-\nabla f(x_k+td_k)\right)\mathrm{d}t \\
&\geq \int_0^{\alpha_k} \|d_k\|^2 \eta(\alpha_k-t)\mathrm{d}t = \frac{1}{2}\|\alpha_k d_k\|^2 \eta \\
&\geq \frac{\rho\eta}{1+\sqrt{M/m}}\|\alpha_k d_k\|^2 .
\end{aligned}
$$

(ii) Presupunem că $d_k^T \nabla f(x_k+\alpha_k d_k) > 0$. Atunci există $0<\alpha^* < \alpha_k$, astfel încât

$$d_k^T \nabla f(x_k+\alpha^* d_k) = 0 .$$

Cu aceasta din uniform convexitatea (4.4.12) rezultă că

$$f(x_k)-f(x_k+\alpha^* d_k) \leq \frac{1}{2} M \|\alpha^* d_k\|^2 \tag{4.4.14}$$

şi

$$f(x_k+\alpha_k d_k)-f(x_k+\alpha^* d_k) \geq \frac{1}{2} m \|(\alpha_k-\alpha^*)d_k\|^2 . \tag{4.4.15}$$

Deoarece $f(x_k+\alpha_k d_k) < f(x_k)$, din (4.4.14) şi (4.4.15) rezultă că

$$\alpha_k \leq \left(1+\sqrt{\frac{M}{m}}\right)\alpha^* .$$

Deci

$$
\begin{aligned}
f(x_k)-f(x_k+\alpha_k d_k) &\geq -\rho\alpha_k d_k^T \nabla f(x_k) \\
&\geq \rho\alpha_k d_k^T \left(\nabla f(x_k+\alpha^* d_k)-\nabla f(x_k)\right) \\
&\geq \eta\rho\alpha_k\alpha^* \|d_k\|^2 \geq \frac{\rho\eta}{1+\sqrt{M/m}}\|\alpha_k d_k\|^2 .
\end{aligned}
$$

Ca atare, (4.4.13) are loc în ambele cazuri. ■

Teoria convergenței algoritmului de direcții descendente cu căutare liniară inexactă prezentată în această secțiune se bazează pe condiția de descendență $\nabla f(x_k)^T d_k < 0$ şi pe presupunerea că direcția d_k este mărginită în normă față de zero. Rezultatele obținute sunt foarte tari, deoarece se bazează pe ipoteze foarte comode, anume că funcția este cel puțin de două ori continuu diferențiabilă şi gradientul acesteia este funcție Lipschitz. Deci pentru funcții suficient de netede, aşa cum se prezintă majoritatea modelelor sistemelor fizice reale, teoria asigură convergența algoritmului 4.4.1.

În cele ce urmează vom considera o altă abordare a convergenței algoritmului general de direcții descendente 4.4.1 cu căutare liniară inexactă pentru

funcții tare convexe în situația în care direcția de căutare satisface o *condiție de descendență suficientă*. Într-adevăr să presupunem că direcția d_k satisface condiția de descendență suficientă:

$$g_k^T d_k \le -c\|g_k\|^2, \tag{4.4.16}$$

unde $c > 0$ este o constantă.

Teorema 4.4.6. *Dacă:*

(i) funcția f este mărginită inferior pe mulțimea de nivel constant $S = \{x \in R^n : f(x) \le f(x_0)\}$, unde $x_0 \in R^n$ este punctul inițial cu care se începe căutarea minimului,

(ii) gradientul $g(x) = \nabla f(x)$ este o funcție Lipschitz continuă pe o mulțime convexă $\bar{S}$ care conține S, adică există $L > 0$ astfel încât $\|g(x) - g(y)\| \le L\|x - y\|$, pentru orice $x, y \in \bar{S}$

şi d_k este o direcție descendentă care verifică $g_k^T d_k < 0$, atunci algoritmul de direcții descendente cu metodele de căutare liniară inexactă Armijo, Goldstein, Wolfe sau Wolfe tare generează un şir infinit $\{x_k\}$ care satisface condiția Zoutendijk

$$\lim_{k\to\infty}\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2 = 0. \tag{4.4.17}$$

Demonstrație. Pentru algoritmul cu căutare liniară inexactă Armijo (4.3.3) fie $K_1 = \{k : \alpha_k = \tau_k\}$ şi $K_2 = \{k : \alpha_k < \tau_k\}$. Atunci din (4.3.3) rezultă că

$$f(x_k) - f(x_k + \alpha_k d_k) \ge -\rho\tau_k g_k^T d_k, \quad k \in K_1,$$
$$f(x_k) - f(x_k + \alpha_k d_k) \ge -\rho\alpha_k g_k^T d_k, \quad k \in K_2.$$

Deoarece pentru orice $k \in K_2$, $\alpha_k / \beta \le \tau_k$, rezultă că

$$f(x_k) - f(x_k + (\alpha_k / \beta) d_k) < -\rho\alpha_k g_k^T d_k / \beta, \; k \in K_2.$$

Utilizând teorema de medie asupra membrului stâng al inegalității de mai sus, există un $\theta_k \in [0,1]$ astfel încât

$$-\alpha_k \nabla f(x_k + \alpha_k\theta_k d_k / \beta)^T d_k / \beta < -\rho\alpha_k g_k^T d_k / \beta, \; k \in K_2,$$

adică pentru orice $k \in K_2$

$$\nabla f(x_k + \alpha_k\theta_k d_k / \beta)^T d_k > \rho g_k^T d_k.$$

Din *(ii)* şi inegalitatea Cauchy-Schwarz, rezultă că pentru orice $k \in K_2$

$$\begin{aligned}\alpha_k L\|d_k\|^2 / \beta &\ge \|\nabla f(x_k + \alpha_k\theta_k d_k / \beta) - \nabla f(x_k)\|\|d_k\| \\ &\ge (\nabla f(x_k + \alpha_k\theta_k d_k / \beta) - \nabla f(x_k))^T d_k \ge -(1-\rho) g_k^T d_k.\end{aligned}$$

Deci, pentru orice $k \in K_2$

$$\alpha_k \geq -\frac{\beta(1-\rho)g_k^T d_k}{L\|d_k\|^2}.$$

Ca atare, pentru orice $k \in K_2$

$$f(x_k) - f(x_k + \alpha_k d_k) > \frac{\beta\rho(1-\rho)}{L}\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2.$$

Dar $\tau_k = -(\nabla f(x_k)^T d_k)/\|d_k\|^2$. Deci pentru orice $k \in K_1$

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \rho\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2.$$

Dacă punem $\eta = min\{\rho, \beta\rho(1-\rho)/L\}$, atunci din relațiile de mai sus obținem

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \eta\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2,$$

care din ipoteza de descendență şi *(i)* implică (4.4.17).

Pentru algoritmul cu căutare liniară dată de regula lui Goldstein (4.3.7) utilizând teorema de medie, precum şi inegalitatea Cauchy-Schwarz, rezultă că există un $\theta_k \in [0,1]$, astfel încât:

$$\begin{aligned}(1-\rho)\alpha_k \nabla f(x_k)^T d_k &\leq f(x_k + \alpha_k d_k) - f(x_k) = \alpha_k \nabla f(x_k + \alpha_k \theta_k d_k)^T d_k \\ &= \alpha_k \left[\nabla f(x_k + \alpha_k \theta_k d_k) - \nabla f(x_k)\right]^T d_k + \alpha_k \nabla f(x_k)^T d_k \\ &\leq \alpha_k \nabla f(x_k)^T d_k + \alpha_k \|\nabla f(x_k + \alpha_k \theta_k d_k) - \nabla f(x_k)\| \|d_k\| \\ &\leq \alpha_k \nabla f(x_k)^T d_k + \alpha_k^2 L \|d_k\|^2.\end{aligned}$$

Deci

$$\alpha_k \geq \frac{\rho}{L}\left(\frac{-g_k^T d_k}{\|d_k\|^2}\right).$$

Acum, ținând seama de (4.3.6), obținem:

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{\rho^2}{L}\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2.$$

Ca atare, din *(i)* rezultă că în cazul căutării liniare cu regula lui Goldstein, relația (4.4.17) este adevărată.

Pentru algoritmul de direcții descendente cu căutare liniară dată de regula lui Wolfe, din *(ii),* (4.3.10) şi ținând seama de inegalitatea Cauchy-Schwarz, obținem:

$$\alpha_k L \|d_k\|^2 \geq \|\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\| \|d_k\|$$

$$\geq \left[\nabla f(x_k + \alpha_k d_k) - \nabla f(x_k)\right]^T d_k \geq -(1-\sigma) g_k^T d_k.$$

Deci

$$\alpha_k \geq \frac{1-\sigma}{L}\left(\frac{-g_k^T d_k}{\|d_k\|^2}\right).$$

Acum din (4.3.6) rezultă:

$$f(x_k) - f(x_k + \alpha_k d_k) \geq \frac{\rho(1-\sigma)}{L}\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2.$$

Ca atare, din *(ii)* rezultă că şi în cazul căutării liniare Wolfe, (4.4.17) este adevărată. ■

În concluzie, pentru *toate* metodele de căutare liniară inexactă de mai sus se poate demonstra că:

1) lungimea pasului de deplasare α_k *este mărginită inferior şi*

2) pentru orice k *:*

$$f(x_k + \alpha_k d_k) - f(x_k) \leq -\eta_0\left(\frac{-g_k^T d_k}{\|d_k\|}\right)^2, \tag{4.4.18}$$

unde $\eta_0 > 0$. *Aceste două rezultate foarte importante le vom utiliza în demonstrația convergenței algoritmului de direcții descendente 4.4.1.*

Pentru a stabili convergența algoritmului conceptual de direcții descendente 4.4.1 ne vom plasa în *clasa funcțiilor tare convexe*. Menționăm că funcțiile tare convexe constituie o clasă foarte naturală de funcții care au proprietăți foarte bune în raport cu care se pot demonstra teoreme de convergență. Dacă f este de două ori continuu diferențiabilă şi tare convexă pe R^n atunci ipotezele *(i)* şi *(ii)* din teorema 4.4.6 sunt satisfăcute, $f(x)$ are un minim unic x^* şi există constantele $0 < m \leq M$ astfel încât pentru orice $x, y \in R^n$:

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2, \tag{4.4.19}$$

$$\frac{1}{2} m\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2} M\|x - x^*\|^2, \tag{4.4.20}$$

$$m\|x - y\|^2 \leq (\nabla f(x) - \nabla f(y))^T (x - y) \leq M\|x - y\|^2, \tag{4.4.21}$$

adică

$$m\|x - x^*\|^2 \leq \nabla f(x)^T (x - x^*) \leq M\|x - x^*\|^2. \tag{4.4.22}$$

Utilizând inegalitatea Cauchy-Schwartz şi teorema de medie în (4.4.22) obținem:

$$m\left\|x-x^*\right\| \le \left\|\nabla f(x)\right\| \le M\left\|x-x^*\right\|. \tag{4.4.23}$$

Teorema 4.4.7. *Presupunem că* $f: R^n \to R$ *este o funcție tare convexă, de două ori continuu diferențiabilă, şi că*

$$g_k^T d_k \le -\tau\left\|g_k\right\|\left\|d_k\right\|, \tag{4.4.24}$$

unde $0 < \tau \le 1$, *atunci algoritmul de direcții descendente cu oricare din metodele de căutare liniară inexactă generează un şir infinit* $\{x_k\}$ *astfel încât:*

$$\lim_{k\to\infty}\left\|g_k\right\| = 0, \tag{4.4.25}$$

şi $\{x_k\}$ *converge cel puțin liniar la* x^*.

Demonstrație. Deoarece f este tare convexă rezultă că există constantele $0 < m \le M$ astfel încât pentru orice $x \in R$ are loc:

$$M\left\|x-x^*\right\| \ge \left\|\nabla f(x)\right\| \ge m\left\|x-x^*\right\|.$$

Din (4.4.21) observăm că $m \le L$. Din (4.4.18), utilizând noua condiție de descendență (4.4.24) putem scrie:

$$f(x_k) - f(x_k+\alpha_k d_k) \ge \eta_0\left(\frac{-g_k^T d_k}{\left\|d_k\right\|}\right)^2 = \eta_0\left(\frac{-g_k^T d_k}{\left\|d_k\right\|\left\|g_k\right\|}\right)^2\left\|g_k\right\|^2$$

$$\ge \eta_0\tau^2\left\|g_k\right\|^2 \ge \eta_0\tau^2 m^2\left\|x_k-x^*\right\|^2 \ge \eta_0\tau^2 m^2\frac{2}{M}\left(f(x_k)-f(x^*)\right).$$

Notăm:

$$\gamma = \sqrt{2\frac{\eta_0\tau^2 m^2}{M}} \quad \text{şi} \quad \zeta = \sqrt{1-\gamma^2}.$$

Deci:

$$f(x_k) - f(x_k+\alpha_k d_k) \ge \gamma^2\left(f(x_k)-f(x^*)\right). \tag{4.4.26}$$

Să arătăm acum că pentru oricare din metodele de căutare liniară prezentate mai sus $\gamma \le 1$.

Pentru algoritmul care utilizează metoda de căutare liniară inexactă Armijo, deoarece

$$\eta_0 = min\{\rho, \beta\rho(1-\rho)/L\} \le \beta\rho(1-\rho)/L$$

rezultă că:

$$\gamma^2 = 2\frac{\eta_0\tau^2 m^2}{M} = \frac{2\beta\rho(1-\rho)\tau^2 m^2}{LM} \le \frac{\beta(1-\rho)\tau^2 m}{M} \le \beta\tau^2(1-\rho) < 1.$$

Pentru algoritmul care utilizează metoda de căutare liniară inexactă Goldstein, $\eta_0 = \rho^2 / L$, atunci

$$\gamma^2 = 2\frac{\eta_0 \tau^2 m^2}{M} = \frac{2\rho^2\tau^2 m^2}{LM} \le \frac{\rho\tau^2 m}{M} \le \tau^2\rho < 1.$$

Pentru algoritmul care utilizează metodele de căutare liniară inexactă Wolfe sau Wolfe tare, $\eta_0 = \rho(1-\sigma)/L$, şi deci

$$\gamma^2 = 2\frac{\eta_0 \tau^2 m^2}{M} = \frac{2\rho(1-\sigma)\tau^2 m^2}{LM} \le \frac{(1-\sigma)\tau^2 m}{M} \le (1-\sigma)\tau^2 < 1.$$

Deci, pentru oricare din metodele de căutare liniară inexactă considerate, avem $\gamma \le 1$. Acum, din (4.4.26) putem scrie:

$$f(x_{k+1}) - f(x_k) \le -\gamma^2 (f(x_k) - f(x^*)),$$

sau

$$f(x_k) - f(x_{k-1}) \le -\gamma^2 (f(x_{k-1}) - f(x^*)).$$

Scăzând $f(x^*)$ din ambii membri ai acestei inegalități, obținem:

$$\begin{aligned} f(x_k) - f(x^*) &\le (1-\gamma^2)(f(x_{k-1}) - f(x^*)) \\ &= \zeta^2 (f(x_{k-1}) - f(x^*)) \le \cdots \le \zeta^{2k} (f(x_0) - f(x^*)). \end{aligned}$$

Dar

$$\begin{aligned} \left\| x_k - x^* \right\|^2 &\le \frac{2}{m}(f(x_k) - f(x^*)) \le \frac{2}{m}\zeta^{2k}(f(x_0) - f(x^*)) \\ &= \frac{2(f(x_0) - f(x^*))}{m}\zeta^{2k} = \omega^2 \zeta^{2k}, \end{aligned}$$

unde

$$\omega^2 = \frac{2(f(x_0) - f(x^*))}{m}.$$

Deci $\left\| x_k - x^* \right\| \le \omega\zeta^k$, ceea ce arată convergența liniară a şirului $\{x_k\}$ la x^*. ■

Teorema de mai sus arată că pentru funcții tare convexe suficient de netede dacă direcția d_k verifică condiția de descendență (4.4.24), atunci algoritmii de metode descendente pentru minimizarea funcției f, echipați cu oricare dintre metodele de căutare liniară inexactă de mai sus, sunt convergenți în sensul lui (4.4.25). Se poate demonstra că teorema este validă şi in cazul în care condiția (4.4.24) este înlocuită cu condiția de descendență suficientă $g_k^T d_k \le -c\left\| g_k \right\|^2$, unde c este o constantă pozitivă.

În concluzie, vedem că pentru metodele de direcții descendente pentru optimizare fără restricții dispunem de tehnici de căutare liniară care asigură convergența algoritmului 4.4.1.

4.5. CONSIDERAȚII PRACTICE

După cum am văzut, pentru determinarea lungimii pasului de-a lungul unei direcții descendente se dispune de o multitudine de metode care se pot clasifica sub forma *căutării liniare exacte* şi a *căutării liniare inexacte* sau *aproximative*. În practică, cu excepția funcțiilor pătratice, se utilizează căutarea liniară inexactă. Cele mai importante metode de căutare liniară inexactă sunt:

1) Căutarea liniară Armijo [Armijo, 1966]**:**
Dându-se scalarii $\beta \in (0,1)$, $\tau_k = -(\nabla f(x_k)^T d_k)/\|d_k\|^2$ şi $\rho \in (0,1/2)$. Se consideră $\alpha_k = \beta^{m_k}\tau_k$, unde m_k este primul întreg nenegativ m pentru care

$$f(x_k) - f(x_k + \beta^m \tau_k d_k) \geq -\rho\beta^m \tau_k \nabla f(x_k)^T d_k, \qquad (4.5.1)$$

adică se încearcă succesiv $m = 0,1,\ldots$ până când inegalitatea de mai sus este satisfăcută pentru un anumit $m = m_k$.

2) Căutarea liniară Goldstein [Goldstein, 1965]**:**
Pentru un $0 < \rho < 1/2$, se determină α_k ca soluție a următorului sistem de inegalități:

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho\alpha\nabla f(x_k)^T d_k, \qquad (4.5.2)$$

$$f(x_k + \alpha d_k) \geq f(x_k) + (1-\rho)\alpha\nabla f(x_k)^T d_k. \qquad (4.5.3)$$

3) Căutarea liniară Wolfe [Wolfe, 1969, 1971]**:**
Dându-se scalarii ρ şi σ, astfel încât $0 < \rho \leq \sigma < 1$, se determină α_k ca soluție a următorului sistem de inegalități:

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho\alpha\nabla f(x_k)^T d_k, \qquad (4.5.4)$$

$$\nabla f(x_k + \alpha d_k)^T d_k \geq \sigma\nabla f(x_k)^T d_k. \qquad (4.5.5)$$

4) Căutarea liniară Wolfe aproximativă [Hager şi Zhang, 2005]**:**
Dându-se scalarii $0 < \rho < 1/2$ şi $\rho \leq \sigma < 1$, se determină α_k ca soluție a următorului sistem de inegalități:

$$(2\rho - 1)\nabla f(x_k)^T d_k \geq \nabla f(x_k + \alpha d_k)^T d_k \geq \sigma\nabla f(x_k)^T d_k. \qquad (4.5.6)$$

5) Căutarea liniară Wolfe tare [Wolfe, 1969, 1971]**:**
Dându-se scalarii ρ şi σ, astfel încât $0 < \rho < \sigma < 1$, se determină α_k ca soluție a următorului sistem de inegalități:

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho\alpha\nabla f(x_k)^T d_k, \qquad (4.5.7)$$

$$\left|\nabla f(x_k + \alpha d_k)^T d_k\right| \leq -\sigma\nabla f(x_k)^T d_k. \qquad (4.5.8)$$

6) Căutarea liniară Wolfe generalizată [Dai şi Yuan, 1996]**:**
Dându-se scalarii ρ, σ_1 şi σ_2, astfel încât $0 < \rho < \sigma_1 < 1$ şi $\sigma_2 \geq 0$, se determină α_k ca soluție a următorului sistem de inegalități:

$$f(x_k + \alpha d_k) \leq f(x_k) + \rho\alpha\nabla f(x_k)^T d_k, \tag{4.5.9}$$

$$\sigma_1 \nabla f(x_k)^T d_k \leq \nabla f(x_k + \alpha d_k)^T d_k \leq -\sigma_2 \nabla f(x_k)^T d_k. \tag{4.5.10}$$

7) Căutare liniară cu backtracking:
Se consideră scalarii $\rho \in (0,1/2)$, $\beta \in (0,1)$ şi se pune $\alpha_k = 1$. Se pune $\alpha_k = \beta\alpha_k$ până când

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \rho\alpha_k \nabla f(x_k)^T d_k \tag{4.5.11}$$

este îndeplinită.

Câteva observații sunt necesare. În anumiți algoritmii de optimizare fără restricții pentru demonstrarea convergenței se utilizează condițiile Wolfe. În alți algoritmi se utilizează condițiile Wolfe tari. Observăm că dacă $\sigma_1 = \sigma_2 = \sigma$, atunci condițiile Wolfe generalizate se reduc la condițiile Wolfe tari. A doua inegalitate din (4.5.6) este chiar a doua condiție Wolfe (4.5.5). În cazul funcțiilor pătratice, prima inegalitate din (4.5.6) este echivalentă cu (4.5.4). În general, când $\varphi(\alpha) = f(x_k + \alpha d_k)$ este înlocuită cu un polinom de interpolare pătratică care interpolează $\varphi(\alpha)$ în $\alpha = 0$ şi $\varphi'(\alpha)$ în $\alpha = 0$ şi $\alpha = \alpha_k$, atunci prima condiție Wolfe (4.5.4) se reduce la prima inegalitate din (4.5.6). Vedem imediat că condițiile Wolfe aproximative au aceeaşi formă ca şi condițiile Wolfe generalizate (4.5.10), dar cu o alegere particulară a lui σ_2. Condiția de reducere a valorii funcției de minimizat (4.5.4) se regăseşte în condițiile Wolfe generalizate, în timp ce în condițiile Wolfe aproximative această condiție este aproximată în prima inegalitate din (4.5.6).

În acest capitol atât pentru căutarea liniară exactă cât şi pentru cea inexactă am demonstrat teoreme care ne asigură că în condiții destul de modeste sistemele de inegalități de mai sus au o soluție care precizează lungimea pasului de-a lungul direcției d_k. Acestea sunt rezultate incontestabile, care odată demonstrate dau certitudinea existenței şi deci a posibilității calculului lui α_k în algoritmii de optimizare fără restricții. Problema care rămâne este aceea a proiectării de algoritmi eficienți care să rezolve sistemele de inegalități de mai sus. De fapt, aceasta este una dintre marile provocări în practica optimizării.

Algoritmi pentru determinarea lui α_k care verifică sistemele de mai sus sunt descrişi, de exemplu, în [Lemaréchal, 1981], [Fletcher, 1987], [Shanno şi Phua, 1987], [Dennis şi Schnabel, 1983], [Schultz, Schabel şi Byrd, 1985], [Moré şi Tuenthe, 1990], [Hager şi Zhang, 2005]. Rezolvarea sistemelor de inegalități algebrice nu este o sarcină uşoară. De aceea, de obicei, algoritmii de căutare liniară bazați pe condițiile de mai sus conțin foarte mult empirism şi se bazează pe tehnicile de interpolare sau cele de reducere a intervalului de incertitudine,

prezentate în capitolul 3. Dintre toți algoritmii de căutare liniară prezentați mai sus, în practică, cei mai utilizați sunt căutarea liniară cu backtracking, căutarea liniară Wolfe, căutare liniară Wolfe tare şi foarte recent căutarea liniară Wolfe aproximativă. Nu vom prezenta algoritmii corespunzători acestor metode de căutare. Dar, pentru a vedea complexitatea, modul în care experiența empirică este utilizată în construcția unor algoritmi de căutare liniară vom prezenta algoritmul de căutare bazat pe condițiile Wolfe. Ceea ce este important de notat este faptul că acesta conține foarte multe rezultate empirice observate de-a lungul utilizării acestuia prin rezolvarea a mii de probleme de optimizare fără restricții.

Să considerăm deci algoritmul pentru determinarea unei valori α care verifică *condițiile Wolfe* (4.5.4) şi (4.5.5). Presupunem că se dispune de punctul curent x_k, de valoarea funcției şi a gradientului în acest punct $f_k = f(x_k)$, şi $g_k = \nabla f(x_k)$, de o direcție descendentă d_k, precum şi de o valoare inițială a lui α cu care se încep calculele. Atunci algoritmul bazat pe condițiile Wolfe este următorul [Shanno şi Phua, 1976], [Birgin şi Martínez, 2001].

Algoritmul 4.5.1. (Căutare liniară Wolfe – Shanno-Phua)

Pasul 1.	*Inițializare.* Se consideră $\alpha_p = 0$, $lsmax = 20$ şi $lsiter = 0$. ($lsmax$ este numărul maxim de iterații admis în acest algoritm.)
Pasul 2.	Se calculează noul punct $x_{new} = x_k + \alpha d_k$, precum şi $f_{new} = f(x_{new})$ şi $g_{new} = \nabla f(x_{new})$.
Pasul 3.	Se verifică dacă condițiile Wolfe sunt satisfăcute. Dacă $\left(\alpha \lVert d_k \rVert > 10^{-30}\right)$ şi $\left(lsiter < lsmax\right)$ şi $\left(g_{new}^T d_k \neq 0 \text{ sau } f_{new} \geq f_k\right)$ şi $\left[\left(f_{new} > f_k + 10^{-4} \alpha g_k^T d_k \text{ sau } \left\lvert g_{new}^T d_k / g_k^T d_k \right\rvert > 0.9\right) \text{ sau } \left(lsiter = 0 \text{ şi } \left\lvert g_{new}^T d_k / g_k^T d_k \right\rvert > 0.5\right)\right]$, atunci se execută pasul 4, altfel stop.
Pasul 4.	Se verifică dacă noul punct x_{new} are o pantă negativă şi o valoare a funcției mai mare decât cea corespunzătoare lui $\alpha = 0$. În acest caz căutarea a trecut printr-un maxim local şi este direcționată către un minim aflat la distanță de minimul căutat. Se reduce α şi se calculează punctul corespunzător acestei valori ale lui α, precum şi valorea funcției şi a gradientului în acest nou punct. Se repetă această procedură până când s-a găsit un punct corespunzător. Dacă $\left(\alpha \lVert d_k \rVert > 10^{-30} \text{ şi } f_{new} > f_k \text{ şi } g_{new}^T d_k < 0\right)$, atunci se execută pasul 5, altfel se continuă cu pasul 6.
Pasul 5.	Se pune $\alpha = \alpha / 3$. Se calculează $x_{new} = x_k + \alpha d_k$, $f_{new} = f(x_{new})$ şi $g_{new} = \nabla f(x_{new})$. Se pune $\alpha_p = 0$ şi se continuă cu pasul 4.
Pasul 6.	Interpolare cubică pentru a determina un nou punct de încercare. Se calculează:

	$a = g_k^T d_k + g_{new}^T d_k - \dfrac{3(f_k - f_{new})}{\alpha_p - \alpha}$, $b = a^2 - (g_k^T d_k)(g_{new}^T d_k)$. Dacă $b > 0$, atunci se pune $b = \sqrt{b}$, altfel se pune $b = 0$. Se calculează $$\alpha_t = \alpha - \frac{(\alpha - \alpha_p)(g_{new}^T d_k + b - a)}{g_{new}^T d_k - g_k^T d_k + 2b}.$$
Pasul 7.	Se verifică dacă minimul este inclus într-un interval. Dacă $\dfrac{g_{new}^T d_k}{g_k^T d_k} \le 0$, atunci se execută pasul 8, altfel se continuă cu pasul 9.
Pasul 8.	Aici minimul este inclus într-un interval. Se verifică dacă punctul de încercare se află suficient de bine inclus în interval. Dacă nu, atunci se alege α_t ca mijlocul intervalului. Dacă $\left[0.99 max\{\alpha, \alpha_p\} < \alpha_t \text{ sau } \alpha_t < 1.01 min\{\alpha, \alpha_p\}\right]$, atunci se pune $\alpha_t = (\alpha + \alpha_p)/2$, altfel: Aici minimul nu se află într-un interval, α_t este prea mic de aceea se dublează cea mai mare lungime anterioară. Dacă $\left(g_{new}^T d_k < 0 \text{ şi } \alpha_t < 1.01 max\{\alpha, \alpha_p\}\right)$, atunci $\alpha_t = 2 max\{\alpha, \alpha_p\}$. Aici α_t este prea mare, de aceea se înjumătățeşte cea mai mică valoare anterioară. Dacă $\left[\left(g_{new}^T d_k > 0 \text{ şi } \alpha_t > 0.99 min\{\alpha, \alpha_p\}\right) \text{ sau } (\alpha_t < 0)\right]$, atunci $\alpha_t = min\{\alpha, \alpha_p\}/2$.
Pasul 9.	Se pune $\alpha_p = \alpha$, $\alpha = \alpha_t$ şi se continuă căutarea. Se calculează noul punct $x_{new} = x_k + \alpha d_k$, $f_{new} = f(x_{new})$ şi $g_{new} = \nabla f(x_{new})$. Se pune $lsiter = lsiter + 1$ şi se continuă cu pasul 3. ♦

Algoritmul nu este simplu. Vedem imediat că condițiile Wolfe sunt uşor de identificat în cadrul algoritmului, dar pe lângă acestea se află o multitudine de alte teste care asigură buna funcționare a acestuia. Toate aceste teste suplimentare sunt introduse în urma experimentelor numerice. Lungimea pasului se determină prin interpolare cubică şi actualizarea intervalului de incertitudine[1].

Mult mai complicat este algoritmul dezvoltat de Moré şi Thuente [1992, 1994] pentru determinarea lungimii pasului prin *condițiile Wolfe tari* (4.5.7) şi (4.5.8). Şi acesta se bazează pe experiența acumulată în rezolvarea unei mari

[1] Detalii asupra acestui algoritm se află în codul Fortran prezentat pe CD-ul anexat lucrării (subrutina WOLFE.FOR). Vezi şi programul LSTEST.FOR, care arată cum se inițializează căutarea liniară Wolfe implementată în această subrutină.

varietăți de probleme de optimizare fără restricții. Problema considerată de Moré şi Thuente este de a găsi un α care să aparțină mulțimii

$$T(\rho)=\{\alpha>0:\varphi(\alpha)\le\varphi(0)+\alpha\rho\varphi'(0),|\varphi'(\alpha)|\le\rho|\varphi'(0)|\},$$

unde $\varphi(\alpha)=f(x_k+\alpha d_k)$. În plus se impune ca α să fie restricționat într-un interval, $0<\alpha_{\min}\le\alpha\le\alpha_{\max}$, condiție care asigură finitudinea algoritmului. Ideea algoritmului este următoarea. Dată o lungime a pasului α_0 dintr-un interval de incertitudine $[\alpha_{\min},\alpha_{\max}]$, algoritmul de căutare generează un lanț de intervale $\{I_k\}$, şi un şir de lungimi $\alpha_k\in I_k\cap[\alpha_{\min},\alpha_{\max}]$. ($I_0=[0,\infty]$.) Intervalele I_k trebuie astfel construite ca intersecția lor cu $T(\rho)$ să fie nevidă. Condițiile impuse asupra capetelor α_l şi α_u ale unui interval I sunt specificate în termenii unei funcții auxiliare $\phi(\alpha)=\varphi(\alpha)-\varphi(0)-\rho\alpha\varphi'(0)$. Dacă capetele intervalului satisfac condițiile

$$\phi(\alpha_l)\le\phi(\alpha_u),\ \ \phi(\alpha_l)\le 0 \ \text{ şi } \ \phi'(\alpha_l)(\alpha_u-\alpha_l)<0, \qquad (4.5.12)$$

atunci există un α^* în I astfel încât $\phi(\alpha^*)\le\phi(\alpha_l)$ şi $\phi'(\alpha^*)=0$. În particular $\alpha^*\in T(\rho)\cap I$. Dată o valoare α_t din intervalul I, atunci capetele α_l^+ şi α_u^+ ale intervalului actualizat I_+ sunt determinate conform următorului algoritm:

U1.	Dacă $\phi(\alpha_t)>\phi(\alpha_l)$, atunci $\alpha_l^+=\alpha_l$ şi $\alpha_u^+=\alpha_t$.
U2.	Dacă $\phi(\alpha_t)\le\phi(\alpha_l)$ şi $\phi'(\alpha_t)(\alpha_l-\alpha_t)>0$, atunci $\alpha_l^+=\alpha_t$ şi $\alpha_u^+=\alpha_u$.
U3.	Dacă $\phi(\alpha_t)\le\phi(\alpha_l)$ şi $\phi'(\alpha_t)(\alpha_l-\alpha_t)<0$, atunci $\alpha_l^+=\alpha_t$ şi $\alpha_u^+=\alpha_l$.

Acest algoritm de actualizare a intervalului se poate utiliza pentru a determina un α care satisface condițiile Wolfe tari exprimate mai sus prin intermediul mulțimii $T(\rho)$. Inițial $\alpha_l=0$ şi $\alpha_u=\infty$. Se consideră o valoare $\alpha_t\in[\alpha_{\min},\alpha_{\max}]$. Dacă cazurile U1 şi U3 sunt îndeplinite, atunci s-a determinat un interval cu capetele α_l şi α_u care satisface (4.5.12). Dacă, pe de altă parte, condiția U2 este îndeplinită atunci procedura se poate repeta cu un α_t^+ din intervalul $[\alpha_t,\alpha_{\max}]$. În acest fel, atât timp cât U2 este îndeplinită continuăm să generăm valori de încercare din intervalul $[\alpha_l,\alpha_{\max}]$. De-a lungul acestui proces impunem ca şi valoarea $\alpha_{\max}$ să fie utilizată ca o valoare de încercare. Aceasta se poate realiza prin alegerea

$$\alpha_t^+\in\left[min\{\delta_{\max}\alpha_t,\alpha_{\max}\},\alpha_{\max}\right], \qquad (4.5.13)$$

unde factorul $\delta_{\max}>1$. Deoarece inițial $\alpha_l=0$ şi $\phi(0)=0$, atunci dacă U2 este satisfăcută, şirul $\alpha_0,\alpha_1,\ldots$ este crescător cu

$$\phi(\alpha_k)\le 0 \ \text{ şi } \ \phi'(\alpha_k)<0,\ \ k=0,1,\ldots. \qquad (4.5.14)$$

Algoritmul de căutare se termină cu $\alpha_{\max}$ dacă $\phi(\alpha_{\max}) \le 0$ şi $\phi'(\alpha_{\max}) < 0$. Astfel, după un număr finit de încercări, algoritmul se termină fie cu $\alpha_{\max}$, fie generează un interval pentru care capetele satisfac (4.5.12).

Cu acest interval care satisface (4.5.12) repetăm procedura de actualizare de mai sus pentru a rafina intervalul. Dacă algoritmul de căutare nu generează un interval I în $[\alpha_{\min}, \alpha_{\max}]$ care să satisfacă (4.5.12), atunci şirul $\{\alpha_k\}$ de valori de încercare este descrescător cu

$$\phi(\alpha_k) > 0 \text{ sau } \phi'(\alpha_k) \ge 0,\ k = 0,1,\ldots. \tag{4.5.15}$$

Dacă se utilizează un α_t cu $\phi(\alpha_t) \le 0$ şi $\phi'(\alpha_t) < 0$, şi ne plasăm în cazurile U2 sau U3, atunci algoritmul de actualizare a intervalului arată că intervalul I_+ se află la dreapta lui α_t. Deoarece $\alpha_t \ge \alpha_{\min}$, atunci intervalul I_+ conține $[\alpha_{\min}, \alpha_{\max}]$. Dacă ne plasăm în cazul U1, atunci $\phi(\alpha_l) < 0$, şi deci $\alpha_l \ge \alpha_{\min}$. Ca atare, intervalul actualizat conține $[\alpha_{\min}, \alpha_{\max}]$. Ca mai sus, în această analiză forțăm algoritmul ca să utilizeze $\alpha_{\min}$ ca valoare de încercare când (4.5.15) are loc şi $\alpha_{\min} > 0$. Aceasta se face prin alegerea

$$\alpha_t^+ \in \left[\alpha_{\min}, max\left\{\delta_{\min}\alpha_t, \alpha_{\min}\right\}\right], \tag{4.5.16}$$

unde factorul $\delta_{\min} < 1$. (Moré şi Thuente consideră $\delta_{\max} = 1.1$ şi $\delta_{\min} = 7/12$.) Observăm că alegerile (4.5.13) şi (4.5.16) sunt asiguratorii în sensul că (4.5.13) este utilizat numai când (4.5.14) are loc, în timp ce (4.5.16) este utilizat doar când (4.5.15) are loc. Dacă algoritmul de căutare generează un interval I din $[\alpha_{\min}, \alpha_{\max}]$, atunci trebuie să introducem o regulă care să garanteze că alegerea lui α_t forțează ca lungimea lui I să tindă la zero. Moré şi Thuente sugerează că dacă lungimea lui I nu se reduce cu un factor $\delta < 1$ după două încercări, atunci se utilizează un pas de bisecție pentru a obține o nouă valoare α_t. (Moré şi Thuente consideră $\delta = 0.66$.) După cum se vede algoritmul este destul de complicat beneficiind de o analiză foarte subtilă a funcțiilor reale de o variabilă reală[2]. Acest algoritm de căutare este utilizat în algoritmii de optimizare fără restricții LBFGS [Liu şi Nocedal, 1989], TNPACK [Schlick şi Fogelson, 1992a,b] şi CGPLUS [Gilbert şi Nocedal, 1992].

Mult mai elaborat este algoritmul de căutare bazat pe *condițiile Wolfe aproximative,* dezvoltat de Hager şi Zhang [2005]. Spre deosebire de algoritmul lui Moré şi Thuente, algoritmul lui Hager şi Zhang în toată analiza utilizează funcția $\varphi(\alpha)$ pentru a determina un interval de căutare $[\alpha_l, \alpha_u]$ care satisface condițiile:

$$\varphi(\alpha) \le \varphi(0) + \varepsilon_k,\ \ \varphi'(\alpha_l) < 0 \text{ şi } \varphi'(\alpha_u) \ge 0, \tag{4.5.17}$$

---

[2] Cititorul interesat poate consulta codul Fortran prezentat pe CD-ul anexat lucrării (subrutina MTLINES.FOR). Vezi şi programul LSTEST.FOR, care arată cum se inițializează căutarea liniară Wolfe tare implementată în această subrutină.

unde $\varepsilon_k \geq 0$ este o estimație a erorii în valoarea funcției f la iterația k. Condițiile (4.5.17) se numesc *condițiile de pantă opusă*. Avantajul utilizării condițiilor (4.5.17) constă în faptul că în precizia finită a calculatorului aceste condiții sunt de mai mare încredere decât condițiile (4.5.12) utilizate de Moré şi Thuente.

Dat un interval $[\alpha_l, \alpha_u]$ care satisface (4.5.17), precum şi un punct α_t Hager şi Zhang prezintă următoarea procedură de actualizare a acestui interval pentru a determina o lungime a pasului care să verifice condițiile Wolfe aproximative. Intrarea în această procedură este intervalul curent $[\alpha_l, \alpha_u]$ şi o valoare α_t generată fie prin procedura secantei, fie prin aceea a bisecției, proceduri pe care le vom explica puțin mai jos. Ieşirea din această procedură este intervalul actualizat $[\alpha_l^+, \alpha_u^+]$. Procedura este următoarea;

$[\alpha_l^+, \alpha_u^+] = update(\alpha_l, \alpha_u, \alpha_t)$

U0.	Dacă $\alpha_t \notin (\alpha_l, \alpha_u)$, atunci $\alpha_l^+ = \alpha_l$ şi $\alpha_u^+ = \alpha_u$.
U1.	Dacă $\varphi'(\alpha_t) \geq 0$, atunci $\alpha_l^+ = \alpha_l$ şi $\alpha_u^+ = \alpha_t$.
U2.	Dacă $\varphi'(\alpha_t) < 0$ şi $\varphi(\alpha_t) \leq \varphi(0) + \varepsilon_k$, atunci $\alpha_l^+ = \alpha_t$ şi $\alpha_u^+ = \alpha_u$.
U3.	Dacă $\varphi'(\alpha_t) < 0$ şi $\varphi(\alpha_t) > \varphi(0) + \varepsilon_k$, atunci $\alpha_l^+ = \alpha_l$, $\alpha_u^+ = \alpha_t$ şi se execută următorii paşi: a) Se pune $\delta = (1-\theta)\alpha_l^+ + \theta\alpha_u^+$. Dacă $\varphi'(\delta) \geq 0$, atunci $\alpha_u^+ = \delta$. b) Dacă $\varphi'(\delta) < 0$ şi $\varphi(\delta) \leq \varphi(0) + \varepsilon_k$, atunci $\alpha_l^+ = \delta$ şi se continuă cu pasul a). c) Dacă $\varphi'(\delta) < 0$ şi $\varphi(\delta) > \varphi(0) + \varepsilon_k$, atunci $\alpha_u^+ = \delta$ şi se continuă cu pasul a).

În procedura *update*, θ este un parametru din $(0,1)$. După execuția paşilor U1-U3 se obține un nou interval $[\alpha_l^+, \alpha_u^+] \subset [\alpha_l, \alpha_u]$ pentru care capetele α_l^+ şi α_u^+ verifică condițiile de pantă opusă (4.5.17). Observăm că ciclul scufundat în U3a-U3c este finit deoarece lungimea intervalului $\alpha_u^+ - \alpha_l^+$ tinde la zero, iar în capetele intervalului următoarele condiții au loc:

$$\varphi'(\alpha_l^+) < 0, \quad \varphi(\alpha_l^+) \leq \varphi(0) + \varepsilon_k,$$
$$\varphi'(\alpha_u^+) < 0, \quad \varphi(\alpha_u^+) > \varphi(0) + \varepsilon_k.$$

Intrarea α_t în procedura de actualizare de mai sus se generează printr-un algoritm special de tip secantă. Dacă α_t se obține prin procedura secantei bazată pe valorile funcției în α_l şi α_u, atunci

$$\alpha_t = \text{secant}(\alpha_l, \alpha_u) = \frac{\alpha_l \varphi'(\alpha_u) - \alpha_u \varphi'(\alpha_l)}{\varphi'(\alpha_u) - \varphi'(\alpha_l)}.$$

Hager şi Zhang utilizează următoarea procedură secantă:

$[\alpha_l^+, \alpha_u^+] = \text{secant}^2(\alpha_l, \alpha_u)$

S1.	$\alpha_t = \text{secant}(\alpha_l, \alpha_u)$ şi $[a,b] = update(\alpha_l, \alpha_u, \alpha_t)$.
S2.	Dacă $\alpha_t = b$, atunci $\alpha_t^+ = \text{secant}(\alpha_u, b)$.
S3.	Dacă $\alpha_t = a$, atunci $\alpha_t^+ = \text{secant}(\alpha_l, a)$.
S4.	Dacă $\alpha_t = a$ sau $\alpha_t = b$, atunci $[\alpha_l^+, \alpha_u^+] = update(a, b, \alpha_t^+)$. Altfel $[\alpha_l^+, \alpha_u^+] = [a,b]$.

Hager şi Zhang [2004] demonstrează că rata de convergența a procedurii secant^2 este $1+\sqrt{2} \approx 2.4$.

În continuare să prezentăm procedura care generează un interval inițial $[\alpha_l, \alpha_u]$ care satisface condițiile de pantă opusă (4.5.17). Se începe cu o estimație inițială $[0, \alpha_t]$. Procedura este următoarea:

$[\alpha_l, \alpha_u] = bracket(\alpha_t)$

B0.	Se inițializează $j = 0$ şi $\alpha_{t0} = \alpha_t$.
B1.	Dacă $\varphi'(\alpha_{tj}) \geq 0$, atunci $\alpha_u = \alpha_{tj}$ şi $\alpha_l = \alpha_{ti}$, unde $i < j$ este cel mai mare întreg astfel încât $\varphi(\alpha_{ti}) \leq \varphi(0) + \varepsilon_k$, stop.
B2.	Dacă $\varphi'(\alpha_{tj}) < 0$ şi $\varphi(\alpha_{tj}) > \varphi(0) + \varepsilon_k$, atunci $[\alpha_l, \alpha_u] = update(0, \alpha_{tj}, \alpha_{tj})$, stop.
B3.	Altfel, se pune $\alpha_{tj+1} = \omega \alpha_{tj}$, $j = j+1$ şi se continuă cu B1.

Parametrul $\omega > 0$ este factorul prin care α_{tj} este crescut la fiecare iterație a procedurii. Creşterea lui α_{tj} este continuată până când fie panta $\varphi'(\alpha_{tj})$ devine negativă, ceea ce înseamnă executarea lui B1, fie valoarea funcției $\varphi(\alpha_{tj})$ este destul de mare pentru a executa B2.

În finalul acestei discuții să prezentăm procedura utilizată pentru a genera valoarea α_t utilizată în procedura *bracket*. Dacă $k = 0$ (deci la prima iterație a algoritmului) şi dacă utilizatorul nu a specificat o valoare inițială în căutarea liniară

(ceea ce nu este cazul pentru căutarea Wolfe, sau căutarea Wolfe tare), atunci se generează una astfel:

a)	Dacă $x_0 \neq 0$, atunci $\alpha_t = \tau \|x_0\|_\infty / \|g_0\|_\infty$, stop.
b)	Dacă $f(x_0) \neq 0$, atunci $\alpha_t = \tau \|f(x_0)\| / \|g_0\|_\infty$, stop.
c)	Altfel, se pune $\alpha_t = 1$, stop.

În această inițializare parametrul $\tau \in (0,1)$ este un factor care reduce puțin valorile utilizate pentru calculul lui α_t. Raționamentul care a stat la baza acestei proceduri este următorul. Dacă $f(x) \approx c_1 + c_2 x^T x$, unde $c_2 > 0$, atunci clar minimul se atinge pentru $x = 0$. Lungimea pasului care realizează acest minim este $\alpha = \|x_0\|_\infty / \|g_0\|_\infty$. Întru-cât această estimație este departe de a fi una bună, atunci o înmulțim cu un scalar mic τ pentru a ne menține la următoarea iterație lângă x_0. Dacă $x_0 = 0$, atunci estimația de mai sus nu este bună şi ca atare considerăm aproximarea $f(x) \approx c_2 x^T x$ pentru care lungimea optimă a pasului este $\alpha = 2|f(x_0)| / \|g_0\|^2$. Din nou înmulțim această estimație cu τ pentru a fi lângă x_0 la următoarea iterație. Dacă, pe de altă parte, $f(x_0) = 0$, atunci din nou această estimație a lui α nu convine şi în acest caz se consideră $\alpha_t = 1$.

Pentru $k > 0$, deci la următoarele iterații ale algoritmului de minimizare a funcției f, Hager şi Zhang sugerează o tehnică utilizată în căutarea liniară Wolfe, anume de a utiliza valoarea de la pasul anterior α_{k-1}. Dacă $\varphi(\tau_1 \alpha_{k-1}) \leq \varphi(0)$, unde $\tau_1 \in (0,1)$ este un factor suficient de mic, şi dacă polinomul de interpolare corespunzător valorilor $\varphi(0)$, $\varphi'(0)$ şi $\varphi(\tau_1 \alpha_{k-1})$ este tare convex cu minimul în α_q, atunci se consideră $\alpha_t = \alpha_q$. Altfel, se pune $\alpha_t = \tau_1 \alpha_{k-1}$. Această idee de a utiliza valorile anterioare ale lungimii pasului pentru a inițializa căutarea la pasul curent a fost studiată de Hirst [1989] în teza lui de doctorat.

Cu aceste preparative, algoritmul de căutare liniară bazat pe condițiile Wolfe aproximative al lui Hager şi Zhang este următorul:

Algoritmul 4.5.2. (Căutarea liniară Wolfe aproximativă Hager-Zhang)

L0.	Se determină α_t prin procedura de inițializare descrisă mai sus. Se pune $[\alpha_{l0}, \alpha_{u0}] = bracket(\alpha_t)$, şi $j = 0$.
L1.	Se calculează $[\alpha_l, \alpha_u] = \text{secant}^2(\alpha_{lj}, \alpha_{uj})$.
L2.	Dacă $\alpha_u - \alpha_l > \gamma(\alpha_{uj} - \alpha_{lj})$, unde $\gamma \in (0,1)$, atunci $\alpha_t = (\alpha_l + \alpha_u)/2$ şi $[\alpha_l, \alpha_u] = update(\alpha_l, \alpha_u, \alpha_t)$.
L3.	Se pune $j = j + 1$, $[\alpha_{lj}, \alpha_{uj}] = [\alpha_l, \alpha_u]$ şi se continuă cu pasul L1.

Vedem că algoritmul este foarte complicat, cu multe interpretări ale diverselor condiții care apar în formularea relațiilor Wolfe aproximative, cu utilizarea tehnicilor de interpolare pătratică sau cubică şi a secantei. Toate acestea se sprijină pe multă cunoaştere empirică obținută din experimente numerice. Procedura de căutare liniară Hager-Zhang este o extensie foarte subtilă a procedurii elaborate de Moré şi Thuente. Pentru ambele proceduri se demonstrează teoreme care asigură buna definire a algoritmilor şi convergența acestora la o lungime pozitivă a pasului de deplasare. Important este faptul că algoritmul de căutare Hager-Zhang determină o lungime a pasului cu o acuratețe de ordinul lui epsilon maşină, față de ceilalți algoritmi (Wolfe, Wolfe tare) care determină lungimea cu acurateța de ordinul lui radical din epsilon maşină [Hager şi Zhang, 2006a][3].

După cum am văzut pentru determinarea lungimii pasului de deplasare α_k de-a lungul direcției d_k se dispune de o multitudine de algoritmi, de diverse complexități, pentru care s-au adus justificări şi demonstrații ale bunei lor definiri şi convergențe. Problema care rămâne, şi care constituie întotdeauna marea provocare a unei teorii, este aceea a eficienței acestor algoritmi, a capabilității lor de a rezolva în mod eficient probleme de optimizare fără restricții – domeniu de care ne ocupăm noi aici. De aceea în cele ce urmează vom considera trei algoritmi de căutare liniară: *backtracking, căutarea liniară Wolfe (standard)* (Shanno-Phua) şi *căutarea liniară Wolfe tare* (Moré şi Thuente) pentru care vom vedea comportarea acestora în ceea ce priveşte rezolvarea unor probleme de optimizare fără restricții. În acest sens, pentru a vedea performanța algoritmilor de căutare liniară de mai sus vom considera cea mai simplă metodă de optimizare fără restricții, metoda pasului descendent, pentru care $x_{k+1} = x_k - \alpha_k g_k$, adică direcția de deplasare este direcția negativă a gradientului în punctul curent, iar lungimea pasului α_k este calculată de cei trei algoritmi de căutare liniară de mai sus[4]. Menționăm că metoda pasului descendent va fi analizată în detalii în capitolul 5.

Exemplul 4.5.1. Fie funcția:

$$f(x) = \sum_{i=1}^{n-1} (x_i - 1)^2 + \left(\sum_{j=1}^{n} x_j^2 - 0.25 \right)^2$$

cu punctul inițial $x_0 = [1, 2, \ldots, n]$, unde $n = 100$. Atunci performanța algoritmilor de mai sus este prezentată în tabelul 4.5.1, unde #iter reprezintă numărul de iterații, #fg este numărul de evaluări ale funcției şi gradientului, CPU(s) este timpul de calcul în secunde, f^* este valoarea funcției în punctul de minim, iar $\|g\|$ este norma gradientului în punctul de minim.

---

[3] Algoritmul 4.5.2 este implementat în programul CG_DESCENT.FOR elaborat de Hager şi Zhang, şi se poate consulta pe CD-ul anexat lucrării (subrutina cg_line).

[4] Cititiorul interesat poate consulta programul LSTEST.FOR care include subrutinele BACK, WOLFE şi MTLINES care implementează algoritmii de căutare liniară corespunzători şi care se găseşte pe CD-ul ataşat lucrării.

Tabelul 4.5.1. Performanța algoritmilor de căutare liniară inexactă. Algoritmul pasului descendent.

Metoda	#iter	#fg	CPU(s)	f^*	$\|g\|$
Backtracking	137	2836	62.47	75	0.953999e-06
Wolfe	14	48	1.18	75	0.680075e-06
Wolfe tare	20	103	2.39	75	0.248833e-06

Exemplul 4.5.2. Fie funcția

$$f(x) = (3x_1 - 2x_1^2)^2 + \sum_{i=2}^{n-1}(3x_i - 2x_i^2 - x_{i-1} - 2x_{i+1} + 1)^2 + (3x_n - 2x_n^2 - x_{n-1} + 1)^2$$

cu punctul inițial $x_0 = [-1, -1, \ldots, -1]$ şi $n = 100$. Tabelul 4.5.2 conține performanțele algoritmilor considerați.

Tabelul 4.5.2. Performanța algoritmilor de căutare liniară inexactă. Algoritmul pasului descendent.

Metoda	#iter	#fg	CPU(s)	f^*	$\|g\|$
Backtracking	607	13665	330.81	0.397067	0.827037e-06
Wolfe	78	195	4.58	0.222e-13	0.826909e-06
Wolfe tare	93	289	6.85	0.1575e-13	0.797402e-06

Exemplul 4.5.3. Fie funcția

$$f(x) = \sum_{i=1}^{n/2}(1.5 - x_{2i-2} + x_{2i-1}x_{2i})^2 + \\ (2.25 - x_{2i-1} + x_{2i-1}x_{2i}^2)^2 + (2.625 - x_{2i-1} + x_{2i-1}x_{2i}^3)^2$$

cu punctul inițial $x_0 = [1, 0.8, \ldots, 1, 0.8]$ şi $n = 100$. Tabelul 4.5.3 conține performanțele acestor algoritmi.

Tabelul 4.5.3. Performanța algoritmilor de căutare liniară inexactă. Algoritmul pasului descendent.

Metoda	#iter	#fg	CPU(s)	f^*	$\|g\|$
Backtracking	1142	19617	477.28	0.1254e-12	0.984514e-06
Wolfe	1087	2218	52.27	0.5976e-12	0.681328e-06
Wolfe tare	300	898	21.73	0.1337e-11	0.928115e-06

În următorul experiment numeric să vedem performanțele celor trei algoritmi de căutare liniară în contextul unui algoritm de gradient conjugat. Algoritmii de gradient conjugat sunt mult mai performanți decât algoritmul pasului descendent şi vor fi tratați în capitolul 8 al acestei monografii. În metoda gradientului conjugat iterațiile sunt determinate utilizând relația $x_{k+1} = x_k + \alpha_k d_k$. Direcția de căutare se calculează sub forma $d_{k+1} = -g_k + \beta_k d_k$, în care parametrul β_k se determină conform unei relații care defineşte algoritmul de gradient conjugat. Lungimea pasului α_k se calculează conform celor cei trei algoritmi de căutare liniară de care discutam mai sus[5]. În acest experiment considerăm algoritmul Hestenes şi Steifel în care $\beta_k = g_{k+1}^T y_k / y_k^T d_k$, unde $y_k = g_{k+1} - g_k$. În următoarele exemple se consideră aceleaşi funcții ca în cele trei exemple prezentate mai sus, dar cu un număr diferit de variabile. În tabelele de mai jos se arată performanțele acestor algoritmi de căutare liniară (număr de iterații, număr de evaluări ale funcției şi gradientului, timpul de calcul cpu) în contextul algoritmilor de gradient conjugat.

Exemplul 4.5.4. Fie funcția din exemplul 4.5.1, în care $n = 10000$. Tabelul 4.5.4 prezintă performanțele algoritmilor de căutare liniară considerați în acest studiu.

Tabelul 4.5.4. Performanța algoritmilor de căutare liniară inexactă.
Algoritmul de gradient conjugat Hestenes-Stiefel.

Metoda	**#iter**	**#fg**	**CPU(s)**	f^*	$\|g\|$
Backtracking	229	7977	132.15	9453.238852	0.606498e-04
Wolfe	115	2912	52.94	9453.238852	0.127230e-05
Wolfe tare	23	107	2.20	9453.238852	0.576972e-05

Exemplul 4.5.5. Fie funcția din exemplul 4.5.2, în care $n = 10000$. Tabelul 4.5.5 prezintă performanțele algoritmilor de căutare liniară considerați în acest studiu.

Tabelul 4.5.5. Performanța algoritmilor de căutare liniară inexactă.
Algoritmul de gradient conjugat Hestenes-Stiefel.

Metoda	**#iter**	**#fg**	**CPU(s)**	f^*	$\|g\|$
Backtracking	264	5070	9.39	1.961211	0.627392e-05
Wolfe	44	82	0.82	0.302616e-13	0.189612e-05
Wolfe tare	68	164	1.48	0.712527	0.322331e-05

[5] Rezultatele prezentate sunt obținute cu programul LSCGTEST.FOR, în care sunt implementați 20 de algoritmi de gradient conjugat, şi care conține subrutinele BACK, WOLFE şi MTLINES care implementează algoritmii de căutare respectivi. Acest program se găseşte pe CD-ul anexat lucrării.

Exemplul 4.5.6. Fie funcția din exemplul 4.5.3, în care $n = 10000$. Tabelul 4.5.6 prezintă performanțele acestor algoritmi.

Tabelul 4.5.6. Performanța algoritmilor de căutare liniară inexactă. Algoritmul de gradient conjugat Hestenes-Stiefel.

Metoda	#iter	#fg	CPU(s)	f^*	$\|g\|$
Backtracking	84	1075	10.87	0.581314e-10	0.691753e-04
Wolfe	12	22	0.39	0.103860e-14	0.319524e-07
Wolfe tare	16	34	0.60	0.147450e-12	0.104290e-05

Din aceste experimente numerice vedem că algoritmii de căutare liniară Wolfe Shanno-Phua (SP) şi Wolfe tare Moré-Thuente (MT) sunt mult mai performanți decât căutarea liniară cu backtracking. Aceasta nu constituie o surpriză deoarece căutarea cu backtracking este extrem de simplă. Ea ține seama numai de valorile funcției. Totuşi, din experimentele numerice de mai sus nu putem concluziona care dintre cei doi algoritmi de căutare liniară: Wolfe (în implementarea Shanno-Phua) sau Wolfe tare (în implementarea Moré-Thuente) este mai performant. De aceea vom efectua următorul studiu numeric în cadrul algoritmilor de gradient conjugat.

Studiu numeric. (Wolfe: Shanno-Phua versus Wolfe tare: Moré-Thuente)
Pentru a compara aceşti algoritmi de căutare liniară în contextul algoritmilor de gradient conjugat am considerat un set de 38 de probleme de optimizare fără restricții şi pentru fiecare problemă am considerat 10 instanțieri în care numărul de variabile n ia valorile: 1000,2000,...,10000. Deci în total am rezolvat 380 de probleme[6]. Atât algoritmul de gradient conjugat Hestenes-Stiefel care utilizează căutarea Wolfe cât şi cel care utilizează căutarea Wolfe tare folosesc acelaşi test de oprire a iterațiilor:

$$\|\nabla f(x_k)\|_2 \le \varepsilon_g$$

unde $\varepsilon_g = 10^{-6}$. Problemele de test sunt din colecția CUTE [Bongartz, Conn, Gould şi Toint, 1995], împreună cu alte probleme de optimizare fără restricții construite în acest scop.

Studiul numeric a fost făcut în următorul context. Fie f_i^{SP} şi f_i^{MT} valorile funcției de minimizat i în punctul de optim calculate de algoritmul de gradient conjugat Hestenes-Stiefel care utilizează căutarea Wolfe (implementarea Shanno-Phua), respectiv de acelaşi algoritm care utilizează căutarea Wolfe tare (implementarea Moré-Thuente), pentru $i = 1,...,380$. Zicem că în cazul particular al problemei i algoritmul SP este mai performant decât algoritmul MT dacă

$$\left| f_i^{SP} - f_i^{MT} \right| < 10^{-3}$$

[6] Problemele au fost rezolvate cu programul LSCGTEST.FOR care se găseşte pe CD-ul anexat acestei monografii. Numele funcțiilor de minimizat, considerate în acest studiu, se află în fişierul LSCGPROB.TXT.

şi numărul de iteraţii, sau numărul de evaluări ale funcţiei şi gradientului sau timpul de calcul al lui W este mai mic decât numărul de iteraţii sau numărul de evaluări ale funcţiei şi gradientului sau timpul de calcul corespunzător lui MT. Din cele 380 de probleme rezolvate de SP şi MT, numai 340 satisfac criteriul de mai sus.

Tabelul 4.5.7 conţine numărul de probleme din cele 340 pentru care SP versus MT a realizat numărul minim de iteraţii (#iter), numărul minim de evaluări ale funcţiei şi gradientului (#fg), respectiv timpul de calcul CPU minim (CPU).

Tabelul 4.5.7. Performanţa algoritmilor de căutare liniară inexactă bazaţi pe condiţiile Wolfe şi Wolfe tare.
Algoritmul de gradient conjugat Hestenes-Stiefel.

	Wolfe (SP)	**Wolfe tare (MT)**	**=**
#iter	187	130	23
#fg	226	106	8
CPU	207	110	23

De exemplu, referitor la numărul de iteraţii, SP a fost mai bun în 187 de probleme (adică a realizat un număr minim de iteraţii în 187 de probleme) faţă de MT care a realizat un număr minim de iteraţii numai în 130 de probleme. SP şi MT au rezolvat un număr de 23 de probleme în acelaşi număr de iteraţii. Din acest tabel vedem clar că SP este mai performant decât MT. Performanţele acestor algoritmi au fost vizualizate utilizând profilul Dolan-Moré [2002], ca în figurile 4.5.1 şi 4.5.2.

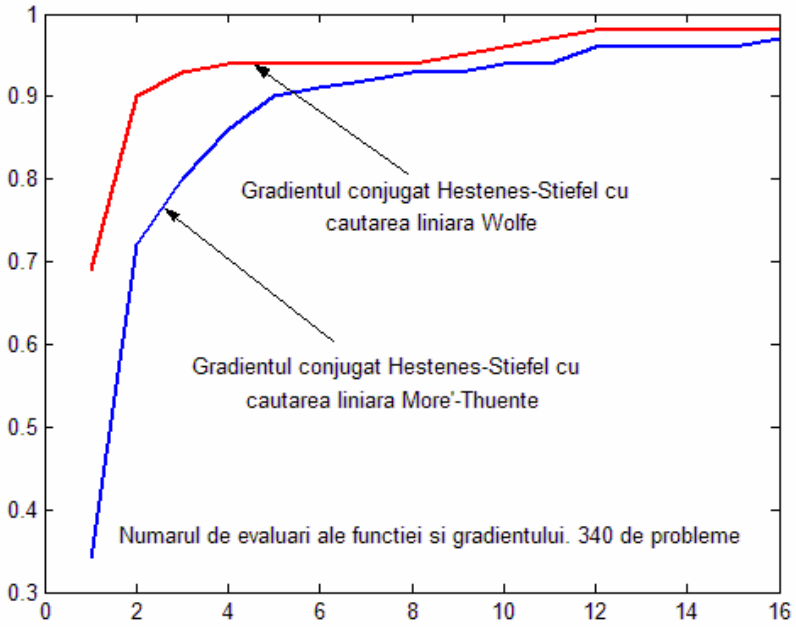


Fig. 4.5.1. Numărul de evaluări ale funcţiei.

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 evaluări ale funcţiei şi gradientului sau a timpului de calcul. Partea din stânga din figurile 4.5.1 şi 4.5.2 arată procentajul din cele 340 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 340 de probleme.

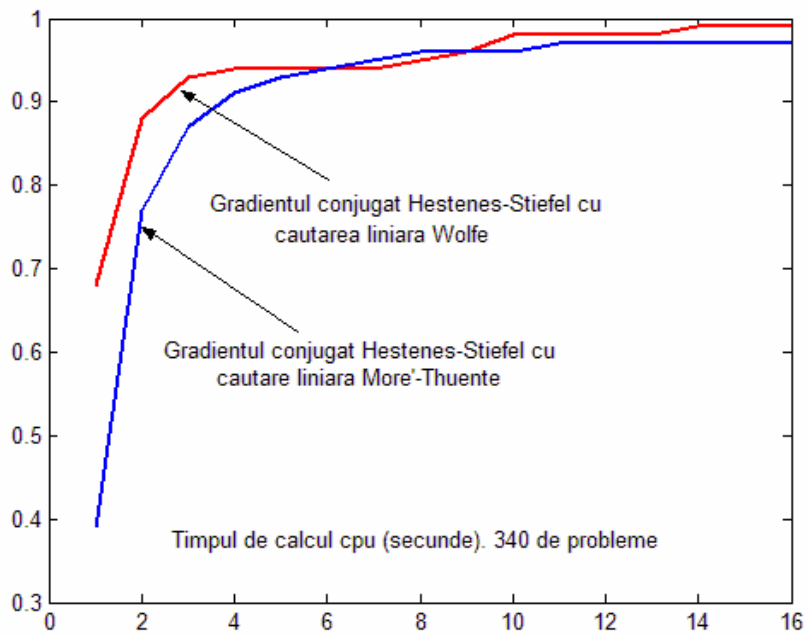


Fig. 4.5.2. Timpul de calcul cpu (secunde).

Concluzia acestui studiu numeric este că în cadrul algoritmului de gradient conjugat Hestenes-Stiefel căutarea liniară bazată pe condiţiile Wolfe este mai performantă decât cea bazată pe căutarea liniară Wolfe tare. Totuşi, acesta nu este un rezultat definitiv. Este foarte posibil ca pentru alţi algoritmi de optimizare fără restricţii să obţinem alte rezultate, în ceea ce priveşte eficienţa căutării liniare. Ceea ce putem spune este că algoritmii de căutare liniară Wolfe (*standard*: Shanno-Phua, *tare*: Moré-Thuente sau *aproximativă*: Hager-Zhang) rămân procedurile de bază utilizate în optimizarea fără restricţii utilizată la nivel industrial.

Capitolul 4 din lucrarea:
Neculai Andrei
„Critica Raţiunii Algoritmilor de Optimizare fără Restricţii”
Iunie 12 –Noiembrie 23, 2006