PROBLEME DESCHISE ÎN ALGORITMII DE GRADIENT CONJUGAT PENTRU OPTIMIZARE FĂRĂ RESTRICŢII

Centenar Eduard Stiefel (1909-1978)

Neculai Andrei

Institutul Naţional de Cercetare – Dezvoltare în Informatică, ICI, Bucureşti
Academia Oamenilor de Ştiinţă din România, Bucureşti

Rezumat. Lucrarea prezintă câteva probleme deschise în algoritmii de gradient conjugat pentru optimizare fără restricţii. Acestea se referă la direcţia iniţială de căutare, condiţia de conjugare, calculul lungimii paşilor de deplasare, noi formule pentru parametrul de conjugare, bazate pe valorile funcţiei de minimizat, influenţa acurateţei procedurii de căutare liniară, cum se poate introduce structura problemei în algoritmii de gradient conjugat, cum se poate considera informaţia de ordinul doi în aceşti algoritmi, care este cea mai bună procedură de restartare, care este cel mai bun algoritm de gradient conjugat hibrid, algoritmi de gradient conjugat scalat, care este cel mai bun criteriu de oprire a iteraţiilor în aceşti algoritmi etc.

Cuvinte cheie: Gradient conjugat, condiţia de conjugare, restartare, informaţia de ordinul doi, structură, gradient conjugat hibrid, scalare.

Abstract. The paper presents some open problems associated to the nonlinear conjugate gradient algorithms for unconstrained optimization. Mainly, these problems refer to the initial direction, the conjugacy condition, the steplength computation, new formula for conjugate gradient parameter computation based on function's values, the influence of accuracy of line search procedure, how we can take the problem's structure on conjugate gradient algorithms, how we can consider the second order information in these algorithms, what the best restart procedure is, what the best hybrid conjugate gradient algorithm is, scaled conjugate gradient algorithms, what the best stopping criterion in conjugate gradient algorithms is etc.

Keywords: Conjugate gradient, conjugacy condition, restart, second order information, structure, hybrid conjugate gradient, scaling.

1. Introducere

Metodele de gradient conjugat reprezintă o contribuţie importantă în panoplia metodelor de optimizare fără restricţii de mari dimensiuni. Algoritmii asociaţi acestor metode sunt caracterizaţi de cerinţe modeste de memorie şi au proprietăţi foarte bune de convergenţă globală. Popularitatea lor este datorată în parte simplităţii expresiei lor algebrice, implementării foarte rapide şi uşoare în programe de calcul, precum şi eficienţei lor în rezolvarea problemelor cu un număr mare de variabile.

Metodele de gradient conjugat au fost proiectate de Magnus Hestenes (1906-1991) şi Eduard Stiefel (1909-1978) în lucrarea *Methods of conjugate gradients for solving linear systems*, J. Research Nat. Bur. Standards Sec. B. 48, 409-436 (1952) în care au prezentat un algoritm pentru rezolvarea sistemelor algebrice liniare, cu matricea simetrică şi pozitiv definită. În 1964, metoda a fost extinsă de Fletcher şi Reeves la probleme de optimizare neliniară fără restricţii. De atunci, un număr foarte mare de algoritmi de gradient conjugat au fost elaboraţi. Chiar dacă algoritmii de gradient conjugat au peste 50 de ani de existenţă, totuşi aceştia continuă să fie de un considerabil interes în particular datorită proprietăţilor lor de convergenţă şi eficienţei în rezolvarea problemelor de optimizare de mari dimensiuni. O scurtă schiţă istorică este foarte interesantă.

Eduard Stiefel (1909-1978)

Istoria acestei metode începe cu cercetările lui Eduard Stiefel la Eidgenössische Technische Hochschule Zürich - Swiss Federal Institute of Technology, şi Cornelius Lanczos (1893-1974) şi Magnus Hestenes (1906-1991) la The Institute for Numerical Analysis (National Applied Mathematics Laboratories of the United States National Bureau of Standards in Los Angeles). În 1951, Eduard Stiefel a proiectat o metodă în $n-$paşi pentru rezolvarea sistemelor algebrice liniare cu matricea simetrică şi pozitiv definită, pe care a publicat-o ca *Über einige Methoden der Relaxationsrechnung.* Z. Angew. Math. Phys., 3 (1952), pp. 1-33. În iulie 1951, invitat de Olga Taussy-Todd (1906-1995), Stiefel a vizitat Institutul pentru Analiză Numerică (INA) de la Biroul Naţional de Standarde – UCLA, Los Angeles, până în februarie 1952. Aici, a găsit că Magnus Hestenes în colaborare cu alţi cercetători de la INA (Lanczos, Forsythe, Motzkin, Rosser, Stein) au elaborat aceeaşi metodă. Hestenes a numit-o metoda de gradient conjugat şi a publicat-o ca un raport intern: *Iterative methods for solving equations*. NAML Report 52-9, National Bureau of Standards, Los Angeles, CA, July 1951. În timpul acestei vizite, Hestenes şi Stiefel şi-au combinat eforturile reuşind să publice lucrarea de 28 de pagini pe două coloane: *Methods of Conjugate Gradients for Solving Linear Systems* în Journal of Research of the National Bureau of Standards, vol.49, No. 6, December 1952 (Research Paper 2379), fiind una dintre cele mai influente lucrări în domeniul analizei numerice. Anumite detalii şi precizări asupra acestei lucrări au fost aduse de Dianne P. O'Leary [O'Leary, D.P., *Conjugate gradients and related KMP algorithms: the beginnings*. in Linear and Nonlinear Conjugate Gradient – Related Methods, L. Adams and J.L. Nazareth (Eds.) SIAM, 1996, pp. 1-8]. Din acel moment, au fost elaborate peste 50 de variante de algoritmi de gradient conjugat şi practic un număr nesfârşit de lucrări au fost publicate în legătură cu această metodă. O privire generală de ansamblu asupra acestei metode a fost dată de Hager şi Zhang [40].

Pentru problema generală de optimizare fără restricţii

$$\min_{x\in R^n} f(x), \tag{1.1}$$

unde $f : R^n \to R$ este o funcţie continuu diferenţiabilă, mărginită inferior, un algoritm de gradient conjugat plecând dintr-un punct iniţial x_0 dat, generează un şir de puncte $\{x_k\}$, conform formulei iterative:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

unde α_k este lungimea pasului de obicei calculată din condiţiile de căutare liniară Wolfe,

$$f(x_k + \alpha_k d_k) - f(x_k) \le \rho\alpha_k g_k^T d_k, \tag{1.3}$$

$$g_{k+1}^T d_k \ge \sigma g_k^T d_k, \tag{1.4}$$

în care $0 < \rho < 1/2 \le \sigma < 1,$ iar direcţiile d_k sunt calculate ca:

$$d_{k+1} = -g_{k+1} + \beta_k s_k, \quad d_0 = -g_0. \tag{1.5}$$

Aici, β_k este un scalar cunoscut ca parametrul algoritmului de gradient conjugat, $g_k = \nabla f(x_k)$ şi $s_k = x_{k+1} - x_k$. În cele ce urmează definim $y_k = g_{k+1} - g_k$. Diferiţi algoritmi de gradient conjugat corespund la diferite alegeri ale parametrului β_k. Deci, un element crucial în orice algoritm de gradient conjugat este formula de definiţie a parametrului β_k. Orice algoritm de gradient conjugat are o structură foarte simplă, ilustrată ca mai jos.

Prototipul Algoritmului de Gradient Conjugat

Pasul 1.	Se selectează un punct iniţial $x_0 \in dom\, f$ şi se calculează: $f_0 = f(x_0)$ şi $g_0 = \nabla f(x_0)$. Se pune $d_0 = -g_0$ şi $k = 0$.
Pasul 2.	Se testează un criteriu de oprire a iteraţiilor. De exemplu, dacă $\|g_k\|_\infty \le \varepsilon$, atunci stop; altfel se continuă cu pasul 3.
Pasul 3.	Se determină lungimea pasului α_k.
Pasul 4.	Se actualizează variabilele ca: $x_{k+1} = x_k + \alpha_k d_k$. Se calculează f_{k+1} şi g_{k+1}. Se calculează $y_k = g_{k+1} - g_k$ şi $s_k = x_{k+1} - x_k$.
Pasul 5.	Se determină β_k.
Pasul 6.	Se calculează direcţia de deplasare: $d_{k+1} = -g_{k+1} + \beta_k s_k$.
Pasul 7.	Criteriul de restartare a iteraţiilor. De exemplu, dacă criteriul de restartare a iteraţiilor al lui Powell $\left\|g_{k+1}^T g_k\right\| > 0.2\|g_{k+1}\|^2$ este satisfăcut, atunci se pune $d_{k+1} = -g_{k+1}$.
Pasul 8.	Se calculează estimaţia iniţială a lungimii pasului, de exemplu $\alpha_k = \alpha_{k-1}\|d_{k-1}\| / \|d_k\|$, se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Acesta este un prototip de algoritm de gradient conjugat, dar se cunosc variante mult mai sofisticate (CONMIN [51], [52], SCALCG [3], [4], [5], [6], ASCALCG [5], [19], ACGHES [18], [21], CG_DESCENT [39]). Aceste variante se concentrează pe proceduri de calcul ale parametrului β_k şi a lungimii pasului α_k.

2. Probleme deschise

În cele ce urmează, vom prezenta 14 probleme deschise în algoritmii de gradient conjugat [20]. Aceste probleme se referă la o serie de aspecte teoretice şi computaţionale ale acestor algoritmi şi, în acest context, constituie direcţii de cercetare în acest domeniu.

Problema 1. De ce direcţia iniţială de căutare $d_0 = -g_0$ este esenţială în algoritmii de gradient conjugat?

Crowder şi Wolfe [29] au prezentat un exemplu de funcţie pătratică tare convexă cu 3 variabile arătând că, dacă direcţia iniţială nu este direcţia pasului descendent, atunci rata de convergenţă a algoritmului de gradient conjugat este liniară. Pe de altă parte, Beale [25] a arătat că, dacă

$$d_{k+1} = -g_{k+1} + \frac{y_0^T g_{k+1}}{y_0^T d_0} d_0 + \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} d_k \tag{2.1}$$

atunci dacă $d_0 \neq -g_0,$ atunci direcţiile astfel calculate sunt conjugate. Deci, (2.1) permite să se obţină direcţii conjugate plecând din orice direcţie iniţială $d_0.$ Totuşi, deoarece d_0 rămâne în formula lui d_{k+1} acesta poate crea dificultăţi în ceea ce priveşte viteza de convergenţă [34].

Mai târziu, Powell [49] a arătat că, dacă $f(x)$ este o funcţie pătratică, atunci utilizând o direcţie iniţială arbitrară d_0 algoritmii de gradient conjugat au o rată de convergenţă liniară. Nazareth [45] a sugerat un algoritm de gradient conjugat ceva mai complicat în care direcţia d_{k+1} este calculată sub forma:

$$d_{k+1} = -y_k + \frac{y_k^T y_k}{y_k^T d_k} d_k + \frac{y_{k-1}^T y_k}{y_{k-1}^T d_{k-1}} d_{k-1}, \tag{2.2}$$

unde $d_0 = 0.$ Nazareth a demonstrat că, dacă $f(x)$ este o funcţie pătratică convexă, atunci pentru orice lungime a pasului α_k direcţiile de căutare sunt conjugate în raport cu Hessianul lui $f.$. Totuşi, departe de minim, dacă $d_0 \neq -g_0,$ atunci d_k poate deveni zero. Deşi interesantă, această inovaţie nu s-a dovedit eficientă în practică. O altă abordare care permite direcţii iniţiale d_0 arbitrare pentru cazul funcţiilor pătratice a fost sugerată de Allwright [1] care a utilizat o schimbare de variabilă bazată pe o factorizare a Hessianului funcţiei $f.$ Observăm că toate aceste observaţii sunt făcute numai pentru funcţii pătratice, pentru funcţii generale nu avem nici un răspuns la această problemă.

Problema 2. Care este cea mai bună condiţie de conjugare?

Condiţia de conjugare este exprimată ca $y_k^T d_{k+1} = 0.$. Recent, Dai şi Liao [30] au introdus o nouă condiţie de conjugare sub forma $y_k^T d_{k+1} = -t s_k^T g_{k+1},$ unde $t \geq 0$ este un scalar. Această nouă condiţie de conjugare este foarte rezonabilă deoarece în practică se utilizează căutarea liniară inexactă. Totuşi, această condiţie este dependentă de parametrul t, pentru care nu cunoaştem încă o formulă de calcul optimă.

Problema 3. De ce în algoritmii de gradient conjugat şirul $\{\alpha_k\}$ ale lungimii pasului au o evoluţie total nepredictibilă şi diferă de 1 cu până la două ordine de mărime?

Studii numerice intensive, cu diferite variante de algoritmi de gradient conjugat, au arătat că lungimea pasului de deplasare α_k diferă de 1 cu până la două ordine de mărime, în funcţie de cum este scalată problema. Mai mult, mărimea pasului are o evoluţie total nepredictibilă. Această comportare a algoritmilor de gradient conjugat este foarte diferită de aceea a metodei Newton sau a metodelor quasi-Newton, care, de obicei, admit lungimi unitare ale lungimii pasului în majoritatea iteraţiilor finale ale procesului de optimizare. Studiile computaţionale cu metoda quasi-Newton cu memorie limitată LBFGS arată că aceasta este foarte eficientă mai ales datorită abilităţii ei de a accepta lungimi unitare ale lui α_k. O încercare de a utiliza această comportare a algoritmilor de gradient conjugat a fost prezentată de Andrei [13], care a sugerat o procedură de accelerare a acestor algoritmi prin modificarea lungimii α_k a pasului (calculată din condiţiile Wolfe) prin intermediul unui parametru η_k, într-o manieră multiplicativă de tipul, $x_{k+1} = x_k + \eta_k \alpha_k d_k$, cu scopul de reducere a valorilor funcţiei de minimizat de-a lungul iteraţiilor. Se demonstrează că schema de accelerare este liniar convergentă. Comparaţii numerice cu această schemă de accelerare sunt prezentate în [19]. O accelerare a algoritmului pasului descendent cu backtracking pentru optimizare fără restricţii este dată în [2].

Problema 4. Care este influenţa acurateţei procedurii de căutare liniară de determinare a lungimii pasului α_k asupra performanţelor algoritmilor de gradient conjugat?

Pentru orice algoritm de optimizare fără restricţii, un element crucial este procedura de calcul a lungimii pasului de deplasare. În acest sens, s-au propus numeroase proceduri. În căutarea liniară exactă, lungimea α_k a pasului se calculează ca soluţie a problemei

$$\alpha_k = \arg\min_{\alpha>0} f(x_k + \alpha d_k), \tag{2.3}$$

unde d_k este o direcţie descendentă. În anumite cazuri speciale (probleme pătratice, de exemplu) este posibil ca α_k să fie calculat analitic, dar pentru vasta majoritate a problemelor α_k se calculează pentru a reduce valorile funcţiei f de-a lungul razei $\{x_k + \alpha d_k : \alpha \geq 0\}.$ În practică, cele mai uzitate sunt procedurile inexacte de calcul ale lui α_k. O multitudine de astfel de proceduri au fost propuse: Goldstein [36], Armijo [24], Wolfe [54], Powell [49], Dennis şi Schnabel [33], Potra şi Shi [48], Lemaréchal [43], Moré şi Thuente [44], Hager şi Zhang [39] şi multe altele. Cele mai utilizate sunt bazate pe condiţiile Wolfe (1.3) şi (1.4). O contribuţie importantă în înţelegerea condiţiilor Wolfe a fost dată de Hager şi Zhang [39], [40], prin introducerea condiţiilor Wolfe aproximative

$$(2\rho - 1) g_k^T d_k \geq g_{k+1}^T d_k \geq \sigma g_k^T d_k. \tag{2.4}$$

Prima inegalitate din (2.4) este o aproximaţie a primei condiţii Wolfe (1.3). Când iteraţiile sunt lângă punctul de optim, aceste aproximaţii date de (2.4) se pot evalua cu o acurateţe mai mare decât condiţiile Wolfe originale, deoarece condiţiile Wolfe aproximative sunt exprimate în termenii derivatei, şi nu ca diferenţe ale valorilor funcţiei. Subliniem faptul că prima condiţie Wolfe (1.3) limitează acurateţea algoritmului de gradient conjugat la ordinul de mărime dat de rădăcina pătrată a preciziei maşinii, în timp ce condiţiile Wolfe aproximative (2.4) realizează o acurateţe de ordinul preciziei maşinii [39]. Se pare că, cu cât lungimea pasului de deplasare este calculată cu o acurateţe mai mare, cu atât convergenţa algoritmului de gradient conjugat este mai pronunţată. De exemplu, algoritmul CG_DESCENT al lui Hager şi Zhang [39] este unul dintre cei mai rapizi algoritmi de gradient conjugat. Totuşi, impunând atât condiţia de descendenţă, cât şi pe cea de conjugare, şi utilizând condiţiile Wolfe originale (1.3) şi (1.4), Andrei [22] a sugerat un alt algoritm de gradient conjugat accelerat mai rapid decât CG_DESCENT. În acest context, o altă chestiune deschisă interesantă este dacă sau nu căutarea liniară nemonotonă a lui Grippo, Lampariello şi Lucidi, [38] este mai bună decât căutarea liniară Wolfe.

Problema 5. Cum se pot utiliza valorile funcţiei f în formula de calcul a parametrului β_k pentru a genera noi algoritmi eficienţi de gradient conjugat?

În general, în algoritmii de gradient conjugat parametrul β_k este calculat utilizând scalarii: $\|g_k\|$, $\|g_{k+1}\|$, $\|y_k\|$, $\|s_k\|$, $y_k^T s_k$, $g_k^T g_{k+1}$, $y_k^T g_{k+1}$ şi $s_k^T g_{k+1}$. După cum se vede în formula pentru β_k diferenţa $f(x_k) - f(x_{k+1})$ nu este utilizată. Yabe şi Takano [55], utilizând un rezultat al lui Zhang, Deng şi Chen [57], au sugerat următoarea formulă pentru β_k:

$$\beta_k^{YT} = \frac{g_{k+1}^T (z_k - t s_k)}{d_k^T z_k}, \tag{2.5}$$

unde $z_k = y_k + \frac{\delta\eta_k}{s_k^T u_u} u_k$, $\eta_k = 6(f(x_k) - f(x_{k+1})) + 3(g_k + g_{k+1})^T s_k$, $\delta > 0$ este o constantă şi $u_k \in R^n$ satisface $s_k^T u_k \neq 0;$ de exemplu $u_k = s_k$. În acelaşi context bazat pe condiţia secantei modificată a lui Zhang, Deng şi Chen [57], cu $u_k = s_k$, Andrei [14] a propus următoarea formulă pentru β_k:

$$\beta_k = \left(\frac{\delta\eta_k}{\|s_k\|^2} - 1 \right) \frac{s_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k} + \frac{y_k^T g_{k+1}}{y_k^T s_k + \delta\eta_k}, \tag{2.6}$$

unde $\delta \geq 0$ este un parametru scalar. O altă posibilitate este prezentată de Yuan [56] ca

$$\beta_k^Y = \frac{y_k^T g_{k+1}}{(f_k - f_{k+1})/\alpha_k - d_k^T g_k / 2}. \tag{2.7}$$

Problema 6. Cum se poate utiliza structura problemei pentru a proiecta algoritmi eficienţi de gradient conjugat neliniar?

Când problema este parţial separabilă, adică aceasta se poate exprima ca o sumă de funcţii elementare fiecare dintre ele având un subspaţiu invariant [28], se poate atunci formula o procedură de actualizare a lui β_k pentru a obţine un algoritm eficient de gradient conjugat? Ideea descompunerii funcţiilor parţial separabile a fost considerată de Conn, Gould şi Toint [27]. Avantajul acestui mod de abordare este că informaţia conţinută într-o astfel de descriere a funcţiei se poate utiliza pentru a explora funcţia numai în anumite direcţii relevante. Ideea este de a ignora anumite subspaţii ale funcţiei f şi de a considera complementul acestora. Chestiunea este dacă acest tip de informaţie se poate utiliza pentru a proiecta noi formule de calcul pentru β_k.

Problema 7. Cum se poate utiliza informaţia de ordinul doi în algoritmii de gradient conjugat?

Andrei [3], [4], [5], [16] a sugerat următoarea formulă de calcul pentru β_k:

$$\beta_k = \frac{s_k^T \nabla^2 f(x_{k+1}) g_{k+1} - s_k^T g_{k+1}}{s_k^T \nabla^2 f(x_{k+1}) s_k}. \tag{2.8}$$

Observăm că, dacă este exactă căutarea liniară, atunci (2.8) este chiar metoda lui Daniel [32]. Ceea ce este interesant în această formulă este prezenţa Hessianului. În practică, pentru probleme de mari dimensiuni, formulele de actualizare a anumitor parametri care nu cer evaluarea matricei Hessian sunt de preferat faţă de cele care implică această matrice. O posibilitate de a utiliza informaţia de ordinul doi, dată de Matricea Hessian, este de a calcula în mod direct produsul Hessian/vector $\nabla^2 f(x_{k+1}) s_k$. Totuşi, experimentele noastre computaţionale au arătat că produsul $\nabla^2 f(x_{k+1}) s_k$ este consumator de timp, mai ales pentru probleme de mari dimensiuni chiar dacă Hessianul este o matrice cu structuri speciale (tridiagonală, bloc diagonală). În plus, ce se întâmplă dacă $s_k \in Ker\nabla^2 f(x_{k+1})$? Într-un efort de a utiliza Hessianul în formula de calcul a lui β_k Andrei [18], [21] a sugerat un algoritm de gradient conjugat neliniar în care produsul Hessian/vector $\nabla^2 f(x_{k+1}) s_k$ este aproximat prin diferenţe finite ca:

$$\nabla^2 f(x_{k+1}) s_k = \frac{\nabla f(x_{k+1} + \delta s_k) - \nabla f(x_{k+1})}{\delta}, \tag{2.9}$$

unde

$$\delta = \frac{2\sqrt{\varepsilon_m}(1 + \|x_{k+1}\|)}{\|s_k\|}, \tag{2.10}$$

şi ε_m este epsilon maşină.

După cum cunoaştem, în metodele quasi-Newton o aproximare B_k a matricei Hessian $\nabla^2 f(x_k)$ este actualizată astfel încât noua matrice B_{k+1} satisface ecuaţia (condiţia) secantei $B_{k+1}s_k = y_k$. Deci, aşa după cum se detaliază în [3], [4], [5], pentru a avea un algoritm de gradient conjugat pentru rezolvarea problemelor de mari dimensiuni, se poate presupune că perechea (s_k, y_k) satisface ecuaţia secantei. Cu această presupunere obţinem

$$\beta_k = \frac{(\theta_{k+1}y_k - s_k)^T g_{k+1}}{y_k^T s_k}, \tag{2.11}$$

unde θ_{k+1} este un parametru. Birgin şi Martínez [26] au ajuns la această formulă pentru β_k, utilizând o interpretare a minimizării funcţiilor pătratice. În [14], [16], se prezintă detalii teoretice şi computaţionale cu un algoritm în care produsul Hessian/vector $\nabla^2 f(x_{k+1})s_k$ este aproximat printr-o tehnică utilizând ecuaţia secantei modificată introdusă de Zhang, Deng şi Chen [57] şi de Zhang şi Xu [58], obţinând β_k ca în (2.6).

Problema 8. Care este cel mai bun algoritm de gradient conjugat scalat?

Anumiţi autori au propus următoarea direcţie de căutare

$$d_{k+1} = -\theta_{k+1}g_{k+1} + \beta_k s_k, \tag{2.12}$$

unde θ_{k+1} este un scalar pozitiv sau o matrice simetrică şi pozitiv definită [3], [4], [5], [26]. Formula (2.12) este cunoscută ca algoritmul de gradient conjugat scalat. Observăm că dacă $\theta_{k+1} = 1$, atunci obţinem algoritmii clasici de gradient conjugat, în funcţie de formula utilizată pentru β_k. Pe de altă parte, dacă $\beta_k = 0$, atunci obţinem o altă clasă de algoritmi, în funcţie de procedura de selecţie a lui θ_{k+1}. Considerând $\beta_k = 0$, atunci dispunem de două posibilităţi pentru θ_{k+1} : un scalar pozitiv sau o matrice simetrică şi pozitiv definită. Dacă $\theta_{k+1} = 1$, atunci avem metoda pasului descendent. Dacă $\theta_{k+1} = \nabla^2 f(x_{k+1})^{-1}$, sau o aproximaţie a acestei matrice, atunci obţinem metoda Newton sau respectiv quasi-Newton. Vedem deci că, în cazul general când $\theta_{k+1} \neq 0$ este selectat ca în metodele quasi-Newton şi $\beta_k \neq 0$, atunci (2.12) reprezintă o combinaţie a metodelor quasi-Newton şi a celor de gradient conjugat. Totuşi, dacă θ_{k+1} este o matrice care conţine informaţii relevante asupra inversei Hessianului funcţiei f, atunci se pare că este mai bine să considerăm $d_{k+1} = -\theta_{k+1}g_{k+1}$ deoarece adăugarea termenului $\beta_k s_k$ în (2.12) poate deteriora caracterul de direcţie descendentă a lui d_{k+1}. Andrei [3], [4], [5], şi Birgin şi Martínez, [26] selectează θ_{k+1} ca inversa câtului Rayleigh. O altă selecţie este bazată pe valorile funcţiei de minimizat în două puncte succesive, aşa numita procedură anticipativă [12].

Problema 9. Care este cel mai bun algoritm hibrid de gradient conjugat?

Algoritmii hibrizi de gradient conjugat au fost proiectaţi pentru a utiliza şi combina proprietăţile (bune) de convergenţă ale algoritmilor clasici de gradient conjugat. Touati-Ahmed şi Storey [53], Hu şi Storey [42], Gilbert şi Nocedal [37] au sugerat algoritmi de gradient conjugat utilizând proiecţii ale algoritmilor Fletcher-Reeves [35], Polak-Ribière [46] şi Polyak [47]. O altă sursă de algoritmi de gradient conjugat hibrizi se bazează pe conceptual de combinaţie convexă a algoritmilor de gradient conjugat clasici. Andrei [7], [8] a introdus această clasă de algoritmi de gradient conjugat hibrid ca o combinaţie convexă a algoritmilor Polak-Ribière-Polyak [46] şi Dai-Yuan [31]. În [9], [10], [15], [17], se prezintă detalii teoretice şi computaţionale asupra algoritmilor hibrizi de gradient conjugat ca nişte combinaţii convexe a algoritmilor Hestenes-Stiefel [41] şi Dai-Yuan [31]. În general, performanţele variantelor hibride bazate pe conceptul de combinaţie convexă sunt superioare algoritmilor care participă în această combinaţie, dar care este cea mai bună alegere a algoritmilor clasici de gradient conjugat, care să participe în această combinaţie convexă rămâne o chestiune deschisă.

Problema 10. Care este cea mai bună procedură de restartare a algoritmilor de gradient conjugat?

Primii algoritmi de gradient conjugat erau restartaţi ori de câte ori $k = n$ sau $k = n+1$. Când n este foarte mare şi valorile proprii ale Hessianului funcţiei de minimizat se grupează într-un număr mic de grupe, această strategie de restartare poate fi ineficientă. Powell [50] a sugerat restartarea algoritmilor de gradient conjugat ori de câte ori

$$\left|g_k^T g_{k+1}\right| \geq 0.2\left\|g_{k+1}\right\|^2 . \tag{2.13}$$

Pentru funcţii pătratice, membrul stâng din (2.13) este un indicator al neconjugării direcţiilor de căutare şi, deci, un semnal că ciclul curent din cadrul procesului de optimizare trebuie încheiat şi altul trebuie început cu direcţia negativă a gradientului în punctual curent. De asemenea, restartarea este recomandată de îndată ce direcţia de căutare încetează de a fi descendentă. Powell sugerează restartarea dacă

$$-1.2\left\|g_k\right\|^2 < d_k^T g_k < -0.8\left\|g_k\right\|^2 \tag{2.14}$$

nu este satisfăcut. Un alt criteriu de restartare a iteraţiilor în algoritmii de gradient conjugat a fost proiectat de Birgin şi Martínez [26]

$$d_{k+1}^T g_{k+1} > -10^{-3}\left\|d_{k+1}\right\|_2 \left\|g_{k+1}\right\|_2 . \tag{2.15}$$

În (2.15), când unghiul dintre d_{k+1} şi $-g_{k+1}$ nu este suficient de ascuţit, atunci algoritmul se restartează cu direcţia negativă a gradientului $-g_{k+1}$. Evident, se pot imagina diferite proceduri de restartare, dar care este cea mai bună rămâne de studiat.

Problema 11. Care este cel mai bun criteriu de oprire a iteraţiilor în algoritmii de gradient conjugat?

În precizie infinită, o condiţie necesară pentru ca x^* să fie minimul exact al funcţiei f este ca $\nabla f(x^*) = 0.$ Într-un algoritm iterativ care lucrează în precizie finită această condiţie trebuie modificată sub forma $\nabla f(x^*) \cong 0.$ Deşi $\nabla f(x^*) = 0$ poate să apară într-un punct de maxim sau într-un punct şa, totuşi procedura de căutare liniară de-a lungul direcţiei de deplasare ne asigură că obţinem un punct de minim al funcţiei $f.$ Deci, $\nabla f(x^*) = 0$ se poate foarte bine considera ca o condiţie necesară şi suficientă pentru ca x^* să fie un minim local al funcţiei $f.$.

Pentru algoritmii de gradient conjugat liniar, diferite criterii de oprire a iteraţiilor au fost analizate de Arioli şi Loghin [23]. Pentru algoritmii de gradient conjugat neliniar, de obicei, algoritmiştii utilizează următoarele criterii de oprire a iteraţiilor

$$\left\|\nabla f(x_k)\right\|_\infty \leq \varepsilon_g , \tag{2.16}$$

$$\alpha_k g_k^T d_k \leq \varepsilon_f \left| f(x_{k+1})\right| , \tag{2.17}$$

$$\left\|\nabla f(x_k)\right\|_\infty \leq \varepsilon_g (1 + \left| f(x_k)\right|) , \tag{2.18}$$

$$\left\|\nabla f(x_k)\right\|_\infty \leq \max\left\{\varepsilon_g , \varepsilon_0 \left\|\nabla f(x_0)\right\|_\infty\right\} , \tag{2.19}$$

$$\left\|\nabla f(x_k)\right\|_2 \leq \varepsilon_g , \tag{2.20}$$

unde, de exemplu $\varepsilon_f = 10^{-20}$, $\varepsilon_g = 10^{-6}$ şi $\varepsilon_0 = 10^{-12}$. Pentru probleme de mari dimensiuni mai convenabil este să se utilizeze norma $\left\|\nabla f(x_k)\right\|_\infty$, dar pentru probleme de mici dimensiuni se pare că mai bine este de a se utiliza norma $\left\|\nabla f(x_k)\right\|_2$. Dintre toate aceste criterii de mai sus, rămâne ca o problemă deschisă care este cel mai bun.

În legătură directă cu acest subiect, deseori ne întrebăm: Ce este mai important: „valoarea optimă locală f^* a funcţiei de minimizat sau punctul de optim local x^*”? În comparaţiile numerice deseori se preferă să se utilizeze valorile optime locale obţinute de diferiţi algoritmi.

Problema 12. Componentele afine ale gradientului

Metoda Newton are o proprietate foarte frumoasă. Pentru orice componentă afină a gradientului $\nabla f(x)$, fiecare iteraţie generată de metoda Newton este o soluţie a acestor componente, aceasta deoarece modelul afin asociat sistemului $\nabla f(x) = 0$ întotdeauna este exact pentru aceste funcţii. Este această proprietate valabilă şi pentru algoritmii de gradient conjugat?

Problema 13. Care este relaţia dintre algoritmii de gradient conjugat şi cei quasi-Newton?

Ambele clase de algoritmi au o anumită maturitate, cu rezultate bine fundamentate matematic şi susţinute computaţional de programe de calcul foarte performante. Chestiunea este aceea că, la ora actuală, nu vedem un progres semnificativ în proiectarea de algoritmi eficienţi şi robuşti, care să utilizeze concepte din ambele aceste clase de algoritmi.

Problema 14. Cum se pot extinde algoritmii de gradient conjugat pentru rezolvarea problemelor de optimizare cu restricţii margini simple?

Să considerăm problema

$$\min_{x \in R^n} \left\{ f(x) \mid l \leq x \leq u \right\}, \tag{2.21}$$

unde l şi u sunt vectori cunoscuţi din R^n. Problema este cum se pot adapta algoritmii de gradient conjugat pentru rezolvarea problemei (2.21). O posibilitate este de a considera tehnici de punct interior şi de a proiecta un algoritm de gradient conjugat în care marginile simple nu sunt manevrate într-o manieră explicită.

3. Concluzie

De mai bine de 50 de ani, algoritmii de gradient conjugat au fost obiectul unor cercetări şi analize teoretice şi computaţionale intensive. Şi în prezent, aceşti algoritmi continuă să reprezinte o componentă importantă a metodelor de optimizare fără restricţii, dovedindu-şi caracterul lor de inepuizabilitate. În această lucrare, am prezentat câteva probleme deschise, care constituie subiecte de meditaţie şi cercetare pentru proiectarea şi implementarea în programe de calcul de noi algoritmi eficienţi de gradient conjugat.

Bibliografie

1. **ALLWRIGHT, J.C.:** Improving the Conditioning of Optimal Control Problems Using Simple Methods. Recent Mathematical Develpments in Control, Edited by O.J. Bell, Academic Press, London, 1972.
2. **ANDREI, N.:** An Acceleration of Gradient Descent Algorithm with Backtracking for Unconstrained Optimization. Numerical Algorithms, No. 42, 2006, pp. 63-73.
3. **ANDREI, N.:** Scaled Conjugate Gradient Algorithms for Unconstrained Optimization. Computational Optimization and Applications, No. 38, 2007, pp. 401-416.
4. **ANDREI, N.:** Scaled Memoryless BFGS Preconditioned Conjugate Gradient Algorithm for Unconstrained Optimization. Optimization Methods and Software, No. 22, 2007, pp. 561-571.
5. **ANDREI, N.:** A Scaled BFGS Preconditioned Conjugate Gradient Algorithm for Unconstrained Optimization. Applied Mathematics Letters, No. 20, 2007, pp. 645-650.
6. **ANDREI, N.:** SCALCG – Scaled Conjugate Gradient Algorithms for Unconstrained Optimization. Technical Report ICI, No. 17/2007. March 30, 2007. (CD inclus)
7. **ANDREI, N.:** New Hybrid Conjugate Gradient Algorithms for Unconstrained Optimization. ICI Technical Report, 26 September, 2007. (Acceptată de Journal of Optimization Theory and Applications, 2009 as: Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization.)
8. **ANDREI, N.:** New Hybrid Conjugate Gradient Algorithms as a Convex Combination of PRP and DY for Unconstrained Optimization. ICI Technical Report, October 1, 2007.
9. **ANDREI, N.:** A Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization as a Convex Combination of Hestenes-Stiefel and Dai-Yuan. ICI Technical Report, 14 January, 2008.
10. **ANDREI, N.:** A Hybrid Conjugate Gradient Algorithm with Modified Secant Condition for Unconstrained Optimization. ICI Technical Report, 6 February, 2008.

11. **ANDREI, N.:** Accelerated Scaled Memoryless BFGS Preconditioned Conjugate Gradient Algorithm for Unconstrained Optimization. ICI Technical Report, 24 March, 2008.
12. **ANDREI, N.:** A Scaled Nonlinear Conjugate Gradient Algorithm for Unconstrained Optimization. Optimization. A journal of Mathematical Programming and Operations Research. Vol. 57, No. 4, August, 2008, pp. 549-570.
13. **ANDREI, N.:** Acceleration of Conjugate Gradient Algorithms for Unconstrained Optimization. ICI Technical Report, 24 October, 2008.
14. **ANDREI, N.:** Accelerated Conjugate Gradient Algorithm with Modified Secant Condition for Unconstrained Optimization. ICI Technical Report, 3 March, 2008.
15. **ANDREI, N.:** A Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization as a Convex Combination of Hestenes-Stiefel and Dai-Yuan. Studies in Informatics and Control, Vol. 17, No. 1, March 2008, pp. 55-70.
16. **Andrei, N.:** A Hybrid Conjugate Gradient Algorithm with Modified Secant Condition for Unconstrained Optimization as a Convex Combination of Hestenes-Stiefel and Dai-Yuan Algorithms. Studies in Informatics and Control, Vol. 17, 2008, pp. 373-392.
17. **ANDREI, N.:** Another Hybrid Conjugate Gradient Algorithm for Unconstrained Optimization. Numerical Algorithms. Vol. 47, No. 2 / February, 2008, pp. 143-156.
18. **ANDREI, N.:** Accelerated Conjugate Gradient Algorithm with Finite Difference Hessian / Vector Product Approximation for Unconstrained Optimization. ICI Techical Report, 4 March, 2008. (Acceptată de Journal of Computational and Applied Mathematics, 2009.)
19. **ANDREI, N.:** ASCALCG – Accelerated Scaled Memoryless BFGS Preconditioned Conjugate Gradient Algorithm for Unconstrained Optimization. Technical Report ICI, No. 17, 2008. (CD inclus).
20. **ANDREI, N.:** Open Problems in Nonlinear Conjugate Gradient Algorithms for Unconstrained Optimization. ICI Technical Report, 30 July, 2008.
21. **ANDREI, N.:** Numerical Experiments with Accelerated Conjugate Gradient Algorithm with Hessian / Vector Product - ACGHES - for Unconstrained Optimization. ICI Technical Report, September 22, 2008.
22. **ANDREI, N.:** An Accelerated Conjugate Gradient Algorithm with Guaranteed Descent and Conjugacy Conditions for Unconstrained Optimization. ICI Technical Report, 3 December, 2008.
23. **ARIOLI, M., D. LOGHIN:** Stopping Criteria for Mixed Finite Element Problems. RAL-TR-2006-010, 2007, pp. 1-15.
24. **ARMIJO, L.:** Minimization of Function Having Lipschitz Continuous First Partial Derivatives. Pacific Journal on Mathematics, 16, 1966, pp. 1-3.
25. **BEALE, E.M.L.:** A Derivation of Conjugate Gradients. În: Lotsma, F.A., (Ed.) Numerical Methods for Nonlinear Optimization. Academic Press, New-York, 1972, pp. 39-43.
26. **Birgin, E., J.M. Martínez:** A Spectral Conjugate Gradient Method for Unconstrained Optimization. Applied Math. and Optimization, No. 43, 2001, pp. 117-128.
27. **CONN, A.R., N.I.M. GOULD, PH.L. TOINT:** LANCELOT – A Fortran Package for Large-scale Nonlinear Optimization (Release A), Springer Verlag, Berlin, 1992.
28. **CONN, A.R., N.I.M. GOULD, PH.L. TOINT:** Improving the Decomposition of Partially Separable Functions in the Context of Large-scale Optimization: A First Approach. În: Hager, W.W., Hearn, D.W. and Pardalos, P.M., (Eds.) Large Scale Optimization: State of the Art, Kluwer Academic Publishers, 1994, pp. 82-94.
29. **CROWDER, H.P., P. WOLFE, P.:** Linear Convergence of the Conjugate Gradient Method. IBM J. Res. Dev., No. 16, 1969, pp. 431-433.
30. **DAI, Y.H., L.Z. LIAO, L.Z.:** New Conjugate Conditions and Related Nonlinear Conjugate Gradient Methods, Appl. Math. Optim., NO. 43, 2001, pp. 87-101.
31. **DAI, Y.H., Y. YUAN, Y.**: A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property, SIAM J. Optim., No. 10, 1999, pp. 177-182.
32. **DANIEL, J.W.:** The Conjugate Gradient Method for Linear and Nonlinear Operator Equations, SIAM J. Numer. Anal., No. 4, 1967, pp. 10-26.
33. **DENNIS, J.E., R.B. SCHNABEL:** Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

34. **DIXON, L.C.W., P.G. DUCKSBURY, P. SINGH:** A New Three-term Conjugate Gradient Method. Journal of Optimization Theory and Applications, No. 47, 1985, pp. 285-300.
35. **FLETCHER, R., C.M. REEVES:** Function Minimization by Conjugate Gradients Comput. J., No. 7, 1964, pp. 149-154.
36. **GOLDSTEIN, A.A.:** On Steepest Descent, SIAM J. Control, Vol. 3, 1965, pp. 147-151.
37. **GILBERT, J.C., J. NOCEDAL:** Global Convergence Properties of Conjugate Gradient Methods for Optimization, SIAM J. Optim., No. 2, 1992, pp. 21-42.
38. **GRIPPO, L., F. LAMPARIELLO, S. LUCIDI:** A Nonmonotone Line Search Technique for Newton's Method, SIAM Journal on Numerical Analysis, No. 23, 1986, pp. 707-716.
39. **HAGER, W.W., H. ZHANG:** A New Conjugate Gradient Method with Guaranteed Descent and an Efficient Line Search, SIAM Journal on Optimization, No. 16, 2005, pp. 170-192.
40. **HAGER, W.W., H. ZHANG:** A Survey of Nonlinear Conjugate Gradient Methods. Pacific Journal of Optimization, No. 2, 2006, pp. 35-58.
41. **HESTENES, M.R., E. STIEFEL:** Methods of Conjugate Gradients for Solving Linear Systems, J. Research Nat. Bur. Standards Sec. B. 48, 1952, pp. 409-436.
42. **HU, Y.F., C. STOREY:** Global Convergence Result for Conjugate Gradient Methods, J. Optim. Theory Appl., No. 71, 1991, pp. 399-405.
43. **LEMARECHAL, C.:** A View of Line Search. În: Auslander, Oettli, J. Stoer (Eds.) Optimization and Optimal Control, Springer, 1981, pp. 59-78.
44. **MORÉ, J.J. D.J. THUENTE:** On Line Search Algorithms with Guaranteed Sufficient Decrease. Mathematics and Computer Science Department Preprint MCD-P153-0590, Argonne National Laboratory, June, 1990.
45. **NAZARETH, J.L.:** A Conjugate Direction Algorithm without Line Search, Journal of Optimization Theory and Applications, No. 23, 1977, pp. 373-387.
46. **POLAK E., G. RIBIÈRE:** Note sur la convergence de methods de directions conjugues, Revue Francaise Informat. Reserche Opérationnelle, No. 16, 1969, pp. 35-43.
47. **POLYAK, B.T.:** The Conjugate Gradient Method in Extreme Problems, USSR Comp. Math. Math. Phys., No. 9, 1969, pp. 94-112.
48. **POTRA, F.A., Y. SHI:** Efficient Line Search Algorithm for Unconstrained Optimization, Journal of Optimization Theory and Applications, Vol. 85, 1995, pp. 677-704.
49. **POWELL, M.J.D.:** Some Convergence Properties of the Conjugate Gradient Method. Math. Programming, No. 11, 1976, pp. 42-49.
50. **POWELL, M.J.D.:** Restart Procedures for the Conjugate Gradient Method, Math. Programming, No. 12, 1977, pp. 241-254.
51. **SHANNO, D.F.:** Conjugate Gradient Methods with Inexact Searches, Mathematics of Operations Research, No. 3, 1978, pp. 244-256.
52. **SHANNO, D.F.:** On the Convergence of a New Conjugate Gradient Algorithm, SIAM J. Numer. Anal., No. 15, 1878, pp. 1247-1257.
53. **TOUATI-AHMED, D., C. STOREY:** Efficient Hybrid Conjugate Gradient Techniques, Journal of Optuimization Theory and Applications, No. 64, 1990, pp. 379-397.
54. **WOLFE, P.:** Convergence Conditions for Ascent Methods, SIAM Rev., No. 11, 1969, pp. 226-235.
55. **YABE, H., M. TAKANO:** Global Convergence Properties of Nonlinear Conjugate Gradient Methods with Modified Secant Condition, Computational Optimization and Applications, No. 28, 2004, pp. 203-225.
56. **YUAN Y.:** Some Problems in Nonlinear Programming. În: Yuan, Y., (Ed.) Numerical Linear Algebra and Optimization, Science Press, Beijing / New York, 2003, pp. 90-110.
57. **ZHANG, J.Z., N.Y. DENG, L.H. CHEN:** New Quasi-Newton Equation and Related Methods for Unconstrained Optimization, J. Optim. Theory Appl., No. 102, 1999, pp. 147-167.
58. **ZHANG, J.Z., C.X. XU:** Properties and Numerical Performance of Quasi-Newton Methods with Modified Quasi-Newton Equations, Journal of Computational and Applied Mathematics, No. 137, 2001, pp. 269-278.

30 ianuarie, 2009