ACADEMY OF ROMANIAN SCIENTISTS

Dr. Neculai ANDREI

METODE AVANSATE
DE
GRADIENT CONJUGAT
PENTRU
OPTIMIZARE FĂRĂ RESTRICŢII

Editura
ACADEMIEI OAMENILOR DE ŞTIINŢĂ DIN ROMÂNIA
Bucureşti, 2009

Tipar:
Editura Academiei Oamenilor de Ştiinţă din România
Splaiul Independenţei nr. 54 sector 5, 050094, Bucureşti, ROMÂNIA
Tel. 00-4021/314.74.91. Fax. 00-4021/314.75.39
Web-site: www.aos.ro, e-mail: aosromania@yahoo.com

Descrierea CIP a Bibliotecii Naţionale a României

ANDREI, NECULAI
Metode avansate de gradient conjugat pentru optimizare fără restricţii/
dr. Neculai Andrei - Bucureşti: Editura Academiei Oamenilor Ştiinţă din România, 2009

ISBN 978-606-92161-0-1

519.8

Referenţi ştiinţifici şi revizia lucrării: Membrii secţiei **Ştiinţa şi Tehnologia Informaţiei** din cadrul **Academiei Oamenilor de Ştiinţă din România**

Şef serviciu Editură: ing. Liviu Mihai **Sima**
Redactor: prof. Andrei D. **Petrescu**
Procesare text: dr. Neculai **Andrei**
Coperta: ec. Ovidiu **Oprea**

C.Z. pentru biblioteci mari: 517.392:512
C.Z. pentru biblioteci mici: 512

ADVANCED CONJUGATE GRADIENT METHODS FOR UNCONSTRAINED OPTIMIZATION

Unconstrained optimization consists in finding values for some variables so that a given function in these variables takes the minimum or maximum possible value. When the number of variables is large enough, this operation turns out to be quite complex. This book presents the state-of-the-art of conjugate gradient algorithms for unconstrained optimization. Both the classical and the latest conjugate gradient methods for solving the unconstrained optimization problems are considered. As we know, unconstrained optimization is not merely an intellectual exercise. Its purpose is to solve practical unconstrained optimization problems on computers. Therefore, the book contains a large number of conjugate gradient algorithms advanced techniques of computational linear algebra which form the kernel of any industrial optimization algorithm.

The monograph is organized in 6 chapters and two annexes. The first chapter has an introductory character, where the main aspects of the descent methods for unconstrained optimization are presented. In chapter 2 the linear conjugate gradient methods for solving linear algebraic systems of equations are discussed. A special attention is given to the discretization of partial differential equations. With chapter 3 the book enters into the subject of nonlinear conjugate gradient methods for unconstrained optimization. The theory of classical conjugate gradient algorithms is presented both of those methods with $\|g_k\|^2$ into the denominator of β_k, and for the methods with $g_{k+1}^T y_k$ into the denominator of β_k. Chapter 4 introduces the hybrid conjugate gradient methods as projections of classical conjugate gradient methods or as convex combinations of these methods. Chapter 5 contains a large number of new conjugate gradient algorithms, some of them based on the very recent results of the author. The last chapter is dedicated to present the computational performances of conjugate gradient algorithms for solving 7 applications from MINPACK2 collection. The first annex reviews a number of 50 nonlinear conjugate gradient algorithms, and the second one articulates some open problems in these methods.

The book addresses to a large number of scientists, theorists, as well as practitioners engineers and decision makers, researchers, computer scientists or professional bodies working in academy, business, industry and government facing design optimization models and algorithms, who are interested in the theory of conjugate gradient methods for unconstrained optimization, in computational methods associated to conjugate gradient algorithms, in numerical experiments and comparisons, as well as in behavior of these algorithms for solving real applications from different areas of activity.

Dr. Neculai Andrei

Din partea autorului

Elaborată într-o manieră riguros matematică cu ilustraţii computaţionale de mare anvergură, monografia constituie o sinteză a celor mai recente rezultate din domeniul metodelor de gradient conjugat pentru optimizare neliniară fără restricţii din care multe aparţin autorului.

Efortul depus în această monografie constă în a sugera şi dovedi cititorului că metodele de gradient conjugat sunt o alternativă foarte serioasă la celelalte metode de optimizare fără restricţii bazate pe conceptul de aproximaţie pătratică, în ceea ce priveşte rezolvarea problemelor de optimizare fără restricţii de mari dimensiuni aşa cum acestea apar în aplicaţiile practice. Ideea este că această clasă de algoritmi beneficiază de rezultate teoretice de convergenţă globală şi complexitate computaţională foarte tari şi în acelaşi timp pot fi cu uşurinţă implementaţi în programe de calcul care implică cerinţe modeste de memorie. Importanţa acestor algoritmi rezidă şi în faptul că aceştia se pot include foarte uşor în alte programe de calcul dedicate rezolvării aplicaţiilor complexe care implică optimizarea fără restricţii.

Autorul mulţumeşte Editurii Academiei Oamenilor de Ştiinţă pentru profesionalismul şi solicitudinea cu care a susţinut apariţia acestei lucrări.

Bucureşti, 20 Iunie 2009

Prefaţă

Optimizarea fără restricţii reprezintă un capitol foarte important al programării matematice. Aceasta se referă la optimizarea unei funcţii (suficient de netedă), în sensul de a calcula acele puncte în care funcţia respectivă îşi atinge valorile extreme, de minim sau maxim. Importanţa optimizării fără restricţii rezultă din faptul că foarte multe fenomene reale din Natură se pot modela ca probleme de acest tip. Pe de altă parte, problemele de optimizare cu restricţii, care constituie o clasă de probleme mult mai generale şi mai bogată, se pot soluţiona prin rezolvarea succesivă a unui şir de probleme de optimizare fără restricţii. De exemplu, metodele barieră din optimizarea cu restricţii au condus la o revigorare a domeniului optimizării fără restricţii, în care metodele de direcţii conjugate ocupă un loc central.

Programarea matematică în general şi optimizarea fără restricţii în particular au atins o anumită maturitate dispunând de un corp de rezultate teoretice şi computaţionale bine stabilit. În acest sens, pentru rezolvarea acestei clase de probleme se cunosc o multitudine de algoritmi bine justificaţi matematic, pentru care se cunosc proprietăţile de convergenţă şi complexitate computaţională şi care au fost implementaţi în programe de calcul deosebit de sofisticate şi eficiente capabile să rezolve probleme complexe de optimizare fără restricţii cu un număr mare de variabile.

*În programarea matematică s-au individualizat două mari clase de probleme: **programarea liniară** şi **programarea matematică neliniară**. Programarea liniară este o problemă în care toate funcţiile (funcţia de optimizat şi restricţiile) sunt liniare. Chiar dacă programarea liniară şi optimizarea neliniară s-au dezvoltat în paralel, rareori şi ocazional având puncte de contact, anul 1984 a constituit începutul unei „revoluţii" în programarea matematică, prin reconsiderarea la un nivel superior a unor tehnici de optimizare bazate pe conceptul de penalizare şi care au devenit cunoscute ca metode de punct interior. Aceasta a constituit **prima unificare din programarea matematică**, în sensul că aceleaşi tehnici pot fi utilizate pentru rezolvarea ambelor clase de probleme. Cel mai important avantaj al metodelor de punct interior asupra metodei simplex pentru programarea liniară este eficienţa cu care pot fi rezolvate probleme de mari dimensiuni şi abilitatea de a utiliza direcţii de căutare aproximative, care permit utilizarea unor metode de gradient conjugat bazate pe structura problemei. Un alt avantaj este că metodele de punct interior se extind foarte bine la problemele de optimizare convexă, în timp ce metoda simplex, care se bazează foarte mult pe structura liniară a problemei, nu. Această observaţie a condus la construcţia unui cadru matematic foarte general al metodelor de punct interior pentru rezolvarea unei mari varietăţi de probleme de optimizare nediferenţiale convexe, ce deschide*

foarte multe perspective teoretice şi computaţionale. Aşa au apărut o serie de noi probleme de optimizare ca programarea semidefinită, programarea pe conuri, optimizarea combinatorială neconvexă, minimizarea normelor matriceale, aproximarea Chebychev logaritmică etc. toate acestea încadrându-se în aşa-numita clasă a ***inegalităţilor matriceale liniare*** *şi care a condus în cele din urmă la* ***a doua unificare dintre optimizare, teoria sistemelor şi control optimal****. Astfel, s-a observat că tehnici şi metode proprii unor domenii foarte diferite ca optimizarea neliniară, teoria sistemelor dinamice sau controlul optimal utilizează aceleaşi concepte matematice cu interpretări proprii.*

Scopul lucrării*. Întregul nostru efort de-a lungul acestei monografii a fost de a prezenta riguros matematic, cu suficiente detalii, într-un mod accesibil şi prietenos, principalele aspecte ale metodelor de gradient conjugat pentru optimizarea fără restricţii. Având în vedere faptul că scopul final al oricărei teorii, în particular al optimizării fără restricţii, este de a dezvolta scheme computaţionale robuste şi de încredere pentru rezolvarea problemelor concrete, lucrarea îşi propune prezentarea principiilor care stau la baza metodelor de gradient conjugat, a algoritmilor corespunzători împreună cu caracteristicile lor de convergenţă şi complexitate, precum şi capabilităţile lor de a rezolva probleme practice reale cu un număr mare de variabile.*

Rigoare şi formalism*. În prezentarea materialului referitor la metodele avansate de gradient conjugat pentru optimizare fără restricţii am adoptat ideea ca în locul unui formalism de tipul definiţie – teoremă – demonstraţie să utilizăm o prezentare narativă în care acest formalism este scufundat şi ilustrat prin comentarii şi exemple numerice relevante, care arată puterea computaţională a algoritmilor. În acest sens am urmărit prezentarea riguroasă a propoziţiilor, lemelor şi teoremelor importante care stabilesc proprietăţile de convergenţă şi complexitate computaţională a algoritmilor, precum şi ilustrarea acestor rezultate prin rezolvarea unor colecţii de probleme de test şi a unor aplicaţii reale din diferite domenii industriale. Principalul efort în organizarea acestui material a fost de a se evita pe cât posibil apelul la concepte definite în secţiunile ulterioare ale lucrării, argumentarea circulară, precum şi utilizarea unor noţiuni, tehnici şi metode considerate ca bine cunoscute în domeniu.*

Structura lucrării*. Cartea este organizată în 6 capitole şi două anexe. În acelaşi timp aceasta conţine o bibliografie cu cele mai relevante şi cele mai recente lucrări din domeniu, precum şi un index de termeni şi un index de autori.*

♦ *Primul capitol are un caracter introductiv. Acesta prezintă conceptele fundamentale cu care se operează în domeniul optimizării fără restricţii, metodele de direcţii descendente utilizate pentru rezolvarea acestei clase de probleme, precum şi câteva rezultate de bază asupra metodelor de direcţii descendente.*

♦ *Capitolul doi prezintă metoda gradientului conjugat pentru rezolvarea sistemelor de ecuaţii algebrice liniare cu matricea coeficienţilor simetrică şi pozitiv definită. Se arată că viteza de convergenţă a metodei de gradient conjugat este puternic dependentă de distribuţia valorilor proprii a matricei sistemului. Totodată, în acest*

capitol se insistă asupra discretizării ecuaţiilor diferenţiale cu derivate parţiale şi a sistemelor liniare obţinute prin discretizare, inclusiv structura de valori proprii.

♦ *Capitolul trei este dedicat metodelor clasice de gradient conjugat pentru optimizare fără restricţii. Principiul pe care se bazează metodele de gradient conjugat constă în faptul că minimizarea unei funcţii* f *pe un subspaţiu dat este echivalentă cu minimizarea succesivă de-a lungul unor direcţii conjugate care generează acel subspaţiu. Aceste metode se reprezintă sub forma* $x_{k+1} = x_k + \alpha_k d_k$, *unde* α_k *este lungimea pasului, iar* d_k *este direcţia de deplasare calculată sub forma* $d_{k+1} = -g_{k+1} + \beta_k d_k$. *Astfel, se prezintă teoria convergenţei algoritmilor de gradient conjugat, separat pentru acele metode cu* $\|g_k\|^2$ *la numărătorul parametrului* β_k *şi respectiv cu* $g_{k+1}^T y_k$ *la numărătorul lui* β_k. *Secţiunea 3.6 a acestui capitol este originală şi prezintă o metodă de accelerare a algoritmilor de gradient conjugat [Andrei, 2008h].*

♦ *Capitolul patru prezintă metodele hibride şi parametrizate. Se cunosc două tehnici de elaborare a algoritmilor hibrizi. Prima utilizează conceptul de proiecţie a algoritmilor clasici de gradient conjugat. A doua tehnică este originală şi se bazează pe conceptul de combinaţie convexă, care deschide o nouă direcţie de cercetare în ceea ce priveşte elaborarea de algoritmi de optimizare fără restricţii [Andrei, 2007g,h,i; 2008a,b,j,k,l,n,s; 2009a,b].*

♦ *În capitolul cinci se prezintă noi metode de gradient conjugat pentru optimizare fără restricţii. Cele mai importante sunt: metode de gradient conjugat BFGS precondiţionate fără memorie date de Shanno [1978a,b], o metodă de gradient conjugat cu descendenţă garantată a lui Hager şi Zhang [2005], metode de gradient conjugat scalat şi scalat accelerate elaborate de Andrei [2007a,b,c], metode de gradient conjugat cu descendenţă suficientă [Andrei, 2006b], metoda gradientului conjugat cu ecuaţia secantei modificată [Andrei, 2005e; 2008b], metoda gradientului conjugat cu aproximaţia produsului Hessian / vector [Andrei, 2008o,u] şi metode de gradient conjugat cu descendenţă şi conjugare garantate [Andrei, 2008v].*

♦ *Capitolul şase prezintă un număr de şase aplicaţii de optimizare fără restricţii rezolvare cu algoritmii de gradient conjugat prezentaţi de-a lungul acestei lucrări, precum şi comparaţii între ei şi algoritmii quasi-Newton BFGS cu memorie limitată şi Newton trunchiat.*

***Proiecte computaţionale**. Ideea pe care am avut-o în elaborarea acestei lucrări a fost de a scrie un text matematic, riguros şi logic prezentat, care în final să se poată utiliza în dezvoltarea unor programe de calcul capabile să rezolve probleme de optimizare fără restricţii, modele ale unor fenomene fizice reale. Pentru realizarea acestui demers algoritmii asociaţi metodelor de gradient conjugat sunt prezentaţi pe paşi cu suficiente detalii computaţionale. În acelaşi timp, lucrarea conţine o multitudine de studii numerice care implică minimizarea unui tren de peste 750 de funcţii de test de optimizare fără restricţii cu numărul de variabile din domeniul* [1000,10000].

Optimizare aplicată. *Există o întreagă dispută asupra termenului „aplicat". Aceasta arată diferenţele conceptuale care există între cei care sunt interesaţi de teoria optimizării şi practiţionişti, interesaţi mai mult de rezolvarea unor probleme concrete. În această monografie am adoptat un punct de vedere rabinic în sensul de a mulţumi şi teoreticianul şi practiţionistul. În acest sens lucrarea prezintă rezultatele furnizate de o multitudine de algoritmi de gradient conjugat în ceea ce priveşte rezolvarea unor aplicaţii concrete ca: torsiunea elasto-plastică a unei bare, distribuţia presiunii într-un lagăr de alunecare (cuzinet), proiectarea optimă a unei bare cu rigiditate torsională maximă, rezolvarea ecuaţiilor Ginzburg-Landau pentru superconductoare neomogene formate din straturi de plumb şi staniu în absenţa unui câmp magnetic, combustia staţionară a unui material solid, configuraţia de energie minimă a unui grup de atomi, etc.*

Adresabilitate. *Parcurgerea şi înţelegerea acestei lucrări cere o anumită maturitate matematică în sensul abilităţii de a opera cu anumite concepte şi obiecte matematice din algebra liniară, calculul diferenţial, analiza convexă, topologie, geometrie diferenţială şi ecuaţii diferenţiale. Ca atare, monografia se adresează celor interesaţi de teoria optimizării fără restricţii şi în special de metodele de gradient conjugat: matematicieni, analişti din domeniul cercetărilor operaţionale şi al ştiinţei managementului, planificatori, programatori, ingineri, cercetători şi doctoranzi în specialităţile care includ utilizarea tehnicilor de optimizare, studenţi interesaţi de aceste probleme.*

Mulţumiri. *Autorul mulţumeşte* Fundaţiei „**Alexander von Humboldt**" *pentru generozitatea şi solicitudinea cu care a ştiut să-l sprijine de-a lungul timpului atât în perioada stagiului de peste trei ani petrecuţi în câteva Universităţi din Germania, cât şi după aceea. Aceasta ne-a permis o implicare directă în cercetarea ştiinţifică de cel mai înalt nivel, posibilitatea audierii şi prezentării unor rezultate din domeniile modelării matematice şi ale optimizării şi, în general, încadrarea în acea clasă rafinată a oamenilor de ştiinţă pentru care intelectualismul ca doctrină prin care cunoaşterea – parţial sau total – se obţine prin raţiune pură, reprezintă un mod de existenţă. Scrierea acestei monografii apare deci ca o datorie de a recompensa, măcar în parte, generozitatea acestei Fundaţii, care în esenţa ei nici nu are nevoie de aşa ceva. De asemenea, autorul mulţumeşte celor câţiva colegi şi prieteni, care de-a lungul timpului cât a durat scrierea acestei monografii au ştiut să-l încurajeze cu solicitudine şi umor. În acelaşi timp, autorul este recunoscător soţiei sale Mihaela, care a înţeles efortul său şi a ştiut să-l scutească de o serie de activităţi lungi şi obositoare, precum şi copiilor Teodor-Marius, Denisa şi Cosmin pentru afecţiunea şi înţelegerea cu care l-au înconjurat şi sprijinit în efortul de a concretiza o muncă depusă pe parcursul a mai bine de 40 de ani.*

Bucureşti
Ianuarie 20, 2009 **Neculai Andrei**

CUPRINS

SIMBOLURI ŞI NOTAŢII

$x \in X$	x aparţine mulţimii X.
$x \notin X$	x nu aparţine mulţimii X.
R	Mulţimea numerelor reale (axa reală).
$\|x\|$	Modulul numărului x.
$\lfloor j \rfloor$	Cel mai mare întreg mai mic decât sau egal cu j.
R^n	Spaţiul real $n-$dimensional.
$ls\{d_1,\ldots,d_k\}$	Subspaţiul liniar generat de vectorii $d_1,\ldots,d_k$.
$K_k(A,b)$	$k-$spaţiul Krylov al matricei A corespunzător vectorului b. $K_k(A,b) = ls\{b, Ab,\ldots,A^{k-1}b\}$.
$x \in R^n$	Vector (coloană) din spaţiul R^n.
x^T	Transpusul vectorului x (vector linie).
$x^T y$	Produsul scalar al vectorului $x \in R^n$ cu vectorul $y \in R^n$, definit ca: $x^T y = \sum_{i=1}^{n} x_i y_i$.
$\cos\langle x, y\rangle$	Cosinusul unghiului dintre vectorii x şi y. $\cos\langle x, y\rangle = \dfrac{x^T y}{\|x\|\|y\|}$.
$x \perp y$	Vectorul x este perpendicular pe vectorul y: $x^T y = 0$.
$\|x\|_2$	Norma l_2 (norma euclidiană) a vectorului x: $\|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$. Deseori se notează sub forma $\|x\|$.
$\|x\|_Q$	Norma vectorului x în raport cu matricea simetrică Q: $\|x\|_Q = (x^T Q x)^{1/2}$, (norma elipsoid).
$\|x-y\|_2$	Distanţa dintre două puncte $x, y \in R^n$, distanţa euclidiană: $\|x-y\|_2 = \sqrt{(x_1-y_1)^2+\cdots+(x_n-y_n)^2}$.
$A=[a_{ij}]$	Matrice cu elementele a_{ij}. a_{ij} este elementul din linia i şi coloana j. Dacă $i=1,\ldots,m$ şi $j=1,\ldots,n$, atunci A este o matrice cu m linii şi n coloane.
I_n	Matricea unitate de ordin n. Se mai notează cu I.

A^T	Transpusa matricei A.
A^{-1}	Inversa matricei A. $AA^{-1} = I$.
$R^{m\times n}$	Mulţimea tuturor matricelor $m\times n$ dimensionale cu elemente numere reale.
$\|A\|_2$	Norma spectrală a matricei A: $\|A\|_2 = \left(\lambda_{\max}(A^T A)\right)^{1/2}$.
$\|A\|_F$	Norma Frobenius a matricei A: $\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n}\|a_{ij}\|^2\right)^{1/2}$.
$\kappa(A)$	Numărul de condiţionare a matricei A: $\kappa(A) = \sigma_{\max}/\sigma_{\min}$, unde $\sigma_{\min}$ şi $\sigma_{\max}$ sunt valoarea singulară minimă, respectiv maximă a matricei A. $\kappa(A) = \|A\|\|A^{-1}\|$.
$diag(a_1,\ldots,a_n)$	Matrice diagonală cu elementele de pe diagonala principală egale cu $a_1,\ldots,a_n$.
$f: X \to Y$	Funcţia f definită pe mulţimea X cu valori în mulţimea Y.
$dom\ f$	Domeniul funcţiei f.
$\nabla f(x)$	Gradientul funcţiei f în x se defineşte ca vectorul $\nabla f(x) = \left[\frac{\partial f(x)}{\partial x_1},\cdots,\frac{\partial f(x)}{\partial x_n}\right]^T$. În limbajul curent gradientul funcţiei f în punctul x_k se notează cu g_k.
$\nabla^2 f(x)$	Hessianul funcţiei f se defineşte ca $n\times n$-matricea simetrică cu elementele: $[\nabla^2 f(x)]_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$, $i, j = 1,\ldots,n$.
$T_k(x)$	Polinomul Cebyshev de prima speţă. $T_k(x) = \cos(k \arccos(x))$.
TST	Matrice Toeplitz, simetrică, tridiagonală.
DFP	Actualizarea quasi-Newton Davidon-Fletcher-Powell.
BFGS	Actualizarea quasi-Newton Broyden-Fletcher-Goldfarb-Shanno.
LBFGS	Actualizarea BFGS cu memorie limitată.
■	Sfârşitul demonstraţiei.

Capitolul 1

METODE DE DIRECŢII DESCENDENTE PENTRU OPTIMIZARE FĂRĂ RESTRICŢII

Scopul acestui capitol este de a prezenta cu suficiente detalii metodele de bază ale optimizării fără restricţii care utilizează informaţiile date de gradientul şi eventual de Hessianul funcţiei de minimizat. Nu vom intra în detalii. De aceea recomandăm cititorului interesat consultarea lucrărilor [Andrei, 1999a,b; 2000a,b; 2002; 2003; 2004a,b; 2009c] unde va găsi o multitudine de aspecte teoretice, computaţionale şi practice privind algoritmii dedicaţi rezolvării acestei clase de probleme.

Orice algoritm de optimizare este compunerea a două aplicaţii: prima determină direcţia de deplasare către minim, o direcţie descendentă. A doua determină lungimea pasului de deplasare de-a lungul direcţiei calculate. În acest capitol vom discuta ambele aceste aspecte ale unui algoritm de optimizare insistând numai asupra acelor metode care s-au dovedit eficiente mai ales pentru rezolvarea problemelor neliniare complexe cu un număr mare de variabile.

Ideea care stă la baza oricărui algoritm de optimizare, inclusiv a celor de optimizare fără restricţii, este foarte simplă. Se consideră un *punct iniţial* din care se demarează calculele. În acest punct se determină o direcţie care este direcţionată către minimul funcţiei de minimizat, numită *direcţie de descendenţă*. De-a lungul acestei direcţii se execută o deplasare convenabilă, numită *căutare liniară*, astfel obţinându-se un nou punct din care se pot relua calculele. **T**oate aceste elemente care definesc o metodă descendentă de optimizare au propriile lor principii şi subtilităţi care ilustrează bogăţia şi maturitatea domeniului optimizării fără restricţii. Ceea ce individualizează un algoritm faţă de altul este modul în care se calculează direcţia de deplasare. Lungimea pasului se calculează într-o manieră standard utilizând anumite condiţii care asigură o reducere suficient de mică a valorilor funcţiei de minimizat de-a lungul direcţiei de descendenţă.

După cum am spus, algoritmii de optimizare fără restricţii, proiectaţi pentru a rezolva problema $\min_{x \in R^n} f(x)$, unde $f : R^n \to R$, suficient de netedă, se identifică după modul în care calculează direcţia de deplasare. Se cunosc o multitudine apreciabilă de algoritmi de optimizare fără restricţii, care se pot organiza în următoarele clase de metode de optimizare:

1. Metode de căutare directă (nederivative)
2. Metoda pasului descendent (Cauc**h**y)
3. Metoda Newton
   - Metoda Newton modificată
   - Metoda Newton discretă
   - Metoda Newton compusă
   - Metoda Newton trunchiată
4. Metode quasi-Newton
   - Metoda quasi-Newton cu memorie limitată
   - Metode quasi-Newton partiţionate
5. Metoda regiunii de încredere
6. Metode de direcţii conjugate

Principii de optimizare fără restricţii

Construcţia algoritmilor de optimizare fără restricţii se bazează pe următoarele două principii fundamentale care stau la baza tuturor metodelor de direcţii descendente de optimizare fără restricţii [Andrei, 2009c].

1. Primul principiu fundamentează ideea că orice problemă de optimizare într-un punct dat din domeniul funcţiei de minimizat se poate foarte bine aproxima printr-o problemă de optimizare pătratică. Cu alte cuvinte, *într-un punct curent* x_k *se calculează o aproximaţie pătratică a problemei* Q_k *care prin rezolvare furnizează un nou punct* x_{k+1} *din care procesul se poate imediat repeta până când un anumit criteriu de oprire a iteraţiilor este îndeplinit.* Această idee este foarte bună şi se bazează pe faptul că modelele matematice ale fenomenelor fizice suficient de netede nu sunt puternic neliniare şi deci se pot aproxima cu o acurateţe bună prin modele pătratice. Reprezentative pentru acest principiu sunt metoda Newton şi variantele sale, metodele quasi-Newton şi metoda regiunii de încredere.
2. Al doilea principiu, complet diferit de primul se bazează pe *conceptul de direcţie conjugată.* Dată o matrice simetrică şi pozitiv definită A, atunci direcţiile $d_1 \neq 0$ şi $d_2 \neq 0$ sunt conjugate în raport cu matricea A, dacă $d_1^T A d_2 = 0.$ Evident că acest concept de conjugare se poate imediat extinde la cazul funcţiilor generale neliniare prin considerarea în locul matricei A a Hessianului acesteia în punctul curent. Această idee a condus la clasa metodelor de direcţii conjugate cu care a demarat optimizarea fără restricţii de mari dimensiuni (milioane de variabile). Metodele de direcţii conjugate pot împrumuta anumite idei din metodele Newton sau quasi-Newton în sensul de a încerca înglobarea informaţiei de ordinul doi dată de Hessianul funcţiei de minimizat sau de o aproximare a acestuia, obţinându-se astfel algoritmi foarte puternici de optimizare fără restricţii.

Să considerăm problema $min\, f(x)$, unde $f : R^n \to R$ este o funcţie de două ori continuu diferenţiabilă pe R^n cu domeniul $dom\, f$. Deoarece f este diferenţiabilă,

atunci o condiţie necesară pentru ca punctul x^* să fie soluţia optimă a problemei este:

$$\nabla f(x^*) = 0 .$$

Aceasta se numeşte *condiţia necesară de optimalitate de primul ordin*. Punctele care o satisfac se numesc *puncte staţionare* sau *critice*. Ca atare, rezolvarea problemei a fost redusă la determinarea unei soluţii pentru sistemul $\nabla f(x^*) = 0$, care, după cum se vede, este un sistem algebric de n ecuaţii în n variabile. Ca atare, minimizarea funcţiilor este intim legată de rezolvarea sistemelor algebrice neliniare. Un punct staţionar nu este întotdeauna un minim al funcţiei. De exemplu, funcţia $f(x) = -x^4$ satisface condiţiile necesare în punctul 0, care după cum se vede este un punct de maxim al lui f. Pentru a obţine un punct de minim trebuie ca derivata de ordinul doi să fie strict pozitivă. Aceasta este *condiţia suficientă de optimalitate.*

De obicei pentru rezolvarea problemei vom folosi o metodă iterativă, care generează un şir de puncte $x_0, x_1, \cdots, x_k, \cdots \in dom\, f$ cu proprietatea că $f(x^k) \to f^*$ când $k \to \infty$. Un astfel de şir cu această proprietate se numeşte *şir minimizant*. Astfel conceput, un algoritm care generează un şir minimizant se opreşte de îndată ce un anumit criteriu de oprire a iteraţiilor este îndeplinit. De exemplu, se pot utiliza următoarele criterii de oprire a iteraţiilor: $\left| f(x_k) - f^* \right| \le \varepsilon_f$, unde $\varepsilon_f > 0$ este o toleranţă specificată, sau distanţa dintre estimaţiile variabilelor $\left\| x_{k+1} - x_k \right\|_2 \le \varepsilon_x$, unde $\varepsilon_x > 0$, sau norma gradientului funcţiei de minimizat $\left\| \nabla f(x_k) \right\|_2 \le \varepsilon_g$ sau $\left\| \nabla f(x_k) \right\|_\infty \le \varepsilon_g$, unde $\varepsilon_g > 0$, de-a lungul iteraţiilor. Evident că se pot utiliza şi combinaţii ale acestor criterii, etc.

1.1. ALGORITMUL GENERAL DE DIRECŢII DESCENDENTE

Să considerăm deci problema

$$min\ f(x) \tag{1.1.1}$$

unde $f : R^n \to R$ este o funcţie continuu diferenţiabilă 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 direcţia descendentă de deplasare d_k, atunci schema iterativă după care se calculează punctul de optim x^* pentru (1.1.1) este

$$x_{k+1} = x_k + \alpha_k d_k , \tag{1.1.2}$$

unde α_k este *lungimea pasului de deplasare* de-a lungul direcţiei d_k. Menţionăm că o direcţie descendentă în punctul x_k satisface condiţia $\nabla f(x_k)^T d_k < 0$, numită

condiţia de descendenţă, precum şi condiţia de a fi mărginită faţă de zero, $\|d_k\| \neq 0$. Prima condiţie înseamnă că aceasta trebuie să facă un unghi ascuţit cu gradientul cu semn schimbat al funcţiei în punctul curent x_k. Dacă $\alpha_k \neq 0$, vedem că a doua condiţie face ca (1.1.2) să se încadreze în *principiul fundamental de construcţie a algoritmilor de analiză numerică, care recomandă ca o estimaţie a unei soluţii a unei probleme să se calculeze ca o mică modificare de mare acurateţe a estimaţiei anterioare.*

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

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

trebuie să fie închisă. Impunerea condiţiei de închidere a mulţimii S asigură eliminarea posibilităţii convergenţei algoritmului într-un punct de pe frontiera lui $dom\, f$. Dacă $dom\, f = R^n$, atunci condiţia este satisfăcută pentru orice x_0. Fie

$$\varphi(\alpha) = f(x_k + \alpha d_k). \tag{1.1.3}$$

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

Dacă α_k se determină astfel încât funcţia f este minimizată în direcţia d_k, adică

$$f(x_k + \alpha_k d_k) = \min_{\alpha > 0} f(x_k + \alpha d_k) \tag{1.1.4}$$

sau

$$\varphi(\alpha_k) = \min_{\alpha > 0} \varphi(\alpha), \tag{1.1.5}$$

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 obţine o reducere acceptabilă a funcţiei 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*.

Menţionăm că deoarece în situaţiile 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 excepţii (cazul funcţiilor 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 funcţiei de minimizat. Este clar că din punctul de vedere al costului computaţional este mult mai bine să ne mulţumim cu o reducere acceptabilă a valorilor funcţiei de-a lungul direcţiei de căutare, decât cu reducerea optimă (maximală). În principiu, structura unui algoritm general de direcţii descendente este următoarea.

Algoritmul 1.1.1. (Algoritm general de optimizare fără restricţii)

Pasul 1.	*Iniţializare*. Se consideră un punct iniţial $x_0 \in R^n$, precum şi toleranţa de terminare a iteraţiilor algoritmului de minimizare ε suficient de mică. Se pune $k = 0$.
Pasul 2.	Se determină *direcţia de descendenţă* d_k.
Pasul 3.	Se calculează *lungimea pasului* α_k astfel încât, de exemplu: $$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k),$$ sau utilizând condiţiile Wolfe (căutare aproximativă).
Pasul 4.	Se pune $$x_{k+1} = x_k + \alpha_k d_k.$$
Pasul 5.	*Continuarea iteraţiilor.* Dacă $\|\nabla f(x_{k+1})\| \leq \varepsilon$, atunci stop; altfel se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Câteva precizări sunt necesare.

1. Observăm că aceasta este structura generală a unui algoritm de determinare a unui optim local al problemei (1.1.1). Determinarea optimului global este o problemă mult mai complicată şi încă nu avem la dispoziţie algoritmi generali pentru rezolvarea acestei probleme. Totuşi, dacă funcţia de minimizat f este convexă, atunci se demonstrează că orice optim local al funcţiei f este de asemenea un optim global.
2. Referitor la iniţializarea algoritmului (vezi pasul 1), în general orice punct x_0 poate fi acceptat ca punct iniţial. Pentru o problemă concretă de optimizare fără restricţii un astfel de punct iniţial se poate preciza din anumite consideraţii fizice, ceea ce constituie o foarte bună metodă de iniţializare. În general, pentru probleme de optimizare (fără restricţii) de obicei odată cu prezentarea problemei se furnizează şi punctul iniţial, care într-un anumit fel sens va caracteriza performanţele algoritmului de optimizare utilizat. Cu alte cuvinte, algoritmii de optimizare fără restricţii se comportă diferit la iniţializări diferite. Cel mai pregnant exemplu este metoda Newton. Bine iniţializată, adică iniţializată lângă soluţie, algoritmul este pătratic convergent la punctul de optim. Iniţializată departe de soluţie, metoda Newton poate fi divergentă. Deseori, problemele sunt însoţite de anumite „puncte patologice" care pun algoritmii de optimizare în dificultăţi majore. Pentru cazul metodelor de optimizare fără restricţii, în afara cazurilor practice când din consideraţii fizice se poate stabili un punct iniţial, de obicei nu se utilizează o procedură de iniţializare care să determine punctul x_0. Această practică se utilizează în cazul problemelor de optimizare (liniare sau neliniare) cu restricţii pentru care se pot imagina proceduri foarte sofisticate de iniţializare.
3. Calculul direcţiei de descendenţă d_k, din pasul 2, este elementul care caracterizează algoritmul de rezolvare a problemei (1.1.1). Altfel spus, metodele de optimizare se clasifică după modul de generare a direcţiei d_k. Cerinţa minimală asupra direcţiei d_k este aceea de descendenţă, adică direcţia trebuie să fie astfel încât o mică deplasare de-a lungul acesteia să conducă la reducerea valorilor funcţiei f. Condiţia de

descendenţă este $\nabla f(x_k)^T d_k < 0.$ Aceasta se poate justifica foarte simplu prin utilizarea aproximării:

$$f(x_k + \alpha d_k) - f(x_k) \cong \alpha \nabla f(x_k)^T d_k,$$

de unde vedem imediat că negativitatea membrului drept din aproximarea de mai sus garantează reducerea valorilor funcţiei de minimizat când ne deplasăm cu un pas suficient de mic α de-a lungul direcţiei d_k.

4. Determinarea lungimii pasului de deplasare α_k din pasul 3 al algoritmului reprezintă un element crucial în ceea ce priveşte eficienţa algoritmului de optimizare. Pentru calculul lui α_k se pot utiliza două tehnici generale: căutare liniară sau metoda regiunii de încredere. În esenţă, metoda de căutare liniară este o problemă de minimizare unidimensională. Procedurile moderne se bazează pe o căutare liniară economică. Acestea sunt descrise într-o serie de lucrări ca: Andrei [1999a (partea a IV-a), 2009c], [Dennis şi Schnabel, 1983], [Peressini, Sullivan şi Uhl, 1988]. Metoda regiunii de încredere încearcă determinarea lungimii pasului printr-o procedură de căutare a acesteia într-un domeniu definit de parametrul regiunii de încredere.
5. Testul de continuare a iteraţiilor din pasul 5 implică utilizarea unor criterii de convergenţă care califică un punct ca soluţia optimă a problemei.

1.2. METODE DE DIRECŢII DESCENDENTE

Algoritmul general de optimizare fără restricţii se concretizează prin precizarea elementelor care-l definesc, în special în ceea ce priveşte direcţia de căutare a optimului. Am văzut mai sus că se cunosc şase metode generale de optimizare fără restricţii. Metodele de căutare directă nederivative încearcă determinarea punctului de minim utilizând numai valorile funcţiei de minimizat. Deşi aceste metode sunt active (vezi de exemplu Powell [1998, 2000, 2001, 2002]), totuşi acestea sunt utilizate numai pentru situaţia în care numărul de variabile este redus, expresia algebrică a gradientului este greu de obţinut sau evaluarea gradientului este consumatoare de timp. Detalii asupra acestor metode se pot găsi în Andrei [2009c]. În cele ce urmează vom prezenta pe scurt metodele de direcţii descendente care şi-au dovedit eficacitatea în ceea ce priveşte rezolvarea problemelor reale de optimizare fără restricţii.

1.2.1. METODA PASULUI DESCENDENT

Aceasta este una dintre cele mai vechi metode de optimizare fără restricţii fiind propusă de Augustin Cauchy în 1848. Importanţa ei este mai mult teoretică decât practică şi într-un anumit sens se află la originea tuturor metodelor de direcţii descendente. La fiecare iteraţie direcţia recomandată de metoda pasului descendent este $-\nabla f(x_k)$, adică a gradientului cu semn schimbat al funcţiei de minimizat. Iteraţiile se calculează sub forma $x_{k+1} = x_k + \alpha_k d_k$, unde α_k este lungimea pasului de deplasare determinată din condiţia de reducere a valorilor funcţiei $\varphi_k(\alpha) = f(x_k - \alpha \nabla f(x_k))$. Metoda este foarte simplu de implementat, cerinţele

de memorie fiind de ordinul $O(n)$, unde n este numărul de variabile ale funcţiei f. Dacă α_k este determinat prin căutare liniară exactă, atunci metoda pasului descendent evoluează în paşi perpendiculari. Adică, dacă $\{x_k\}$ este şirul de puncte generat de metodă, atunci vectorul care uneşte x_k cu x_{k+1} este ortogonal celui care uneşte x_{k+1} cu x_{k+2}. Această comportare în paşi perpendiculari face ca metoda să fie lentă şi deci computaţional ineficientă. Pentru cazul funcţiilor pătratice convexe, rata de convergenţă este liniară. În general, rata de convergenţă nu este mai mare decât $\left[(\kappa-1)/(\kappa+1)\right]^2$, unde κ este numărul de condiţionare a Hessianului funcţiei de minimizat [Andrei, 2004a]. O variantă a algoritmului este pasul descendent relaxat în care $x_{k+1} = x_k + \theta_k \alpha_k d_k$, unde θ_k este o variabilă aleatoare uniform distribuită în intervalul $[0,1]$ [Andrei, 2004c; 2005a,b]. În 1988 Barzilai şi Borwein au propus un nou algoritm de gradient descendent care pentru lungimea pasului utilizează o aproximaţie în două puncte a ecuaţiei secantei din cadrul metodelor quasi-Newton. Utilizând o strategie de globalizare bazată pe căutarea liniară nemonotonă, Raydan [1997] a arătat convergenţa globală a acestui algoritm pentru funcţii nepătratice. Utilizând căutarea liniară cu backtracking, metoda a fost modificată şi accelerată prin modificarea lungimii pasului ca în [Andrei, 2006a].

1.2.2. METODA NEWTON ŞI VARIANTELE SALE

Toate metodele Newton se bazează pe principiul de aproximare locală a funcţiei de minimizat printr-un model pătratic. În punctul x_k de-a lungul direcţiei d_k modelul pătratic al funcţiei f are expresia

$$f(x_k + d) \cong f(x_k) + g_k^T d + \frac{1}{2} d^T G_k d, \tag{1.2.1}$$

unde $g_k = \nabla f(x_k)$ şi $G_k = \nabla^2 f(x_k)$. Minimul membrului drept din (1.2.1) este realizat de vectorul d_k care minimizează funcţia pătratică

$$m_p(d) = g_k^T d + \frac{1}{2} d^T G_k d. \tag{1.2.2}$$

Deci, direcţia Newton d_k satisface sistemul algebric liniar de n ecuaţii cu tot atâtea necunoscute

$$G_k d = -g_k, \tag{1.2.3}$$

cunoscut ca *sistemul Newton.* Această direcţie este bine definită dacă matricea G_k este nesingulară. Când punctul iniţial x_0 cu care se demarează calculele este suficient de aproape de soluţia x^* a problemei se demonstrează că metoda Newton este pătratic convergentă [Andrei, 2004a,b]. Aceasta înseamnă că numărul de cifre semnificative precise din soluţie se dublează la fiecare iteraţie. Principalele avantaje ale metodei Newton sunt:

1. Iniţializată într-un punct suficient de apropiat de soluţia optimă şi dacă G_k este nesingulară, atunci metoda converge pătratic la soluţie.
2. Pentru funcţii pătratice, metoda Newton furnizează o soluţie exactă într-o singură iteraţie.
3. Pentru componentele afine ale gradientului funcţiei de minimizat, la fiecare iteraţie, metoda este exactă.

Dezavantajele metodei Newton sunt:

1. Pentru multe probleme nu este global convergentă.
2. La fiecare iteraţie implică evaluarea Hessianului funcţiei de minimizat.
3. Implică rezolvarea la fiecare iteraţie a unui sistem algebric liniar care poate fi prost condiţionat.

Aceste caracteristici ale metodei Newton se pot utiliza pentru elaborarea de noi algoritmi de optimizare fără restricţii. Astfel, orice combinaţie a următoarelor trei tehnici poate constitui o modificare a metodei Newton:

1. Introducerea la fiecare iteraţie a unei căutări liniare de-a lungul direcţiei Newton, adică adoptarea iteraţiei $x_{k+1} = x_k + \alpha_k d_k$, unde α_k este lungimea pasului determinată printr-o procedură de căutare liniară. Aceasta este o modificare de globalizare a metodei Newton.
2. Considerarea la fiecare iteraţie a unei aproximaţii (simetrice şi pozitiv definite) a matricei Hessian, sau a unei aproximaţii a inversei matricei Hessian. Acestea au condus la metodele quasi-Newton.
3. Rezolvarea inexactă la fiecare iteraţie a sistemului Newton, adică determinarea aproximativă direcţiei Newton d_k din sistemul Newton. Aceasta a condus la metoda Newton trunchiată sau inexactă.

Menţionăm faptul că metoda Newton este foarte rar utilizată în aplicaţiile practice concrete. Mai importante sunt variantele acesteia pe care le vom discuta imediat.

Metoda Newton modificată

Dacă Hessianul G_k al funcţiei de minimizat nu este o matrice pozitiv definită, atunci vom considera o modificare a acestei matrice de forma $E(x_k) = G_k + \mu_k I$, unde parametrul μ_k este ales astfel încât $E(x_k)$ să fie pozitiv definită. Pentru a determina μ_k observăm că pentru orice $\mu > 0$, $x^T(G_k + \mu I) = x^T G_k x + \mu \|x\|^2$. Dacă definim

$$\hat{\mu}_k = max_{\|x\|=1} \left| x^T G_k x \right|, \tag{1.2.4}$$

atunci se poate arăta că $\hat{\mu}_k$ este egal cu valoarea absolută a celei mai mari valori proprii a Hessianului funcţiei de minimizat. Dacă $\mu_k > \hat{\mu}_k$ şi $x \neq 0$, atunci

$$x^T(G_k + \mu_k I)x = \|x\|^2 \left[\frac{x}{\|x\|} G_k \frac{x}{\|x\|} + \mu_k \right] > \|x\|^2 \left(-\hat{\mu}_k + \mu_k \right) > 0,$$

ceea ce arată că $E(x_k)$ este pozitiv definită.

Metoda Newton discretă

După cum se vede metoda Newton (standard) aşa cum a fost prezentată, la fiecare iteraţie implică evaluarea a n funcţii (componentele gradientului) şi $n(n+1)/2$ funcţii care definesc Hessianul G_k. Metoda Newton discretă înlocuieşte evaluarea funcţiilor Hessianului prin determinarea unei aproximări prin diferenţe finite a elementelor Hessianului. Fiecare coloană i a lui G_k se poate aproxima prin vectorul

$$\frac{g(x_k + h_i e_i) - g(x_k)}{h_i},$$

unde h_i este un pas convenabil ales pentru a balansa erorile de rotunjire (proporţionale cu $1/h_i$) cu erorile de trunchiere (proporţionale cu h_i) [Andrei, 1999a, capitolul 15]. Într-o aritmetică exactă metoda Newton discretă converge pătratic la soluţie dacă fiecare h_i tinde la zero odată cu norma gradientului. Totuşi, erorile de rotunjire limitează mărimea lui h_i şi deci acurateţea soluţiei. Când gradientul devine foarte mic este posibilă o pierdere a acurateţei prin deteriorarea numerică a numărătorului expresiei de mai sus. Deci, metoda Newton discretă nu este recomandată pentru rezolvarea problemelor de mari dimensiuni, cu excepţia cazurilor în care Hessianul are o structură de matrice rară cunoscută şi această structură este utilizată în schema de aproximare a elementelor Hessianului.

Metoda Newton compusă

Dacă nu ţinem seama de efortul de calcul implicat de evaluarea funcţiei f, a gradientului g_k şi a Hessianului G_k, atunci numărul de operaţii algebrice pe iteraţie este de ordinul $O(n^3)$ fiind reclamat de factorizarea matricei G_k sau a unei modificări a ei $G_k + \mu_k I$. Evident, pentru valori mari ale lui n factorizarea este o problemă foarte serioasă. O tehnică de reducere a efortului de calcul pe iteraţie este dată de aşa numita *metoda Newton simplificată* în care la fiecare iteraţie se rezolvă sistemul algebric liniar $\nabla^2 f(x_0) d_k = -g_k$ în privinţa direcţiei d_k. Evident, în cazul metodei Newton modificate se rezolvă sistemul $(\nabla^2 f(x_0) + \mu_0 I) d_k = -g_k$. Observăm că metoda Newton simplificată implică factorizarea iniţială a lui $G_0 = \nabla^2 f(x_0)$ şi apoi utilizarea acestei factorizări de-a lungul tuturor iteraţiilor. În acest caz efortul de calcul este de ordinul $O(n^2)$ operaţii algebrice, dar preţul plătit este că algoritmul are o convergenţă liniară.

Pentru a depăşi situaţiile extreme date de metoda Newton şi metoda Newton simplificată este foarte natural să imaginăm o variantă a metodei Newton care între doi paşi Newton consideră m paşi Newton simplificaţi. Astfel, prin *metoda Newton compusă de ordinul* m înţelegem un proces iterativ în care pentru $i = 0, 1, ..., m$ se rezolvă sistemele algebrice liniare

$$\nabla^2 f(x_k)d_i = -g(x_k + d_0 + d_1 + ... + d_{i-1}) \qquad (1.2.5)$$

în privinţa lui d_i, şi estimaţia soluţiei de la pasul $k+1$ se calculează sub forma

$$x_{k+1} = x_k + \alpha_k(d_0 + d_1 + ... + d_m), \qquad (1.2.6)$$

pentru $k = 0,1,....$ Ortega şi Rheinboldt [1970, capitolul 10] demonstrează că în condiţii standard metoda Newton compusă este superliniar convergentă. O variantă interesantă este dată de *metoda Newton compusă de ordinul 1* în care un pas al metodei Newton este compus cu un pas al metodei Newton simplificate sub forma: pentru $k = 0,1,...$ se rezolvă sistemul liniar $\nabla^2 f(x_k)d_N = -g(x_k)$ în privinţa lui d_N, după care se rezolvă sistemul $\nabla^2 f(x_k)d_S = -g(x_k + d_N)$ în privinţa lui d_S cu care se calculează următoarea estimaţie $x_{k+1} = x_k + \alpha_k(d_N + d_S)$. Această variantă a metodei Newton compusă este atribuită lui Traub [1964]. Ortega şi Rheinboldt [1970] demonstrează convergenţa cubică a acestei metode.

Metoda Newton compusă este eficientă mai ales pentru cazurile în care n este mare şi efortul de calcul pentru evaluarea funcţiei f este mic. O situaţie de acest fel este întâlnită în cazul metodelor de punct interior de urmărire a traiectoriei pentru rezolvarea problemelor de programare liniară sau neliniară [Andrei, 1999a,b; 2009c].

Observăm că fiecare iteraţie a metodei Newton compusă de ordinul m se poate privi ca o iteraţie majoră şi este rezultatul a $(m+1)$ iteraţii minore (interioare). Numărul mediu de operaţii algebrice pe o iteraţie interioară este de ordinul $O((n^3 + mn^2)/(m+1))$. Vedem că pentru m mare acesta este $O(n^2)$. Mai mult, viteza medie de convergenţă pentru iteraţiile interioare este $(m+2)^{1/(m+1)}$, care pentru m mare tinde la 1. Deci, pentru valori mari ale lui m metoda Newton compusă de ordinul m se comportă asemănător metodei Newton simplificate. Ca atare, implementările eficiente în programe de calcul ale metodei Newton compuse nu numai că nu trebuie să modifice pe m la fiecare iteraţie, dar trebuie să menţină pe m relativ mic [Potra şi Rheinboldt, 1986], [Potra, 1987].

Metoda Newton trunchiată

Această variantă a metodei Newton a fost introdusă de Dembo, Eisenstat şi Steihaug [1982] şi a fost analizată de Dembo şi Steihaug [1983], Deuflhard [1990] şi implementată în programe foarte sofisticate de Nash [1985], Nash şi Sofer [1990], Schlick şi Overton [1987] şi Schlick şi Fogelson [1992a, b]. Ideea de bază a acestei metode este că departe de soluţie nu este necesar să rezolvăm sistemul Newton cu o foarte mare acurateţe. Ca atare, iteraţiile care definesc o soluţie a sistemului Newton se pot trunchia de îndată ce s-a obţinut o soluţie aproximativă a acestui sistem. De aici provine şi numele de trunchiată sau inexactă a metodei. Deci pentru $k = 0,1,\ldots$ se determină d_k astfel încât

$$\left\|\nabla^2 f(x_k)d_k + g(x_k)\right\| \le \eta_k \left\|g(x_k)\right\|, \qquad (1.2.7)$$

după care se calculează următoarea estimaţie $x_{k+1} = x_k + \alpha_k d_k$.

Problemele privind analiza acestei variante a metodei Newton sunt legate de modul de definire a şirului de „forţare" $\{\eta_k\}$. Acesta lucrează ca un „regulator" care controlează reziduul $\left\|\nabla^2 f(x_k)d_k + g(x_k)\right\|$. Analiza dată de Dembo, Eisenstat şi Steihaug [1982] a condus la următorul rezultat. Fie f o funcţie continuu diferenţiabilă şi $\{\eta_k\}$ un şir de numere reale cu $0 \le \eta_k \le \eta < 1$. Atunci metoda Newton trunchiată, iniţializată într-un punct aflat într-o vecinătate a soluţiei, care generează şirul $\{x_k\}$ definit de

$$\nabla^2 f(x_k)d_k = -g(x_k) + r_k, \tag{1.2.8}$$

unde

$$\frac{r_k}{\|g(x_k)\|} \le \eta_k, \quad x_{k+1} = x_k + d_k$$

este bine definit şi converge cel puţin liniar la x^*. Dacă $\{\eta_k\}$ converge la zero, atunci $\{x_k\}$ converge superliniar la x^*. Dacă $\{\eta_k\} = O(\|g(x_k)\|^p)$, pentru $0 \le p \le 1$, atunci $\{x_k\}$ converge la x^* cu ordinul $1+p$.

Rezultatul de mai sus garantează convergenţa pătratică dacă la fiecare iteraţie pasul Newton aproximativ d_k verifică relaţia

$$\frac{\left\|\nabla^2 f(x_k)d_k + g(x_k)\right\|}{\|g(x_k)\|} \le min\{1, \|g(x_k)\|\}. \tag{1.2.9}$$

În practică deseori cantitatea din membrul stâng din (1.2.9), numită „reziduu relativ", este calculată în timpul rezolvării aproximative a sistemului Newton. Astfel (1.2.9) furnizează un criteriu convenabil de oprire a iteraţiilor în procesul de rezolvare a sistemului Newton. Una dintre cele mai bune metode de rezolvare aproximativă a sistemului Newton este metoda de gradient conjugat precondiţionată, pe care o vom prezenta cu detalii în capitolele următoare ale acestei lucrări. Cea mai bună alegerea lui η_k este necunoscută. Singura cerinţa este ca $\eta_k \to 0$. Practic s-au utilizat următoarele două alegeri: $\eta_k = 1/k$ sau $\eta_k = \|g(x_k)\|$. A doua alegere implică o convergenţă pătratică [Dembo şi Steihaug, 1983]. În aplicaţiile practice ambele aceste alegeri sunt implementate. Metoda Newton trunchiată cunoaşte două abordări.

A) *Abordarea Schlick-Fogelson* [1992a,b]. Rezolvarea aproximativă a sistemului Newton (1.2.3) precondiţionat, în care matricea Hessian este calculată analitic sau prin diferenţe finite.
B) *Abordarea Nash* [1985]. Aproximarea matricei Hessian utilizând actualizarea BFGS şi rezolvarea aproximativă a sistemului Newton corespunzător.

În ambele abordări rezolvarea aproximativă a sistemului **N**ewton se face prin metode de direcţii conjugate. O justificare a acestei selecţii este dată de faptul că metoda de direcţii conjugate are proprietatea că modelul pătratic asociat funcţiei de minimizat în punctul curent este minimizat pe subspaţiul generat de aceste direcţii.

În continuare să prezentăm un algoritm de tip Newton trunchiat în care la fiecare iteraţie externă se calculează o aproximaţie B_k a Hessianului printr-o schemă de actualizare de tip BFGS, de exemplu – cunoscută ca abordarea Nash.

Algoritmul 1.2.1. (TN: Newton trunchiat general – Nash)

Pasul 1.	*Iniţializare.* Se consideră un punct iniţial x_0. Se pune $k = 0$. Se selectează toleranţa ε de calcul a minimului funcţiei f, precum şi parametrii ρ şi σ din condiţiile de căutare liniară Wolfe.
Pasul 2.	Se calculează: $f(x_k)$, $g_k = \nabla f(x_k)$, precum şi o aproximaţie B_k a Hessianului $\nabla^2 f(x_k)$ (de exemplu BFGS).
Pasul 3.	Se execută testul de oprire a iteraţiilor. Dacă $\|g_k\|_\infty \le \varepsilon(1+\|f(x_k)\|)$ sau $\|g_k\|_\infty \le \varepsilon$, atunci stop, altfel se continuă cu pasul 4.
Pasul 4.	*Iteraţiile interioare.* Se calculează o soluţie aproximativă d_k a sistemului $B_k d_k = -g_k$ dată de algoritmul 1.2.2.
Pasul 5.	*Determinarea lungimii pasului.* Se calculează α_k care verifică condiţiile Wolfe $$f(x_k + \alpha d_k) \le f(x_k) + \rho\alpha\nabla f(x_k)^T d_k,$$ $$\nabla f(x_{k+1})^T d_k \ge \sigma\nabla f(x_k)^T d_k.$$
Pasul 6.	Se calculează $x_{k+1} = x_k + \alpha_k d_k$, se pune $k = k+1$ şi se continuă cu pasul 2. ♦

În pasul 4 al algoritmului 1.2.1, care implementează iteraţiile interioare, direcţia d_k se calculează prin algoritmul de gradient conjugat în care B_k acţionează ca o matrice de precondiţionare.

Algoritmul 1.2.2. (Iteraţia interioară: Gradient conjugat)

Pasul 1.	Se consideră: $d_k^0 = 0$, $r_0 = -g_k$, $p_0 = r_0$ şi $\delta_0 = r_0^T r_0$. Se pune $i = 0$.
Pasul 2.	Se calculează $q_i = B_k p_i$. Dacă $p_i^T q_i \le \varepsilon\delta_i$, atunci dacă $i = 0$ se pune $d_k = p_0$, sau altfel dacă $i \ne 0$, atunci se pune $d_k = d_k^i$, stop.
Pasul 3.	Se calculează:

	$a_i = \dfrac{r_i^T r_i}{p_i^T q_i}, \qquad d_k^{i+1} = d_k^i + a_i p_i, \qquad r_{i+1} = r_i - a_i q_i.$ Dacă $\dfrac{\|r_{i+1}\|}{\|g_k\|} \le \eta_k$, atunci se pune $d_k = d_k^i$ şi stop.
Pasul 4.	Continuarea iteraţiilor de gradient conjugat. Se calculează $b_i = \dfrac{r_{i+1}^T r_{i+1}}{r_i^T r_i}, \quad p_{i+1} = r_{i+1} + b_i p_i, \quad \delta_{i+1} = r_{i+1}^T r_{i+1} + b_i^2 \delta_i,$ se pune $i = i+1$ şi se continuă cu pasul 2. ♦

Alegerea şirului $\{\eta_k\}$ este una dintre cele mai dificile probleme din cadrul algoritmului Newton trunchiat. Dembo şi Steihaug [1983] demonstrează că dacă

$$\eta_k = min\left\{\frac{1}{k}, \|g_k\|^t\right\},$$

unde $0 < t \le 1,$ atunci metoda Newton trunchiată converge cu o rată de convergenţă de ordinul $O(1+t).$ Această alegere a şirului $\{\eta_k\}$ conduce la un algoritm adaptiv pentru rezolvarea problemei de minimizare a funcţiei $f.$ Când suntem departe de soluţie $\|g_k\|$ este mare şi ca atare criteriul $\|r_{i+1}\| \le \eta_k \|g_k\|$ cere un efort de calcul redus. Când $g_k \to 0$, ceea ce implică $\eta_k \to 0$, criteriul de oprire a iteraţiilor interne $\|r_{i+1}\| \le \eta_k \|g_k\|$ forţează ca direcţia dată de algoritmul de gradient conjugat 1.2.2 să fie apropiată de direcţia Newton atât ca modul cât şi ca direcţie propriu-zisă. Aceasta asigură convergenţa „aproape pătratică". Observăm simplitatea algoritmului 1.2.2. Acesta nu implică decât produse scalare şi un produs matrice-vector, în care apare aproximarea B_k a matricei Hessian.

1.2.3. METODE QUASI-NEWTON

Să notăm cu d_k^N direcţia Newton, adică $d_k^N = -\nabla^2 f(x_k)^{-1} g_k.$ Un rezultat clasic, foarte important, stabilit de Dennis şi Moré [1974, 1977] arată că algoritmul $x_{k+1} = x_k + \alpha_k d_k$ este superliniar convergent dacă şi numai dacă

$$\alpha_k d_k = d_k^N + o(\|d_k^N\|). \tag{1.2.10}$$

Deci, pentru a atinge o viteză mărită de convergenţă este necesar ca algoritmul să aproximeze asimptotic pasul Newton. Acesta este principiul lui Dennis şi Moré. Problema este cum putem face aceasta fără a evalua la fiecare iteraţie matricea Hessian. Răspunsul a fost dat de Davidon [1959] şi apoi dezvoltat şi popularizat de Fletcher şi Powell [1963]. Ideea este de a demara calculele cu o aproximare oarecare a matricei Hessian şi la fiecare iteraţie să se actualizeze această matrice prin utilizarea curburii problemei măsurată de-a lungul direcţiei de deplasare. Făcută cu grijă această actualizare conduce la metode deosebit de eficiente şi

robuste, cunoscute sub numele de *metode quasi-Newton* sau încă *metode de metrică variabilă*. Detalii asupra acestor metode se găsesc în [Andrei, 2009c].

Această schemă computaţională a fost una novatoare care a revoluţionat domeniul optimizării fără restricţii, furnizând o alternativă foarte serioasă pentru metoda Newton. De-a lungul timpului s-au propus mai multe variante de metode quasi-Newton, dar din 1970 s-a observat că, în general, metoda BFGS (Broyden, Fletcher, Goldfarb, Shanno) este cea mai bună. Algoritmul metodei BFGS este următorul.

Algoritmul 1.2.3. (Algoritmul BFGS)

Pasul 1.	*Iniţializare*. Se consideră un punct iniţial $x_0 \in R^n$, o matrice simetrică şi pozitiv definită B_0, care aproximează într-un anumit sens matricea Hessian, precum şi toleranţa de terminare a iteraţiilor algoritmului de minimizare ε suficient de mică. Se pune $k = 0$.
Pasul 2.	Se determină direcţia de căutare d_k ca soluţie a sistemului $B_k d_k = -g_k$.
Pasul 3.	Se determină lungimea pasului α_k şi se calculează $x_{k+1} = x_k + \alpha_k d_k$
Pasul 4.	Se testează un criteriu de oprire a iteraţiilor. De exemplu dacă $\|\nabla f(x_{k+1})\| \le \varepsilon$, atunci stop.
Pasul 5.	Se actualizează aproximaţia matricei Hessian B_k sub forma $$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \qquad (1.2.11)$$ unde $s_k = x_{k+1} - x_k$ şi $y_k = g_{k+1} - g_k$. Se pune $k = k+1$ şi se continuă cu pasul 2. ♦

Observăm că cele două matrice de corecţie care apar în (1.2.11) au fiecare rangul unu. Ca atare, teorema de deplasare a valorilor proprii [Wilkinson, 1965] ne arată că prima matrice de corecţie de rang unu care se scade, descreşte valorile proprii – zicem că le deplasează la stânga. A doua care se adună deplasează valorile proprii la dreapta. Deci, trebuie să existe un anumit echilibru în ceea ce priveşte deplasarea valorilor proprii, altfel aproximaţia (1.2.11) ar putea deveni fie singulară, fie arbitrar de mare ceea ce conduce la eşecul metodei. Convergenţa globală a metodei BFGS se obţine din studiul acestor deplasări ale valorilor proprii. Utilizând operatorii urma şi determinantul unei matrice, Powell [1976] a măsurat efectul celor două matrice de corecţie asupra lui B_k arătând că dacă f este convexă atunci pentru orice matrice simetrică şi pozitiv definită B_0 şi orice punct iniţial x_0 metoda BFGS realizează $\liminf_{k\to\infty} \|g_k\| = 0$. Dacă în plus şirul $\{x_k\}$ converge la o soluţie a problemei în care Hessianul este o matrice pozitiv definită, atunci rata de convergenţă a met**o**dei este superliniară.

Analiza lui Powell a fost extinsă de Byrd, Nocedal şi Yuan [1987] la clasa Broyden a metodelor quasi-Newton în care (1.2.11) este înlocuit de o expresie ceva mai generală:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \Phi(s_k^T B_k s_k) v_k v_k^T, \qquad (1.2.12)$$

unde scalarul $\Phi \in [0,1]$ şi

$$v_k = \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}. \qquad (1.2.13)$$

Dacă în (1.2.12) considerăm $\Phi = 0,$ atunci obţinem metoda (sau actualizarea) BFGS, în timp ce $\Phi = 1$ furnizează metoda DFP (Davidon, Fletcher, Powell). Pentru cazul problemelor convexe, Byrd, Nocedal şi Yuan [1987] au demonstrat convergenţa globală şi superliniară a tuturor metodelor din clasa Broyden cu excepţia metodei DFP. Modul de abordare eşuează când $\Phi = 1$ şi lasă cazul nerezolvat. Studiile numerice intensive au arătat că metoda quasi-Newton BFGS este cea mai bună în clasa Broyden.

Metoda quasi-Newton cu memorie limitată

Metoda quasi-Newton cu memorie limitată se referă în principal la metoda BFGS cu memorie limitată. Ideea este că în loc de a memora matricele H_k care aproximează inversa matricei Hessian, se memorează un anumit număr m de perechi de vectori (s_i, y_i) care le definesc. Produsul $H_k g_k$ care defineşte direcţia de căutare este obţinut prin executarea unui număr de produse scalare care implică vectorul g_k şi cele mai recente m perechi de vectori (s_i, y_i). După calculul unei noi iteraţii, cea mai veche pereche din mulţimea perechilor (s_i, y_i) se elimină şi una nouă este adăugată. Astfel, algoritmul întotdeauna păstrează cele mai recente m perechi (s_i, y_i) cu care se construieşte aproximarea inversei Hessianului la iteraţia următoare.

Să detaliem procesul de actualizare cu memorie limitată. Presupunem că iteraţia curentă este x_k şi se dispune de m perechi (s_i, y_i), $i = k-m, \ldots, k-1$. Pentru început, aşa după cum a recomandat Oren şi Spedicato [1976], se formează matricea „*iniţială*" $H_k^{(0)} = \gamma_k I$, unde

$$\gamma_k = \frac{s_k^T y_k}{y_k^T y_k}. \qquad (1.2.14)$$

Pentru $k+1 > m$ actualizarea H_{k+1} se calculează astfel:

$$\begin{aligned} H_{k+1} = {} & (V_k^T V_{k-1}^T \cdots V_{k-m+1}^T) H_k^{(0)} (V_{k-m+1} \cdots V_{k-1} V_k) \\ & + \rho_{k-m+1} (V_k^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_k) \\ & + \rho_{k-m+2} (V_k^T \cdots V_{k-m+3}^T) s_{k-m+2} s_{k-m+2}^T (V_{k-m+3} \cdots V_k) \end{aligned}$$

$$+\cdots$$
$$+\rho_{k-1}V_k^T s_{k-1}s_{k-1}^T V_k$$
$$+\rho_k s_k s_k^T, \qquad (1.2.15)$$

unde

$$V_k = I - \rho_k y_k s_k^T \quad \text{şi} \quad \rho_k = \frac{1}{y_k^T s_k}.$$

Formula de actualizare (1.2.15) este cunoscută ca formula L-BFGS. Nash şi Nocedal [1991] arată că valorile lui m din intervalul $3 \le m \le 7$ dau cele mai bune rezultate. Algoritmul acestei actualizări este următorul.

Algoritmul 1.2.4. (Algoritmul general L-BFGS cu actualizarea matricei H_k)

Pasul 1.	*Iniţializare.* Se alge $x_0 \in R^n$, parametrul m, şi matricea $H_0 \in R^{n\times n}$, precum şi toleranţa $\varepsilon > 0$ de oprire a iteraţiilor. Se pune $k = 0$.
Pasul 2.	Dacă $\|g_k\| \le \varepsilon$, stop.
Pasul 3.	Se calculează: $d_k = -H_k g_k,$ $x_{k+1} = x_k + \alpha_k d_k,$ unde α_k este lungimea pasului de deplasare, determinată prin căutare liniară exactă sau inexactă (de exemplu Wolfe).
Pasul 4.	Fie $\hat{m} = min\{k, m-1\}$. Dacă $y_k^T s_k \le 0$, atunci punem $H_{k+1} = I$ (pasul descendent) şi eliminăm toate perechile (s_i, y_i), $i = k-\hat{m},\ldots,k$. Dacă $y_k^T s_k > 0$, atunci punem $H_{k+1} = (V_k^T V_{k-1}^T \cdots V_{k-\hat{m}}^T) H_0 (V_{k-\hat{m}} \cdots V_{k-1} V_k)$ $+\rho_{k-\hat{m}}(V_k^T \cdots V_{k-\hat{m}+1}^T) s_{k-\hat{m}} s_{k-\hat{m}}^T (V_{k-\hat{m}+1} \cdots V_k)$ $+\cdots$ $+\rho_{k-1} V_k^T s_{k-1} s_{k-1}^T V_k$ $+\rho_k s_k s_k^T.$
Pasul 5.	Punem $k = k+1$ şi se continuă cu pasul 2. ♦

Metodele quasi-Newton cu memorie limitată se pot privi ca extensii ale metodelor gradientului conjugat în ceea ce priveşte accelerarea convergenţei acestora. Principalul avantaj al acestor metode constă în faptul că utilizarea memoriei poate fi controlată de utilizator. În acelaşi timp, metodele quasi-Newton cu memorie limitată se pot privi şi ca metode quasi-Newton în care utilizarea memoriei este restricţionată. Simplitatea acestor metode se manifestă în aceea că acestea nu implică cunoaşterea structurii matricei Hessian, sau a separabilităţii acesteia.

Metoda quasi-Newton partiţionată

Griewank şi Toint [1982a] au demonstrat că orice funcţie f cu Hessianul matrice rară este parţial separabilă, adică se poate scrie sub forma:

$$f(x) = \sum_{i=1}^{ne} f_i(x), \tag{1.2.16}$$

unde fiecare din cele ne funcţii „elementare" f_i depind numai de câteva variabile. Aceasta constituie fundamentul pentru aşa numitele metode quasi-Newton partiţionate pentru optimizare de mari dimensiuni. Ideea care stă la baza acestei metode este de a exploata structura parţial separabilă (1.2.16) şi de a actualiza o aproximaţie B_k^i a Hessianului fiecărei funcţii elementare f_i. Aceste matrice se pot asambla pentru a defini aproximaţia B_k a Hessianului funcţiei f. Complicaţia cu această abordare este că chiar dacă $\nabla^2 f(x^*)$ este pozitiv definită, câteva funcţii elementare pot fi concave, astfel încât metoda BFGS nu poate fi întotdeauna aplicată. Direcţiile de căutare în cadrul metodei Newton partiţionate sunt determinate ca soluţii ale sistemului liniar

$$\left(\sum_{i=1}^{ne} B_k^i \right) d_k = -g_k, \tag{1.2.17}$$

în interiorul unei regiuni de încredere, utilizând o iteraţie de gradient conjugat trunchiată. Dacă se detectează o direcţie de curbură negativă, atunci iteraţiile de gradient conjugat sunt oprite (în aceasta constă trunchierea) şi d_k este selectat ca această direcţie de curbură negativă.

Metodele quasi-Newton partiţionate lucrează bine în practică şi reprezintă o contribuţie importantă în domeniul optimizării neliniare. Multe probleme practice sunt formulate ca în (1.2.16), sau se pot transforma la această structură. În cazul problemelor convexe convergenţa globală a acestor metode este similară cu a metodei BFGS, tot ceea ce trebuie să presupunem este convexitatea tuturor funcţiilor elementare f_i. În această ipoteză, Griewank [1991] arată că metoda quasi-Newton partiţionată este global convergentă c**h**iar dacă sistemul (1.2.17) este rezolvat inexact.

1.2.4. METODA REGIUNII DE ÎNCREDERE

După cum am văzut, ideea pe care se bazează metoda Newton este de a aproxima funcţia de minimizat $f(x)$ în jurul punctului curent x_k printr-un model pătratic

$$q^{(k)}(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T G_k s, \tag{1.2.18}$$

şi de a utiliza minimul s_k al lui $q^{(k)}(s)$ pentru a defini următoarea aproximaţie a minimului x^* sub forma

$$x_{k+1} = x_k + s_k. \tag{1.2.19}$$

Astfel proiectată metoda garantează numai convergenţa locală, adică dacă s este suficient de mic, atunci metoda este global convergentă. Prin introducerea căutării liniare metoda devine global convergentă. În acest context, vedem că metoda Newton este o compunere de două aplicaţii. Prima utilizează modelul pătratic $q^{(k)}(s)$ pentru a determina o direcţie de căutare. A doua calculează o lungime convenabilă α_k a pasului de deplasare de-a lungul acestei direcţii. Deşi acest mod de abordare este foarte bun asigurând o bună rată de convergenţă atât pentru metoda Newton cât şi pentru variantele ei, totuşi aceasta nu utilizează suficient de bine modelul pătratic. Cu alte cuvinte, modelul pătratic nu este utilizat în determinarea lungimii pasului. Pe de altă parte, după cum am văzut, acest mod de operare eşuează dacă Hessianul nu este o matrice pozitiv definită.

Metoda regiunii de încredere nu numai că înlocuieşte căutarea liniară pentru a obţine convergenţa globală, dar depăşeşte problemele cauzate de nepozitiv definirea matricei Hessian. În plus, aceasta produce o reducere mult mai pronunţată în valorile funcţiei de minimizat decât o face căutarea liniară.

În metoda regiunii de încredere pentru început se defineşte o *regiune* în jurul punctului curent

$$\Omega_k = \left\{ x : \|x - x_k\| \le \Delta_k \right\}, \tag{1.2.20}$$

unde Δ_k este *raza* regiunii Ω_k, în care există încrederea că modelul pătratic $q^{(k)}(s)$ constituie o bună aproximare a funcţiei f. În continuare se determină un pas s_k care aproximează minimul modelului pătratic $q^{(k)}(s)$ în regiunea de încredere de mai sus. Cu alte cuvinte, se determină s_k astfel încât $x_k + s_k$ este cea mai bună aproximaţie a minimului lui $q^{(k)}(s)$ pe sfera generalizată

$$\left\{ x_k + s : \|s\| \le \Delta_k \right\} \tag{1.2.21}$$

centrată în punctul x_k de rază Δ_k. Dacă pasul s_k nu este acceptabil, atunci se reduce raza regiunii de încredere şi se procedează ca mai sus pentru a determina un nou minim în noua regiune de încredere. Observăm că astfel definită metoda reţine rata de convergenţă locală rapidă a metodelor Newton sau quasi-Newton având în acelaşi timp proprietatea de convergenţă globală. Deoarece pasul este restricţionat la regiunea de încredere, metoda se mai numeşte *metoda pasului restricţionat*.

Cu acestea, un model al subproblemei metodei regiunii de încredere este

$$\begin{aligned} &\min q^{(k)}(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s \\ &\text{referitor la} \\ &\|s\| \le \Delta_k, \end{aligned} \tag{1.2.22}$$

unde $\Delta_k > 0$ este raza regiunii de încredere şi B_k este o aproximaţie simetrică a matricei Hessian G_k. De obicei, în problema (1.2.22) se utilizează norma l_2 (norma euclidiană), aşa încât s_k este minimul lui $q^{(k)}(s)$ pe sfera centrată în x_k

de rază Δ_k. Dacă se utilizează alte norme, atunci se obţin alte forme ale regiuni**i** de încredere. Dacă în (1.2.22) se pune $B_k = \nabla^2 f(x_k)$, atunci se obţine chiar *metoda Newton a regiunii de încredere*.

Astfel prezentată, metoda regiunii de încredere ridică mai multe chestiuni: cum se alege Δ_k, cum se rezolvă subproblema (1.2.22), care sunt criteriile de oprire a iteraţiilor, etc. detalii descrise în [Andrei, 2009c].

1.3. STUDIU NUMERIC (NEWTON TRUNCHIAT VERSUS L-BFGS)

În această secţiune vom prezenta un studiu numeric în care vom detalia profilul de performanţă Dolan-Moré [2002] al algoritmului Newton trunchiat în abordarea B a lui Nash în comparaţie cu algoritmul quasi-Newton BFGS cu memorie limitată (LBFGS). Metoda Newton trunchiată este implementată în pachetul TN elaborat de Nash [1985]. Metoda quasi-Newton BFGS cu memorie limitată (LBFGS) este prezentată de Liu şi Nocedal [1989] şi implementată de Nocedal [1980]. În LBFGS condiţiile Wolfe de căutare liniară sunt implementate în maniera dată de Moré şi Thuente [1990, 1992, 1994]. Detalii privind pachetele TN şi LBFGS, modul de utilizare a acestora, precum şi experienţa computaţională cu aceste pachete se găsesc în lucrările [Andrei, 2007k] şi respectiv [Andrei, 2008r].

Pentru aceasta considerăm un set de 75 de funcţii de test, în formă generalizată sau extinsă [Andrei, 2008d]. Pentru fiecare funcţie considerăm un număr de 10 experimente numerice în care numărul de variabile este crescător: $n = 1000, 2000, ..., 10000$. Deci în total rezolvăm un tren de 750 de probleme de test de optimizare fără restricţii. Toţi algoritmii sunt implementaţi în Fortran şi utilizează aceeaşi procedură de determinare a lungimii pasului dată de condiţiile Wolfe, precum şi acelaşi criteriu de oprire a iteraţiilor care se referă la norma infinit a gradientului: $\|g_k\|_\infty \leq 10^{-6}$. Numărul maxim de iteraţii admis de fiecare algoritm este limitat la 10000. În acest studiu numeric vom compara LBFGS pentru 4 valori ale parametrului m cu TN. Din cele 750 de probleme le reţinem numai acelea pentru care diferenţa dintre valorile optime ale funcţiei obiectiv, determinate de fiecare algoritm în parte, este mai mică decât 10^{-3}, adică dacă f_i^{LBFGS} şi f_i^{TN} sunt valorile optime ale funcţiei f_i, $i = 1, \ldots, 750$, determinate de algoritmul LBFGS şi respectiv TN, atunci în comparaţiile numerice vom reţine numai acele experimente numerice pentru care $\left| f_i^{LBFGS} - f_i^{TN} \right| \leq 10^{-3}$.

Figurile 1.3.1 – 1.3.4 arată profilurile de performanţă Dolan–Moré [2002] ale acestor algoritmi. În figura 1.3.1 vedem profilul de performanţă al lui LBFGS ($m = 3$) versus TN. În tabelul din această figură vedem că din cele 682 de probleme care satisfac criteriul de mai sus, din punctul de vedere al timpului de calcul, algoritmul LBFGS ($m = 3$) a fost mai bu**n** în 470 de probleme.

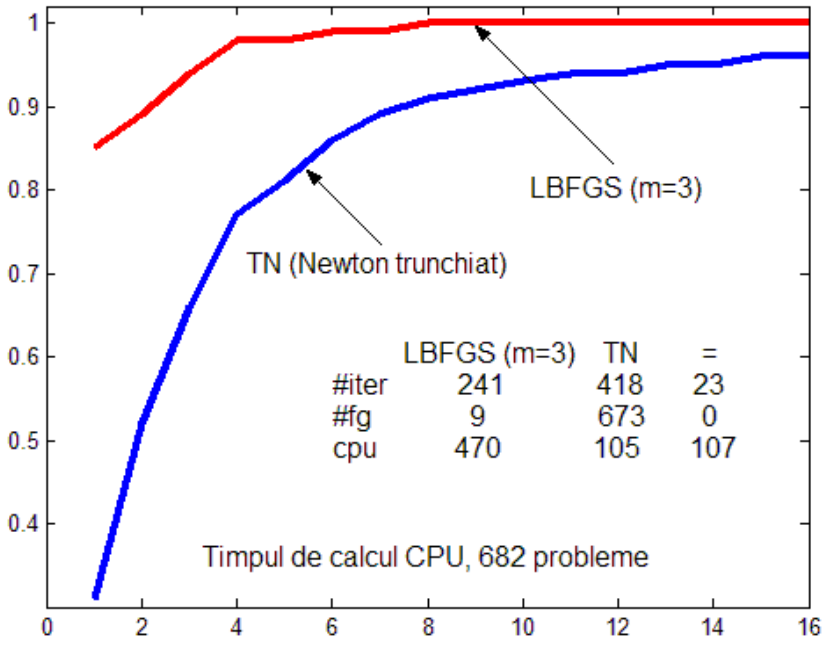

Fig. 1.3.1. Profilul de performanţă L-BFGS ($m = 3$) versus TN (Newton Trunchiat).

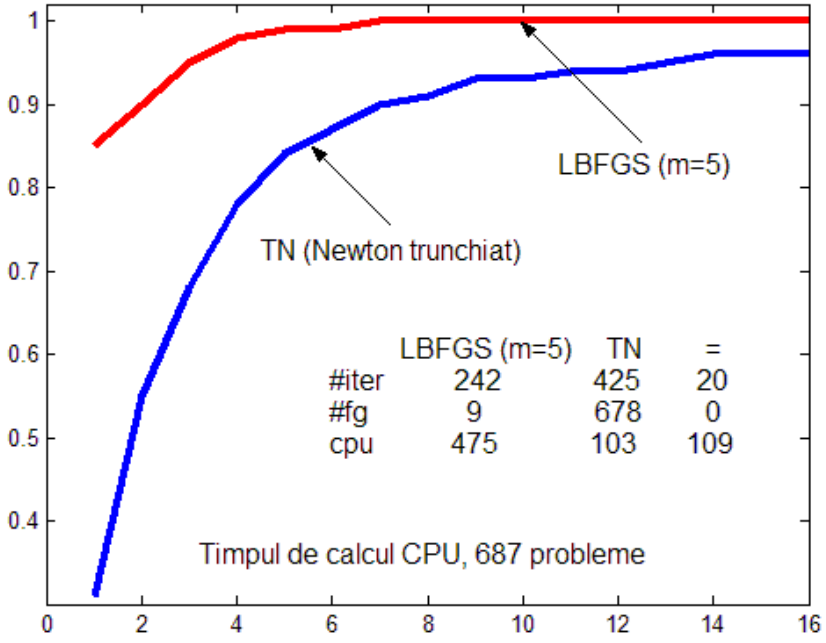

Fig. 1.3.2. Profilul de performanţă L-BFGS ($m = 5$) versus TN (Newton Trunchiat).

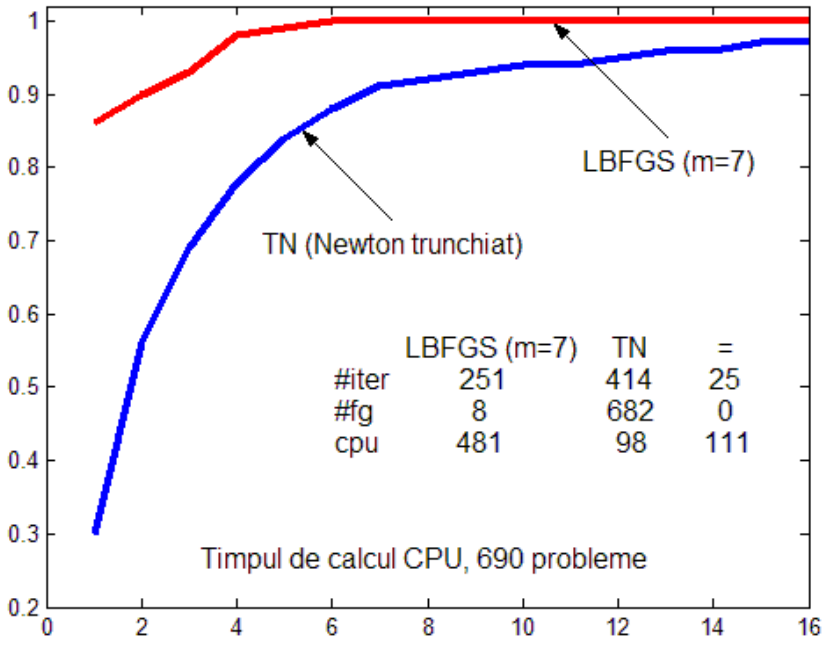

Fig. 1.3.3. Profilul de performanţă L-BFGS ($m = 7$) versus TN (Newton Trunchiat).

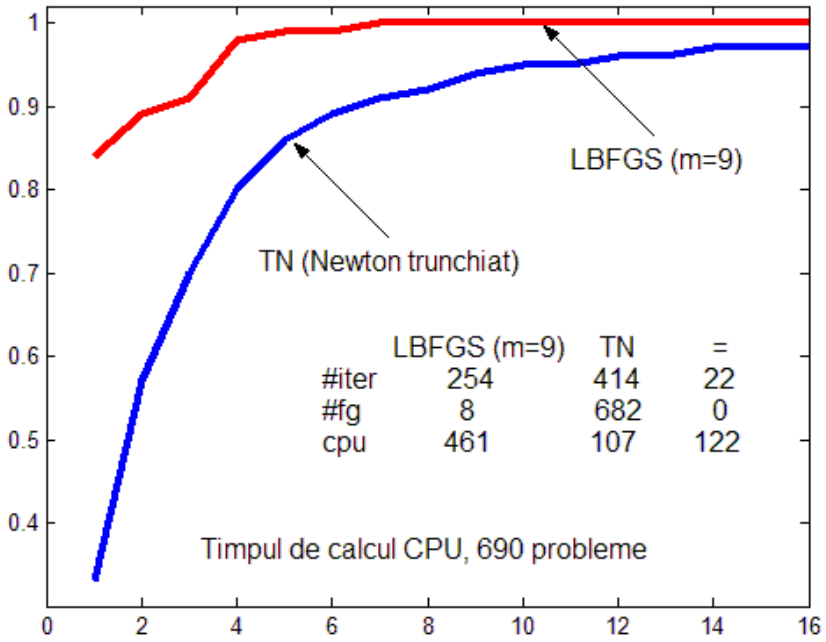

Fig. 1.3.4. Profilul de performanţă L-BFGS ($m = 9$) versus TN (Newton Trunchiat).

Algoritmul TN a fost mai bun în doar 105 probleme, ambii algoritmi au realizat acelaşi timp de calcul în 107 probleme.

Din punctul de vedere al iteraţiilor şi al numărului de evaluări ale funcţiei de optimizat şi a gradientului ei, vedem că metoda TN este mai bună. Din figura 1.3.1 notăm că referitor la numărul de iteraţii algoritmul TN a fost mai bun în 418 probleme, adică pentru 418 probleme din cele 682 algoritmul TN le-a rezolvat într-un număr de iteraţii mai mic decât LBFGS ($m = 3$). Metoda TN cere mai puţine iteraţii, dar timpul de calcul pe iteraţie este mult mai mare.

Partea din stânga din figurile de mai sus arată procentajul din problemele considerate în acest studiu pentru care un algoritm a fost mai bun. Partea din dreapta arată procentajul problemelor care au fost rezolvate cu succes de fiecare algoritm în parte. Curba superioară corespunde al**g**oritmului care a rezolvat cele mai multe probleme într-un timp de calcul care este un multiplu al celui mai bun timp de calcul. Vedem că, cel puţin pentru acest set de probleme de test, în privinţa timpului de calcul, algoritmul LBFGS pentru toate cele 4 valori ale lui m este superior algoritmului TN. Diferenţele între aceşti doi algoritmi sunt semnificative.

Este important de observat faptul că valori mari ale lui m nu conduc la o creştere a performanţei metodei quasi-Newton cu memorie limitată LBFGS. Creşterea numărului de perechi (s_i, y_i) implică un timp de calcul mai mare în implementarea lui (1.2.15). Aceasta arată că metoda LBFGS are un anumit caracter conservativ în sensul că în produsul $-H_{k+1}g_{k+1}$ care defineşte direcţia d_{k+1} nu este nevoie de toată actualizarea inversei matricei Hessian, câteva perechi de vectori (s_i, y_i) sunt suficiente.

În capitolele următoare ale acestei lucrări vom utiliza în mod constant aceşti algoritmi în sensul de a prezenta profilurile de performanţă Dolan-Moré ale algoritmilor de gradient conjugat versus LBFGS şi TN.