Univerzitet u Nišu
Prirodno-Matematički fakultet
Departman za matematiku

Milena J. Petrović

DVOSMERNI I DVOKORAČNI UBRZANI METODI ZA BEZUSLOVNU OPTIMIZACIJU

Doktorska disertacija

Niš, 2014

University of Niš
Faculty of Science and Mathematics
Department of mathematics

Milena J. Petrović

ACCELERATED DOUBLE DIRECTION AND DOUBLE STEP SIZE METHODS FOR UCONSTRAINED OPTIMIZTION

PhD thesis

Niš, 2014

Mentor:

Prof. dr Predrag Stanimirović

redovni profesor Prirodno-Matematičkog fakulteta Univerziteta u Nišu.

Članovi komisije:

1. **Prof. dr Gradimir Milovanović**

   redovni član SANU,

2. **Prof. dr Vladimir Rakočević**

   redovni profesor Prirodno-Matematičkog fakulteta Univerziteta u Nišu,

3. **Prof. dr Dragana Ilić-Cvetković**

   redovni profesor Prirodno-Matematičkog fakulteta Univerziteta u Nišu,

4. **Prof. dr Predrag Rajković**

   redovni profesor Mašinskog fakulteta Univerziteta u Nišu.

Datum odbrane:

Anastasiji i Vasiliju

Прилог 4/1

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:	
Идентификациони број, **ИБР**:	
Тип документације, **ТД**:	**монографска**
Тип записа, **ТЗ**:	**текстуални**
Врста рада, **ВР**:	**докторска дисертација**
Аутор, **АУ**:	**Милена Ј. Петровић**
Ментор, **МН**:	**Предраг Станимировић**
Наслов рада, **НР**:	**ДВОСМЕРНИ И ДВОКОРАЧНИ УБРЗАНИ МЕТОДИ ЗА БЕЗУСЛОВНУ ОПТИМИЗАЦИЈУ**
Језик публикације, **ЈП**:	**Српски**
Језик извода, **ЈИ**:	**Енглески**
Земља публиковања, **ЗП**:	**Србија**
Уже географско подручје, **УГП**:	**Србија**
Година, **ГО**:	**2014**
Издавач, **ИЗ**:	**ауторски репринт**
Место и адреса, **МА**:	**Ниш, Вишеградска 33.**
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)	
Научна област, **НО**:	**Математика**
Научна дисциплина, **НД**:	**Операциона истраживања**
Предметна одредница/Кључне речи, **ПО**:	**Операциона истраживања, нумеричка анализа математичко програмирање, оптимизација**
УДК	**519.6:519.863:519.853 (043.3)**
Чува се, **ЧУ**:	**библиотека**
Важна напомена, **ВН**:	
Извод, **ИЗ**:	Дисертација је настала на основу оригиналних објављених радова и на основу резултата који се у тези први пут појављују. Полазећи од постојећих градијентних метода за безусловну оптимизацију у дисертацији су предложени и описани нови убрзани модели специфичне двосмерне и двокорачне форме за решавање проблема из ове области. Акценат је при томе стављен на убрзана својства ових модела.Урађена анлиза конвергенције и резултати нумеричких експеримената са новим итеративним шемама потврђују теоријски значај нових метода.
Датум прихватања теме, **ДП**:	**08.07.2014.**
Датум одбране, **ДО**:	
Чланови комисије, **КО**:	Председник:
	Члан, ментор:

Образац Q4.09.13 - Издање 1

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	**monograph**
Type of record, **TR**:	**textual**
Contents code, **CC**:	**doctoral dissertation**
Author, **AU**:	**Milena J. Petrović**
Mentor, **MN**:	**Predrag Stanimirović**
Title, **TI**:	**ACCELERATED DOUBLE DIRECTION AND DOUBLE STEP SIZE METHODS FOR UNCONSTRAINED OPTIMIZATION**
Language of text, **LT**:	**Serbian**
Language of abstract, **LA**:	**English**
Country of publication, **CP**:	**Serbia**
Locality of publication, **LP**:	**Serbia**
Publication year, **PY**:	**2014**
Publisher, **PB**:	**author's reprint**
Publication place, **PP**:	**Niš, Višegradska 33.**
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	
Scientific field, **SF**:	**mathematics**
Scientific discipline, **SD**:	**Operational research**
Subject/Key words, **S/KW**:	**Operational research, numerical analysis, mathematical programming, optimization**
UC	**519.6:519.863:519.853 (043.3)**
Holding data, **HD**:	**library**
Note, **N**:	
Abstract, **AB**:	This thesis is based on the originally published papers as well as on the results first introduced herein. Taking into account known gradient methods for non-conditional optimization, new accelerated models with specific double direction and double step size formulations for related problem solving are proposed and described in this paper. The emphasis was put on accelerated parameters of these models. The conducted convergence analysis and numerical experiments' results with new iterative patterns confirm theoretical importance of these new methods.
Accepted by the Scientific Board on, **ASB**:	**08.07.2014.**
Defended on, **DE**:	
Defended Board, **DB**: President:	
Member:	
Member, Mentor:	

Predgovor

ISTRAŽIVANJA predložene doktorske disertacije prevashodno su bazirana na teoriji nelinearne optimizacije i numeričke analize. Teorija optimizacije, kao naučna disciplina primenjene matematike i operacionih istraživanja, predstavlja proces nalaženja ekstremnih vrednosti postavljene funkcije cilja. Shodno svom zadatku, optimizacioni procesi su našli veliku primenu kako u raznim naukama tako i u inženjerstvu, ekonomiji, vojnoj i avio industriji.

Za početak razvoja nelinearnog programiranja uzima se 1950. godina koja se vezuje za nastanak metoda konjugovanih gradijenata i kvazi-Njutnovih (*quasi-Newton*) metoda. Može se slobodno reći da je otada razvoj nelinearne optimizacije u ekspanziji. Danas je ovo vrlo aktuelna tema koja se odlikuje efikasnim modernim metodima kojih je sve više.

Gradijentnim metodima smatramo metode sistematskog traženja i izračunavanja rešenja koji u svojim formulacijama sadrže gradijent objektne funkcije. Usvojena je podela gradijentnih metoda na gradijentne metode prvog reda (koji koriste samo prvi izvod ciljne funkcije) i gradijentne metode drugog reda (koji koriste i prvi i drugi izvod ciljne funkcije). Rezultati ovog istraživanaja su gradijentni metodi prvog reda.

Numerička analiza kao oblast matematike koja se bavi pronalaženjem približnog numeričkog rešenja određenog problema nalazi primenu u raznim naučnim i inženjerskim disciplinama. Osnove numeričke analize doprinele su razvoju mnogih optimizacionih procesa i dovele do interesantnih numeričkih rezultata i usavršavanja postojećih modela. Prilikom konstruisanja novih optimizacionih iterativnih metoda neophodno je dobro oceniti konvergenciju postavljenih metoda. Pri tome je primena teorije numeričke analize veoma značajna i u izradi ove disertacije je od ključne važnosti.

Moja zainteresovanost za ovu granu matematike započela je od osnovnih studija na Prirodno-matematičkom fakultetu u Beogradu, na smeru numerička matematika i optimizacija. Kasnije, na katedri za numeričku analizu na Univerzitetu u Lundu gde sam završila master studije, bila sam u prilici da proširim znanje iz ove naučne oblasti. Proučavajući razne pristupe rekonstrukcija funkcija i njihovih primena na metode konačnih zapremina, imala sam priliku da sarađujem sa izuzetnim stručnjacima iz ove oblasti od kojih izdvajam svog mentora i tadašnjeg šefa katedre profesora Akima Šrola (*Achim Schroll*) i profesora Klausa Firera (*Clauss Führer*). Neki rezultati ovih istraživanja sa širokom praktičnom primenom opisani su u master radu pod nazivom 'Šema konačnih zapremina trećeg reda tačnosti na četvorougaonoj mreži', (*A truly third order finite volume scheme on the quadrilateral mesh*).

O proširivanju teorijskog znanja, kao i o njegovoj primeni koja se ogleda kroz razvoj novih savremenih metoda bezuslovne nelinearne optimizacije, svedoče brojni naučni radovi. Svakako najznačajniji za izradu ove disertacije jesu *Accelerated gradient descent methods with line search* od P.S. Stanimirovića i M.B. Miladinovića [1], *A multi-step curve search algorithm in nonlinear optimization: Nondifferentiable convex case* od N.I. Djuranović-Miličić i M. Gardaševic-Filipović [2] kao i mnogi drugi [3, 4, 5].

Disertacija je nastala na osnovu originalnih objavljenih radova [6, 7] i na osnovu rezultata koji se u tezi prvi put pojavljuju, [8]. Polazeći od postojećih opšte prihvaćenih gradijentnih metoda za bezuslovnu optimizaciju u disertaciji su predloženi i opisani novi ubrzani modeli specifične forme za rešavanje problema iz ove oblasti. Predložena su dva pristupa u definisanju novih gradijentnih metoda i to sa dva smera traženja, takozvani dvosmerni metodi, i sa druge strane metoda sa dva iterativna koraka, takozvanih dvokoračnih metoda. Akcenat je pri tome stavljen na ubrzana svojstva ovih modela.

Dvosmerni metod je predstavljen u radu *Accelerated double direction method for solving unconstrained optimization problems* [6]. Pored novog načina definisanja gradijentne iteracije za bezuslovnu optimizaciju, rezultati prikazani u [6] potvrđuju značajno smanjenje u broju potrebnih iteracija u odnosu na neke novije gradijentne modele. Bitan doprinos ovog istraživanja je svakako i potvrda ubrzanih karakterisika predloženog algoritma, koja je zasnovana na konstrukciji neubrzavajuće verzije originalnog metoda i njihovih poređenja.

U radu [7], *An Accelerated double step size model in unconstrained optimization* predložen je dvokoračni ubrzani gradijentni metod bezuslovne optimizacije. Koristeći ubrzana svojstva potvrđena u [6], definisana gradijentna iteracija sa dve veličine iterativnog koraka, shodno izvedenim numeričkim testiranjima, još je efikasnija u odnosu na dvosmerni model. Tačnije dvokoračni algoritam prevazilazi nedostatke dvosmernog posebno po pitanju smanjenja broja određivanja vrednosti objektne funkcije i CPU vremena izvršenja numeričkih testiranja.

Teorijski značaj ostvaren posebno dvokoračnom iteracijom, dala su ideju za razmatranje mogućnosti primene dvokoračnog modela za transformisanje na jednokoračni postavljanjem određene relacije koja povezuje iterativne korake. Ovaj pristup je opisan u radu *Transformation of the accelerated double step size method* [8]. Dobijeni numerički rezultati pokazuju daleko veću efikasnost jednokoračnog gradijentnog ubrzanog metoda nastalog redukcijom dvokoračnog modela u odnosu na klasičan jednokoračni gradijentni ubrzani metod predložen u [1].

Urađena analiza konvergencije predstavljenih dvosmernih i dvokoračnih metoda potvrđuje njihovu dobru definisanost. Numerički eksperimenti uradjeni sa novim iterativnim šemama su takodje prikazani u disertaciji. Da su novodefinisani dvosmerni i dvokoračni modeli od teorijskog zanačaja potvrđuju ostvareni numerički rezultati koji i daju prostor za dalja proučavanja ovih iteracija. U zaključnim razmatranjima predloženo je definisanje klase ubrzanih dvosmernih i/ili dvokoračnih algoritama.

U izradi predstavljenih radova kao i same disertacije u mnogome je pomoglo iskusno mentorstvo redovnog profesora Predraga Stanimirovića. Dragoceni saveti i predlozi akademika Gradimira Milovanovića i redovnih profesora Vladimira Rakočevića, Dragane Cvetković-Ilić i Predraga Rajkovića takođe su od neprocenjivog značaja unapredili predložena istraživanja. Kolege sa Prirodno-matematičkog fakulteta Univerziteta u Prištini, sa privremenim sedištem u Kosovskoj Mitrovici, redovni profesor Dojčin Petković, docenti Stefan Panić i Negovan Stamenković i asistenti Nataša Kontrec i Časlav Stefanović svojom moralnom, stručnom i tehničkom podrškom doprineli su izradi ove disertacije. Ovim putem se svima srdačno zahvaljujem na nesebičnoj pomoći.

u Nišu, decembra 2014. *Autor*

Sadržaj

Glava 1

Uvodna razmatranja

Optimizacija kao proces nalaženja optimalnog rešenja uizvesnom smislu postoji već dugi niz godina u raznim naukama. Za njeno osamostaljivanje kao posebne naučne grane smatra se kraj četvrte decenije prošlog veka sa nastankom Dantzigovog simpleks metoda prilagođenog rešavanju zadataka linearnog programiranja. Potom sledi razvoj nelinearnog programiranja sa nastankom kvazi-Njutnovih metoda i metoda konjugovanih gradijenata, a potom i sve intenzivniji razvoj raznih savremenih optimizacionih metoda za rešavanje širokog spektara manje ili više složenih problema u mnogim naučnim oblastima.

Nekoliko poslednjih decenija, posebna pažnja se poklanja razvoju naučne oblasti nelinearnog programiranja i u njenim okvirima poboljšavanju postojećih i pronalaženju novih metoda [9, 10, 11, 12, 13, 14, 15]. Fokus istraživanja ove disertacije jesu oni metodi nelinearne optimizacije koji u svojim formulacijama sadrže gradijent, tzv. gradijentni metodi. Postoji mnoštvo naučnih radova na ovu temu od kojih su samo neki od bitnijih za naša istraživanja predstavljena u ovoj doktorskoj disertaciji [3, 5, 16, 17, 18, 1]. Opšte je prihvaćeno da gradijentne metode delimo u dve grupe: gradijentne metode prvog reda, one koji koriste samo prvi izvod date funkcije, i na gradijentne metode drugog reda koji koriste i prvi i drugi izvod ciljne funkcije. Takođe je poznata činjenica da se najveća tendencija povećanja, u slučaju traženja maksimuma, ili smanjenja, ukoliko se određuje minimalna vrednost ciljne funkcije, postiže upravo pomeranjem po gradijentnom pravcu. U radu [2], predstavlja se iterativni metod sa dva pravca pretraživanja od kojih generalno, ni jedan ne mora nužno biti gradijent. Ovako definisan dvosmerni metod za nediferencijabilne funkcije bezuslovne optimizacije, otvorio je put za dalje proučavanje i analizu dvosmernih metoda pod određenim uslovima kao i mogućnost primene predstavljene ideje u slučaju kada je jedan od pravaca pretraživanja gradijentni. Ovo i jeste jedna od tema koja će biti razmatrana u ovoj doktorskoj disertaciji. Motivacijski, iz ove tematike proistekla je ideja za proučavanje gradijentnih metoda bezuslovne optimizacije koje definišu dva iterativna koraka, tzv. dvokoračni metodi, što je takođe sastavni istraživački deo ove teze.

U radu [3] autor predlaže ubrzani model kvazi-Njutnovog metoda i prema dobijenim rezultatima unapređuje dotadašnje gradijentne metode bezuslovne optimizacije. Predrag Stanimirović i Marko Miladinović dali su novi doprinos radom [1] po pitanju brzine konvergencije, broja iterativnih koraka i broja evaluacija, odnosno broja određivanja vrednosti posmatrane funkcije. Oni takođe u istom radu daju i predlažu jednostavan i elegatan način izračunavanja ubrzavajućeg faktora, i grupišu metode sličnih svojstava identifikujući novu klasu ubrzanih gradijentnih metoda za rešavanje problema bezuslovne optimizacije. U ovoj disertaciji su primenjene predložene ideje, a na osnovu njih su na sličan način ubrzani konstruisani dvosmerni i dvokoračni metodi bezuslovne nelinearne optimizacije.

1.1 Proces optimizacije

Veoma aktuelan i čest zadatak kako u mnogobrojnim naučnim oblastima tako i u savremenim industrijskim granama je nalaženje optimalnog rešenja zadatog problema. Matematički formulisano, ovaj postupak predstavlja određivanje minimuma ili maksimuma date ciljne funkcije pod određenim uslovima. Drugim rečima, optimizacija kao široko primenjena naučna disciplina matematike i operacionih istraživanja predstavlja proces nalaženja uslova za dostizanje ekstremnih vrednosti postavljene funkcije cilja.

U okviru svog zadatka optimizacioni proces koristi metode raznih naučnih disciplina od kojih se mogu izdvojiti numerička analiza, matematičko programiranje, varijacioni račun i optimalno upravljanje. Svim optimizacionim procesima opšta karakteristika je nalaženje najprihvatljivijeg rešenja pri datim uslovima. Ipak, u zavisnosti od prirode funkcije cilja i zadatih ograničenja, mogu se izdvojiti dve suštinske grupe optimizacionih procesa: linearni i nelinearani. Kako sam naziv sugeriše, kod linearne optimizacije figurišu linearana funkcija cilja i linearna ograničenja, dok je opšte prihvaćeno da se nelinearnom optimizacijom smatra optimizacioni proces u kome je makar jedna od datih funkcija, bilo da je funkcija cilja ili neka od funkcija iz skupa ograničena, nelinearna. Predmet ove disertacije jesu nelinearni optimizacioni problemi.

1.1.1 Optimizacioni zadatak i faze procesa optimizacije

U cilju ostvarivanja što boljeg rešenja određenog problema pod datim uslovima, važno je sagledati sve relevantne pretpostavke koje definišu jedan optimizacioni zadatak. Osnovno polazište je *objekat optimizacije* pod kojim se podrazumeva posmatrani proces, delatnost, vreme, resursi, dobit i sl. Pod *kriterijumom optimalnosti* podrazumevamo definisanu funkciju cilja, odnosno analiziranu efikasnost. Pri tom se najbolja vrednost kriterijuma optimalnosti naziva optimalna vrednost ili ekstremum. Pravilno i korektno definisana funkcija cilja je preduslov za ispravnu realizaciju optimizacije. Nužna pretpostavka u optimizacionom procesu je da objekat koji se optimizuje bude *upravljiv*, što praktično znači postojanje izvesnog stepena slobode. Da bi se definisao skup različitih stanja objekta optimizicije, iz kojih se izdvaja optimalno stanje, potrebno je da se parametri objekta optimizacije menjaju nezavisno jedni od drugih.
Konačno, shodno definisanom upravljivom objektu optimizacije sa postavljenim kriterijumom optimalnosti, optimalno rešenje tražimo putem dobro odabranog *metoda optimizacije*. Kako ne postoji univerzalni optimizacioni metod, za dati problem on se bira u skladu sa prirodom objekta optimizacije i sa postavljenom ciljnom funkcijom.

Suštinski, optimizacioni proces se bavi rešavanjem određenog matematičkog modela a to rešavanje se ogleda u određivanju minimuma ili maksimuma ciljne funkcije pri zadatim ograničenjima. Sledi kratak opis uopštenog matematičkog modela optimizacionog procesa.

Kao što je već rečeno, za objekat koji se optimizuje podrazumevamo da je upravljiv i da je definisan ulaznim, tj. takozvanim upravljačkim $x = (x_1, x_2, \ldots, x_n)$ i izlaznim $y = (y_1, y_2, \ldots, y_n)$ parametrima kao i matematičkim modelom koji ih povezuje:

$$y_j = f_j(x), \quad j = 1, \ldots, m \tag{1.1}$$

Na osnovu prethodne jednakosti, može se zaključiti da kriterijum optimalnosti, odnosno ciljna funkcija označena kao $Q = Q(x,y)$, koja jeste funkcija svih ulaznih i izlaznih parametara, zavisi samo od ulaznih, tj. upravljačkih veličina, odnosno važi:

$$Q = Q(x,y) = f_j(x) = Q(x) = Q(x_1, x_2, \ldots, x_n), \quad j = 1, \ldots, m \tag{1.2}$$

Podrazumeva se da je skup upravljačkih parametara ograničen, odnosno da postoji domen D_x ovih veličina tako da je svaka upravljačka promenjliva unutar tog skupa, tj. važi $x \in D_x$. U nekim slučajevima upravljački parametri zadovoljavaju takozvana funkcijska ograničenja tipa jednakosti ili tipa nejednakosti. U tom slučaju važe neke od sledećih relacija:

$$\mu_l(x_1,x_2,\ldots,x_n) = \mu_{0l}, \quad l = 1,\ldots,m_1 < n$$

$$\nu_k(x_1,x_2,\ldots,x_n) = \nu_{0k}, \quad k = 1,\ldots,m_2$$

gde su μ_l i ν_k date funkcije.

Generalno, zadatak matematičkog programiranja ima sledeću formulaciju:

$$\begin{aligned} &\min \setminus \max f(x) \\ &g_i(x) \geq 0, \quad i = 1,\ldots,m \\ &h_j(x) = 0, \quad j = 1,\ldots,p \end{aligned}$$

Ovde je $f(x)$ ciljna funkcija dok su g_i i h_j funkcije ograničenja. Kao što je već naglašeno, ukoliko su sve navedene funkcije (f, g_i i h_j) linearne, radi se o problemu linearnog programiranja. U protivnom, ako je makar jedna od datih funkcija nelinearna govorimo o zadatku nelinearnog programiranja. Ako su u formulaciji zadatka izostavljene funkcije ograničenja (g_i i h_j) radi se o bezuslovnoj optimizaciji. Upravo će problemi bezuslovne optimizacije biti predmet proučavanja ovog rada. Jednostavnom jednakošću:

$$\min f(x) = -\max f(-x)$$

se proces nalaženja maksimuma prevodi u postupak traženja minimuma ciljne funkcije. Obrnuto se postiže analognim izrazom: $\max f(x) = -\min f(-x)$. Shodno pomenutoj analizi, u daljem izlaganju posmatrani optimizacioni problemi biće prikazani kroz zadatak traženja minimuma.

Optimizaciju kao proces definiše više svojstvenih faza. Neke značajnije su:

1. *Definisanje zadatka optimizacije*, predstavlja prvu i jednu od najvažnijih etapa optimizacije od koje zavise sve ostale faze optimizacionog procesa. Postavljanje zadatka optimizacije, koje je već izloženo u opštem smislu, je početna faza bilo kog optimizacionog procesa u kojoj se definiše sam željeni cilj optimizacije i određuju nezavisni parametri u datom procesu kao i eventualna ograničenja. Optimizacione zadatke delimo na više vrsta prema različitim kriterijumima. Tako na primer, postoje zadaci statičke optimizacije u kojima se ciljna funkcija optimizuje u datom nepromenljivom stanju. Zadaci statičke optimizacije mogu biti formulisani kao linerani ili nelinearni optimizacioni problemi. Nasuprot statičkim postoje zadaci dinamičke optimizacije gde postavljena ciljna funkcija zavisi od parametara koje sadrži i koji se u opštem slučaju me-

njaju u prostoru i u vremenu. Prema broju ulaznih (upravljačkih) veličina, optimizacione zadatke delimo na zadatke jednodimenzionalne optimizacije $n = 1$ ili na zadatke višedimenzionalne optimizacije $(n > 1)$. Kod zadataka višedimenzionalne optimizacije, ukoliko je broj ulaznih parametara $4 \leq n \leq 5$ radi se o zadacima manjih dimenzija. U slučaju $5 < n < 20$ u pitanju su zadaci srednjih dimenzija, dok za $n \geq 20$ govorimo o optimizacionim zadacima velikih dimenzija. Zadaci optimizacije se takođe razlikuju i po tome da li je zadata ili nije početna (startna) tačka.

2. *Izbor kriterijuma optimizacije*. U ovoj etapi se na osnovu postavljenog zadatka formuliše funkcija cilja. Kriterijumi optimizacije su ili potpuno definisani ili, u slučaju složenijih procesa, mogu proizvoditi više ishoda. Kod složenijih modela je nemoguće postaviti opštu formulaciju funkcije cilja datu u vidu univerzalnog kriterijuma. Razlog tome je konstantno uvećanje broja parametara, komplikovanje računskih aparata što dovodi do stalnog preispitivanja adekvatnosti i pouzdanosti univerzalnog matematičkog modela. Zbog toga generalno kod složenijih optimizacionih modela funkcija cilja uglavnom nije matematički postavljena već je neformalno opisana nizom aspekata koji rezultiraju kako iz matematičkog modeliranja tako iz sprovedenih eksperimentalnih testiranja. Kada je izostavljena matematička formulacija kriterijuma optimalnosti radi se o takozvanim relativnim kriterijumima optimalnosti. Kriterijumi se prema načinu vrednovanja dele na determinističke kriterijume, kriterijume statističke verovatnoće i kriterijume za uslove konfliktnih situacija. Uopšte, u većini optimizacionih procesa kvalitet rešenja se ocenjuje na osnovu većeg broja kriterijuma što doprinosi boljoj objektivnosti celokupnog optimizacionog procesa i ostvarenoj optimalnoj vrednosti.

3. *Postavljanje matematičkog modela optimizacije*. Složenost optimizacionog procesa determiniše složenost odgovarajućeg matematičkog modela. Sam matematički model optimizacionog procesa u nekim situacijama nije čisto matematički već u zavisnosti od prirode problema može sadržati eksperimentalne rezultate, neke intuitivne pretpostavke kao i mnoge druge faktore uticaja.

4. *Izbor metoda optimizacije* zavisi od ostalih optimizacionih faza, a pre svega od prve etape, od definisanog zadatka tj. problema optimizacije i njegove prirode koja može biti deterministička , stohastička, dinamička itd. Takođe, metod kojim će se rešavati zadati optimizacioni problem zavisi od matematičke formulacije datog zadatka koji može biti linearan ili nelinearan, definisan sa određenim ograničenjima ili bez njih, u čijoj se formulaciji mogu javljati izvodi ili ne i sl. U zavisnosti od broja postavljenih kriterijuma

razlikujemo jednokriterijumsku i višekriterijumsku optimizaciju pa se i optimizacioni metodi razlikuju i prema ovoj karakteristici. U vezi datog optimizacionog kriterijuma, definisana funkcija cilja može biti data analitički sa jasno definisanim matematičkim izrazom za ciljnu funkciju ili eksperimentalno gde ne postoji matematička formulacija funkcije cilja. Svi ovi aspekti svakako nužno determinišu izbor metoda optimizacije. Sa izborom algoritma (softverskog rešenja) datog problema, okončava se i faza izbora metoda optimizacije posmatranog procesa. Kako je jedan od osnovnih fokusa ove disertacije razmatranje i analiza postojećih optimizacionih metoda kao i predstavljanje novih, poboljšanih u izvesnom smislu, u narednom poglavlju biće izložen detaljniji opis i podela optimizacionih metoda.

5. *Formiranje algoritma i programa.* Na osnovu postavljenog algoritma rešava se dati problem softverski tj. pristupa se izradi odgovarajućeg programa a zatim sledi numeričko testiranje programiranog modela.

6. *Programska realizacija.* U ovoj etapi procesa optimizacije vrši se računarska realizacija optimizacione procedure. Ovo je ujedno i završna faza u optimizaciji koja se uglavnom sprovodi na računarima sa bržim hardverskim uređajima kojima se ostvaruje traženi kapacitet obrade i obezbeđuje očekivana numerička tačnost.

7. *Interpretacija i analiza dobijenog rešenja.* Dobijeni rezultati mogu biti ilustrovani na više načina kao što su tabelarni, grafički prikaz i sl. Predstavljeni izlazni podaci se analiziraju, upoređuju na osnovu čega se donose preliminarni zaključci o datom procesu.

8. *Korekcije postavljenog modela i algoritma.* Za očekivati je da postavljeni algoritam posmatranog optimizacionog zadatka relativno brzo konvergira, da bude što univerzalnijeg karaktera, da ispunjava sva zadata ograničenja ako takva postoje i da zauzima što manje memorijskog prostora. Ukoliko neke od ovih ili sličnih pretpostavki nisu zadovoljene u očekivanoj meri pristupa se korigovanju postavljnog algoritma sa ciljem poboljšanja potrebnih karakteristika.

9. *Tesiranje i primena optimalnog rešenja.* Svrha svakog optimizacionog zadatka jeste njegova praktična primena. Zato je testiranje i primena optimalnog rešenja kao poslednja etapa optimizacionog procesa veoma značajna jer se upravo kroz ovu fazu praktične primene proverava primenljivost i efikasnost urađenog optimizacionog modela.

1.1.2 Klasifikacija metoda optimizacije

Postoji više podela metoda optimizacije koje zavise od nekih ključnih karakteristika samog procesa optimizacije. U ovom poglavlju su navedene neke od važnijih podela.

1. Prema broju i prirodi zadatih ciljnih funkcija optimizacioni metodi se dele na:

    - metode jednokriterijumske optimizacije koji se koriste za optimizaciju jedne jedinstvene ciljne funkcije;
    - metode višekriterijumske optimizacije pomoću kojih se optimizuje više ciljnih funkcija;
    - metode za diferencijabilne i nediferencijabilne ciljne funkcije. U slučaju kada je ciljna funkcija diferencijabilna koriste se takozvani gradijentni metodi. Ovakvi metodi koriste izvod ciljne funkcije. Gradijentni metodi se, kao što je već napomenuto, dele na gradijentne metode prvog reda (koriste samo prvi izvod ciljne funkcije), i na gradijentne metode drugog reda (koriste i prvi i drugi izvod ciljne funkcije). U istraživanjima opisanim u ovoj disertaciji obrađeni su prevashodno gradijentni metodi prvog reda. Za optimizaciju nediferencijabilnih funkcija cilja koriste se takozvani negradijentni metodi koji ne koriste izvode zadate funkcije.
    - metode kod kojih je zadata ili nije zadata tačnost lokalizacije ekstremuma;
    - metode za jednoekstremalnu ili višeekstremalnu funkciju cilja;
    - metode za analitički ili eksperimentalno zadatu funkciju cilja;

2. Prema eventualnim postavljenim uslovima (ograničenjma) metode optimizacije delimo na:

    - metode bezuslovne optimizacije tj. optimizacije bez ograniěnja;
    - metode uslovne optimizacije odnosno optimizacije sa ograničenjima koja su zadata linearnim jednačinama i (ili) nejednačinama;
    - metode optimizacije sa funkcionalnim ograničenjima;

3. Opšta podela optimizacionih metoda:

    - *Analitički metodi*. Primenjuju se ukoliko je moguće naći izvod ciljne funkcije a potom ispitati osobine izvoda putem matematičke analize. Dakle, primenom ovih metoda određivanje ekstremnih vrednosti zadate funkcije cilja f sastoji se u pronalaženju takvih vektora x za koje je ispunjena jednakost $f^{'}(x) = 0$. Jasno je da u slučaju složenijih nelinearnih problema ovakvi metodi nisu primenljivi.

- *Grafički metodi.* Ovi metodi služe za grafičko određivanje ekstremuma ciljne funkcije uz pomoć grafičkog predstavljanja funkcije cilja i zadatih ograničenja. Iako se ovim putem dobija jasan i pregledan prikaz datog problema, grafički metod je moguće primeniti jedino u slučaju kada ciljna funkcija zavisi od jedne ili dve ulazne veličine.
- *Eksperimentalni metodi.* Baziraju se na izvršenim eksperimentima vezanim za dati problem. Na osnovu dobijenih eksperimentalnih rezultata pretpostavlja se (prognozira) ekstermum ciljne funkcije. Eksperimentalni metodi ne koriste matematički model i ovakvim metodima se pribegava upravo onda kada nije moguće postaviti pogodan matematički model za posmatrani optimizacioni proces .
- *Numerički (iterativni) metodi.* Kao što sam naziv sugeriše suština ovakvih metoda je u korektnom postavljanju numeričkih iteracija kojim se ostvaruje dovoljno dobro aproksimativno rešenje analiziranog problema. Jedna od prednosti ovakvih metoda je u mogućnosti relativno jednostavnog programiranja postavljenog algoritma metoda i kasnije testiranju istog. Metode ove vrste delimo na gradijentne iterativne metode i na negradijentne iterativne metode. Kako je osnovni predmet proučavanja ovog rada baziran na analizi postojećih i pronalaženju novih efikasnih numeričkih iterativnih metoda nelinearne bezuslovne optimizacije, u nastavku su navedeni neki prihvaćeni kriterijumi za prekid iterativnog metoda optimizacije. Kako ne postoji univerzalni kriterijum za zaustavljanje numeričkog optimizacionog metoda u daljem će biti predstavljeni najčešće korišćeni.

  Jedan od njih je dat sledećom nejednakošću:

$$\frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)}\|} \leq \delta \tag{1.3}$$

  Ovaj kriterijum u osnovi sadrži normu razlike dve uzastopne vrednosti upravljačkih parametara dobijenih iz dve uzastopne iteracije. Loša strana ovog kriterijuma je moguće prevremeno zaustavljanje iterativnog postupka u slučaju kada je funkcija cilja jako osetljiva na ekstremum. Tada se događa da iako vrednost $\|x^{(k+1)} - x^{(k)}\|$ postane vrlo mala, razlika između vrednosti ciljne funkcije u tim tačkama bude znatna.

  Sledeći kriterijum koristi upravo vrednosti ciljne funkcije u dve uzastopne iteracije.

Nepovoljnost ovog kriterijuma ogleda se u tome što se može dogoditi da se prilikom prekida algoritma, ispunjenjem prethodne nejednakosti, dobijena vrednost dosta razlikuje od optimalne u slučaju kada je ciljna funkcija slabo osetljiva na ekstremum. Ovo se dešava zato što je u tom slučaju veličina $\|x^{(k+1)} - x^{(k)}\|$ daleko veća od veličine $f(x^{(k+1)}) - f(x^{(k)})$.

Postoji takođe i kriterijum za ocenu tačnosti lokalizacije ekstremuma. Kod ovog kriterijuma upravljački parametri se menjaju pomoću unapred zadatih minimalnih koraka $h_{min_i}, i = 1, ..., n$. Uslov za prekid iteracije je

$$h_i \leq h_{min_i}, \quad i = 1, ..., n \tag{1.4}$$

Nažalost, i kod ovog kriterijuma se dešava da rešenje bude neprihvatljivo ukoliko je funkcija cilja izrazito osetljiva na ekstremum.

Da bi se izbegle predočene 'nus-pojave' prethodnih kriterijuma često je u upotrebi takozvani dvostruki kriterijum koji je definisan sledećim nejednakostima:

$$\begin{array}{ll} 1. & h_i \leq h_{min_i}, \quad i = 1, ..., n \\ 2. & 1 - \frac{f^{(1)} - f^{(2)}}{f^*} \leq \varepsilon \end{array}$$

Ovde su $f^{(1)}$ i $f^{(2)}$ dve najbliže vrednosti ciljne funkcije vrednosti f^* koja predstavlja njenu optimalnu vrednost. Numeričke iteracije traju do ispunjenja oba kriterijuma. Prvo se proverava prvi kriterijum i sa njegovim ispunjenjem se započinje sa proverom drugog kriterijuma. Kada se i drugi kriterijum ispuni iterativni postupak se završava. Kao što se iz izraza za drugi kriterijum može uočiti, ovaj dvostruki kriterijum je nepovoljan u slučaju kada je $f^* \approx 0$. Tada se koristi druga verzija dvostrukog kriterijuma koja je definisana na sledeći način:

$$\begin{array}{ll} 1. & h_i \leq h_{min_i}, \quad i = 1, ..., n \\ 2. & |\frac{\Delta_1 - \Delta_2}{2}| \leq \varepsilon \end{array}$$

gde su $\Delta_1 = f^* - f^{(1)}$ i $\Delta_2 = f^* - f^{(2)}$.

1.1.3 Bezuslovna optimizacija

Zadatak bezuslovne optimizacije, kao što sama reč sugeriše, jeste zadatak koji ne sadrži dodatna ograničenja. On se jednostavno formuliše na sledeći način: za datu ciljnu funkciju f naći

$$\min f(x), \quad x \in \mathbb{R}^n \tag{1.5}$$

Poznato je da tačka x^* koja zadovoljava prethodni izraz ispunjava optimalne uslove prvog i drugog reda koji su formulisani sledećim definicijama:

Definicija 1.1. *Tačka x^* naziva se tačkom lokalnog minimuma ako postoji okolina te tačke $U(x^*)$, takva da je za svaki vektor $x \in U(x^*)$ ispunjena sledeća nejednakost*

$$f(x^*) \leq f(x). \tag{1.6}$$

Kaže se da je tačka x^* tačka striktnog lokalnog minimuma ako postoji okolina te tačke $U(x^*)$ takva da je za svaki vektor $x \neq x^*$ i $x \in U(x^*)$ važi

$$f(x^*) < f(x). \tag{1.7}$$

Definicija 1.2. *Tačka x^* jeste tačka globalnog minimuma ako važi nejednakost:*

$$f(x^*) \leq f(x) \quad \forall x \in \mathbb{R}^n. \tag{1.8}$$

Tada postoji okolina te tačke $U(x^*)$ takva da je za svaki vektor $x \neq x^*$ i $x \in U(x^*)$ važi

$$f(x^*) < f(x) \quad \forall x \in \mathbb{R}^n. \tag{1.9}$$

Svakako je lakše i u praksi dosta češće, nalaženje tačke lokalnog minimuma. Ta tačka, u opštem slučaju, ne mora istovremeno biti i globalni minimum. Iz tog razloga postupak traženja globalnog minimuma se može svesti na određivanje više lokalnih minimuma od kojih će se odabrati najmanji i proglasiti za globalni minimum.

U cilju efikasnog traženja lokalnog minimuma potrebno je pretraživanje vršiti po pravcu opadanja koji se definiše na sledeći način:

Definicija 1.3. *Vektor $d \in \mathbb{R}^n$ nazivamo vektorom opadajućeg pravca u tački $x \in \mathbb{R}^n$ funkcije*

$f :\to \mathbb{R}$ koja je diferencijabilna u toj tački, ukoliko važi

$$\langle \nabla f(x), d \rangle < 0. \tag{1.10}$$

Kako iz Tejlorovog razvoja imamo da važi

$$f(x_k + td) = f(x_k) + t\nabla f(x_k)^T d + o(t) \tag{1.11}$$

sledi da postoji $\delta > 0$ takvo da je uslov $f(x_k + td) < f(x_k)$ ispunjen za svako $t \in (0, \delta)$ ako i samo ako je d vektor opadajućeg pravca funkcije f u tački x.

Potrebni uslovi prvog reda, tj. činjenica da je gradijent funkcije jednak nuli u tački lokalnog minimuma, formulisani su narednom teoremom:

Teorema 1.1. *Neka je D otvoren skup i $f : D \subset \mathbb{R}^n \to \mathbb{R}$ neprekidno diferencijabilna funkcija. Ukoliko je x^* tačka lokalnog minimuma zadatka* (1.5) *onda važi*

$$\nabla f(x^*) = 0 \tag{1.12}$$

U narednoj teoremi, koja daje potrebne uslove drugog reda, pretpostavka da je data funkcija f dva puta neprekidno diferencijabilna i x^* tačka lokalnog minimuma takođe potvrđuje da funkcija f ima nula nagib u toj tački ali i nenegativno zakrivljenje.

Teorema 1.2. *Ukoliko je $f : D \subset \mathbb{R}^n \to \mathbb{R}$ dva puta neprekidno diferencijabilna na otvorenom skupu D i x^* lokalni minimum zadatka* (1.5) *tada važi*

$$\nabla f(x^*) = 0 \quad i \quad \nabla^2 f(x^*) \quad je\, pozitivno\, semidefinitna. \tag{1.13}$$

Sledeća teorema opisuje dovoljne uslove drugog reda.

Teorema 1.3. *Neka je D otvoren skup i $f : D \subset \mathbb{R}^n \to \mathbb{R}$ dva puta neprekidno diferencijabilna funkcija i neka postoji $x^* \in D$ takva da je*

$$\nabla f(x^*) = 0 \quad i \quad \nabla^2 f(x^*) \quad jeste\, pozitivno\, definitna, \tag{1.14}$$

onda je tačka x^* striktni lokalni minimum funkcije f.

Pojam stacionarne tačke je opisan sledećom definicijom:

Definicija 1.4. *Tačka $x^* \in \mathbb{R}^n$ je stacionarna tačka diferencijabilne funkcije f ako je ispunjen uslov*

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

Iz teoreme (1.1) jednostavno sledi zaključak da ukoliko je x^* tačka lokalnog ekstremuma onda je ona istovremeno i stacionarna tačka dok obrnuto ne mora nužno da važi. Ukoliko stacionarna tačka nije i tačka lokalnog ekstremuma onda je nazivamo tačkom nagomilavanja.

Kako se dobar deo proučavanja koja su bila potrebna za izradu ove disertacije odnosi na konveksne i diferencijabine konveksne funkcije, za završetak ovog poglavlja, izdvajamo ove skupove funkcija kao i uslove za postojanje globalnog ekstremuma nad tim skupovima. Kod diferencijabilnih konveksnih funkcija su sve stacionarne tačke ujedno i tačke lokalnih ekstremuma. Još preciznije, za konveksne funkcije važi da je lokalni minimum istovremeno i globalni minimum a kod diferencijabilnih konveksnih funkcija svaka stacionarna tačka je ujedno i tačka globalnog minimuma. Ova tvrđenja opisuju sledeće dve teoreme.

Teorema 1.4. *Neka $f : D \subset \mathbb{R}^n \to \mathbb{R}$ gde je D neprazan konveksan skup i neka je $x^* \in D$ tačka lokalnog minimuma funkcije f.*

- *Ako je f konveksna funkcija onda je x^* globalni minimum funkcije f;*
- *Ako je f striktno konveksna funkcija onda je x^* jedinstveni globalni minimum funkcije f.*

Teorema 1.5. *Neka $f : \mathbb{R}^n \to \mathbb{R}$ diferencijabilna konveksna funkcija. Tada je $x^* \in \mathbb{R}^n$ tačka globalnog minimuma funkcije f ako i samo ako je $\nabla f(x^*) = 0$.*

1.2 Osvrt na ubrzane gradijentne metode i njihova svojstva

Za rešavanje problema (1.5) čija je ciljna funkcija diferencijabilna vrlo često se koriste gradijentni metodi prvog reda. Ukoliko je funkcija cilja dva puta neprekidno diferencijabilna onda se ne retko koriste gradijentni metodi drugog reda. Usvojene oznake za gradijent i Hesijan funkcije cilja su:

$$g(x) = \nabla f(x),\ G(x) = \nabla^2 f(x),\ g_k = \nabla f(x_k),\ G_k = \nabla^2 f(x_k).$$

Najčešće posmatrana iterativna šema za rešavanje ovakvih problema je oblika:

$$x_{k+1} = x_k + t_k d_k. \tag{1.15}$$

Ovde je x_{k+1} tekuća iterativna tačka, x_k prethodna vrednost, pozitivna veličina t_k predstavlja dužinu iterativnog koraka dok je d_k vektor traženja. Osnovni zadatak kod ovakvih iterativnih metoda je odrediti zadovoljavajuću dužinu koraka i veličinu vektora pravca traženja. Za vektor d_k se pretpostavlja da zadovoljava uslov opadanja, tj. da važi:

$$g_k^T d_k < 0.$$

Polazeći od Njutnovog metoda kod koga se nalaženje vektora pravca ostvaruje rešavanjem jednačine $G_k^T d = 0$, preko gradijentnog metoda u kome se uzima da je $d_k^T = g_k$, u radovima novijeg datuma predloženo je više načina za određivanje vektora pravca. Iz modela konjugovanih gradijentnih metoda javilo se nekoliko varijanti za računanje vektora d_k pomoću definisanog pravila:

$$d_k = \begin{cases} -g_k, & k = 0; \\ -g_k + \beta_k d_{k-1}, & k \geq 1, \end{cases}$$

β_k je ovde konstanta čija se vrednost u datoj iteraciji računa u Fletcher-Reeves metodu [19]

$$\beta_k^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}},$$

dok se kod Polak-Ribiére-Polyak metoda, [20, 21] određuje na sledeći način

$$\beta_k^{PRP} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}}.$$

U [22, 23] su takođe predložene određene procedure za izbor vektora pravca. U radu [2] predstavljen je novi algoritamski predlog za definisanje vektora pravca i to za nediferencijabilne ciljne funkcije.

Sledeća komponenta neophodna da bi iteracija (1.15) bila dobro definisana je dužina koraka u tekućoj iteraciji. Razvijena je čitava klasa procedura pod nazivom *linijsko traženje* koje predstavljaju efektivne načine za određivanje iterativnog koraka. Ovakvi algoritmi su uglavnom bazirani na uslovu opadanja ciljne funkcije u svakoj iteraciji:

$$f(x_k + t_k d_k) < f(x_k).$$

Generalno, ovakve procedure se dele na procedure tačnog i netačnog linijskog traženja. U ovoj doktorskoj tezi su prevashodno analizirane procedure netačnog linijskog traženja, posebno

Armijeva backtracking koja je zajedno sa mnogim drugim procedurama linijskog traženja opisana u narednoj glavi disertacije.

Barzilai i Borvein u radu [5] opisuju svoj takozvani *BB* algoritam i nešto drugačiji predlog za konstrukciju iterativnog koraka koji ima sledeću prezentaciju:

$$x_{k+1} = x_k - \frac{1}{\gamma_k^{BB}} g_k, \quad \gamma_k^{BB} = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}. \tag{1.16}$$

gde je $s_{k-1} = x_k - x_{k-1}$ i $y_{k-1} = g_k - g_{k-1}$. Više kasnije objavljenih radova predstavljaju, uslovno rečeno, modifikovane verzije *BB* metoda [24, 25, 26, 27, 28, 29, 30, 31, 18, 32].

Može se reći da metodi tipa (1.15) datiraju još od 1847. godine sa nastankom Košijevog metoda pada gradijenta. Za vektor pravca Koši je odabrao negativan gradijent i ovaj metod prilagodio diferencijabilnim funkcijama. Posebno interesantan pristup izneo je Nekulaj Andrej u radu [33] kada je u izrazu (1.15) za vektor pravca izabrao negativan gradijent $-g_k$, a iterativni korak t_k pomnožio parametrom $\theta_k > 0$, za koji se pokazalo da ima ubrzavajuća svojstva. Na taj način pomenuti autor je formulisao iterativni metod oblika

$$x_{k+1} = x_k - \theta_k t_k d_k. \tag{1.17}$$

Svakako, ideja za konstrukciju ovakvog metoda proistekla je iz klasičnog Njutnovog metoda

$$x_{k+1} = x_k - t_k G_k^{-1} g_k$$

i bila motivisana time da se izbegne neželjeno računanje Hesijana. Sa tim u vezi Andrej nudi aproksimaciju inverza Hesijana, matricu θ_k. Koristeći kvazi-Njutnovu jednačinu

$$S_{k+1} y_k = s_k,$$

gde je

$$s_k = x_{k+1} - x_k, \;\; y_k = g_{k+1} - g_k$$

dok S_k predstavlja simetričnu matricu formata $n \times n$ koja aproksimira inverz Hesijana. Primenivši sledeću aproksimaciju

$$S_k = \gamma_k^{-1} I, \quad \gamma_k \in \mathbb{R}, \tag{1.18}$$

uz pomoć koje se modifikovana Njutnova iteracija

$$x_{k+1} = x_k - t_k S_k g_k, \tag{1.19}$$

prevodi u

$$x_{k+1} = x_k - t_k \gamma_k^{-1} g_k, \tag{1.20}$$

Andrej Nekulaj predlaže aproksimaciju Hesijana baziranu na vrednostima funkcije i njenog gradijenta u dve uzastopne iterativne tačke. Dobijeni metod je linearno konvergentan. Izraz za skalar γ_k u Andrejevom metodu koji pomnožen sa jediničnom matricom I aproksimira Hesijan je

$$\gamma_{k+1} = \frac{2}{g_k^T g_k} \frac{1}{t_k^2} \left[f(x_{k+1}) - f(x_k) + t_k g_k^T g_k \right]. \tag{1.21}$$

Jedan od glavnih doprinosa ovog metoda, u daljem skraćeno označen sa *AGD* $(AcceleratedGradientDescent)$, je ubrzanje osnovnog metoda pada gradijenta.

Ideja koju su postavili auotori P.Stanimirović i M. Miladinović u radu [1] u osnovi je nastala na temeljima modifikovanog Njutnovog metoda uz korišćenje analize koje je primenio A. Nikolaj u konstruisanju *AGD* metoda. Kao rezultat istraživanja ovih i sličnih oblasti pomenuti autori su definisali i istestirali novi ubrzani metod oblika (1.20) sa znatno poboljšanim karakteristikama u odnosu na *AGD* iterativnu šemu. Izbor za parametar ubrzanja kod pomenutog takozvanog *SM* metoda je formulisan sledećom jednakošću

$$\gamma_{k+1} = 2\gamma_k \frac{\gamma_k [f(x_{k+1}) - f(x_k)] + t_k \|g_k\|^2}{t_k^2 \|g_k\|^2}. \tag{1.22}$$

Isti autori su takođe koristeći skalarnu aproksimaciju inverza Hesijana i kvazi-Njutnovo svojstvo konstruisali vrstu dvotačkastog metoda opadajućeg gradijenta. U ovoj šemi se radni parametar γ_{k+1} aktivira nakon ažuriranja inverza Hesijana B_k u svakom iterativnom koraku i to na sledeći način

$$B_{k+1} = B_k + a_k I = (\gamma_k + a_k) I = \gamma_{k+1} I$$

gde veličina $a_k = a_k(s_k, \gamma_k) \in \mathbb{R}$ zavisi od vrednosti dve uzastopne iterativne tačke x_k i x_{k+1} i odgovarajućih vrednosti gradijenata u tim tačkama, g_k i g_{k+1}. Parametar γ_{k+1} se određuje iz jednakosti

$$s_k = \gamma_{k+1} y_k$$

i njegova vrednost iznosi

$$\gamma_{k+1} = \gamma_k + \frac{\|s_k - \gamma_k y_k\|^2}{(s_k - \gamma_k y_k)^T y_k} = \frac{s_k^T r_k}{y_k^T r_k} \tag{1.23}$$

gde je sa r_k označen vektor $s_k - \gamma_k y_k$.

Autori u [1, 4] su opravdano i smisleno iterativne metode koji sadrže multiplikativni parametar ubrzanja, definisali kao klasu ubrzanih metoda opadajućih gradijenata (*accelerated gradient descent methods*). Pomenuti metodi ove klase detaljnije su opisani u trećoj glavi.

1.3 Osobine i karakteristike dvosmernih i dvokoračnih ubrzanih metoda

U radu [2] autori predstavljaju zanimljiv predlog iterativne šeme za rešavanje problema bezuslovne optimizacije koja u svojoj formulaciji sadrži dva različita vektora pravca. Ovaj numerički metod je definisan generalno za nediferencijabilne funkcije i opisan je sledećim izrazom

$$x_{k+1} = x_k + \alpha_k s_k + \alpha_k^2 d_k \tag{1.24}$$

U prethodnoj formulaciji α_k predstavlja dužinu iterativnog koraka koja se računa posebno definisanim algoritmom. Takođe, i dva vektora pravca s_k i d_k u relaciji (1.24) se izračunavaju prema zadatim algoritamskim šemama. Osnovni nedostak rada [2] je izostanak implementacije definisanog metoda, što je donekle i razumljivo shodno postavljenim uslovima. U svakom slučaju ovakva ideja je bila motivacija za pokušaj implementacije slično definisane iterativne šeme pod redukovanim uslovima. U radu [6] autori su analizirali slučaj diferencijabilnih funkcija i shodno tome modifikovali algoritme za računanje koraka i dva vektora pravca. Za određivanje dužine iterativnog koraka predložena je Armijeva (*backtracking*) procedura.
Vektor pravca koji je u (1.24) označen sa s_k je zamenjen sledećim izrazom

$$s_k = -\gamma_k g_k,$$

gde je γ_k ubrzavajući parametar dobijen iz Tejlorovog razvoja postavljene iteracije. Za izračunavanje drugog vektora pravca iz (1.24), d_k, predložena je diferencijabilna verzija originalnog algoritma za računanje d_k iz (1.24). Uzevši u obzir pomenute zamene, šema (1.24) je transformisana u sledeću šemu

$$x_{k+1} = x_k - \alpha_k \gamma_k^{-1} g_k + \alpha_k^2 d_k \tag{1.25}$$

prilagođenu skupu diferencijabilnih funkcija. Autori su u [6] implementirali iterativni metod (1.25). Dobijeni su izuzetno povoljni numerički rezultati po pitanju smanjenja broja itertivnih koraka. U istom radu dokazana je linerana konvergencija ovog dvosmernog metoda za uniformno konveksne i striktno konveksne funkcije. Da bi dokazali ubrzavajuća svojstva predložene iteracije autori su konstruisali neubrzavajuću verziju ovog metoda i uporedili je sa originalnom šemom. Testiranja su potvrdila da su ubrzavajuće karakteristike koje proizilaze iz multiplikativnog parametra γ_k više nego evidentne. Iz tog razloga konstruisana šema (1.25) ne predstavlja samo gradijentni metod već se može podvesti pod pomenutu klasu ubrzanih gradijentih metoda. U skladu sa navedenim konstruisani dvosmerni metod je nazvan *ADD* metod (*Accelerated Double Direction Method*). *ADD* metod je upoređen sa *SM* metodom. Numerička testiranja su potvrdila poboljšanje po pitanju broja iterativnih koraka.

Dobijeni rezultati za ovako definisanu ubrzanu dvosmernu šemu otvorili su novo pitanje: kakve bi osobine i karakteristike pokazao ubrzani metod sa dva iterativna koraka. Shodno ovoj ideji u radu [7] je predstavljen ubrzani dvokoračni metod za rešavanje problema bezuslovne optimizacije koji je nazvan *ADSS* metod (*Accelerated Double Step Size Method*). Koristeći, upšteno gledano, šemu sličnu (1.24) sa sledećim izmenama

$$s_k = -\gamma_k g_k$$

$$d_k = -g_k$$

i uvodeći dva različita iterativna koraka α_k i β_k, konstruisan je ubrzani dvokoračni iterativni model za rešavanje problema bezuslovne optimizacije

$$x_{k+1} = x_k - \alpha_k \gamma_k^{-1} g_k - \beta_k g_k = x_k - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) g_k. \tag{1.26}$$

Po konstrukciji, *ADSS* metod sadrži parametar ubrzanja γ_k koji je dobijen na sličan način kao i kod *ADD* i *SM* metoda, iz Tejlorovog razvoja koji odgovara postavljenoj iteraciji. U radu [7] numerički rezultati su pokazali višestruku prestižnost *ADSS* šeme u poređenju sa *SM* i *ADD* metodima. Prateći tri glavne karakteristike: broj iterativnih koraka, CPU vreme i broj evaluacija ciljne funkcije, pokazana je ubedljiva dominatnost *ADSS* šeme po pitanju sva tri aspekta. Na sličan način kao što je to urađeno u radovima [6, 1] i u [7] je dokazana linearna konvergencija *ADSS* metoda.

Detaljan opis pomenuta dva ubrzana dvosmerna i dvokoračna metoda prikazana su u četvrtoj glavi ove disertacije. Ostvareni rezultati ovih radova otvorili su mogućnost za istraživanje klase dvosmernih i dvokoračnih metoda i proučavanje osobina pojedinih metoda ove klase u zavisnosti od promene vektora pravaca i parametara iterativnih koraka. Sa druge strane, postavljanje određenih relacija između veličina iterativnih koraka dovodi do transformisanja dvokoračnog ubrzanog modela na ubrzani gradijentni metod sa jednim iterativnim korakom. Tako se, postavljanjem sledećeg uslova:

$$\alpha_k + \beta_k = 1$$

koji povezuje dva iterativna koraka iz *ADSS* metoda, dobija da se u svakoj iteraciji ubzana dvokoračna *ADSS* šema svodi na jednokoračnu

$$x_{k+1} = x_k - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) g_k. \tag{1.27}$$

koja je takođe ubrzana, zbog toga što sadrži parametar ubzanja γ_k dobijen iz Tejlorovog razvoja šeme (1.27). Pošto je nastala redukcijom dva koraka polazne *ADSS* iteracije na jedan, šema (1.27) je nazvana *TADSS* (*Transformed ADSS*). Ova iteracija je opisana u radu [8]. Iako su testiranja pokazala blago poboljšanje u odnosu na dvokoračnu *ADSS* iteraciju, ponašanje ova dva metoda se može smatrati približno jednakim po pitanju sva tri testirana aspekta. *TADSS* algoritam kao i razne predočene mogućnosti za dalje proučavanje i ispitivanje dvosmernih i dvokoračnih metoda su sadržaj poslednje glave disertacije.

1.4 Plan doktorske disertacije

Doktorska disertacija se sastoji iz sledećih glava od kojih je svaka podeljena na više poglavlja:

1. Uvod i uvodna razmatranja

2. Pregled i analiza osnovnih gradijentnih metoda i tehnika linijskog traženja

3. Prezentovanje i opis novijih gradijentnih i ubrzanih gradijentnih metoda za bezuslovnu optimizaciju

4. Dvokoračni i dvosmerni gradijentni metodi sa faktorom ubrzanja

U tekućoj, prvoj i uvodnoj glavi izloženi su osnovni pojmovi nephodni za formulaciju optimizacionog zadatka, kao i generalne podele i osobine metoda optimizacije. U ovoj glavi, kao što se moglo videti, dat je pregled matematičke osnove potrebne za izučavanje problema optimizacije, osnovne definicije i opšte osobine nelinearnog programiranja uz poseban akcenat koji je stavljen na bezuslovnu nelinearnu optimizaciju. Uz opštu formulaciju zadatka nelinearnog programiranja kod bezuslovne optimizacije

$$\min f(x),\ x \in \mathbb{R}^n,$$

naglašena je specifičnost zadatka nelinearne optimizacije, njegov 'neuniverzalni' karakter i uslovljenost matematičkim modelom i dimenzijom.

U drugoj glavi su opisani metodi linijskog pretraživanja po pravcu (*line-search*) koji se generalno definišu opštim iterativnim procesom

$$x_{k+1} = x_k + t_k d_k$$

gde je x_k aproksimacija ekstremuma funkcije f u tekućoj iteraciji, x_{k+1} aproksimacija ekstremuma u narednoj iteraciji, d_k pravac pretraživanja dok je t_k dužina koraka u pravcu vektora d_k.

Pored opisivanja načina određivanja intervala traženja, klasifikacije metoda, analizirani su neki poznatiji algoritmi linijskog traženja kao što su algoritmi Armija (*Armijo*), Goldštajna (*Goldstein*) i Volf-Pauela (*Wolf-Powel*).

Ista glava sadrži pregled postojećih gradijentnih metoda. Praćen je hronološki razvoj gradijentnih metoda počev od Njutnovih metoda, modifikovanih Njutnovih metoda, kvazi Njutnovih metoda kao i nekih savremenijih gradijentnih metoda.

U sadržaju naredne, treće glave poseban akcenat je stavljen na analizu novijih gradijentnih i klasu ubrzanih gradijentnih metoda, opšte formulacije

$$x_{k+1} = x_k - \theta_k t_k g_k.$$

koju je detektovao Nekulaj Andreji (*N. Andrei*) u radu [34]. Pored korišćenja koraka dužine t_k ovaj iterativni model se odlikuje parametrom ubrzanja $\theta_k > 0$ koji poboljšava ponašanje algoritma opadajućeg gradijenta. Ovakvom algoritamskom šemom su ostvareni bolji numerički rezultati u odnosu na klasičan metod opadajućih gradijenata. Pomenuti metod koristi Armijevo pravilo ili tzv. backtracking proceduru za izračunavanje veličine koraka t_k.

Sledeću ideju, koja nastavlja evaluaciju ubrzanih gradijentnih metoda, karakteriše kombinacija Andrejevog pristupa iz [3] i klasičnog Njutnovog metoda sa linijskim pretraživanjem

$$x_{k+1} = x_k - t_k G_k^{-1} g_k,$$

gde je G_k Hesijan date funkcije. Ideja se ogleda u zameni inverza Hesijana njegovom skalarnom aproksimacijom. Iz ovog zanimljivog pristupa nastao je rad [1] u kome je opisana nova algoritamska šema bezuslovne optimizacije. Ovaj metod opadajućih gradijenata je u prethodnom poglavlju već označen kao *SM*-metod. *SM*-metod donosi poboljšanje ubrzanih gardijentnih šema što je potvrđeno numeričkim rezultatima koji pokazuju prestiž *SM*-metoda u odnosu na Andrejev metod iz [34] (*AGD*-metod), po pitanju broja iteracija i potrebnog procesorskog vremena.

U ovom delu disertacije takođe je ukazano na značaj gradijentnog metoda Barzilai i Borwein, tzv. *BB*-metoda iz [5], Rajdanovog globalnog *BB*-metoda [17], tzv. *GBB*-metod, kao i gardijentnog metoda sa skalarnom korekcijom koraka opisanog u [4], *SC*-metod.

Četvrta glava je po prikazu originalnih rezultata vezanih za izradu disertacije od suštinskog značaja. U njenom početnom delu dat je opis specifične formulacije iterativne šeme za bezuslovnu optimizaciju koja sadrži dva vektora pravca. Iteracija oblika

$$x_{k+1} = x_k + \alpha_k s_k + \alpha_k^2 d_k \tag{1.28}$$

opisana je u [2]. U ovom modelu α_k predstavlja iterativni korak dok su sa s_k i d_k označena dva opadajuća smera iteracije. Svaka od ovih komponenti je definisana odgovarajućim algoritmima. Predložen metod se odnosi na nediferencijabilne funkcije i njegova implementacija u [2] nije data. Jedna od osnovnih motivacionih ideja koja je dala nove rezultate u oblasti ubrzanih metoda za bezuslovnu optimizaciju, bila je modifikacija šeme (1.28) odgovarajućim algoritmima za svaku od potrebnih komponenti koji su prilagođeni određenim diferencijabilnim slučajevima. Sledeći cilj je bila implementacija tako definisanog metoda i on je uspešno ostvaren.

U nastavku četvrte glave disertacije predstavljeni su originalni doprinosi urađenog naučnog istraživanja. Nova dva ubrzana metoda za bezuslovnu optimizaciju koja su opisana u ovom delu, utemeljena su prevashodno na idejama i rezultatima opisanim u [1, 2].

Najpre je dat detaljan opis konstrukcije, konvergencije i numerički testovi novog ***dvosmernog*** ubrzanog iterativnog modela za nelinearnu bezuslovnu optimizaciju, nazvanog *Accelerated Double Direction Method* ili skraćeno *ADD*-metod. Bazična ideja za nastanak metoda sa dva vektora pravca potiče iz [2]. Transformisanjem metoda iz [2], definisanog za nediferencijabilne slučajeve, na uslove koji podrazumevaju diferencijabilnost objektne funkcije, nastao je njegov 'diferencijabilan pandam':

$$x_{k+1} = x_k - \alpha_k \gamma_k^{-1} g_k + \alpha_k^2 d_k, \tag{1.29}$$

u kome je pravac s_k iz (1.28) zamenjen gradijentom g_k a drugi pravac, d_k, je definisan novim modifikovanim algoritmom. Korak α_k, se za razliku od predloženog algoritma za određivanje iterativnog koraka iz [2], u iteraciji (1.29) računa *backtracking* procedurom. Pokazuje se dobra definisanost ovakvog metoda kao i njegova konvergencija za određene klase diferencijabilnih funkcija. Parametar γ_k je dobijen iz Tejlorovog razvoja na sličan način kao kod *SM*-metoda u [1] i predstavlja parametar ubrzanja. Da bi se dokazale prednosti ubrzavajuće karakteristike koje donosi parametar γ_k, konstruisana je i testirana *neubrzavajuća* verzija *ADD*-metoda. *Nonaccelerated Double Direction Method* (*NADD*-metod) u svom izrazu ne sadrži faktor ubrzanja γ_k:

$$x_{k+1} - x_k = \alpha_k^2 d_k - \alpha_k g_k.$$

Dobijeni rezultati testiranja potvrđuju apsolutnu prednost *ADD*-metoda u odnosu na njegovu neubrzavajuću verziju.

Osnovni cilj konstruisanog *ADD*-metoda bio je smanjenje broja iteracija u poređenju sa postojećim metodama. Kako je u [1] već pokazana dominantnost *SM*-metoda u odnosu na metode *AGD* i *GD* (*gradient descent method*), dovoljno je bilo uporediti metode *ADD* i *SM*. Rezultati numeričkog testiranja pokazali su napredak po pitanju smanjena broja iteracija koji je ostvaren primenom *ADD*-metoda.

U istoj glavi je takođe predstavljen još jedan ubrzani metod za bezuslovnu optimizaciju. Ostvarena poboljšanja u pogledu broja iteracija primenom *ADD*-metoda dovela je do ideje da se konstruiše metod slične forme, ali koji bi imao dva koraka u datoj iteraciji i jedan smer traženja. Zamenom vektora s_k vektorom $-\gamma_k^{-1} g_k$ u (1.28) i uvođenjem još jednog iterativnog koraka β_k dobija se sledeća iterativna šema:.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \gamma_k^{-1} \mathbf{g}_k - \beta_k \mathbf{g}_k = \mathbf{x}_k - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) \mathbf{g}_k. \tag{1.30}$$

Shodno formulaciji, ovako konstruisan *dvokoračni* ubrzani gradijentni metod nazvan je *Accelerated Double Step Size Method* ili u skraćenoj formi, *ADSS*-metod. Iterativni koraci α_k

i β_k određuju se putem dve različite backtracking procedure linijskog traženja. Za smer pada izabran je gradijent posmatrane funkcije, dok se parametar ubrzanja γ_k određuje prema izrazu za datu iteraciju, putem Tejlorovog razvoja, kao kod *SM* i kod *ADD-metoda* . Istraživanja vezana za *ADSS* metod objavljena su u [7].

Osnovni motiv za konstrukciju *ADSS*-metoda bila je pored daljeg smanjenaja broja iteracija, ovog puta i smanjenje broja evaluacija funkcije kao i smanjenje potrebnog procesorskog vremena. Dobijeni numerički rezultati koji su prikazani, pokazuju da *ADSS* iterativna šema daleko prevazilazi, po pitanju sve tri karakteristike, i *SM* i *ADD* metode. Linearna konvergencija *ADSS* metoda, kao i kod *ADD* i *SM* iteracija, pokazana je za uniformno konveksne funkcije i striktno konveksne kvadratne funkcije pod određenim uslovima.

Izuzetne performanse dvokoračnog *ADSS* modela dovele su do ideje o postavljanju sledećeg uslova koji povezuje dva iterativna koraka u *ADSS* iteraciji

$$\alpha_k + \beta_k = 1.$$

Primenjen na 1.30, prethodna relacija redukuje dvokoračni ubrzani gardijentni *ADSS* metod na jednokoračnu takođe ubrzanu gradijentnu iteraciju

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \big(\alpha_k(\gamma_k{}^{-1} - 1) + 1\big)\,\mathbf{g}_k. \tag{1.31}$$

Model dobijen transformisanjem dvokoračne *ADSS* šeme na jednokoračnu opravdano je nazvan *Transformed ADSS*, tj. *TADSS*-metod, i predstavljen je takođe u četvrtoj glavi disertacije. Ovako definisan metod je zadržao dobra svojstva svog originala, *ADSS* -metoda , i čak pokazuje blaga poboljšanja u odnosu na njega. Sa druge strane, kao jednokoračni ubrzani gradijentni metod uporediv je sa takođe jednokoračnim ubrzanim gradijentnim *SM*- metod, pri čemu su potvrđena značajna poboljšanja u korist *TADSS* algoritma. I ova poslednja istraživanja doprinose značaju *ADSS* modela.

Razmatranja i generalni zaključci o prikazanim rezultatima u doktorskoj disertaciji čine sadržaj poslednjeg dela četvrte glave. Analizirana je mogućnost definisanja čitave klase dvosmernih i dvokoračnih metoda za različite izbore koraka i vektora pravca kao i mogućnosti detektovanja većeg broja metoda te klase. Izneti su i predlozi za dalja istraživanja i razvoj sličnih ubrzanih iterativnih šema nelinearne bezuslovne optimizacije. Izveden je zaključak o doprinosu dobijenih rezultata sa naučne tačke gledišta.

Glava 2

Gradijentni metodi i tehnike linijskog traženja

U ovoj glavi govoriće se o gradijentnim metodima bezuslovne optimizacije. Već je pomenuto da pod gradijentnim metodima podrazumevamo one šeme koje u svojim formulacijama sadrže samo prvi izvod objektne funkcije i tada je reč o gradijentnim metodima prvog reda. Šeme koje koriste i prvi i drugi izvod date ciljne funkcije nazivamo gradijentnim metodima drugog reda. Najpre će biti opisane tehnike linijskog traženja. Definisanje samog pojma, ilustrovanje usvojene podele kao i detaljan opis tačnog i netačnog linijskog traženja može se naći u raznoj literaturi a neka značajnija izdanja su [35, 36, 37, 38, 39, 40]. Ova tema je ujedno i sadržina prva četiri poglavlja ove glave. Razmatranja i analiza mnogih pojedinačnih često upotrebljivanih modela kao što su Armijevo, Volf-Pauelovo, Goldštajnovo pravilo su takođe u sastavu pomenutih poglavlja. U ostalim poglavljima prikazani su značajniji gardijentni metodi. Detaljno su opisane i analizirane gradijentne iterativne šeme i njihova konvergencija počev od Košijevog metoda najstrmijeg pada [41], osnovnog gradijentnog metoda i njegovih modifikacija, pa do Njutnovog metoda, modifikovanog Njutnovg i kvazi-Njutnovog metoda.

2.1 Pojam i klasifikacija linijskog pretraživanja

Za rešavanje bezuslovnog višedimenzionog minimizacionog problema

$$\min f(x), \quad x \in \mathbb{R}^n$$

pri čemu je objektna funkcija $f : \mathbb{R}^n \to \mathbb{R}$ glatka, aproksimacija traženog minimuma kod metoda linijskog traženja po gradijentu *(line search methods)* ostvaruje se iteracijama koje se generalno formulišu kao

$$x_{k+1} = x_k + t_k d_k, \quad k = 0, 1, 2, \ldots$$

Prema već usvojenim oznakama, u prethodnoj relaciji x_k je tekuća aproksimacija minimuma, x_{k+1} aproksimacija minimuma u narednoj iteraciji, t_k dužina iterativnog koraka i d_k vektor pravca pretraživanja. Očigledno je konstruisanje nove aproksimacije minimuma, koja je bolja od prethodne, obezbeđeno ukoliko su određene vrednosti parametara t_k i d_k, odnosno vrednosti veličine koraka i pravca pretraživanja. Iz ovoga zaključujemo da je osnovni preduslov dobre definisanosti metoda linijskog traženja adekvatan izbor odnosno računanje ova dva parametra. Uglavnom se kod metoda linijskog traženja parametar pravca d_k odabira, dok se veličina iterativnog koraka t_k određuje rešavanjem jednodimenzionog optimizacionog problema. Uopštena algoritamska šema metoda linijskog traženja ima sledeći prikaz.

Algoritam 2.1.1 Opšti algoritam metoda linijskog pretraživanja

1: $k := 0$
2: Ako je $\|\nabla f(_{k+1})\| \leq \varepsilon$ zaustaviti algoritam
3: Izračunati vektor d_k pravca traženja
4: Naći veličinu koraka $t_k > 0$ nekom od procedura linijskog traženja
5: $x_{k+1} = x_k + t_t k$
6: $k := k + 1$
7: izlazne veličine x_{k+1}, $f(x_{k+1})$

Već je napomenuto da se kod minimizacionih modela bezuslovne optimizacije diferencijabilne ciljne funkcije f čiji je gradijent g, vektor pravca pretraživanja uglavnom bira tako da zadovoljava uslov pada u svakoj iteraciji

$$d_k g_k < 0.$$

Jedna od podela metoda linijskog traženja je prema odabiru smera pretraživanja. Sa tim u vezi, navode se neki poznatiji izbori vektora pravca kod metoda linijskog traženja u bezuslovnoj optimizaciji kao i jedan novi algoritam za definisanje smera pretraživanja.

- *Smer negativnog gradijenta.* Za smer negativnog gradijenta

$$d_k = -g_k = -\nabla f(x_k),$$

se slobodno može reći da prethodi svim ostalim izborima vektora pravca. Takođe nije pogrešno definisati ga kao neizostavni sastavni faktor skoro svih drugih poznatih pravaca pretraživanja. Često se u literaturi može naći da se pod gradijentnim metodima, u užem smislu, podrazumevaju upravo oni gradijentni metodi kojima je smer pretraživanja u svakoj iteraciji jednak vrednosti negativnog gradijenta u tekućoj tački.

Poznato svojstvo gradijenta objektne funkcije jeste da je pravac gradijentnog vektora u svakoj tački x_k prostora normalan na površ sa konstantnom vrednošću date funkcije f i pri tom sadrži datu tačku. U praktičnom smislu, ova osobina gradijenta da upravo smer gradijenta objektne funkcije u tački x_k, u oznaci g_k, definiše smer najbržeg rasta funkcije cilja f ka njenom maksimumu polazeći od tačke x_k. Sa druge strane, obrnuti smer, odnosno smer suprotan gradijentnom smeru, predstavlja smer najbržeg pada ciljne funkcije f. Dakle, smer negativnog gradijenta jeste smer najbržeg opadanja vrednosti objektne funkcije i kretanja ka njenoj minimalnoj vrednosti. Konvergencija metoda linijskog traženja sa pravcem negativnog gradijenta je spora u slučaju pojave takozvanog cik-cak fenomena koji takođe dovodi i do divergencije. Algoritam metoda linijskog traženja sa smerom negativnog gradijenta ima sledeći prikaz:

Algoritam 2.1.2 Metod linijskog traženja sa smerom negativnog gradijenta

1: $k := 0$
2: Ako važi $\|g_k\| \leq \varepsilon$ kraj algoritma.
3: $d_k = -g_k = -\nabla f(_k)$
4: Naći veličinu koraka $t_k > 0$ nekom od procedura linijskog traženja
5: $x_{k+1} = x_k + t_k d_k$
6: $k := k + 1$
7: $x_{k+1}, f(x_{k+1})$

- *Njutnov smer.* Ovaj izuzetno važan smer pretraživanja pored gradijenta zavisi i od Hesijana G_k ciljne funkcije i definiisan je jednakošću

$$d_k = -G_k{}^{-1} g_k.$$

Ovako definisan Njutnov smer traženja zasniva se na pretpostavci da je Hesijan pozitivno definitan. Ideja iz koje je proistekao ovako konstruisan pravac bazirana je na Tejlorovoj teoremi i Tejlorovom kvadratnom razvoju analizirane funkcije. Njutnov pravac traženja doprineo je razvoju klase Njutnovih metoda koji se odlikuju jako brzom konvergencijom koja ide do kvadratnog reda. Međutim, negativna strana Njutnovog smera ogleda se u situacijama kada Hesijan ciljne funkcije nije pozitivno definitan kao i kada je računanje Hesijana komplikovano ili teško ostvarivo. To je uticalo na razvoj mnogobrojnih modifikacija Njutnovog metoda a samim tim i modifikacija Njutnovog smera. Najveće dve grupe ovakvih metoda su modifikovani Njutnovi metodi i kvazi-Njutnovi metodi.

- *Kvazi-Njutnov smer*. Ovaj smer traženja se primenjuje generalno kod kvazi-Njutnovih metoda i zasniva se na relaciji
$$d_K = -B_k g_k,$$
gde je matrica B_k predstavlja aproksimaciju inverza Hesijana. Matrica B_k se odlikuje svojstvima simetričnosti i pozitivne definitnosti.

- *Novodefinisani smer iz [6]*. Smer traženja generalno može biti proizvoljno definisan shodno potrebama razmatranog problema. Ovde je predstavljen jedan primer specifično definisanog smera traženja iz [6]. Za njegovo određivanje potrebno je prethodno znati vrednost veličine iterativnog koraka α_k. Potom se traženi vektor pravca nalazi rešavanjem određenog pomoćnog problema, što je prikazano u narednom algoritmu

Algoritam 2.1.3 Izračunavanje pravca vektora d_k.

Require: Veličina iterativnog koraka α_k.

1: $\alpha = \alpha_k$.

2:
$$d_k(\alpha) = \begin{cases} d_k^*, & k \leq m-1 \\ \sum_{i=2}^{m} \alpha^{i-1} d_{k-i+1}^*, & k \geq m, \end{cases}$$

gde je d_k^* rešenje problema $\min_{x\in\mathbb{R}} \Phi_k(d)$, pri čemu je
$$\Phi_k(d) = g_k^T d + \frac{1}{2}\gamma_{k+1} I.$$

Kada je jednom pravac pretraživanja d_k odabran, pristupa se određivanju drugog ključnog parametra metoda linijskog traženja, parametru koji karakteriše veličinu iterativnog koraka. U tom cilju definisana je pomoćna funkcija Φ

$$\Phi(t) = f(x_k + t d_k). \tag{2.1}$$

Funkcija Φ treba da zadovoljava uslov

$$\Phi(t_k) < \Phi(0)$$

koji datom metodu linijskog traženja obezbeđuje napredovanje u odabranom pravcu ka minimumu ciljne funkcije. Ukoliko je uslov (2.1) ispunjen, radi se o monotonom linijskom pretraživanju .

Problem nalaženja dužine iterativnog koraka kod metoda linijskog traženja dovodi do najopštije podele ovih metoda, i to na metode tačnog linijskog traženja (*exact line search*) i metode netačnog linijskog traženja (*inexact line search*). Kod tačnog linijskog traženja vrednost iterativnog koraka predstavlja rešenje jednodimenzionog minimizacionog problema

$$f(x_k + t_k d_k) = \min_{t>0} f(x_k + t d_k).$$

Primenu ovog načina nalaženja koraka t_k uz izbor pravca negativnog gradijenta $d_k = -g_k$ nalazimo kod Košijevog metoda najstrmijeg pada (*The steepest descent method*). Nije teško zaključiti da je rešavanja pomenutog jednodimenzionog optimizacionog problema u svakoj iteraciji nepraktično, iziskuje previše vremena i memorijskog prostora. Zbog ove činjenice razvijene su tehnike netačnog linijskog traženja koje su mnogo primenljivije u praksi. Kod procedura netačnog linijskog traženja nije nužno potrebno rešiti jednodimenzionu minimizaciju tj. odrediti tačnu vrednost minimuma postavljenog problema već je dovoljno da objektna funkcija zadovolji postavljen uslov. U zavisnosti od definisanog uslova izdvaja se veći broj tehnika netačnog linijskog traženja. Poznatije i često upotrebljivane su procedure Armija (*Armijo*) ili takozvana *backtracking* procedura, Goldštajna (*Goldstein*), Volfa (*Wolfe*), Pauela (*Powel*), Flečer (*Fletcher*) i druge [42, 43, 44, 32, 45]. Pomenuti modeli se baziraju na relaciji

$$f(x_{k+1}) < f(x_k), \quad k = 0, 1, \ldots \tag{2.2}$$

tj. podrazumevaju monotono opadanje ciljne funkcije u svakoj iteraciji. Ovakvi metodi pripadaju klasi monotonog netačnog linijskog traženja. Metodi kod kojih uslov (2.2) ne mora da bude ispunjen u svakoj iteraciji pripadaju klasi nemonotonog netačnog linijskog traženja [46, 47, 48, 49, 50]. U opštem slučaju i monotoni i nemonotoni metodi netačnog linijskog traženja odlikuju se dobrom konvergencijom.

2.1.1 Određivanje intervala pretraživanja

Kod metoda linijskih traženja potrebno je odrediti početni interval koji sadrži optimalnu vrednost x^* ciljne funkcije i unutar koga će se vršiti pretraživanje. Ovakav interval se naziva interval pretraživanja ili interval neizvesnosti.

Definicija 2.1. *(Interval pretraživanja) Za funkciju*

$$\Phi(t^*) = \min_{t\geq 0} \Phi(t) = \min_{t\geq 0} f(x_k + td_k)$$

zatvoren interval $[a,b] \subset [0,\infty]$ *kome pripada* t^**, ukoliko takav postoji, nazivamo interval traženja ili interval neizvesnosti za jednodimenzioni minimizacioni problem* $\min_{t\geq 0} \Phi(t)$.

Metod za određivanje početnog intervala traženja ogleda se u nalaženju tri tačke x_1, x_2, x_3 koje zadovoljavaju uslov

$$x_1 < x_2 < x_3 \qquad \Phi(x_1) > \Phi(x_2) \qquad \Phi(x_2) < \Phi(x_3).$$

Za pokretanje ovakve procedure potrebna je početna tačka x_0 i početna dužina koraka t_0. Algoritam započinje traženjem tačke koja je veće vrednosti od početne vrednosti x_0, dakle ide se *napred* i nastavlja se u tom smeru dok god je pretraživanje uspešno. Uprotivnom, ide se *nazad*. To praktično znači da se ispituje uslov

$$\Phi(t_0 + h_0) < \Phi(t_0) \tag{2.3}$$

Ukoliko je ispunjena prethodna nejednakost pretraživanje ide u smeru napred sve dok vrednost funkcije cilja opada. Kada se znak nejednakosti u relaciji (2.3) promeni, odnosno kada se ispuni uslov

$$\Phi(t_0 + h_0) > \Phi(t_0)$$

pretraživanje se nastavlja u smeru unazad sve dok se povećava vrednost ciljne funkcije. Ovo su osnove takozvanog *napred-nazad* algoritma (*forward-backward*) za određivanje inicijalnog intervala traženja. Izlazne veličine algoritma napred-nazad su krajnje tačke intervala neizvesnosti koji ima osobinu da sadrži minimalnu vrednost ciljne funkcije.

Algoritam 2.1.4 Algoritam napred-nazad

Require: (Ulaz) funkcija cilja f početna tačka $x_0 \in \mathbb{R}^n$, početni korak $t_0 > 0$, $h_0 > 0$ koeficijent množenja $\eta > 1$ (najčešće se uzima $\eta = 2$).

1: $t_{pr} := t_0, \quad t := t_0 + h_0$

2: Ako je $\Phi(t) > \Phi(t_{pr})$ onda $h := -h_0, \quad t := t_0 + h$

3: Sve dok je $\Phi(t) < \Phi(t_{pr})$ računati $t_{pr} := t, \quad h := h\eta, \quad t := t_{pr} + h$

4: $a := \min\{t, t_0\}, \quad a := \max\{t, t_0\}$

5: izlazne veličine a, b

2.2 Tačno linijsko traženje

Već je napomenuto da se procedura tačnog linijsko traženja zasniva na rešavanju jednodimenzionog minimizacionog problema

$$f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + t d_k) \tag{2.4}$$

Rešenje ovog zadatka predstavlja optimalnu dužinu koraka u datoj iteraciji, pri čemu se podrazumeva da je određen ili odabran pravac pretraživanja, vektor d_k. Ukoliko je moguće analitički odrediti minimum problema (2.4) tehnika tačnog linijskog traženja je podobna i primenljiva za rešavanje postavljenog optimizacionog zadatka i daje dobra konvergentna svojstva. Nažalost, u praksi je to redak slučaj. Samim tim, primena tehnike tačnog linijskog pretraživanja postaje neprihvatljiva, bilo zbog nemogućnosti rešavanja jednodimenzionog zadatka (2.4) bilo zbog neefikasnosti ovog metoda koja proističe iz znatnog gubitka vremena potrebnog za rešavanje problema (2.4), memorijskog prostora itd. Ipak, metod tačnog linijskog traženja je od velikog teorijskog značaja. Ispitivanje konvergencije dalo je korisne rezultate. Iako je osnova tehnike tačnog linijskog traženja rešavanje zadatka (2.4), koje se svodi na određivanje parametra koraka tekuće iteracije, iz analize konvergencije ovog modela izvodi se zaključak da konvergencija metoda tačnog linijskog traženja u mnogome zavisi od izbora vektora pravca d_k.

Algoritam 2.2.1 Algoritam metoda tačnog linijskog pretraživanja

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1$.

1: $k := 0$
2: Dok važi $\|g_k)\| > \varepsilon$ preći na sledeći korak
3: Izračunati vektor d_k pravca traženja
4: Naći veličinu koraka $t_k > 0$ tako da je ispunjen uslov

$$f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + t d_k)$$

5: $x_{k+1} = x_k + t_k d_k$
6: $k := k + 1$
7: izlazne veličine x_k, $f(x_k)$

Ako označimo jednodimenzionu funkciju Φ kao $\Phi(t) = f(x_k + td_k)$, uzevši u obzir jednakost (2.4) očigledno važe relacije

$$\Phi(0) = f(x_k), \quad \Phi(t) \leq \Phi(0)$$

a prvu stacionarnu tačku funkcije Φ definiše izraz

$$t_k = \min\{t \geq 0 \mid \nabla f(x_k + td_k)^T d_k = 0\}. \tag{2.5}$$

Sada se preciznije može reći da je metod tačnog linijskog pretraživanja određen relacijama (2.4) i (2.5) obzirom da relacija (2.4) definiše globalni minimum a relacija (2.5) stacionarnu tačku funkcije Φ.

2.3 Netačno linijsko traženje

U prethodnom poglavlju opisana je procedura tačnog linijskog traženja kod koje se određivanje iterativnog koraka bazira na rešavanju minimizacionog problema

$$t_k = \arg\min_{t>0} f(x_k + td_k). \tag{2.6}$$

Ovo praktično znači da se korak kod metoda tačnog linijskog traženja određuje minimizacijom funkcije f duž zraka $\{x + t\triangle x | t \geq 0\}$. Dati postupak je u praksi teško ostvariv i zahteva previše procesorskog vremena i memorije. Stoga je određivanje iterativnog koraka putem aproksimativne minimizacije funkcije $f(x+t\triangle x)$ duž zraka $\{x+t\triangle x|t \geq 0\}$ praktično gledano daleko prihvatljivije. Takva minimizacije je osnova tehnike netačnog linijskog traženja.

Značaj metoda sa netačnim linijskim traženjem je višestruk. Sam postupak tačnog rešavanja jednodimenzione minimizacije (2.6) i neretka kompleksnost ovog problema dovela je do potrebe za alterativnim rešenjem a samim tim do nastanka netačnog linijskog pristupa u rešavanju postavljene optimizacije. Pri tome, stopa konvergencije uglavnom nije smanjena u poređenju sa metodima tačnog linijskog traženja. Mnogobrojnim istraživanjima je potvrđeno da konvergencija i red konvergencije metoda linijskog traženja ne zavisi od toga da li se koristi tačno ili netačno linijsko traženje.

Opšta iteracija metoda netačnog linijskog traženja ima istu formu kao kod metoda tačnog pretraživanja

$$x_{k+1} = x_k + t_k d_k$$

pri čemu je x_{k+1} nova iterativna tačka, x_k prethodna a ključni parametri u ovoj šemi koje treba odrediti, a koji definišu efikasnost samog metoda, jesu zapravo smer traženja d_k i parametar koraka t_k.

Za smer pretraživanja može se odabrati bilo koji od predloženih izbora iz Poglavlja 2.1 ove glave. Kako nam je fokus na rešavanju minimizacionog problema, bitno je da odabran smer bude smer opadanja, odnosno da zadovoljava uslov pada

$$g_k^T d_k < 0$$

o kome je već bilo reči.

Za efikasnost metoda kao i za njegovu konvergenciju jednako je važan i parametar iterativnog koraka, odnosno određivanje njegove optimalne dužine. Sa tim u vezi, u ovom poglavlju je opisano više tehnika netačnog linijskog traženja koje imaju za cilj određivanje što optimalnije dužine koraka. Sve takve procedure generalno imaju zajedničko polazište a to je dovoljno smanjenje vrednosti objektne funkcije u svakoj iteraciji, tj. ako upotrebimo oznaku $\ll$ za obeležavanje dovoljno male razlike posmatranih vrednosti onda je osnova metoda linijskog traženja relacija

$$(\forall k \in \mathbb{N}) \qquad f(x_k + t_k d_k) \ll f(x_k),$$

a opšta algoritamska šema ima sledeći prikaz

Algoritam 2.3.1 Algoritam metoda netačnog linijskog pretraživanja

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \le \varepsilon \ll 1$.

1: $k := 0$

2: Dok je ispunjen uslov $\|g_k)\| > \varepsilon$ preći na naredne korake

3: Izračunati vektor d_k pravca traženja

4: Naći veličinu koraka $t_k > 0$ tako da je ispunjen uslov

$$f(x_k + t_k d_k) \ll f(x_k)$$

5: $x_{k+1} = x_k + t_k d_k$

6: $k := k+1$

7: izlazne veličine x_k, $f(x_k)$

U nastavku su opisane šeme netačnog linijskog traženja koje su predložili Armijo, Goldštajn, Volf-Pauel, kao i linijsko traženje unazad takozvana *backtracking* procedura koja je korišćena u novim istraživanjima opisanim u četvrtoj Glavi ove disertacije.

2.3.1 Armijev algoritam linijskog traženja unazad-*Backtracking* procedura

Opšti oblik Armijevog pravila je

$$f(x_k) - f(x_k + \tilde{\beta}^{m_k} t d_k) \ge -\rho \tilde{\beta}^{m_k} t g_k^T d_k \tag{2.7}$$

pri čemu je $\tilde{\beta} \in (0,1)$, $\rho \in (0, \frac{1}{2})$, $t > 0$ dok je m_k je najmanji nenegativan ceo broj za koji je ispunjena prethodna nejednakost. Za nadjenu najmanju celobrojnu nenegativnu vrednost m_k prethodni uslov se može preformulisati u

$$f(x_k) - f(x_k + t d_k) \ge -\rho t g_k^T d_k \tag{2.8}$$

ukoliko se uzmu u obzir sledeće dve relacije

$$\tilde{\beta}^{m_k} \equiv \beta$$

i

$$t \to \beta t \tag{2.9}$$

gde za β, ρ važe ista ograničenja kao u (2.7) a iz poslednje relacije lako se zaključuje da i t zadržava ista definisana svojstva iz nejednakosti (2.7), tj. $t > 0$. Na ovaj način opšte Armijevo pravilo netačnog linijskog pretraživanja (2.7) transformisano je u takozavanu *backtracking* proceduru ili proceduru linijskog traženja unazad čiji je izlazni kriterijum ispunjenje uslova

$$f(x_k + td_k) \leq f(x_k) + \rho t g_k^T d_k. \tag{2.10}$$

Iz prethodne analize jednostavno se zaključuje da backtracking procedura praktično zavisi od dva parametra $\rho \in (0, \frac{1}{2})$ i $\beta \in (0,1)$ a imajući u vidu izlazni uslov (2.10) nije teško formulisati algoritam backtracing procedure

Algoritam 2.3.2 Algoritam Backtracking

Require: (Ulaz) funkcija cilja f, $\rho \in (0, \frac{1}{2})$, $\beta \in (0,1)$

1: Početni korak $t := 1$

2: Dok važi $f(x_k + td_k) > f(x_k) + \rho t g_k^T d_k$, računati $t := \beta t$,

3: izlazne veličine x_k, $f(x_k)$

Relacija (2.9) jasno opravdava naziv Backtracking procedure. Kako se kreće od najveće vrednosti iterativnog koraka, $t = 1$ nastavlja se putem smanjenja ove veličine (u smeru 'unazad') sve do ispunjenja izlaznog kriterijuma, odnosno nejednakosti (2.10). Da je određivanje zadovoljavajuće veličine koraka definisano backtracking algoritmom ostvarivo pod datim uslovima potvrđuje naredna analiza. Kako važe nejednakosti

$$g_k^T d_k < 0$$

i

$$0 < \rho < 0.5$$

iz Tejlorovog razvoja prvog reda imamo da je ispunjeno

$$f(x_k + td_k) \approx f(x_k) + t g_k^T d_k < f(x_k) + \rho t g_k^T d_k.$$

Implementacija Armijevog metoda je vrlo jednostavna pa je zbog toga ovaj model često korišćen u praksi. Algoritam opšteg Armijevog pravila je zbog sličnosti sa Goldštajnovim pravilom prikazan u objedinjenoj algoritamskoj šemi u narednom poglavlju.

2.3.2 Goldštajnovo pravilo

Pravilo Goldštajna se zasniva na izlaznoj nejednakosti Armijeve backtracking procedure (2.10) i njoj dopunske nejednakosti. Radi preglednosti navedena su oba uslova jedan za drugim:

$$f(x_k + td_k) \leq f(x_k) + \rho t g_k^T d_k$$

$$f(x_k + td_k) \geq f(x_k) + (1-\rho) t g_k^T d_k \tag{2.11}$$

pri tome se u obe nejednakosti pretpostavlja isto ograničenje parametra ρ koje je predložio Armijo, $\rho \in (0, \frac{1}{2})$. Kako je uslov (2.10) ekvivalent uopštenog Armijevog pravila (2.7), može sa pravom konstatovati da je Goldštajnovo pravilo uopštenje Armijevog pravila, definisano striktnijim uslovima. Iz tog razloga sledi prikaz zajedničkog uopštenog Algoritma Armijevog i Goldštajnovog metoda:

Algoritam 2.3.3 Algoritam Armijevog i Goldštajnovog metoda netačnog linijskog traženja

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $\rho \in (0, \frac{1}{2})$, $0 \leq \varepsilon \ll 1$.

1: $k := 0$
2: Sve dok je ispunjen uslov $\|g_k)\| > \varepsilon$ preći na naredne korake
3: Izračunati vektor d_k pravca traženja
4:
- Naći veličinu koraka $t_k > 0$ tako da je ispunjen uslov (2.7) Armijevog metoda;
- Naći veličinu koraka $t_k > 0$ tako da su ispunjeni uslovi (2.10) i (2.11) Goldštajnovog metoda;

5: $x_{k+1} = x_k + t_k d_k$
6: $k := k + 1$
7: izlazne veličine x_k, $f(x_k)$

2.3.3 Pravilo Volf-Pauela Wolfe-Powel

Samo pravilo (2.10) ne garantuje da će parametar koraka t koji se njime generiše ostati unutar intervala u kome se nalazi minimum funkcije $\Phi(t)$. Koncepcija Volf-Pauelovog pravila je da se obezbedi ovo svojstvo. Pomenuti autori su predložili da se uslov (2.11) zameni uslovom

$$[\nabla f(x_k + t_k d_k)]^T d_k \geq \sigma g_k^T d_k, \quad \sigma \in (\rho, 1) \tag{2.12}$$

koji predstavlja osnovu Volf-Pauelovog pravila, dok zajedno, relacije (2.10) i (2.12) čine Volf-Pauelov metod netačnog linijskog traženja. U nejednakosti (2.12) konstanta σ zadovoljava uslov $\sigma \in (0,1)$. Od važnosti je naglasiti da interval u kome se pretražuje veličina parametra koraka t generisan Volf-Pauelovom šemom sadrži minimum funkcije Φ.

Sam Volf-Puelov uslov (2.12) proizilazi iz uslova (2.11) i teoreme o srednjoj vrednosti, jer za t_k koje zadovoljava (2.11) važi

$$t_k[\nabla f(x_k + \theta_k t_k d_k)]^T d_k = f(x_k + t_k d_k) - f(x_k) \geq (1-\rho) t_k \nabla f(x_k)^T d_k$$

što predstavlja nejednakost (2.12) za $\theta_k = 1$ i $1 - \rho = \sigma$. Sa druge strane, na osnovu teoreme o srednjoj vrednosti i uslova (2.10) važi

$$\tilde{t}_k[\nabla f(x_k + \theta_k \tilde{t}_k d_k)]^T d_k = f(x_k + \tilde{t}_k d_k) - f(x_k) = \rho \tilde{t}_k \nabla f(x_k)^T d_k$$

pri čemu je $\theta_k \in (0,1)$. Kako je $g_k^T d_k < 0$, za $\rho < \sigma < 1$ imamo

$$[\nabla f(x_k + \theta_k \tilde{t}_k d_k)]^T d_k = \rho \nabla f(x_k)^T d_k > \sigma \nabla f(x_k)^T d_k$$

a ovo jeste nejednakost (2.12) za $t_k = \theta_k \tilde{t}_k$.

Prethodna analiza potvrđuje postojanje parametra t_k koji zadovoljava oba uslova Volf-Pauelovog metoda, (2.10) i (2.12). Očigledno su postavljena ograničenja nad konstantama ρ i σ, $\rho < \sigma < 1$, neophodan uslov za egzistenciju takvog parametra t_k.

Sledi opšti algoritam Volf-Pauelovog metoda, baziranog na relacijama (2.10) i (2.12).

Algoritam 2.3.4 Algoritam opšteg Volf-Pauelovog metoda netačnog linijskog traženja

---

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $\beta \in (0,1)$, $\rho \in (0, \frac{1}{2})$, $0 \leq \varepsilon \ll 1$.

1: $k := 0$
2: Dok važi $\|g_k)\| > \varepsilon$ preći na naredne korake
3: Izračunati vektor d_k pravca traženja
4: Naći veličinu koraka $t_k > 0$ tako da su ispunjeni uslovi (2.10) i (2.12) Volf-Pauelovog metoda;
5: $x_{k+1} = x_k + t_k d_k$
6: $k := k + 1$
7: izlazne veličine x_k, $f(x_k)$

2.3.4 Konvergencija metoda netačnog linijskog traženja

Na početku ovog poglavlja naveden je opšti algoritam metoda sa netačnim linijskim traženjem.

Algoritam 2.3.5 Opšti algoritam metoda netačnog linijskog traženja

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1$.

1: $k := 0$
2: Dok važi $\|g_k)\| > \varepsilon$ računati
3: Izračunati vektor d_k pravca traženja tako da važi

$$d_k^T \nabla f(x_k) < 0$$

4: Naći veličinu koraka $t_k > 0$ primenom nekog od metoda netačnog linijskog traženja Armija, Goldštajna ili Volf-Pauela
5: $x_{k+1} = x_k + t_k d_k$
6: $k := k+1$
7: izlazne veličine x_k, $f(x_k)$

U prethodnom algoritmu se pretpostavlja da odabrani smer pretraživanja, d_k, zadovoljava uslov pada

$$d_k^T g_k < 0,$$

a da je dužina iterativnog koraka, t_k, nađena nekim od opisanih metoda netačnog linijskog traženja. Uopšte, kod izbora smera vektora pravca traženja, princip je izbeći smerove koji su skoro ortogonalni na smer negativnog gradijenta, $-g_k$. Preciznije, za smer $s_k = t_k d_k$, ugao koji smer s_k gradi sa smerom negativnog gradijenta $\theta_k \in [0, \frac{\pi}{2}]$ i konstantu $\mu > 0$ bira se da važi

$$\forall k, \quad \theta_k \leq \frac{\pi}{2} - \mu, \tag{2.13}$$

i pri tom je ispunjeno

$$\cos\theta_k = \frac{-g_k^T s_k}{(\|g_k\| \|s_k\|)}. \tag{2.14}$$

Navedeni uslov (2.13) potreban je iz razloga što bi u suprotnom bilo $g_k^T s_k \to 0$ a to bi praktično značilo da smer s_k skoro da nije smer pada.

Analizu konvergencije metoda sa netačnim linijskim traženjem započinjemo ocenom veličine pada vrednosti objektne funkcije f u svakoj iteraciji odabranog metoda sa netačnim pretraživanjem.

Teorema 2.1. *Neka je $f(x)$ uniformno konveksna funkcija, odnosno neka postoji konstanta*

$\eta > 0$ takva da važi

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

ili postoje konstante m i M, $0 < m < M$ koje zadovoljavaju nejednakosti

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

tada važi

$$f(x_k) - f(x_k + t_k d_k) \geq \frac{\rho\eta}{1+\sqrt{\frac{M}{m}}}\|t_k d_k\|^2,$$

pri čemu je ρ i t_k zadovoljavaju relaciju (2.10).

Sledi teorema koja pod izvesim uslovima garantuje globalnu konvergenciju Algoritma (2.3.5).

Teorema 2.2. *Pretpostavimo da je parametar koraka t_k u Algoritmu* (2.3.5) *generisan Goldštajnovim pravilom* (2.10)-(2.11) *ili Volf-Pauelovim pravilom* (2.10)-(2.12). *Ukoliko za objektnu funkciju f postoji na skupu $\{x|f(x) \leq f(x_0)\}$ uniformno neprekidna funkcija ∇f, tada važi jedan i samo jedan od sledećih uslova:*

- $\nabla f(x_k) = 0$ *za neko k;*
- $f(x_k) \to -\infty$*;*
- $\nabla f(x_k) \to 0$

Naredne dve teoreme govore o konvergenciji metoda netačnog linijskog traženja kod kojih je veličina iterativnog koraka generisana Volf-Pauelovim pravilom.

Teorema 2.3. *Neka je parametar koraka t_k definisan pomoću relacija* (2.10)-(2.12) *primenjenih nad neprekidno diferencijabilnom funkcijom $f : \mathbb{R}^n \to \mathbb{R}$ koja je ograničena odozdo. Neka je pritom gradijent funkcije f, ∇f, neprekidna funkcija na skupu*

$$\{x|f(x) \leq f(x_0)\},$$

tada za niz generisan Algoritmom (2.3.5) *važi*

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

odnosno

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

Teorema 2.4. *Pretpostavimo da je funkcija $f : \mathbb{R}^n \to \mathbb{R}$ neprekidno diferencijabilna i da njen gradijent ∇f zadovoljava Lipšicov uslov*

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|x - y\|.$$

Neka je uz to veličina iterativnog koraka t_k u Algoritmu (2.3.5) *određena Volf-Puelovom šemom* (2.10)-(2.12). *Ukoliko važi uslov* (2.14)*, tada za niz $\{x_k\}$ generisanog Algoritmom* (2.3.5) *važi jedna i samo jedna od navedenih relacija:*

- $\nabla f(x_k) = 0$ *za neko k;*
- $f(x_k) \to -\infty$*;*
- $\nabla f(x_k) \to 0$.

2.4 Metod najstrmijeg pada

Već je napomenuto da je tvorac metoda najstrmijeg pada Koši (*C*auchy) i da je ovaj metod jedan od prvih metoda iz takozvane klase gradijentnih metoda, koji za smer pretraživanja koristi smer negativnog gradijenta. Drugi neophodan parametar u definisanju iteracija metoda najstrmijeg pada je parametar iterativnog koraka koji se u ovom metodu određuje primenom procedure tačnog linijskog traženja. Pomenute dve najbitnije karakteristike metoda najstrmijeg pada generišu algoritam ovog metoda.

Već su pomenute negativne strane metoda sa pravcem negativnog gradijenta, pre svega pojava 'cik-cak' fenomena. Kod metoda najstrmijeg pada dodatne poteškoće stvara određivanje iterativnog koraka koje je generisano procedurom tačnog linijskog traženja odnosno rešavanjem jednodimenzionog minimizacionog problema (2.4) u svakoj iteraciji. Ponekad se veličina iterativnog koraka može odrediti tačno analitičkim putem, ali to se dešava u pojedinim slučajevima, uglavnom kod nekih posebnih kvadratnih problema. Generalno, najčešći način za određivanje dužine iterativnog koraka jeste aproksimativnom minimizacijom ciljne funkcije f u pravcu zraka $\{x_k + td_k : t > 0\}$. Uopšte posmatrano, metod najstrmijeg pada se smatra relativno efikasnim za funkcije koje su dobro uslovljene ali je praktično neupotrebljiv za funkcije sa slabijom uslovljenošću. Takođe, efikasnost ovog metoda opada čak i kod kvadratnih funkcija

Algoritam 2.4.1 Algoritam metoda najstrmijeg pada

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1$.

1: $k := 0$
2: Dok je ispunjen uslov $\|g_k)\| > \varepsilon$ računati
3: $d_k = -g_k = -\nabla f(x_k)$
4: (Tačno linijsko traženje:) Naći veličinu koraka $t_k > 0$ tako da važi

$$f(x_k + t_k d_k) = \min_{t \geq 0} f(x_k + t d_k)$$

5: $x_{k+1} = x_k + t_k d_k$
6: $k := k + 1$
7: izlazne veličine x_k, $f(x_k)$

sa porastom kondicionog broja matrica. Svoje dobre performanse pokazuje u prvim iteracijima dok se sa približavanjem ka stacionarnoj tački znatno usporava i manifestuje pojavu 'cik-cak' fenomena. Vršena su ramatranja na temu da li je neefikasnost i uopšte loše performanse metoda najstrmijeg pada uslovljena izborom pravca pretraživanja ili načinom određivanja dužine iterativnog koraka. Istraživanja su pokazala da loš uticaj proističe iz izračunavanja iterativnog koraka a ne od smera negativnog gradijenta. Neki rezultati na ovu temu objavljeni su u radu [51]. Ovakav zaljučak, prirodno je potstakao dalje razvijanje gradijentnih metoda sa netačnim linijskim traženjem dužine iterativnog koraka i uopšte minimizacionih bezuslovnih metoda sa smerom pretraživanja duž negativnog gradijenta ali sa drugačijim tehnikama za nalaženje optimalne dužine iterativnog koraka. Premda je globalno konvergentan, metod najstrmijeg pada zbog svih navedenih razloga nema posebnu praktičnu primenu. Međutim sa aspekta teorije konvergencije, analiza konvergencije ovog metoda je od izuzetne važnosti.

2.5 Osnovni gradijentni metod i njegove modifikacije

Kao što je već pomenuto, utvrđeno je da loše performanse metoda najstrmijeg pada proističu zbog načina određivanja iteratinog koraka, tj. uglavnom zbog primene procedure tačnog linijskog traženja . Sa druge strane, analiza konvergencije ovog metoda prikazana u [40] potvrđuje globalnu konvergenciju ovog metoda i otvara mogućnost da se računanje iterativnog koraka u algoritmu metoda najstrmijeg pada izvrši nekom od tehnika netačnog linijskog pretraživanja. Negativna strana i ovog slučaja je opet vreme koje je potrebno uložiti za rešavanje jednodimenzione minimizacije u svakoj iteraciji. Sa ciljem uštede vremena, javila se potreba za traženjem drugačijeg rešenja prilikom određivanja iterativnog koraka. Tako je nastala modifikacija metoda najstrmijeg pada kod koje se računanje iterativnog koraka putem rešavanja jednodimenzione optimizacije u svakoj iteraciji zamenjuje izborom konstantnog parametra koraka

$$t_k = h, \quad \forall k = 0, 1, \ldots$$

Ovako definisanim korakom konstruisan je metod pod nazivom *osnovni gradijentni metod*. Dakle, iteracije osnovnog gradijentnog metoda su definisane izrazom

$$x_{k+1} = x_k - hg_k, \quad k = 0, 1, \ldots.$$

Očigledno je da je ovako definisan metod jednostavniji za upotrebu, lakše se kodira zato što se ne zahteva prevođenje objektne funkcije $f(x)$ u pomoćnu funkciju $\phi(x)$ potrebne u rešavanju jednodimenzione optimizacije.

Izbor konstantnog koraka otvara pitanje veličine te konstante. Korak treba da ima dovoljno malu vrednost da bi norma gradijenta ostala manja od unapred zadate tačnosti sve do lokalizacije ekstremuma, ali ta vrednost treba da je istovremeno i dovoljno velika ukoliko je optimalna tačka dosta udaljena od početne. Usvojen je sledeći kriterijum za određivanje koraka osnovnog gradijentnog metoda:

$$0 < h < \frac{2}{\lambda_M},$$

pri čemu je λ_M najveća sopstvena vrednost Hesijana funkcije f. Ovakav izbor koraka daje sporiju konvergenciju od metoda najstrmijeg pada. Algoritam izgleda:

Algoritam 2.5.1 Algoritam osnovnog gradijentnog metoda

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1.$, korak $h > 0$

1: $k := 0$
2: Ako je $\|g_k)\| \leq \varepsilon$ kraj algoritma.
3: $d_k = -g_k = -\nabla f(x_k)$
4: $x_{k+1} = x_k + hd_k$; $k = k + 1$
5: izlazne veličine x_k, $f(x_k)$

U cilju daljeg smanjenja broja računskih operacija predloženo je smanjenje broja izračunavanja izvoda ciljne funkcije. Tako je nastala modifikovana verzija osnovnog gradijentnog metoda. Umesto izračunavanja gradijenta koje se u osnovnom gradijentnom metodu obavlja u svakoj iteraciji, kod njegovog modifikovanog modela koristi se jedan isti gradijentni pravac u više iteracija. Tačnije, tekući gradijentni vektor se koristi sve dok se ne dostigne maksimum za dati pravac. Zatim sledi izračunavanje novog gradijentnog pravca za čiji smer se lokalizuje maksimalna vrednost. Kao i kod osnovnog gradijentnog metoda, upotreba konstantnog koraka h i kod modifikovanog metoda ostaje na snazi. Na ovaj način znatno je redukovan broj izračunavanja vrednosti ciljne funkcije potrebnih u računanju njenih parcijalnih izvoda. Broj izračunavanja iznosi $n+1$ za funkiju od n parametara. Ovakav model ima prednosti prilikom implementiranja jer se simboličko diferenciranje koristi manji broj puta. Sledi Algoritam modifikovanog osnovnog gradijentnog metoda.

Algoritam 2.5.2 Algoritam modifikovanog osnovnog gradijentnog metoda

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1.$, korak $h > 0$

1: $k := 0$
2: Dok je ispunjen uslov $\|g_k)\| > \varepsilon$ računati
3: $d_k = -g_k = -\nabla f(x_k)$
4: $x_{k+1} = x_k + hd_k$; $k = k+1$
5: Dok važi $f(x_{k+1}) < f(x_k)$ preći na sledeći korak
6: $x_{k+1} = x_k + hd_k$; $k = k+1$
7: izlazne veličine x_k, $f(x_k)$

2.6 Dva gradijentna modela sa automatskom korekcijom koraka

Izbor konstantnog koraka kod osnovnog gradijentnog metoda sa jedne strane pojednostavljuje implementaciju ove šeme ali sa druge strane predstavlja njen osnovni nedostatak. Sa manjim vrednostima konstantnog koraka broj iteracijija potrebnih za dostizanje ekstremuma prilično raste a samim tim i broj evaluacija ciljne funkcije. U slučaju kada se odabere veća vrednost konstantnog koraka rizikuje se preskakanje ekstremuma posebno ako je ekstremum jako izražen. U ovom poglavlju su prikazane dve šeme kod kojih se korak menja automatski u zavisnosti od rasta ili opadanja vrednosti ciljnje funkcije u dve, odnosno tri poslednje iteracije.

1. Kod prvog modela izabrani iterativni korak h_k ostaje na snazi sve dok je vrednost funkcije u datoj iteraciji manja od vrednosti funkcije u prethodnoj iterativnoj tački. U protivnom

se tekući korak duplo smanjuje. Dakle,

$$h_{k+1} = \begin{cases} h_k, & f(x_k) < f(x_{k-1}) \\ \\ \frac{1}{2}h_k, & f(x_k) \geq f(x_{k-1}), \end{cases}$$

Ovaj model, može koristiti dva kriterijuma za prekid algoritma. Jedan od poznatih uslova prekida je kada norma gradijenta u tekućoj iteraciji postane manja od zadate tačnosti. Drugi mogući uslov prekida je kada veličina koraka postane manja od unapred postavljene vrednosti h_{min}. Oba kriterijuma mogu dovesti do preranog prekida pretraživanja (daleko od ekstremuma). Pokazalo se da je konvergencija ovog metoda brža ukoliko se algoritam startuje sa većim početnim korakom. Generalno, loša strana svih metoda kod kojih se veličina iterativnog koraka automatski određuje u zavisnosti od smanjenja vrednosti objektne funkcije u prethodnim iteracijama je ta što se može desiti da se lokalizacija ekstremne vrednosti izvrši dalje od samog ekstremuma. Ovo se dešava zbog prevremenog ispunjenja uslova $h_k < h_{min}$ ukoliko je izabran ovaj kriterijum.

Algoritam 2.6.1 Algoritam metoda sa automatskom korekcijom koraka 1

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in R^n$, $0 \leq \varepsilon \ll 1$, korak $h_0 > 0$

1: $k := 0$
2: Dok važi $\|g_k)\| > \varepsilon$ računati
3: $d_k = -g_k = -\nabla f(x_k)$
4: $x_{k+1} = x_k + h_k d_k$; $k = k+1$
5: Ako je $f(x_k) \geq f(x_{k-1})$ onda $h_k := \frac{h_k}{2}$;
6: izlazne veličine x_k, $f(x_k)$

2. Kod drugog gradijentnog modela sa automatskom korekcijom koraka, veličina koraka se određuje na osnovu vrednosti ciljne funkcije u tri poslednje iteracije i to na sledeći način:

$$h_{k+1} = \begin{cases} 2h_k, & f(x_k) < f(x_{k-1}) < f(x_{k-2}) \\ \\ \frac{1}{2}h_k, & f(x_k) \geq f(x_{k-1}) \\ \\ h_k, & \text{inače.} \end{cases}$$

Korak se uvećava duplo ukoliko su poslednja tri koraka bila uspešna. U slučaju da je poslednji korak bio neuspešan korak se smanjuje duplo, dok u ostalim slučajevima ostaje isti. Algoritamski, predstavljen model prikazujemo sledećim koracima:

Algoritam 2.6.2 Algoritam metoda sa automatskom korekcijom koraka 2

Require: (Ulaz) funkcija cilja f, inicijalna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1$, korak $h_0 > 0$

1: $k := 0$
2: Dok važi $\|g_k)\| > \varepsilon$ računati
3: $d_k = -g_k = -\nabla f(x_k)$
4: $x_{k+1} = x_k + h_k d_k$; $k = k + 1$
5: ako je $f(x_k) \geq f(x_{k-1})$ do onda $h_k := \frac{h_k}{2}$;
6: inače, ako je $f(x_k) < f(x_{k-1}) < f(x_{k-2})$ onda $h_k := 2h_k$;
7: izlazne veličine x_k, $f(x_k)$

2.7 Metodi konjugovanih gradijenata

Za metode konjugovanih gradijenata može se reći da su poboljšali svojstva kako metoda najstrmijeg pada tako i Njutnovog metoda. Sa ovim metodima postignta je brža konvergencija u odnosu na metod najstrmijeg pada. S druge strane, izbegnuto je određivanje Hesijana i njegovih aproksimacija, neophodnog kod Njutnovog metoda, što doprinosi dosta čestoj primeni metoda konjugovanih gradijenata u praksi. Iteracija koja generalno definiše metod konjugovanih gradijenata je oblika (1.15)

$$x_{k+1} = x_k + t_k d_k.$$

pri čemu dužina koraka predstavlja rešenje jednodimenzionog minimizacionog problema

$$\min_{t>0} f(x_k + t d_k)$$

dok se vektor pravca određuje po sledećoj šemi:

$$\begin{aligned} d_0 &= -\nabla f(x_0) \\ d_k &= -\nabla f(x_k) + p_k d_{k-1} \\ p_k &= \frac{\|f(x_k)\|^2}{\|f(x_{k-1})\|^2} \end{aligned}$$

odnosno,

$$d_k = -g_k + \frac{\langle g_k, g_k \rangle}{\langle g_{k-1}, g_{k-1} \rangle} d_{k-1}$$

U prethodnoj relaciji sa $\langle \cdot \rangle$ je označen skalarni proizvod vektora. Algoritamskom šemom metod konjugovanih gradijenata se predstavlja na sledeći način:

Algoritam 2.7.1 Metod konjugovanih gradijenata

Require: (ULAZ) Početna tačka x_0, $\varepsilon > 0$, $k := 0$.
1: Izračunati $g_0 = \nabla f(x_0)$ i $d_0 = -\nabla f(x_0)$.
2: Ako je $\|g_k\| \leq \varepsilon$, kraj.
3: Odrediti t_k rešavanjem jednodimenzionog problema $f(x_k + t_k d_k) = \min_{t>0} f(x_k + t d_k)$.
4: $x_{k+1} = x_k + t_k d_k$.
5: $d_{k+1} = -\nabla f(x_k) + p_k d_k$
   $p_0 = 0$
   $p_k = \frac{\|f(x_k)\|^2}{\|f(x_{k-1})\|^2}$
6: $k := k+1$, preći na korak 2.

Specijalno, u slučaju kada je objektna funkcija f kvadratna

$$f(x) = \frac{1}{2} x^T A x + b^T x + C \tag{2.15}$$

gde je A simetrična, pozitivno definitna matrica reda $n \times n$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}$, izvođenje metoda konjugovanih gradijenata se realizuje na sledeći način. Uzevši u obzir početni uslov $d_0 = -g_0$ i da je za kvadratnu funkciju (2.15)

$$g(x) = Ax + b$$

dobija se prva iterativna tačka

$$x_1 = x_0 + t_0 d_0.$$

Odavde je

$$g_1^T d_0 = 0.$$

Dalje je

$$d_1 = -g_1 + \beta_0 d_0 \tag{2.16}$$

gde je β_0 parametar koji zadovoljava uslov

$$d_1^T A d_0 = 0.$$

Množeći (2.16) sa $d_0^T A$ uz korišćenje poslednjeg uslova dobija se izraz za β_0:

$$\beta_0 = \frac{g_1^T g_1}{g_0^T g_0}.$$

Jednačina (2.16) se može generalizovati:

$$d_k = -g_k + \sum_{i=0}^{k-1} \beta_i d_i \tag{2.17}$$

gde se β_i bira tako da važi

$$d_k^T A d_i = 0 \quad \forall i = 1, 2, \cdots, k-1.$$

Na sličan način pokazuje se da je

$$\beta_{k-1} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}. \tag{2.18}$$

Iz svega navedenog, može se zaključiti da je iterativni postupak metoda konjugovanih gradijenata za slučaj kvadratne funkcije definisan sledećim relacijama:

$$\begin{aligned} x_{k+1} &= x_k + t_k d_k \\ d_k &= -g_k \beta_{k-1} d_{k-1} \end{aligned} \tag{2.19}$$

što zajedno sa dobijenom vrednošću za β_{k-1} u relaciji (2.18) predstavlja Flečer-Rivsovu formulaciju $(F-R)$ metoda konjugovanih gradijenata. U nastavku su navedeni neke od poznatijih formulacija metoda konjugovanih gradijenata u zavisnosti od izraza za β_{k-1}:

$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})}$, (H-S, tj. K-V, Hestens-Stivelova, odnosno Krauden-Volfova formula)

$\beta_{k-1} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}}$, (P-R-P, Polak-Riberi-Poljakova formula)

$\beta_{k-1} = -\frac{g_k^T g_k}{d_{k-1}^T g_{k-1}}$, (Diksonova formula)

$\beta_{k-1} = \frac{g_k^T g_k}{d_{k-1}^T (g_k - g_{k-1})}$, (D-J, Dan-Juanova formula)

2.8 Njutnov metod

Poznato je da je osnovni Njutnov metod za rešavanje problema bezuslovne optimizacije baziran, sa jedne strane, na kvadratnoj aproksimaciji, označenoj sa $q^{(k)}$, dva puta neprekidno diferencijabilne funkcije cilja f, čiji je Hesijan pozitivno definitna funkcija, u k-toj iterativnoj tački x_k i sa druge strane, na minimizciji ove aproksimacije. Formalni zapis ove aproksimacije je predstavljen na sledeći način:

$$f(x+s) \approx q^{(k)}(s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s_T \nabla^2 f(x_k) s$$

gde je $s = x - x_k$. Nakon minimizacije aproksimacije $q^{(k)}(s)$ dobija se jedna od najpoznatijih metoda za rešavanje problema bezuslovne optimizacije - Njutnova formula:

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

ili, zapisano sa usvojenim oznakama za gradijent i Hesijan objektne funkcije:

$$x_{k+1} = x_k - G_k^{-1} g_k. \tag{2.20}$$

Poslednja jednakost nam definiše veličinu

$$d_k = x_{k+1} - x_k = -G_k^{-1} g_k$$

koja predstavlja Njutnov smer pada. Da je ovako određen Njutnov smer pada zaista smer pada potvrđuje sledeća relacija

$$g_k^T d_k = -g_k^T G_k^{-1} g_k = -\|g_k\|^2_{G_k^{-1}} \leq 0$$

a ova nejednakost važi obzirom da je G_k pozitivno definitna funkcija. Kako sadrži Hesijan u svom izrazu, Njutnov smer pada generalno pruža bolje karakteristike od smera kod metoda najstrmijeg pada koji koristi samo gradijent. Njutnov smer pada se smatra jednim od najznačajnijih smerova. ovaj smer je vrlo koristan i zgodan za primenu u slučajevima kada se objektna funkcija ne razlikuje puno od njene kvadratne aproksimacije dobijene iz Tejlorovog razvoja. Obično Njtutnovom smeru odgovara iterativni korak $t_k = 1$ za svako k. Sledi prikaz algoritma Njutnovog metoda.

Algoritam 2.8.1 Njutnov metod

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$.

1: $k := 0$
2: Izračunati $g_k = \nabla f(x_k)$. Ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma.
3: Odrediti $G_k s = -g_k$ po s_k.
4: $x_{k+1} = x_k + s_k$.
5: $k := k+1$, preći na korak 2.

Kako je u relaciji

$$f(x_k + d) \approx f(x_k) + g_k^T d,$$

d_k je zapravo rešenje minimizacionog problema

$$\min_{d \in \mathbb{R}^n} \frac{g_k^T d}{\|d\|}$$

koje zavisi od same norme, tako u l_2 normi, imamo da je rešenje $d_k = -g_k$ što predstavlja smer metoda najstrmijeg pada. U eliposoidnoj normi, $\|\cdot\|_{G_k}$,

$$d_k = -G_k^{-1} g_k$$

a to je Njutnov smer pada.

U slučaju kada je primenljiv, Njutnov metod ima kvadratnu konvergenciju i smatra se brzim metodom. Uopšte, važi sldeća ocena

$$\frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^2} \leq M$$

za dovoljno veliko k, pri čemu je x^* tačno rešenje posmatranog minimizacionog zadatka. Njutnov metod je posebno brz u slučaju kvadratnih funkcija dok generalno kod nekvadratnih funkcija, Njutnov metod ne mora obavezno da konvergira. Kada je početna tačka relativno blizu optimalne i ukoliko se nekvadratna objektna funkcija aproksimira odgovarajućom kvadratnom aproksimacijom, Njutnov metod i u ovom slučaju ima kvadratnu aproksimaciju.

Sa druge strane, ukoliko početna tačka nije dovoljno blizu rešenja neizvesno je da je d_k Njutnov smer pada, takođe je neizvesna pozitivna definitnost Hesijana, pa je i sama konvergencija Njutnovog metoda neizvesna.

Za pokazivanje globalne konvergencije Njutnovog metoda koristi se tehnika linijskog traženja. Algoritam Njunovog metoda sa linijskim traženjem koristi činjenicu da je red konvergencije Njutnovog metoda kvardatni kada niz iterativnih koraka t_k teži jedinici i temelji se na sledeće dve definisane jednakosti

$$d_k = -G_k^{-1} g_k$$

$$x_{k+1} = x_k + t_k d_k$$

Algoritam 2.8.2 Njutnov metod sa linijskim traženjem

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$.

1: $k := 0$
2: Izračunati $g_k = \nabla f(x_k)$.
3: Ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma. Izlaz x_k, inače korak 4.
4: Rešiti $G_k d = -g_k$ po d_k.
5: (Linijsko traženje): naći t_k tako da važi $f(x_k + t_k d_k) = \min_{t>0} f(x_k + t d_k)$.
6: $x_{k+1} = x_k + t_k d_k$.
7: $k := k + 1$, preći na korak 2.

2.9 Modifikacije Njutnovog metoda

Poznato je da najveći nedostatak Njutnovog metoda proističe upravo iz onog faktora koji mu i obezbeđuje najbolje karakteristike- Hesijana. U slučaju kada Hesijan nije pozitivno definitna matrica ili kada je izračunavanje Hesijana i njegovog inverza vrlo komplikovano, Njutnov metod postaje neupotrebljiv. Ovakvi problemi su postali motivacijske osnove za razvoj modifikovanih Njutnovih metoda i kvazi-Njutnovih metoda. Kod modifikovanih Njutnovih metoda navedeni problemi se rešavaju svođenjem Njutnovog metoda na metod najstrmijeg pada ili na osnovni gradijentni metod prvog reda. Za razliku od takvog pristupa, kvazi-Njutnovi metodi nude podesno konstruisanje aproksimacija Hesijana ili aproksimacija njegovog inverza. Modifikovani Njutnovi metodi daju rešenje za slučaj kada Hesijan nije pozitivno definitan ili je nedefinitan pri čemu je objektna funkcija neograničena.

2.9.1 Modifikovani Njutnovi metodi: Goldštajn-Prajsov metod i metod Markuarda

Jedna modifikacija Njutnovog metoda je takozvani metod Goldštajna i Prajsa. Ideja ovog metoda sastoji se u zameni Njutnovog smera u slučaju kada Hesijan nije pozitivno definitan smerom metoda najstrmijeg pada uz primenu pravila ugla. Odnosno, smer vektora traženja se bira po sledećem modelu:

$$d_k = \begin{cases} -G_k^{-1} g_k, & \theta \geq \eta; \\ -g_k, & \text{inače}, \end{cases}$$

gde je η pozitivna konstanta a $\theta < \frac{\pi}{2} - \mu$ za neko $\mu > 0$.

Druga modifikovana varijanta Njutnovog metoda naznačena je kao metod Goldštajna i ostalih a u nekoj literaturi ova šema je nazvana metodom Markuarda. Ovaj model definiše sledeća iteracija

$$x_{k+1} = x_k - (G_k + \lambda_k I)^{-1} g_k \tag{2.21}$$

U prethodnoj relaciji, veličina λ_k predstavlja parametar kretanja prema ekstremumu. Očigledno je da za $\lambda_k = 0$ izraz (2.21) predstavlja Njutnov metod. U obrnutom slučaju, za dovoljno veliko λ_k, dijagonalna matrica $\lambda_k I$ postaje dominantan faktor u relaciji (2.21) pa se tada metod Markuarda ponaša približno isto kao gradijentni metodi prvog reda. Strategija sprovođenja ovog metoda je takva da se isprva uzima parametar λ_0 koji ima veliku vrednost pa se traženje minimuma ostvaruje na način sličan primeni gradijentnih metoda prvog reda. Napredovanjem ka optimumu, vrednost parametra λ_k opada pa metod Markuarda poprima karakteristike Njutnovog metoda. Sledi prikaz k-tog koraka algoritma metoda Markuarda.

Algoritam 2.9.1 k-ti korak modifikovanog Njutnovog metoda Markuarda

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$.

1: Postaviti $\bar{G}_k = G_k + \lambda_k I$, gde je $\lambda_k = 0$ ako je G_k pozitivno definitna matrica, inače $\lambda_k > 0$.
2: Izračunati $\bar{G}_k d = -g_k$ po d_k.
3: $x_{k+1} = x_k + d_k$.

Naredna teoreme potvrđuje konvergenciju modifikovanog Njutnovog metoda.

Teorema 2.5. *Neka je D otvoren skup i $f : D \subset \mathbb{R}^n \to \mathbb{R}$ dva puta neprekidno diferencijabilna* *na D. Neka je*

$$\Omega = \{x \in | f(x) \leq f(x_0)\}$$

kompaktan skup. Ako je niz x_k generisan modifikovanim Njutnovim metodom na skupu Ω tada važi

$$\lim_{k \to \infty} g_k = 0.$$

2.9.2 Kvazi-Njutnovi metodi

Razni problemi od kojih su samo neki: neefikasnost, komplikovane računske operacije, prevelik utrošak vremena, cena, koji se mogu javiti prilikom izračunavanja Hesijana dobrim delom su otklonjeni kvazi-Njutnovim metodima. Ovakvi i slični problemi kod kvazi-Njutnovih metoda se rešavaju konstrukcijom niza aproksimacija Hesijana ili konstrukcijom niza aproksimacija inverza Hesijana. Pri tome, svaka od aproksimacija zadovoljava takozvanu kvazi-Njutnovu jednačinu koja garantuje dobru definisanost tih aproksimacija. Kvazi-Njutnova jednačina i opis izvođenja kvazi-Njutnovih metoda je opisan u ovom poglavlju. Ovakvi metodi su od izuzetnog značaja ne samo zato što otklanjaju nedostatke Njutnovog metoda, već zato što istovremeno zadržavaju dobre karakteristike po pitanju konvergencije. Kvazi-njutnovi metodi u svojim formulacijama koriste jedino vrednosti ciljne funkcije i njenog gradijenta. To znači da koriste samo prvi izvod date funkcije i spadaju u gradijentne metode prvog reda. Dakle, u praktičnom smislu, izbegnuto je računanje Hesijana neophodnog kod klasičnog Njutnovog metoda.

Aproksimacije Hesijana, obično označene sa B_k, konstruišu se prema sledećim postvakama:

- Svaka od matrica B_k koja pripada nizu aproksimacija $\{B_k\}_{k \in N_0}$ jeste pozitivno definitna.
- Vektor pravca koji se određuje iz relacije

  $$d_k = B_k^{-1} g_k$$

  ima smer pada. Kao što je ranije pomenuto, da bi neki vektor traženja d_k imao smer pada date funkcije f potrebno je da zadovoljava uslov

  $$g_k^T d_k < 0.$$

  U cilju potvrde globalne konvergencije, nekada se traži da vektor d_k zadovoljava i sledeće nejednakosti

$$g_k^T d_k \leq -c_1 \|g_k\|^2$$

$$\|d_k\| \leq c_2 \|g_k\|$$

pri čemu su konstante $c_1, c_2 > 0$.

- Izračunavanje ovih matrica je relativno jednostavno. Uopšte, ideja zamene Hesijana njegovom adekvatno konstruisanom aproksimacijom povlači sa sobom i jednostavnost kako u izračunavanju samih aproksimacija tako i u računskim operacijama nad njima.

Prilikom izvođenja kvazi Njutnove jednačine koja ujedno i definiše uslove koje aproksimacije Hesijana i njegovog inverza treba da zadovoljavaju, polazi se od Tejlorovog razvoja drugog reda dva puta neprekidno diferencijabilne funkcije $f : \mathbb{R}^n \to R$ na otvorenoma skupu $D \subset \mathbb{R}^n$. Koristeći već usvojene oznake za gradijent i Hesijan objektne funkcije u tački x_{k+1} u kojoj se vrši aproksimacija, $g_{k+1} = \nabla f(x_{k+1})$ i $G_{k+1} = \nabla^2 f(x_{k+1})$, imamo:

$$f(x) \approx f(x_{k+1}) + g_{k+1}^T (x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T G_{k+1} (x - x_{k+1}).$$

Nakon diferenciranja prethodne relacije dobijamo

$$g(x) \approx g_{k+1} + G_{k+1}(x - x_{k+1}) \tag{2.22}$$

Za $x = x_k$ označimo razlike vrednosti dve uzastopne iterativne tačke i razlike vrednosti gradijenata u tim tačkama na sledeći način

$$s_k = x_{k+1} - x_k$$

$$y_k = g_{k+1} - g_k$$

Zamenom poslednje dve jednakosti u (2.22) dobija se

$$G_{k+1}^{-1} y_k \approx s_k \tag{2.23}$$

Specijalno, kada je f kvardatna funkcija sa Hesijanom G, prethodna relacija postaje

$$G^{-1} y_k = s_k$$

odnosno

$$y_k = Gs_k.$$

Relacija (2.23) je ujedno i osnovni motivacijski model u postavljanju kvazi-Njutnove jednačine. Polazeći od (2.23), očekuje se da aproksimacije inverza Hesijana, označene sa H_k, zadovoljavaju

$$H_{k+1}y_k = s_k \tag{2.24}$$

što zapravo znači da aproksimacije Hesijana, matrice B_k zadovoljavaju sledeću jednačinu

$$B_{k+1}s_k = y_k. \tag{2.25}$$

Relacije (2.24) i (2.25) nazivaju se kvazi-Njutnove jednačine ili kvazi-Njutnovi uslovi. Jednačina (2.24) se odnosi na uslov koji treba da zadovoljavaju aproksimacije inverza Hesijana H_k, dok jednačina (2.25) definiše uslov koji treba da ispunjavaju aproksimacije Hesijana B_k. Iz jedne jednačine se može izvesti druga i pri tom je jasno da je $B_{k+1} = H_{k+1}^{-1}$.

Neka je sa

$$m_{k+1}(x) = f(x_{k+1}) + g_{k+1}^T(x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T B_{k+1}(x - x_{k+1}) \tag{2.26}$$

označena model-funkcija u tački x_{k+1} koja ispunjava sledeće uslove

1. $m_{k+1}(x_{k+1}) = f(x_{k+1})$
2. $\nabla m_{k+1}(x_{k+1}) = g_{k+1}$
3. $\nabla m_{k+1}(x_k) = g_k$.

Poslednji uslov postavljen je umesto interpolacioniog uslova

$$\nabla^2 m_{k+1}(x_{k+1}) = G_{k+1}$$

koji odgovara Njutnovom metodu.

Imajući u vidu model-funkciju (2.26) kao i postavljne uslove koje treba da zadovoljava, dobija se

$$g_k = g_{k+1} + B_{k+1}(x_k - x_{k+1})$$

odnosno

$$B_{k+1}(x_{k+1} - x_k) = g_{k+1} - g_k$$

što prema usvojenim oznakama predstavlja kvazi-Njutnovu jednačinu (2.25). Kako se baš iz jednačine (2.25) dobija

$$s_k^T B_{k+1} s_k = s_k^T y_k \tag{2.27}$$

ukoliko se ista pomnoži vektorom s_k^T sa leve strane, jasno je da će matrica B_{k+1} biti pozitivno definitna ukoliko je ispunjen uslov

$$s_k^T y_k > 0. \tag{2.28}$$

Ovaj uslov se naziva uslovom prevoja za matrice H_{k+1} i B_{k+1} koje zadovoljavaju kvazi-Njutnove jednačine (2.24) i (2.25). Sledi ilustracija opšteg algoritma kvazi-Njutnovih metoda.

Algoritam 2.9.2 Kvazi-Njutnov metod-H

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n \times n}$, $0 \leq \varepsilon \leq 1$.
1: $k := 0$
2: Ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma.
3: Izračunati $s_k d = -H_k g_k$ po d_k.
4: Naći $t_k > 0$ nekom od procedura linijskog traženja.
5: $x_{k+1} = x_k + t_k s_k$.
6: Ažurirati H_k u H_{k+1} tako da važi kvazi-Njutnova jednačina (2.24).
7: $k := k + 1$, preći na korak 2.

U prikazanom algoritmu se najčešće za prvu aproksimaciju inverza Hesijana uzima identična matrica $H_o = I$. To praktično znači da prva iteracija predstavlja metod najstrmijeg pada. Varijanta algoritma 2.9.2 koji se bazira na primeni aproksimacija Hesijana, B_k razlikuje se u trećem i šestom koraku od prethodnog.

Algoritam 2.9.3 Kvazi-Njutnov metod -B

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, $H_0 \in \mathbb{R}^{n\times n}$, $0 \leq \varepsilon \leq 1$.

$k := 0$

Ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma.

Izračunati $B_k d = -g_k$ po d_k.

Naći $t_k > 0$ nekom od procedura linijskog traženja.

$x_{k+1} = x_k + t_k s_k$.

Ažurirati B_k u B_{k+1} tako da važi kvazi-Njutnova jednačina (2.25).

$k := k+1$, preći na korak 2.

Izborom aproksimacionih matrica H_k i B_k tako da zadovoljovaju uslove (2.24) i (2.25) obezbeđena je njihova pozitivna definitnost u ažuriranju. Ovu osobinu Hesijan objektne funkcije u klasičnom Njutnovom metodu ne mora nužno da ispunjava. To je ujedno i jedna od osnovnih prednosti kvazi-Njutnovih metoda u poređenju sa osnovnim Njutnovim metodom. Već je napomenuto da u svojim izračunavanjima kvazi-Njutnove iteracije koriste samo vrednosti ciljne funkcije i njenog gradijenta a ne i vrednosti Hesijana koji je neophodan u Njutnovom metodu. Samim tim i broj množenja u svakoj iteraciji, koji kod Njutnovog metoda iznosi $o(n^3)$, je sveden na $o(n^2)$ kod kvazi-Njutnovih metoda.

Kao što je pokazano u drugom poglavlju ove glave, Njutnov metod sa elipsoidnom normom $\|\cdot\|_{G_k}$ predstavlja metod najstrmijeg pada. Na sličan način se može pokazati da kvazi-Njutnov metod sa metrikom $\|\cdot\|_{B_k}$, gde je B_k aproksimacija Hesijana G_k takođe predstavlja metod najstrmijeg pada. Rešenje minimizacionog problema

$$\min_{\|d\|_{B_k \leq 1}} g_k^T d$$

jeste d_k, a kako je zadovoljena sledeća nejednakost

$$(g_k^T d)^2 \leq (g_k^T B_k^{-1} g_k)(d^T B_k d),$$

zaključujemo da je $g_k^T d_k$ najmanje ukoliko je

$$d_k = -B_k^{-1} g_k = -H_k g_k.$$

U nastavku su opisana dva značajna modela matrica koje predstavljaju aproksimacije inverza Hesijana i zadovoljavaju jednačinu (2.24). Takođe je opisano i njihovo ažuriranje. Najpre je opisana simetrična matrica za ažuriranje čiji je rang 1. Ovu matricu označavamo sa $SR1$

i ona predstavlja aproksimacionu matricu inverza Hesijana. Algoritam (2.9.2) postaje efikasna varijanta kvazi-Njutnovog metoda primenom $SR1$ matrice ažuriranja. Uz pretpostavku da je H_k aproksimacija inverza Hesijana G_k^{-1} u $k-toj$ iteraciji cilj je da se uz pomoć ove aproksimacije dobije aproksimacija inverza Hesijana u narednoj iteraciji, H_{k+1}. Ovaj postupak se naziva postupak ažuriranja matrice H_k u H_{k+1} i sprovodi se prema relaciji

$$H_{k+1} = H_k + E_k$$

gde se za E_k uzima neka matrica koja je najčešće nižeg ranga od ranga matrice H_k. Kako je $SR1$ ažuriranje ranga 1 uzima se da je

$$E_k = uv^T, \quad u, v \in \mathbb{R}^n.$$

Tada je

$$H_{k+1} = H_k + uv^T \tag{2.29}$$

odnosno

$$H_{k+1}y_k = (H_k + uv^T)y_k = s_k$$

ili

$$(v^T y_k)u = s_k - H_k y_k \tag{2.30}$$

Iz poslednje relacije se može zaključiti da je smer vektora u vektor $s_k - H_k y_k$. Kako je

$$s_k - H_k y_k \neq 0$$

jer bi u protivnom matrica H_k zadovoljavala kvazi-Njutnovu jednačinu, pretopstavimo da važi i $v^T y_k \neq 0$. Onda je jednakost

$$H_{k+1} = H_k + \frac{1}{v^T y_k}(s_k - H_k y_k)v^T$$

direktna posledica relacija (2.29) i (2.30). Kako je simetričnost uslov koji treba da zadovoljava matrica H_k uzima se da je

$$v = s_k - H_k y_k.$$

Na osnovu toga , dobija se izraz za $SR1$ ažuriranje

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \tag{2.31}$$

Naredna teorema opisuje $SR1$ ažuriranje i potvrđuje njen kvadratni završetak.

Teorema 2.6. *(Svojstva SR1 ažuriranja) Za kvadratnu funkciju čiji je Hesijan pozitivno definitan i za linearno nezavisne korake $s_0, s_1, \cdots, s_{n-1}$, $SR1$ ažuriranje se završava u $n+1$ koraka, odnosno $H_n = G^{-1}$.*

Pored toga što simetrična matrica ažuriranja ima kvadratni završetak, ona zadovoljava i relaciju

$$H_i y_j = s_j, \quad j < i.$$

Bitno je naglasiti da simetrična matrica ažuriranja ranga 1 ne zadržava osobinu pozitivne definitnosti matrice H_k. Preciznije, nejednakost

$$(s_k - H_k y_k)^T y_k > 0$$

je ispunjena ako i samo ako $SR1$ ažuriranje zadržava osobinu pozitivne definitnosti matrice H_k. Ukoliko je veličina $(s_k - H_k y_k)^T y_k$, koja predstavlja imenilac $SR1$ ažuriranja, bliska nuli može doći do prekida algoritma.

Zbog navedenih svojstava javila se potreba za unapređivanjem $SR1$ algoritma. Cilj je da se pored kvadratnog završetka obezbedi i pozitivna definitnost. Predložena je sledeća modifikacija ažuriranja $SR1$.

- Dok važi

  $$\mid (s_i - H_i y_i)^T y_i \mid \geq r \parallel s_i - H_i y_k \parallel \parallel y_i \parallel, \quad za \quad r \in (0,1) \tag{2.32}$$

  koristiti $SR1$ ažuriranje

- inače $H_{i+1} = H_i$

Da je predložena modifikacija dobro postavljena, u smislu obezbeđivanja dobrih aproksimacija Hesijana, potvrđuje teorema koja sledi.

Teorema 2.7. *Za dva puta neprekidno diferencijabilnu funkciju f čiji je Hesijan ograničen i Lipšic neprekidan u okolini tačke x^* kojoj teži niz x_k, pretpostavimo da važi uslov* (2.32)*pri* *čemu su koraci s_k uniformno linearno nezavisni. Tada niz aproksimacija inverza Hesijana H_k koji je generisan simetričnom matricom ažuriranja ranga* 1 *zadovoljava jednakost*

$$\lim_{i\to\infty} \|H_i - (\nabla^2 f(x^*))^{-1}\| = 0.$$

Sledeći predlog za matricu ažuriranja koji će biti opisan u ovom poglavlju je takozvano DFP ažuriranje koje je generisano matricom ažuriranja ranga 2. Ovakav model kvazi-Njutnovog metoda je prvi predložio Dejvidon ($Davidon$) a kasnije su ovaj koncept upotpunili Flečer ($Fletcher$) i Pauel ($Powell$). Koncept ove šeme je formiranje aproksimacije inverza Hesijana H_{k+1} na osnovu aproksimacije inverza Hesijana iz prethodne iteracije H_k. Konstrukcija nove aproksimacije inverza Hesijana bazira se dodavanju dve simetrične matrice nižeg ranga tekućoj aproksimaciji inverza Hesijana. Kako je ova matrica ažuriranja ranga 2, matrice koje se koriste u njenoj formulaciji su ranga 1, što se može ilustrovati kao

$$H_{k+1} = H_k + auu^T + bvv^T \tag{2.33}$$

pri čemu su $u, v \in \mathbb{R}^n$, dok su realne konstante a i b nepoznate koje treba odrediti. Zamenom (2.33) u (2.24) dobija se

$$H_k y_k + auu^T y_k + bvv^T y_k = s_k.$$

Imajući u vidu prethodnu relaciju za u i v se biraju sledeće vrednosti

$$u = s_k, \qquad v = H_k y_k.$$

Sada se iz (2.33) jednostvano određuju veličine parametara a i b

$$a = \frac{1}{{s_k}^T y_k}, \qquad b = -\frac{1}{{v_k}^T y_k} = -\frac{1}{{y_k}^T H_k y_k}$$

a odavde proizilazi konačan izraz DFP modela

$$H_{k+1} = H_k + \frac{s_k {s_k}^T}{{s_k}^T y_k} - \frac{H_k y_k {y_k}^T H_k}{{y_k}^T H_k y_k} \tag{2.34}$$

na kojoj se zasniva DFP algoritam

Algoritam 2.9.4 Algoritam DFP

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, simetrična i pozitivno definitna matrica $H_0 \in R^{n \times n}$, $\varepsilon > 0$.

1: $k := 0$

2: $(k - ti$ korak) za $k = 0, 1, \cdots$

1. ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma;
2. izračunati $d_k = -H_k g_k$ po d_k;
3. naći veličinu koraka $t_k > 0$;
4. postaviti
   - $s_k = t_k d_k$
   - $x_{k+1} = x_k + s_k$
   - $y_k = g_{k+1} - g_k$
   - $H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$
5. $k := k + 1$, preći na korak 1.

Kao i u slučaju $SR1$ ažuriranja i DFP procedura ima osobinu kvadratnog završetka za kvadratne funkcije i svojstvo da važi $H_i y_j = s_j, \quad j < i$. Uz to, DFP metod generiše konjugovane smerove a kada se za početnu aproksimaciju inverza Hesijana uzme da je $H_0 = I$ onda DFP šema generiše konjugovane gradijente. I za ostale funkcije, ne nužno kvadratne, značajno je da DFP algoritam čuva pozitivnu definitnost ažuriranja matrica. Broj množenja u svakoj iteraciji iznosi $3n^2 + o(n)$. Uopšte, DFP metod je superlinrano konvergentan a za striktno konveksne funkcije sa tačnim linijskim traženjem DFP je i globalno konvergentan. Jedna od važnijih prednosti DFP algoritma je čuvanje pozitivne definitnosti što je bitan preduslov stabilnosti jednog numeričkog metoda a takođe je važan faktor za globalnu knvergenciju definisanog metoda. Modeli koji zadržavaju svojstvo pozitivne definitnosti, odnosno oni modeli koji iz pozitivne definitnosti matrica H_k ili B_k generišu pozitivno definitne matrice H_{k+1} tj. B_{k+1} nazivaju se pozitivno definitnim metodima. O ovoj karakteristici DFP algoritma govori naredna teorema.

Teorema 2.8. *DFP ažuriranje* (2.34) *zadržava pozitivnu definitnost ako i samo ako važi*

$$s_k^T y_k > 0.$$

Posledica 2.1. *Svaka aproksimaciona matrica H_k konstruisana algoritmom* (2.9.4) *jeste pozitivno definitna a smerovi $d_k = -H_k g_k$ jesu smerovi pada.*

O kvadratnom završteku DFP ažuriranja govori sledeća teorema.

Teorema 2.9. *Neka kvadratna funkciaja f čiji je Hesijan pozitivno definitan koristi tačno linijsko traženje. Tada niz s_j generisan DFP metodom za sve vrednosti $i = 0, 1, \cdots, m, \quad m \leq n-1$ ima sledeća svojstva*

1. $H_{i+1} y_j = s_j, \quad j = 1, \cdots, i$
2. $s_i{}^T G s_j = 0, \quad j = 1, \cdots, i-1$
3. *metod konvergira u $m+1 \leq n$ koraka*

Ako je $m = n-1$, tada je $H_n = G^{-1}$.

Glava 3

Prezentovanje i opis novijih gradijentnih i ubrzanih gradijentnih metoda za bezuslovnu optimizaciju

PREDSTOJEĆA glava daje detaljan prikaz i analizu nekoliko gradijentnih modela bezuslovne optimizacije novijeg datuma. Opisani su oni metodi koji su bili od posebnog značaja u istraživanjima sprovedenim sa ciljem izrade disertacije.

Najpre su opisana tri, u opštem smislu važna, metoda predstavljena u [5, 52, 4]. U Poglavlju 3.1 opisan je često citiran i široko poznat metod Barzilai i Borveina, takozvani *BB*-metod. Naredno Poglavlje 3.2 govori o jednom od gradijentnih metoda definisanih od strane Neku-lai Andreia. Treći gradijentni metod od opšteg značaja predstavljen u Poglavlju 3.3 je metod skalarne korekcije, takozvani *SC*-metod, naših poznatih autora Marka Miladinovića, Predraga Stanimirovića i Slađane Miljković.

U drugom delu ove glave, akcenat je stavljen na ubrzane gradijentne algoritme. Izdvojena su dva bitnija gradijentna metoda sa ubrzanim svojstvima i opisana u četvrtom i petom pot-poglavlju tekuće glave disertacije. U Poglavlju 3.4 opisan je još jedan gradijentni metod autora Andreia Nekulaia sa ubrzanim karakteristikama. Poslednje potpoglavlje tekuće glave rezervisano je za ubrzani gradijentni *SM* metod koji je od ključnog značaja za istraživanja urađene disertacije, autora Predraga Stanimirovića i Marka Miladinovića.

3.1 Metod Barzilai-Borveina i primena tehnike nemonotonog linijskog traženja

Godine 1988 naučnici Barzilai i Borvein u radu *Two point step size gradient method* uvode novi pristup u određivanju veličine iterativnog koraka kod gradijentnog metoda. Koristeći u osnovi kvazi- Njutnov koncept iteracije, dužinu itertaivnog koraka Barzilai i Borvein su odredili putem aprksimacije jednačine sečice date kroz dve sukcesivne tačke. Kako u opštoj gradijentnoj itretativnoj šemi

$$x_{k+1} = x_k - H_k g_k \tag{3.1}$$

pomenuti autori koriste kvazi-Njutnov pristup, za aproksimaksionu matricu inverza Hesijana biraju matricu oblika:

$$H_k = t_k I.$$

U prethodnoj jednakosti se parametar t_k određuje rešavanjem minimizacije

$$\min_t \|s_{k-1} - t y_{k-1}\|^2,$$

pri čemu su veličine s_k i y_k definisane kao kod kvazi-Njutnovog metoda, tj.

$$s_{k-1} = x_k - x_{k-1}$$

i

$$y_{k-1} = g_k - g_{k-1}.$$

Kao rezultat ove aproksimacije šema (3.1) postaje

$$x_{k+1} = x_k - t_k g_k$$

pri čemu je

$$t_k = \frac{\langle s_{k-1}, y_{k-1} \rangle}{\langle y_{k-1}, y_{k-1} \rangle}, \tag{3.2}$$

gde je sa $\langle \cdot, \cdot \rangle$ označen skalarni proizvod vektora.

Uzevši u obzir uslov simetrije, može se minimizirati i izraz

$$\|t s_{k-1} - y_{k-1}\|^2,$$

što rezultira drugom vrednošću za parametar t_k:

$$t_k = \frac{\langle s_{k-1}, s_{k-1} \rangle}{\langle s_{k-1}, y_{k-1} \rangle}. \tag{3.3}$$

Konačno, iteracija koja predstavlja krajnju formulaciju gradijentnog metoda Barzilai i Borveina, takozvanog *BB* metoda, je oblika

$$x_{k+1} = x_k - t_k^{BB_{1,2}} g_k, \tag{3.4}$$

gde je

$$t_k^{BB_1} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \quad t_k^{BB_2} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}.$$

Algoritam *BB* metoda ima sledeći jednostavan prikaz:

Algoritam 3.1.1 BB-metod

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \leq 1$.

1: $k := 0$
2: Ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma, u protivnom $d_k = -g_k$.
3: Ukoliko je $k = 0$, naći t_0 linijskim traženjem, inače izračunati t_k pomoću relacija (3.2) i (3.3).
4: $x_{k+1} = x_k + t_k d_k$.
5: $k := k+1$, preći na korak 2.

U radu *Two point step size gradient method*, autori su pokazali nestandardnom teorijskom analizom superlinearnu konvergenciju Algoritma (3.1.1) za dvodimenzionalne kvadratne funkcije. Takođe su uporedili konstruisanu *BB* iterativnu šemu sa metodom najstrmijeg pada i pokazali ubedljivu prednost *BB* metoda. Potvrđeno je da je stopa konvergencije *BB* metoda viša sa porastom slabe uslovljenosti Hesijana.

Doprinos opisanog *BB* metoda je višestruk i može se slobodno reći da je ovakav pristup u definisanju iterativnog koraka predočio mnoge nove mogućnosti u proučavanju gradijentnih metoda. Istraživanja u vezi *BB* metoda, njegove modifikacije i unapređenja kako sa praktičnog tako i sa teorijskog aspekta, opisana su u mnogim radovima [24, 25, 26, 27, 28, 29, 30, 31, 18, 32, 53]. U [29] autori su potvrdili *R*-linearnu konvergenciju *BB* metoda za proizvoljno dimenzionalne striktno konveksne kvadratne funkcije a takođe je dokazano da je metod *R*-linearno konvergentan uopšteno za objektne funkcije.

Jedan od autora koji se posebno pozabavio analizom, konvergencijom i klasifikovanjem *BB* metoda je svakako Rajdan. Uz pomoć strategije globalizacije baziranoj na nemonotonom linijskom traženju iz [46], ovaj autor je u radu [17] dokazao globalnu konvergenciju *BB* metoda za nekvadratne funkcije i svrstao ga u red metoda nemonotonog linijskog traženja, [17, 18]. U pomenutim istraživanjima priloženi su numerički dokazi o svojstvima *BB* metoda za probleme sa do 10^4 promenljivih. Takođe je potvrđeno da se *BB* metod nalazi u rangu važnijih metoda konjugovanih gradijenata kao što je recimo konjugovani metod Polak-Riberija (*Polak-Ribière*), predstavljen u [21]. Zajedno sa Molinom, Rajdan je u [16] pod strogim pretpostavkama potvrdio Q-linearnu konvergenciju 'preuslovljenog' *BB* metoda. Probleme na ovu temu Rajdan je obrađivao sa Glantom (*Glunt*) i Hejdenom (*Hayden*) u [54, 55]. Razlog što je ovaj metod i dalje aktuelan u savremenim proučavanjima gradijentnih metoda i tehnika nemonotonog linijskog traženja, je pre svega njegova efikasnost i jednostavnost ali i dobra konvergencija. Zanimljivo istraživanje u vezi *BB* metoda dao je i Fletčer (*Fletcher*) u radu [56].

Praktična istraživanja su pokazala da nije uvek neophodno u svakoj iteraciji zahtevati monotono opadanje objektne funkcije f,

$$f(x_{k+1}) < f(x_k), \quad \forall k \in N.$$

Već je pomenuto da su osnove nemonotonog linijskog traženja postavljene u [46]. Datim istaraživanjem uopšteno je Armijevo pravilo tako da se dopušta povećanje vrednosti ciljne funkcije a da to ne utiče na svojstva konvergencije metoda. U pomenutom radu [46], autori Gripo (Grippo), Lamparielo (Lampariello) i Lucidi (Lucidi) pretpostvljaju da objektna funkcija zadovoljava Armijevu relaciju u svakoj novoj iteraciji, ali pod uslovom da je maksimalna vrednost funkcije postignuta u prethodnim iteracijama. Generalizacija Armijevog pravila ostvarena je definisanjem uslova

$$f(x_k + td_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \sigma t g_k^T d_k, \tag{3.5}$$

pri čemu je $m(0) = 0$, $0 \leq m(k) \leq \min\{m(k-1)+1, M-1\}$ za neki pozitivan ceo broj M, dok je σ parametar preuzet iz Armijevog pravila (2.7).

U slučaju kada objektna funkcija f nije kvadratna, primenom *BB* metoda može se dogoditi da su vrednosti koraka t_k, određene relacijama (3.2) i (3.3), ili previše male ili previše velike. Tada se, kao što je pomenuto, primenjuje strategija globalizacije koja je bazirana na nemonotonom linijskom traženju. U tom slučaju se uvode unapred zadata ograničenja nad parametrom koraka t_k:

$$0 < t^l \leq t_k \leq t^u, \quad \forall k.$$

Umesto iteracije (3.6), pod navedenim okolnostima koristi se iterativna šema

$$x_{k+1} = x_k - \frac{1}{t_k} g_k = x_k - \lambda_k g_k, \tag{3.6}$$

pri tom je

$$\frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \quad i \quad \lambda_k = \frac{1}{t_k}.$$

Kako je

$$-\lambda_k g_k = -\frac{1}{t_k} g_k = x_{k+1} - x_k \equiv s_k,$$

zaključujemo da je

$$t_{k+1} = \frac{s_k^T y_k}{y_k^T y_k} = -\frac{\lambda_k g_k^T y_k}{\lambda_k^2 g_k^T g_k} = -\frac{g_k^T y_k}{\lambda_k g_k^T g_k}.$$

Na osnovu prethodne analize i novouvedenih oznaka algoritam *BB* metoda sa nemonotonim linijskim traženjem sadži sledeće korake:

Algoritam 3.1.2 BB metod sa nemonotonim linijskim traženjem

Require: (ULAZ) Početna tačka $x_0 \in \mathbb{R}^n$, $0 \leq \varepsilon \ll 1$, $\rho \in (0,1)$, $\delta > 0$, $0 < \sigma_1 < \sigma_2 < 1$, t^l, t^u, ceo broj M.

1: $k := 0$.
2: Ako je $\|g_k\| \leq \varepsilon$ onda prekid algoritma.
3: Ukoliko je $t_k \leq t^l$ ili $t_k \geq t^u$, postaviti $\alpha_k = \delta$.
4: Postaviti $\lambda = \frac{1}{t_k}$.
5: (nemonotono linijsko traženje). Ako je

$$f(x_k - \lambda g_k) \leq \max_{0 \leq j \leq \min(k,M)} f(x_{k-j}) - \rho \lambda g_k^T g_k,$$

staviti $\lambda_k = \lambda$, $x_{k+1} = x_k - \lambda_k g_k$ i preći na korak 7.
6: Izabrati $\sigma \in [\sigma_1, \sigma_2]$, postaviti $\lambda = \sigma\lambda$, preći na korak 5.
7: Postaviti $\alpha_{k+1} = -\frac{g_k^T y_k}{\lambda_k g_k^T g_k}$, $k := k+1$, preći na korak 2.

Shodno principima nemonotonog linijskog pretraživanja, za kraj ovog poglavlja ilustrovana je algoritamska procedura linijskog traženja primenljiva na *BB*-metod i metod pada gradijenata

Algoritam 3.1.3 Nemonotono linijsko traženje

Require: (ULAZ) Objektna funkcija $f(x)$, smer traženja d_k, brojevi $0 < \sigma < 0.5$, $\beta \in (0,1)$, $a \in \mathbb{R}$, $m \in N_0$.

1: $t := a$.

2: Dok je ispunjeno $f(x_k + td_k) \leq \max_{0 \leq j \leq m} f(x_{k-j}) + \sigma t g_k^T d_k$, uzeti $t := t\beta$.

Napomena 3.1. *Za vrednost $m = 0$ u prethodnom algoritmu, definisano nemonotono linijsko traženje svodi se na Armijevo linijsko traženje.*

3.2 Novi gradijentni metod Andreia Nekulaia

Autor Nekulai Andrei *(Neculai Andrei)* u radu *A new gradient descent method for unconstrained optimization*, [52], predstavlja novi koncept definisanja iterativnog koraka koji se bazira na aproksimaciji dijagonalnom skalarnom matricom. Određivanje vodećeg skalara aproksimacione matrice bazirano je na Tejlorovom razvoju drugog reda. Pri tome se u izračunavanjima koriste samo vrednosti funkcije i njenog gradijenta u dve uzastopne iterativne tačke. Ovakav pristup je dao izuzetno dobre rezultate, otvorio mogućnosti za razvoj različitih gradijentnih metoda baziranih na pretpostavkama koje je uveo Andrei, te i motivaciju za izdvajanje čitave klase metoda sa slično određenjim iterativnim korakom, koju su detetkovali autori u radu [1] i nazvali je klasom ubrzanih metoda opadajućeg gradijenta, (*a class of accelerated gradient descent methods*).

U samom radu *A new gradient descent method for unconstrained optimization* Andrei vrši poređenje novog modela sa Košijevim metodom pada po gradijentu i Barzilai-Borveinovim *BB*-metodom. Iako koristi svega $3n$ memorijskih lokacija, klasični Košijev metod najstrmijeg pada koji koristi smer negativnog gradijenta a veličinu iterativnog koraka određuje primenom procedure tačnog linijskog traženja, je izuzetno spor i neefikasan zbog čega se vrlo retko primenjuje u praksi. O tome je bilo više reči u u prethodnoj glavi. Iz tog razloga Andrei u svom radu bira da jedan od poredbenih metoda bude Košijev metoda pada po gradijentu, u kome se vrednost iterativnog koraka određuje tehnikom netačnog linijskog traženja i to backtracking procedurom. Ovakav izbor definisanja koraka iteracije je odabran verovatno iz razloga što i sam autor u svom novopredloženom modelu određuje iterativni korak na isti način. Za parametre korišćene u backtracking proceduri ($\rho \in (0, 0.5)$ i $\beta \in (0,1)$) Andrej je uzeo vrednosti $\rho = 0.0001$ i $s = 0.8$. Sledi algoritam metoda pada po gradijentu sa korišćenjem backtracking procedure koji će u nastavku biti označen kao *GD* metod.

Algoritam 3.2.1 GD algoritam

Require: (ULAZ) Objektna funkcija $f(x)$, početna tačka $x_0 \in domf$

1: $k = 0$
2: Izračunati pravac pretraživanja $d_k = -\nabla f(x_k)$.
3: Odrediti veličinu iterativnog koraka primenom backtracking procedure linijskog traženja.
4: Izračunati novu iterativnu tačku $x_{k+1} = x_k + t_k d_k$.
5: Proveriti kriterijum završetka. Ako je ispunjen, onda KRAJ algoritma, inače $k = k+1$ i preći na korak 2.

Veliki pomak po pitanju efikasnosti i brzine konvergencije koji su učinili autori Barzilai i Borvein konstrukcijom *BB* metoda bio je inspiracija i Nekulaiu za traženje novog rešenja u definisanju iterativnog koraka koji bi otklonio evenetualne nedostatke *BB* šeme. Na svojstven i originalan način Nekulai Andrei je unapredio *BB* metod i predložio svoj gradijentni model. Novodefinisani gardijentni metod, koji će u nastavku biti označavan kao *NGD* metod (od *new gradient method* iz naslova rada [52]), se prema sprovedenim numeričkim eksperimentima ponaša približno kao *BB* algoritam ili ga blago prevazilazi. Pri tome je od velikog značajna činjenica da se u situacijama kada se *BB* ne može upotrebiti, moguće je Barzilai-Borveinov parametar $t^{BB} = \gamma^{BB}$ zameniti Andrejevom veličinom γ_A. U svom modelu Andrei koristi Armijevo linijsko traženje unazad. Iz sprovedene analize, bazirane na Telorovom razvoju drugog reda, određuje se iterativna konstanta $\gamma_A(x_{k+1})$ koju potom Andrei koristi za računanje vrednosti inicijalnog koraka u backtracking proceduri. U nastavku je opisana analiza postavljanja *NGD* metoda, kovergencija, numerički eksperimenti i poređenja *NGD* modela sa metodom najstrmijeg pada i *BB* metodom.

Nekulaieva procedura polazi od odabrane inicijalne tačke x_0, izračunate vrednosti funkcije u toj tački, $f(x_0)$ i vrednosti njenog gradijenta, $g_0 = \nabla f(x_0)$. Uz korišćenje Armijevog backtracking pravila sa početnim korakom $t = 1$ izračunava se dužina iterativnog koraka prve iteracije, t_0, čija vrednost daje prvu iterativnu tačku:

$$x_1 = x_0 - t_0 g_0.$$

Zatim se iznova računa $f(x_1)$ i $g_1 = \nabla f(x_1)$. U $k+1$ iterativnoj tački

$$x_{k+1} = x_k - t_k g_k, \quad k = 1, 2, \cdots$$

uzima se vrednost funkcije iz Tejlorovog razvoja drugog reda

$$f(x_{k+1}) = f(x_k) - t_k g_k^T g_k + \frac{1}{2} t_k^2 g_k^T \nabla^2 f(z) g_k, \quad z \in [x_k, x_{k+1}].$$

Za dovoljno malo rastojanje između uzastopnih tačaka x_k i x_{k+1} može se uzeti da je $z = x_{k+1}$. Istovremeno se Hesijan $\nabla^2 f(z)$ može zameniti aprkosimacionom dijagonalno-skalarnom matricom

$$\nabla^2 f(z) \approx \gamma_{k+1} I, \quad \gamma_{k+1} \in \mathbb{R}.$$

Iz predloženih pretpostavki dobija se vrednost parametra γ_{k+1}

$$\gamma_A(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{t_k^2} [f(x_{k+1}) - f(x_k) + t_k g_k^T g_k]. \tag{3.7}$$

U cilju izračunavanja naredne iterativne tačke

$$x_{k+2} = x_{k+1} - t_{k+1} g_{k+1},$$

potrebno je odrediti veličinu koraka t_{k+1} u toj iteraciji. Sa tim u vezi analizira se funkcija

$$\Phi_{k+1}(t) = f(x_{k+1}) - t g_{k+1}^T g_{k+1} + \frac{1}{2} t^2 \gamma_{k+1} g_{k+1}^T g_{k+1} \tag{3.8}$$

koja ima sledeća svojstva:

$$\Phi_{k+1}(0) = f(x_{k+1}) \tag{3.9}$$

$$\Phi_{k+1}'(0) = -g_{k+1}^T g_{k+1} < 0 \tag{3.10}$$

Iz (3.9) i (3.10) sledi zaključak da je funkcija $\Phi_{k+1}(t)$ konveksna za svako $t \geq 0$ i ima minimalnu vrednost ukoliko je $\gamma_{k+1} > 0$. Uzimajući u obzir ovaj uslov kao i uslov da je izvod funkcije Φ_{k+1} jednak nuli u tački minimuma, $\Phi_{k+1}(t^*) = 0$, dobija se da je tačka

$$\bar{t}_{k+1} = \frac{1}{\gamma_{k+1}}$$

tačka minimuma funkcije $\Phi_{k+1}(t)$. Dalje je

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

što potvrđuje da se vrednost funkcije smanjuje za $\gamma_{k+1} > 0$. Na ovaj način je definisana gornja granica veličine iterativnog koraka t_{k+1}. Preciznije, dužina novog iterativnog koraka t_{k+1} određuje se backtracking procedurom tako da je

$$t_{k+1} = \arg\min_{t \le \bar{t}_{k+1}} f(x_{k+1} - t g_{k+1})$$

Andrej Nekulai razmatra i slučaj $\gamma_{k+1} < 0$. Ukoliko se dogodi da je

$$f(x_{k+1}) - f(x_k) + t_k g_k^T g_k < 0$$

tada je $f(x_k) - f(x_{k+1}) > t_k g_k^T g_k$. Autor predlaže da se u tom slučaju veličini tekućeg koraka t_k doda vrednost η_k tako da bude ispunjeno

$$f(x_{k+1}) - f(x_k) + (t_k + \eta_k) g_k^T g_k > 0.$$

Prethodna nejednakost je ispunjena ukoliko je

$$\eta_k = \frac{1}{g_k^T g_k}[f(x_{k+1}) - f(x_k) + t_k g_k^T g_k] + \delta \tag{3.11}$$

za neko $\delta > 0$. Tada je vrednost novog (pozitivnog) parametra $\gamma_A(x_{k+1})$

$$\gamma_A(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{(t_k + \eta_k)^2}[f(x_{k+1}) - f(x_k) + (t_k + \eta_k) g_k^T g_k]. \tag{3.12}$$

Algoritam 3.2.2 NGD Algoritam

Require: (ULAZ) Objektna funkcija $f(x)$, početna tačka $x_0 \in domf$

1: Izračunati $f(x_0)$, $g_0 = \nabla f(x_0)$ i $t_0 = \arg\min_{t<1} f(x_0 - tg_0)$, $f(x_1)$ i $g_1 = \nabla f(x_1)$. Postavi $k = 0$.

2: Proveriti kriterijum završetka. Ako je ispunjen, onda KRAJ algoritma, inače preći na korak 3.

3: Izračunati skalarnu aproksimaciju Hesijana funkcije f u tački x_{k+1}, $\gamma_A(x_{k+1})$ prema formuli (3.7).

4: Ako je $\gamma_A(x_{k+1}) < 0$, izabrati $\delta > 0$ i izračunati $\gamma_A(x_{k+1})$ prema formuli (3.12) gde je η_k definisano relacijom (3.11).

5: Odrediti veličinu inicijalnog koraka primenom

$$\bar{t}_{k+1} = \frac{1}{\gamma_A(x_{k+1})}$$

sa kojom se startuje backtracking procedura u narednom koraku 6.

6: Koristeći proceduru backtracking linijskog traženja odrediti korak t_{k+1}:

$$t_{k+1} = \arg\min_{t \leq \bar{t}_{k+1}} f(x_{k+1} - tg_{k+1})$$

7: Izračunati narednu iterativnu tačku $x_{k+2} = x_{k+1} + t_{k+1}g_{k+1}$, postaviti $k = k+1$ i preći na korak 2.

U numeričkim testiranjima Andrejev *NGD* metod je upoređen sa *GD* i *BB* metodima. Urađene su komparacije nad 12 test funkcija. Ilustrovani primeri uglavnom pokazuju da nova *NGD* šema po svojim performansama daleko prevazilazi *GD* model a da se u odnosu na *BB* algoritam ponaša pribižno jednako ili poboljšano. Primeri uglavnom ptvrđuju da je Andrejeva konstanta γ_A veća ili jednaka od konstante Barziali-Borveina u *BB* metodu $t^{BB} = \gamma^{BB}$.

U numeričkim testiranjima korišćena su dva kriterijuma završetka

$$\|\nabla f(x_k) \leq \varepsilon_g\|$$

ili

$$\frac{|(f(x_{k+1}) - f(x_k)|}{1 + |f(x_k)|} \leq \varepsilon_f$$

gde u konstante $\varepsilon_g = 10^{-6}$ i $\varepsilon_f = 10^{-16}$. Generalno se može zaključiti da su oba metoda, i *BB* i *NGD*, linearno konvergentni. Stopa konvergencije je blago uslovljena kondicionalnim brojem matrice Hesijana minimizirajuće funkcije. Za dobro uslovljene konveksne funkcije i *BB* i *NGD* algoritmi predstavljaju napredak u odnosu na *GD* metod. Za loše uslovljene funkcije ovi algoritmi su poput drugih gradijentnih metoda vrlo spori tako da u tim slučajevima nemaju praktičnu primenu. Kako uglavnom važi

$$\gamma_A \geq \gamma^{BB},$$

pokazuje se da je inicijalni korak backtracking procedure kod *NGD* algoritma manji od odgovarajućeg inicijalnog koraka *BB* šeme. Ali u toku iteracija γ_A je blisko γ^{BB}. Iz ove činjenice je proistekla ideja da se u slučaju kada je $\gamma^{BB} < 0$ ova konstanta u *BB* metodu modifikuje Andrejevom konstatntom γ_A.

U drugoj teoremi Andrej dokazuje strogu pozitivnost konstante γ_A.

Teorema 3.1. *Realan broj γ_A generisan NGD algoritmom jeste strogo pozitivna konstanta.*

Dokaz. Kako je izlazni uslov backtracking procedure

$$f(x_{k+1}) - f(x_k) + t_k g_k^T g_k > 0$$

vazi $f(x_k) - f(x_{k+1}) < t_k g_k^T g_k$ a odavde zamenom u (3.7)

$$\gamma_A(x_{k+1}) = \frac{2}{t_k} - \frac{2(f(x_k) - f(x_{k+1}))}{t_k^2(g_k^T g_k)} > \frac{2}{t_k} - \frac{2t_k(g_k^T g_k)}{t_k^2(g_k^T g_k)} = 0$$

□

Prethodna teorema potvrđuje dobru definisanost inicijalnog koraka $\bar{t}_{k+1}$ backtracking procedure koju je Andrej predložio u 5.koraku *NGD* algoritma.

Za kraj ovog poglavlja sledi prikaz kumulativnih rezultata koji pokazuju broj iteracija tesiranih funkcija *BB* i *NGD* metodima. U prvoj tabeli ilustrovani su dobijeni rezultati testiranih funkcija sa brojem parametata: $n = 1000, 2000, 3000, 4000, 5000$. Druga tabela sadrzi rezultate testiranih funkcija sa brojem parametara: $n = 10000, 20000, 30000, 40000, 50000$.

Tabela 3.1. *Broj iteracija BB i NGD metoda za* $n = 1000, 2000, 3000, 4000, 5000$.

Nr. Ex.	**BB**	**NGD**
Ex2	12370	12304
Ex3	887	320
Ex4	291	344

Nr. Ex.	**BB**	**NGD**
Ex5	1037	984
Ex6	439	408
Ex7	182	187
Ex8	1072	1033
Ex9	79	60
Ex10	9420	3602
Ex11	230	315
Ex12	1381	127
TOTAL	27388	19684

Tabela 3.2. *Broj iteracija BB i NGD metoda za* $n = 10000, 20000, 30000, 40000, 50000$.

Nr. Ex.	**BB**	**NGD**
Ex1	18723	18675
Ex2	7003	5528
Ex3	644	547
Ex4	236	261
Ex5	1087	970
Ex6	430	388
Ex7	169	168
Ex8	1147	1021
Ex9	75	60
Ex10	7832	3367
Ex11	230	315
Ex12	1026	125
TOTAL	38602	31425

3.3 Metod skalarne korekcije

Na osnovama nemonotonog linijskog traženja autori Marko Miladinović, Predrag Stanimirović and Sladjana Miljković u radu *Scalar Correction Method for Solving Large Scale Unconstrained Minimization Problems*, [4], su konstruisali gradijentni metod koristeći kvazi-Njutnovu jednačinu i odgovarajuću skalarnu aproksimaciju inverza Hesijana. U ovom radu primarno se pristupa rešavanju minimizacionog problema bezuslovne optimizacije

$$\min f(x), \quad x \in \mathbb{R}^n \tag{3.13}$$

za datu objektnu funkciju $f : \mathbb{R}^n \to \mathbb{R}$. Polazište je najčešće upotrebljivana iteracija za ovu vrstu problema

$$x_{k+1} = x_k + t_k d_k, \quad k = 0, 1 \ldots \tag{3.14}$$

koja je više puta pominjana i analizirana u ovoj tezi. Dalje u [4], pomenuta iteracija (3.14) se poredi sa iteracijom koja definiše kvazi-Njutnovnu šemu

$$x_{k+1} = x_k - B_k g_k \tag{3.15}$$

u kojoj B_k predstavlja odgovarajuću aproksimaciju inverza Hesijana. Postoje različiti izbori aproksimacija inverza Hesijana. Uglavnom se taj izbor bazaira na što jednostavnijoj formi aproksimacione matrice. Poznatiji primeri ovakvih aproksimacionih matrica opisani su u [57]. Neki od izbora su:

$$B_k = \lambda_k I$$

gde je λ_k skalar ili dijagonalna matrica

$$B_k = diag(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

Poslednja aproksimacija suštinski zahteva veliki memorijski prostor zbog skladištenja celokupne aproksimacione matrice inverza Hesijana u svakoj iteraciji, pa ovakvi izbori nisu pogodni za optimizacione probleme većih razmera. Ovo je jedan od razloga zbog koga su autori u [4] odabrali aproksimaciju inverza Hesijana u obliku sklarne matrice

$$B_k = \gamma_k I \approx G_k^{-1}, \quad \gamma_k > 0.$$

Sa ovakvim izborom aproksimacione matrica inverza Hesijana, kvazi-Njutnova šema (3.15) sve svodi na iteraciju

$$x_{k+1} = x_k - \gamma_k g_k.$$

Sličnim razmatranjima su se prethodno bavili naučnici Barzilai i Borvein što je rezlutirao konstruisanjem *BB* metoda opisanog u [5]. Podsetimo se da su rešavanjem minimizacije

$$\gamma_k = \arg\min_t \|t^{-1} s_{k-1} - y_{k-1}\|^2,$$

Barzilai i Borvein odredili iterativni korak (3.16) i iteratini korak (3.17) u simetricnom obliku

$$\gamma_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}, \tag{3.16}$$

$$\hat{\gamma}_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}. \tag{3.17}$$

U prethodnim relacijama prema već usvojenim oznakama je $s_k := x_{k+1} - x_k$ i $y_k := g_{k+1} - g_k$. Iterativni model opisan u [4] po svojim računskim karakteristikama i veličini potrebnog memorijskog prostora je uporediv sa *BB* algoritmom. Autori su novokonstruisani metod nazvali *metod skalarne korekcije* ili u skraćenoj formi *SC* metod (*scalar correction method*). Za *SC* metod se generalno može reći da predstavlja neku vrstu poboljšanog pandama *BB* metoda koji je, prema sprovedenim testiranjima, unapređen od strane *SC* algoritma po svim praćenim karakteristikama: broju iteracija, potrebnog procesorskog vremena i broju određivanja vrednosti ciljne funkcije.

Pri numeričkim eksperimentima *SC* metod je upoređen sa *BB* metodom. Oba algoritma se baziraju na tehnici nemonotonog linijskog traženja ilustrovanog algoritmom (3.1.3). Parametar a iz ovog algoritma koji predstavlja inicijalnu veličinu koraka je određen relacijom (3.16) kod *BB* metoda , odnosno relacijom (3.30), koja će kasnije biti navedena, kod *SC* metoda.

Već je bilo reči da se tehnika nemonotonog linijskog traženja bazira na upotrebi celobrojne konstante M takvoj da se u svakoj iteraciji veličina koraka t određuje saglasno nejednakosti

$$f(x_k + td_k) \leq \max_{0 \leq j \leq m(k)} f(x_{k-j}) + \sigma t g_k^T d_k, \tag{3.18}$$

pri čemu je $m(0) = 0$, $0 \leq m(k) \leq \min\{m(k-1)+1, M-1\}$, a σ je parametar iz Armijevog pravila (2.7).

Podsetimo se da su autori u [46] dokazali globanu konvergenciju metoda sa nemonotonim linijskim traženjem za dva puta neprekidno diferencijabilne funkcije kod kojih vektor traženja zadovoljava uslove u kojima su konstatne c_1 i c_2 pozitivne

$$g_k^T d_k \leq -c_1 \|g_k\|^2 \tag{3.19}$$

$$\|d_k\| \leq c_2 \|g_k\|. \tag{3.20}$$

Isti autori su takođe potvrdili globalnu konvergenciju pomenutih metoda sa nemonotonim linijskim traženjem i pod oslabljenim uslovima (3.19) i (3.20). Imajući u vidu definicuju *R* linearne konvergencije koja upravo sledi, autori L. Grippo, F. Lampariello, S. Lucidi su u radu [46] dokazali da je svaki iterativni metod koji koristi tehniku nemonotonog linijskog traženja *R* linerano konvergentan za uniformne konveksne funkcije.

Definicija 3.1. *Neka je $x_k \in \mathbb{R}^n$ neki niz koji konvergira ka x^*. Ako važi*

$$0 < \limsup_{k \to \infty} \|x_k - x^*\| < 1,$$

onda kažemo da niz $\{x_k\}$ R linearno konvergira ka x^*.

Propozicija 3.1. *Pretpostavimo da je f glatka i uniformno konveksna funkcija. Neka je* (3.14) *bilo koji iterativni metod kod koga vektor pravca d_k zadovoljava uslove* (3.19) *i* (3.20) *i neka je iterativni korak t_k određen algoritmom nemonotonog linijskog traženja* (3.1.3)*. Tada postoje konstante $c_3 > 0$ i $c_4 \in (0,1)$ takve da*

$$f(x_k) - f(x^*) \leq c_3 c_4^k \left[f(x_1) - f(x^*) \right].$$

U ovoj disertaciji pomenuto je već da je Rajdan koristeći strategiju globalizacije zasnovanu na nehomogenom linijskom traženju koja je predstavljena u [46] dokazao globalnu konvergenciju *BB* metoda za nekvadratne funkcije. Autori u [4] koriste ove Rajdanove rezultate objavljene u [17]. Oni porede *BB* metod sa nemonotonim linijskim traženjem, koji su označili kao *GBB* metod, sa skalarnim metodom predstavljenim u [4] i označenim kao *SC* metod. Isti autori ilustruju opšti algoritam za takozvane dvotačkasto-koračne gradijentne metode *(two-point step size gradient methods)* koji koriste tehniku nemonotonog linijskog traženja za rešavanje mnogobrojnih problema bezuslovne optimizacije. Pomenuti algoritam, čiji prikaz sledi, pripada klasi kvazi-Njutnovih metoda sa nemonotonim linijskim traženjem kod kojih se aproksimacija inverza Hesijana definiše odgovarajućom skalarnom matricom.

Algoritam 3.3.1 Opšti gradijentni algoritam sa nemonotonim linijskim traženjem

Require: Ciljna funkcija $f(x)$, izabrana početna tačka $x_0 \in dom(f)$, positivna celobrojna konstanta M i realne konstante $0 < \xi_1 < \xi_2$.

1: Postaviti $k = 0$, izračunati $f(x_0)$, $g_0 = \nabla f(x_0)$ i uzeti $\gamma_0 = 1$, $m(0) = 0$.
2: Ako je ispunjen kriterijum završteka, onda zaustaviti iteracije; inače preći na sledeći korak.
3: Naći t_k putem Algoritma 3.1.3 sa ulaznim veličinama $d_k = -g_k$, $m = m(k)$ i $a = \gamma_k$.
4: Izračunati $x_{k+1} = x_k - t_k g_k,\ f(x_{k+1})$, g_{k+1}, $s_k := x_{k+1} - x_k$, $y_k := g_{k+1} - g_k$.
5: Izračunati inicijalnu veličinu koraka traženja γ_{k+1} prema postavljenom metodu. Akoje $\gamma_{k+1} > \xi_2$ ili $\gamma_{k+1} < \xi_1$, postaviti $\gamma_{k+1} = 1$.
6: Postaviti $k = k+1$, $m(k) = \min\{m(k-1)+1, M-1\}$, i preći na korak 2.
7: Izlazne veličine x_{k+1} i $f(x_{k+1})$.

Korak 5 iz prethodnog algoritma obezbeđuje ograničenost niza γ_k. Ovaj korak zavisi od metoda koji se primenjuje u Algoritmu 3.3.1. U slučaju primene *BB* metoda, γ_{k+1} iz koraka 5 uglavnom je definisano relacijom (3.16) iz razloga što je utvrđeno da ovaj izbor inicijalnih veličina koraka daje bolje rezultate u praktičnim izračunavanjima nego izbor (3.17). Dakle, u slučaju primene *BB* metoda γ_{k+1} iz koraka 5 Algoritma 3.3.1 postaje

$$\gamma_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}.$$

Sada je moguće definisati algoritam *BB* gradijentnog metoda sa nemonotonim linijskim traženjem, koji je korišćen u [4] kao poredbeni algoritam.

Algoritam 3.3.2 BB gradijentni algoritam sa nemonotonim linijskim traženjem

Step 5: Izračunati inicijalni korak tražena γ_{k+1} pomoću (3.16). Ako je $\gamma_{k+1} > \xi_2$ ili $\gamma_{k+1} < \xi_1$, postaviti $\gamma_{k+1} = 1$.
Ostali koraci su identični sa koracima u Algoritmu 3.3.1.

Ako u Algoritmu 3.3.1 želimo da primenimo *SC* metod inicijalni korak traženja γ_{k+1} iz koraka 5 ovog algoritma se definiše na osnovama analize koja sledi i koja istovremeno predstavlja bazične ideje u konstruisanju *SC* metoda. Kako je u kontruisanju *SC* metoda izabrabna skalrna matrica za aproksimaciju inverza Hesijana

$$B_k = \gamma_k I,$$

imamo da je aproksimacija inverza Hesijana u narednom koraku

$$B_{k+1} = B_k + a_k I = (\gamma_k + a_k)I = \gamma_{k+1} I,$$

a odavde se jednostavno zaključuje da je veličina inicijalnog koraka traženja γ_{k+1} koji će biti primenjen u koraku 5 Algoritma 3.3.1

$$\gamma_{k+1} = \gamma_k + a_k. \tag{3.21}$$

Kako parametar a_k zavisi od s_k i y_k, $a_k = a_k(s_k, y_k)$, a kako je $s_k := x_{k+1} - x_k$, $y_k := g_{k+1} - g_k$ zaključujemo da ovaj parametar zapravo zavisi od dve vrednosti uzastopne itertivne tačke x_k i x_{k+1} i vrednosti gradijenata u tim tačkama g_k i g_{k+1}. Primenivši sekantnu jednačinu na tačke x_k i x_{k+1} dobija se

$$s_k = \gamma_{k+1} y_k. \tag{3.22}$$

odakle se zamenom (3.21) dobija

$$s_k - \gamma_k y_k - a_k y_k = 0.$$

Na sličan način kao kod *BB* matoda, autori u [4] nalaze a_k putem minimizacije normi

$$\min_a \|a^{-1}(s_k - \gamma_k y_k) - y_k\|^2 \quad \text{i} \quad \min_{\hat{a}} \|(s_k - \gamma_k y_k) - \hat{a} y_k\|^2. \tag{3.23}$$

što rezultira određivanjem simetričnih vrednosti parametra a_k

$$a_k = \frac{(s_k - \gamma_k y_k)^T (s_k - \gamma_k y_k)}{(s_k - \gamma_k y_k)^T y_k} \quad \text{i} \quad \hat{a}_k = \frac{(s_k - \gamma_k y_k)^T y_k}{y_k^T y_k}. \tag{3.24}$$

Autori od prethodna dva rešenja biraju prvo, odnosno a_k, kao sinonim izbora veličine (3.16) umesto (3.17) kod *BB* metoda. U slučaju kada je $(s_k - \gamma_k y_k)^T y_k = 0$ bira se $a_k = 0$. Zamenom a_k iz (3.24) u (3.21) dobija se

$$\gamma_{k+1} = \gamma_k + \frac{(s_k - \gamma_k y_k)^T (s_k - \gamma_k y_k)}{(s_k - \gamma_k y_k)^T y_k} = \gamma_k + \frac{\|s_k - \gamma_k y_k\|^2}{(s_k - \gamma_k y_k)^T y_k}, \tag{3.25}$$

ili $\gamma_{k+1} = \gamma_k$ u slučaju kada je $(s_k - \gamma_k y_k)^T y_k = 0$. Upravo zbog ovako definisane skalarne korekcije metod je dobio naziv *scalar correction method*.

Uvođenjem pomoćnog vektora

$$r_k = s_k - \gamma_k y_k$$

prethodni izraz za γ_{k+1} postaje

$$\gamma_{k+1} = \gamma_k + \frac{\|r_k\|^2}{y_k^T r_k} = \frac{s_k^T r_k}{y_k^T r_k}, \tag{3.26}$$

Ovakvim definisanjem inicijalnog koraka autori u [4] su pronašli u izvesnom smislu njegovu optimalnu veličinu: njegova vrednost je dovljno velika ali nije i prevelika (toliko da 'preskoči' ekstremum) što je obezbeđeno upotrebom nemonotonog linijskog traženja. Sa ciljem potvrde da je ovako određena veličina inicijalnog koraka najadekvatnija, u smislu efikasnosti i brzine konstruisanog modela, izbor (3.26) se poredi sa još dva izbora inicijalnog koraka koji su takođe dobijeni iz dvotačkaste aproksimacije sekantne jednačine kvazi-Njutnovog metoda. Jedan od tih izbora je korak koji su predložili Barzilai i Borvein γ_{k+1}^{BB} koji je definisan relacijom (3.17). Drugi je

$$\gamma_{k+1}^{SS} = \frac{\|s_k\|}{\|y_k\|}. \tag{3.27}$$

Relacija (3.27) je dobijena normiranjem jednačine (3.22).

Dalje je dokazana sledeća lema kojom se tvrdi da je izbor γ_{k+1} iz (3.26) najadekvatniji kada je $y_k^T r_k > 0$ u poređenju sa ostala dva predložena izbora.

Lema 3.1. *Za korake* (3.26), (3.16) *i* (3.27) *važe sledeće nejednakosti*

$$\gamma_{k+1}^{BB} \le \gamma_{k+1}^{SS} \le \gamma_{k+1}, \qquad y_k^T r_k > 0, \tag{3.28}$$

$$\gamma_{k+1} \le \gamma_{k+1}^{BB} \le \gamma_{k+1}^{SS}, \qquad y_k^T r_k \le 0. \tag{3.29}$$

Korisreći prethodnu lemu, konačan izbor inicijalnog koraka u *SC* metodu je definisan na sledeći način:

$$\gamma_{k+1}^{SC} := \begin{cases} \frac{s_k^T r_k}{y_k^T r_k}, & y_k^T r_k > 0, \\ \frac{\|s_k\|}{\|y_k\|}, & y_k^T r_k \le 0. \end{cases} \tag{3.30}$$

Imajući u vidu prethodnu analizu, algoritam *SC* metoda je definisan narednim koracima:

Algoritam 3.3.3 Algoritam SC metod opadanja po gradijentu sa nemonotonim linijskim traženjem

Ulaz: Objektna funkcija $f(x)$, izabrana inicijalna tačka $x_0 \in dom(f)$, pozitivni ceo broj M i realne konstante $0 < \xi_1 < \xi_2$.

Korak 5: Izračunati inicijalni korak traženja γ_{k+1} pomoću 3.30. Ako je $\gamma_{k+1} > \xi_2$ ili $\gamma_{k+1} < \xi_1$, postaviti $\gamma_{k+1} = 1$.

Ostali koraci su identični koracima iz Algoritma 3.3.1.

U opisima numeričkih eksperimenata dat je pregled dobijenih rezultata testiranja konstruisanog *SC* metoda i *BB* metoda. Bitna zajednička karakteristika oba testirtana modela je da je na njima primenjena tehnika nemonotonog linijskog traženja. Parametri koji su korišćeni u nemonotonom traženju uzimaju vrednosti $\sigma = 0.0001$ i $\beta = 0.8$. Za vrednosti ostalih potrebnih parametara u Algoritmu 3.3.1 uzeto je $\xi_1 = 10^{-5}$, $\xi_2 = 10^5$ i $M = 10$.

Testirano je 40 test funkcija predloženih u [34]. Analizirano je deset različitih vrednosti broja promenljivih za svaku test funkciju: 100, 500, 1000, 2000, 3000, 5000, 7000, 8000, 10000 i 15000. Pri tome su korišćena dva kriterijuma završetka

$$\|g_k\| \leq 10^{-6} \quad \text{i} \quad \frac{|f(x_{k+1}) - f(x_k)|}{1 + |f(x_k)|} \leq 10^{-16}.$$

U prikazanoj numeričkoj analizi, praćene su tri osnovne karakteristike: broj iteracija, CPU vreme i broj određivanja vrednosti objektne funkcije za svaki test problem. U Tabeli 3.3 dat je prikaz vrednosti sve tri karakteristike za 40 testiranih funkcija metodima *SC* i *GBB*. Ova tabela takođe sadrži optimalne vrednosti testiranih funkcija kada je broj parametara 100. Iz priloženog se može videti da *SC* algoritam daje uglavnom bolje ili jednako dobre rezultate u poređenju sa *GBB* algoritmom. Preciznije, po pitanju broja iteracija *SC* metod prevazilazi *GBB* u slučaju 21 testirane funkcije dok 9 test funkcija daju iste rezultate. U analizi broja određivanja vrednosti testiranih funkcija, *SC* je optimalniji u 22 slučaja dok je *GBB* bolji u 9 testiranja.

Tabela 3.3. *Prikaz numeričkih rezultata* GBB *i* SC *metoda testiranih na* 40 *test funckcija.*

Test function	Br. iteracija		CPU vreme		Br. eval. funk.		Min. funk.	
	GBB	SC	GBB	SC	GBB	SC	GBB	SC
Extended Freud. and Roth	417	590	2.98	3.653	1734	1970	0	2E-16
Extended Rosenbrock	858	440	8.995	3.621	4253	1430	0	0

Extended White and Holst	619	580	5.026	5.088	1998	1840	7.66E-13	0
Extended Beale	444	693	5.322	8.261	1028	1676	2E-13	2.45E-13
Extended Penalty	506	501	6.792	7.058	2066	2056	75	75
Perturbed Quadratic	18072	6893	155.231	45.182	62401	18899	4.6E-15	2.6E-15
Raydan 1	8197	3894	52.089	20.557	27040	10556	505	505
Raydan 2	75	75	0.387	0.307	160	160	100	100
Diagonal 1	7066	3313	43.479	21.855	15798	8874	-15706.7	-15706.7
Diagonal 2	10525	4871	111.168	37.292	41163	13778	15.74	15.74
Diagonal 3	8397	3457	83.809	29.903	28014	9501	-4605.8	-4605.8
Hager	950	887	6.808	6.228	2523	2059	-653.08	-653.08
Generalized Tridiagonal 1	280	276	2.169	2.387	640	632	97.21	97.21
Extended Tridi-agonal 1	372	359	1.683	2.074	864	808	2.73E-9	2.73E-9
Extended Three Exponential Terms	100	90	0.996	0.715	300	280	127.96	127.96
Diagonal 4	70	65	0.543	0.48	330	320	0	2E-16
Diagonal 5	50	50	0.59	0.495	110	110	69.31	69.31
Extended Him-melblau function	135	120	1.481	1.324	410	380	2.4E-15	0
Generalized PSC1	7344	4688	193.87	105.322	26099	13002	98.72	98.72
Extended PSC1	150	160	2.293	2.447	460	480	38.66	38.66
Extended Powell	10	10	0.03	0.015	30	30	0	0
Extended Block Diagonal BD1	178	166	1.918	1.854	416	392	8.2E-15	8.2E-15
Extended Maratos	200	305	2.058	2.965	730	1000	-50.03	-50.03
Quadratic Diago-nal Perturbed	5891	5494	51.934	39.652	23921	17054	3.1E-12	3.1E-12
Quadratic QF1	18175	7407	145.512	45.121	61728	20355	-0.005	-0.005
Extended Quadratic Penalty QP1	171	189	3.714	4.09	740	776	390.06	390.06
Quadratic QF2	13383	7504	140.136	74.198	45740	20972	-1.00	-1.00
Extended EP1	48	48	1.152	1.215	356	356	793.18	793.18

Extended Tridiagonal-2	401	485	1.51	1.808	892	1112	38.58	38.58
ARWHEAD	140	170	4.119	4.293	710	797	-2.1E-14	-2.1E-14
Almost Perturbed Quadratic	20336	7131	170.088	41.511	69489	19472	1.32E-13	1.32E-13
ENGVAL1	333	392	4.963	5.309	836	956	109.09	109.09
QUARTC	212	212	1.09	1.12	474	474	5.97E-9	5.97E-9
Diagonal 6	75	75	0.371	0.262	160	160	1.11E-14	1.11E-14
LIARWHD	579	677	12.104	11.465	2615	2425	0	0
Generalized Quartic GQ1	216	223	2.995	2.714	525	538	1.56E-14	0
Diagonal 7	70	70	0.497	0.465	160	160	-81.68	-81.68
Diagonal 8	70	70	0.729	0.589	190	190	-48.05	-48.05
Diagonal 9	6553	5329	32.528	29.37	22417	14986	-15346.2	-15346.2
HIMMELH	20	20	0.121	0.091	60	60	-62.5	-62.5
AVERAGE	3229,4[illegible]	3780,9[illegible]	[illegible]490244	17,16978049	11191,56098	6180,829268		

U narednoj tabeli 3.4 autori prikazuju broj numeričkih eksperimenata od ukupnih 400 u kojima posmatrni metodi dostižu minimalni broj iteracija, minimalno CPU vreme i minimalni broj određivanja vrednosti objektne funkcije. U poslednjoj koloni tabele 3.4 prikazan je broj testiranja u kojima oba metoda daju iste vrednosti.

Tabela 3.4. *Poredbene performanse* SC *i* GBB *metoda u* 400 *numeričkih eksperimenata.*

Karakteristike	SC	GBB	oba metoda
Broj iteracija	199	102	99
CPU vreme (sec)	186	84	130
Broj evaluacija funkcije	208	100	92

U trećoj tabeli 3.5 autori ilustruju srednje vrednosti rezultata koje ostvaruju oba metoda u svih 400 testiranja po pitanju sve tri analizirane karakteristike. Vrednosti ove tabele potvrđuju da *SC* metod daje duplo bolje rezultate po pitanju broja iteracija, 2.2 puta bolje rezultate u pogledu utrošenog procesorskog vremena i broja evaluacija ciljnih funkcija.

Tabela 3.5. *Srednje vrednosti numeričkih rezultata za* 40 *test funkcija prilikom* 10 *numeričkih eksperimenta u svakoj iteraciji.*

Srednje vrednosti performansi	SC	GBB
Broj iteracija	169.9	329.2
CPU vreme (sec)	1.43	3.16
Broj evaluacija funkcije	477.7	1123.9

Autori su takođe prikazali i grafičke ilustracije performansi poredbenih *SC* i *GBB* algoritama. Opšti zaključak je da je *SC* algoritam efikasniji, jednostavniji i zauzima manje memorijskog prostora od *GBB* modela sa kojim je upoređen.

3.4 Andrejev ubrzani gradijentni algoritam

Za rad *An acceleration of gradient descent algorithm with backtracking for unconstrained optimization*, [3], Andreja Nekulaja koji je opisan u ovom poglavlju se slobodno može reći da predstavlja motivacijsku osnovu za definisanje klase ubrzanih gradijentnih metoda. Prepoznatljiva karakteristika metoda ove klase je dobro definisan multiplikativni faktor gradijenta i iterativnog koraka. Preciznije, poznato je da je najopštija iteracija bezuslovne minimizacije oblika

$$x_{k+1} = x_k + t_k d_k, \tag{3.31}$$

pri čemu je sa d_k označen vektor traženja a sa t_k iterativni korak. Andrej u [3], u prethodnoj iterativnoj šemi za vektor traženja bira negativan gradijent i uvodi parametar ubrzanja $\theta_k > 0$ tako da iteracija 3.31 postaje

$$x_{k+1} = x_k - \theta_k t_k g_k, \tag{3.32}$$

Pre opisa procedura određivanja parametra ubrzanja θ_k, važno je istaći da u izračunavanju veličine iterativnog koraka autor koristi Armijevu backtracking proceduru. Backtracking parametri uzimaju sledeće vrednosti: $\sigma = 0.0001$ i $\beta = 0.8$. U narednoj propoziciji, definiše se donja granica vrednosti iterativnog koraka određenoj backtracking procedurom.

Propozicija 3.2. *Neka je d_k vektor traženja koji ima smer pada i neka $\nabla f(x)$ zadovoljava Lipšicov uslov $\|\nabla f(x) - \nabla f(xy\| \leq L\|x - y\|$ za sve x i y iz skupa $\{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$, pri čemu* *je L pozitivna konstanta. Ako linijsko traženje zadovoljava Armijev uslov tada važi*

$$t_k \geq \min\left\{1, \frac{\beta(1-\sigma)}{L}\frac{-g_k^T d_k}{\|d_k\|^2}\right\}. \tag{3.33}$$

U određivanju veličine faktora ubrazanja θ_k, autor polazi od dve ključne nejedankosti. Prva predstavlja izlazni uslov Armijeve backtracking procedure

$$f(x_k - tg_k) \leq f(x_k) - \sigma t g_k^T g_k.$$

Druga koristi Tejlorovoj razvoj primenjen na iteraciju 3.32

$$f(x_k - \theta t_k g_k) = f(x_k) - \theta t_k g_k^T g_k + \frac{1}{2}\theta^2 t_k^2 g_k^T \nabla^2 f(x_k) g_k + o\left(\|\theta t_k g_k\|^2\right). \tag{3.34}$$

U poređenju sa Tejlorovim razvojem u tački $z = x_k - t_k g_k$

$$f(x_k - t_k g_k) = f(x_k) - t_k g_k^T g_k + \frac{1}{2} t_k^2 g_k^T \nabla^2 f(x_k) g_k + o\left(\|t_k g_k\|^2\right), \tag{3.35}$$

izvodi se zaljučak da je

$$f(x_k - \theta t_k g_k) = f(x_k - t_k g_k) + \Psi_k(\theta). \tag{3.36}$$

Uporedivši prethodne dve jednakosti, lako je odrediti izraz funkcije $\Psi_k(\theta)$

$$\Psi_k(\theta) = (1-\theta) t_k g_k^T g_k - \frac{1}{2}(1-\theta^2) t_k^2 g_k^T \nabla^2 f(x_k) g_k + \theta^2 t_k o\left(t_k \|g_k\|^2\right) - t_k o\left(t_k \|g_k\|^2\right). \tag{3.37}$$

Andrej dalje uvodi oznake :

- $a_k = t_k g_k^T g_k \geq 0$
- $b_k = t_k^2 g_k^T \nabla^2 f(x_k) g_k$
- $\varepsilon = o\left(t_k \|g_k\|^2\right)$

kojima se relacija 3.37 transformiše u

$$\Psi_k(\theta) = (1-\theta) a_k - \frac{1}{2}(1-\theta^2) b_k + \theta^2 t_k \varepsilon - t_k \varepsilon. \tag{3.38}$$

Analizirajući funkciju $\Psi_k(\theta)$ iz prethodne relacije, pomoću definisanih veličina a_k i b_k Andrej određuje vrednost parametra ubrzanja θ. Nije teško uočiti da je funkcija $\Psi_k(\theta)$ konveksna za $b_k \geq 0$. Dalje važi

$$\Psi'_k(\theta) = (b_k + 2t_k\varepsilon)\theta - a_k \quad \text{i} \quad \Psi'_k(\theta_m) = 0$$

za

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

Kako je $\Psi'_k(0) = -a_k < 0$, funkcija $\Psi_k(\theta)$ je konveksna kvadratna funkcija sa minimumom u tački θ_m i pri tom je

$$\Psi'_k(\theta_m) = -\frac{(a_k - (b_k + 2t_k\varepsilon))^2}{2(b_k + 2t_k\varepsilon)} \leq 0.$$

Zamenivši tačku θ tačkom θ_m u relaciji 3.36 a pod uslovom da je $b_k \geq 0$ pokazuje se da vrednost funkcje f u narednoj iterativnoj tački, primenom iteracijom (3.32) ima manju vrednost od vrednosti funkcije f u narednoj iterativnoj tački iteracije 3.31 (u kojoj je $d_k = -g_k$):

$$f(x_k - \theta_m t_k g_k) = f(x_k - t_k g_k) - \frac{(a_k - (b_k + 2t_k\varepsilon))^2}{2(b_k + 2t_k\varepsilon)} \leq f(x_k - t_k g_k).$$

Daljom analizom, koja se ovom prilikom izostavlja, autor pokazuje da se postavljenom iteracijom 3.32 ostvaruje pad vrednosti funkcije, odnosno da važi $f(x_{k+1}) \leq f(x_k)$. Bližim određivanjem vrednosti veličine b_k

$$b_k = t_k^2 g_k^T \nabla^2 f(x_k) g_k = -t_k y_k^T g_k, \tag{3.39}$$

gde je $y_k = g_{k+1} - g_k$, i zanemarivanjem konstante ε u relaciji koja definiše θ_m dobija se konačan izraz za izračunavanje parametra ubrzanja:

$$\theta_k = \theta_m = \frac{a_k}{b_k}.$$

Kako su definisani svi potrebni parametri sada se može ilustrovati Algoritam Andrejevog ubrzanog gradijentnog metoda, skraćeno označenog kao *AGD* algoritam:

Algoritam 3.4.1 *AGD* metod

Require: Objektna funkcija $f(x)$, odabrana inicijalna tačka $x_0 \in dom(f)$.

1: Postaviti $k = 0$, izračunati $f(x_0)$ i $g_0 = \nabla f(x_0)$.
2: Pronaći korak t_k Backtracking procedurom linijskog traženja.
3: Izračunati: $z = x_k - t_k g_k, g_z = \nabla f(z)$ i $y_k = g_z - g_k$.
4: Izračunati: $a_k = t_k g_k^T g_k, b_k = -t_k y_k^T g_k$ i $\theta_k = \frac{a_k}{b_k}$.
5: Postaviti: $x_{k+1} = x_k - \theta_k t_k g_k$ i izračunati f_{k+1} i g_{k+1}.
6: Proveriti kriterijum završetka. Ukoliko je zadovoljen, zaustaviti algoritam; inače postaviti $k = k+1$ i preći na korak 2.

Napomena 3.2. *Iz prethodnog algoritma* (3.4.1) *se može izvesti algoritam metoda pada po gradijentu, GD metoda, ukoliko se zanemare koraci* 3 *i* 4 *i postavi* $\theta_k = 1$ *u koraku* 5.

Konvergenciju *AGD* metoda potvrđuje sledeća propozicija:

Propozicija 3.3. *Pretpostavimo da je* f *strogo konveksna funkcija na skupu* $S = \{x|f(x) \leq f(x_0)\}$. *Tada niz* x_k *generisan AGD algoritmom konvergira linearno ka* x^*.

U numeričkim testiranjima sprovedenim radi ispitivanja osobina definisanog *AGD* metoda i njegovog upoređivanja sa srodnim algoritmima, analize su izvršene za 34 test funkcije. Izbran je sledeći desetočlan skup vrednosti broja promenljivih:$\{100, 200, 300, 400, 500, 600, 700, 800, 900, 1000\}$. Korišćena su sledeća dva kriterijuma završetka svakog testiranja:

$$\|g_{k+1}\|_\infty \leq \varepsilon_g \quad \text{ili} \quad t_k|g_k^T d_k| \leq \varepsilon_f |f(x_{k+1})|,$$

pri čemu je $\varepsilon_g = 10^{-6}$ i $\varepsilon_f = 10^{-20}$.

Andrejev ubrzani *AGD* metod je upoređen sa metodom pada po gradijentu, *GD* metodom. Praćene su vrednosti koje prikazuju broj ostvarenih iteracija, potrebno CPU vreme kao i broj evaluacija objektne funkcije. Potvrđen je ubedljiv prestiž *AGD* algoritma po pitanju sve tri karakteristike u odnosu na *GD* metod. Naredne dve tabele ilustruju ovu tvrdnju. U Tabeli (3.6) prikazane su prosečne vrednosti broja iteracija, CPU vremena u sekundama i broja evaluacija funkcija pri testiranju *AGD* i *GD* metodima respektivno. U Tabeli 3.7 prikazan je broj testiranja u kojima svaki od analiziranih metoda ostvaruje minimalno iteracija, minimalno CPU vreme i minimalnan broj evaluacija funkcije.

Tabela 3.6. *Prosečne vrednosti numeričkih rezultata za* 340 *testiranja.*

Ukupne vrednosti	AGD	GD
Ukupan broj iteracija	657915	1598990
Ukupno CPU vreme (sec)	17432902	40979772
Ukupan broj evaluacija funkcije	454780	1004326

Tabela 3.7. *Performanse AGD i GD algoritama u* 340 *problema.*

Kriterijum performansi	Broj problema
Minimalan broj ostvarenih iteracija *AGD* metodom	320
Minimalan broj ostvarenih iteracija *GD* metodom	36
Jednak broj iteracija ostvarenih *AGD* i *GD* metodama	16
Minimalan broj ostvarenih evaluacija funkcija *AGD* metodom	322
Minimalan broj ostvarenih evaluacija funkcija *GD* metodom	18
Minimalno CPU vreme postignuto *AGD* metodom	325
Minimalno CPU vreme postignuto *GD* metodom	28
Jednako CPU vreme postignuto *AGD* i *GD* metodama	13

Nesumnjivi značaj *AGD* metoda u odnosu na klasični metod pada po gradijentu ogleda se u drastičnom smanjenju potrebnih iteracija i broja evaluacija funkcija kao i potrebnog procesorskog vremena. Pored ovih izuzetnih rezultata ostvarenih primenom Andrejevog ubrzanog gradijentnog metoda, možda je još veći doprinos u samom detektovanju i načinu definisanja ubrzavajuće komponente θ_k. Ovakav pristup dao je jasnu smernicu i motivaciju mnogim autorima za nastavak proučavanja gradijentnih metoda sa ubrzavajućim karakteristikama, [4, 1, 6, 7, 8].

3.5 Ubrzani gradijentni SM metod

U radu *Accelerated gradient descent methods with line search*, [1], autori Predrag Stanimirović i Marko Miladinović, definišu novi ubrzani gradijentni metod. Svojstvo ubrzanosti je definisano parametrom ubrzanja γ_k čije je definisanje proisteklo iz ideja rada [3] opisanog u prethodnom poglavlju. Naime, istraživanja su potvrdila činjenicu da veličina koju je najpre odredio Nekulai Andrei u [3] iz Tejlorovog razvoja drugog reda objektne funkcije, korišćenjem odgovarajuće aproksimacije Hesijana, ima ubrzavajuća svojstva. Ovu osobinu su eksperimenalno dokazali i autori u [6] numeričkim testranjem i upoređivanjem metoda sa i bez ubrzavajućeg parametra γ_k. Tom prilikom dobijeni rezultati su nedvosmisleno potvrdili ubrzvajuće karakteristike gradijentnih metoda koji u svojim formulacijama kao dodatni multiplikativni faktor gradijenta i iterativnog koraka imaju ovako određenu veličinu γ_k. Pri tom se pretpostavlja da je iterativni korak određen nekom od procedura linijskog traženja. Time je i potvrđena ispravnost definisanja takozvane *klase ubrzanih gradijentnih metoda* od strane autora u [1], tj. onih metoda koji u svojim formulacijama sadrže ubrzavajući parametar γ_k.

Polazeći od uobičajne iteratine šeme

$$x_{k+1} = x_k + t_k d_k, \quad k = 0, 1 \dots \tag{3.40}$$

autori u [1] za polazište svog istraživanja uzimaju opšti algoritam metoda sa linijskim traženjem, Algoritam 2.1.1, ilustrovanog u Poglavlju 2.1. Dakle, dve osnovne pretpostavke koje se odnosne na ključne veličine u iteraciji (3.40) su:

- vektor traženja d_k ima smer pada;
- iterativni korak t_k je određen tehnikom linijskg traženja.

Istraživanja predstavljena u [1] nadovezuju se na rezultate ostvarene Andrejevom iteracijom iz [3]

$$x_{k+1} = x_k - \theta_k t_k g_k, \tag{3.41}$$

u kojoj je sa $\theta_k > 0$ označen faktor ubrzanja. Metod (3.41) se upoređuje sa opštom iteracijom koja formuliše modifikovan Njutnov metod [23]

$$x_{k+1} = x_k - t_k S_k g_k, \tag{3.42}$$

U relaciji (3.42) aproksimaciona matrica inverza Hesijana S_k jeste pozitivno definitna i ne mora nužno da zadovoljava kvazi-Njutnovu jedačinu. Za razliku od Andrejevog $\theta_k > 0$ autori u [1] faktor ubrzanja određuju iz iteracije (3.42), uz pretpostavku da je aproksimacija inverza Hesijana definisana sa

$$S_k = \gamma_k^{-1} I, \quad \gamma_k \in \mathbb{R}. \tag{3.43}$$

Time se modifikovana Njutnova iteracija (3.42) transformiše u

$$x_{k+1} = x_k - t_k \gamma_k^{-1} g_k, \tag{3.44}$$

Primenom Tejlorovog razvoja na šemu (3.44) dobija se relacija

$$f(x_{k+1}) = f(x_k) - t_k g_k^T \gamma_k^{-1} g_k + \frac{1}{2} t_k^2 (\gamma_k^{-1} g_k)^T \nabla^2 f(\xi) \gamma_k^{-1} g_k, \tag{3.45}$$

Ovde $\xi \in [x_k, x_{k+1}]$ predstavlja tačku

$$\xi = x_k + \alpha(x_{k+1} - x_k) = x_k - \alpha t_k \gamma_k^{-1} g_k, \; 0 \le \alpha \le 1. \tag{3.46}$$

Praktično, za tačku ξ prihvatljiv je izbor $\xi = x_{k+1}$ ukoliko se odabere $\alpha = 1$ obzirom da je udaljenost između tačaka x_k i x_{k+1} dovoljno mala. Onda je logičan izbor za aproksimaciju Hesijana u ovoj tački upravo

$$\nabla^2 f(\xi) = \gamma_{k+1} I. \tag{3.47}$$

Zamenom (3.47) u (3.45) dobija veličina parametra *SM* metoda

$$\gamma_{k+1} = 2\gamma_k \frac{\gamma_k [f(x_{k+1}) - f(x_k)] + t_k \|g_k\|^2}{t_k^2 \|g_k\|^2}. \tag{3.48}$$

Neophodan uslov koji ovako definisan parametar ubrzanja mora da zadovolji je njegova pozitivnost, $\gamma_{k+1} > 0$. Ovaj uslov proističe iz neophonog i dovoljnog uslova optimalnosti drugog reda. Neophodan uslov drugog reda potvrđuje da ukoliko je x_k tačka lokalnog minimuma glatke funkcije onda njen gradijent teži nuli i matrica Hesijana je pozitivno semidefinitna. Sa druge strane, prema dovoljnom uslovu drugog reda ukoliko gradijent objektne funkcije u nekoj tački teži nuli i ukoliko je njen Hesijan pozitivno definitan, tada ta funkcija dostiže minimum u toj tački. Dakle, zadovoljenje neophodnog i dovoljnog uslova optimalnosti zavisi od pozitivnosti parametra γ_{k+1}. U slučaju da se dogodi da je $\gamma_{k+1} < 0$, autori jednostavno dodeljuju vrednost $\gamma_{k+1} = 1$. Objašnjenje za ovakav izbor je sledeće: ukoliko se dogodi da Hesijan G_k nije pozitivno definitan onda se izborom $\gamma_{k+1} = 1$ postiže da se vrednost funkcije u narednoj iterativnoj tački x_{k+2} izračunava uobičajnim metodom pada po gradijentu: $x_{k+2} = x_{k+1} - t_{k+1}g_{k+1}$.

Da bi se odredila vrednost funkcije u narednoj iterativnoj tački $x_{k+2} = x_{k+1} - t_{k+1}\gamma_{k+1}^{-1}g_{k+1}$, pristupa se izračunavanju veličine iterativnog koraka t_{k+1} pomoću Armijeve backtracking procedure. Za pokretanje ove procedure potrebno je odrediti inicijalnu veličinu koraka, odnosno njenu gornju granicu, pa se iz tog razloga analizira funkcija

$$\Phi_{k+1}(t) = f(x_{k+1}) - tg_{k+1}^T\gamma_{k+1}^{-1}g_{k+1} + \frac{1}{2}t^2(\gamma_{k+1}^{-1}g_{k+1})^T\nabla^2 f(\xi)\gamma_{k+1}^{-1}g_{k+1},$$

pri čemu je $\xi \in [x_{k+1}, x_{k+2}]$, $t \geq 0$ and $\gamma_{k+1} > 0$. U slučaju kada je $\xi \approx x_{k+1}$, važi

$$\begin{aligned}
\Phi_{k+1}(t) &= f(x_{k+1}) - t\gamma_{k+1}^{-1}\|g_{k+1}\|^2 + \frac{1}{2}t^2\gamma_{k+1}\gamma_{k+1}^{-2}\|g_{k+1}\|^2 \\
&= f(x_{k+1}) - t\gamma_{k+1}^{-1}\|g_{k+1}\|^2 + \frac{1}{2}t^2\gamma_{k+1}^{-1}\|g_{k+1}\|^2.
\end{aligned}$$

Očigledno da je za obezbeđenu pozitivnost parametra γ_{k+1} funkcija Φ konveksna, i da važe jednakosti $\Phi_{k+1}(0) = f(x_{k+1})$ i $\Phi'_{k+1}(t) = (t-1)\gamma_{k+1}^{-1}\|g_{k+1}\|^2$. Takodje je poznato da funkcija Φ opada kada je $\Phi'_{k+1}(t) < 0$ što je ispunjeno za $t \in (0,1)$. Kako važi

$$\Phi'_{k+1}(t) = 0 \Leftrightarrow \bar{t}_{k+1} = 1, \tag{3.49}$$

zaključujemo da je minimum funkcije Φ_{k+1} postignut u tački $t = 1$ pa se ova vrednost inicijalnog koraka i uzima za startovanje backtracking procedure. Uzevši u obzir zamenu (3.47) upotrebljenu u (3.45) zaključuje se da je opadanje objektne funkcije f obezbeđeno ukoliko je $\gamma_k > 0$ u svakoj iteraciji.

Algoritam 3.5.1 Backtracking procedura sa početnim korakom $t = 1$.

Require: Objektna funkcija $f(x)$, vektor traženja d_k u tački x_k, brojevi $0 < \sigma < 0.5$ i $\beta \in (0,1)$.

1: $t = 1$.
2: Dok važi $f(x_k + td_k) > f(x_k) + \sigma t g_k^T d_k$, uzeti $t := t\beta$.
3: Izlaz $t_k = t$.

Obzirom na prethodno opisanu analizu, algoritam *SM* metoda je opisan sledećim koracima:

Algoritam 3.5.2 *SM* metod

Require: Objektna funkcija $f(x)$ odabrana inicijalna tačka $x_0 \in dom(f)$.

1: Postaviti $k = 0$, izračunati $f(x_0)$, $g_0 = \nabla f(x_0)$ i uzeti $\gamma_0 = 1$.
2: Ako je kriterijum završetka ispunjen, obustaviti iteracije; inače, preći na naradni korak.
3: (Backtracking:) Pronaći korak $t_k \in (0,1]$ pomoću Algoritama (3.5.1) za $d_k = -\gamma_k^{-1} g_k$.
4: Izračunati $x_{k+1} = x_k - t_k \gamma_k^{-1} g_k,\ f(x_{k+1})$ i $g_{k+1} = \nabla f(x_{k+1})$.
5: Odrediti skalarnu aproksimaciju γ_{k+1} Hesijana funkcije f u tački x_{k+1} pomoću relacije (3.48).
6: Ako je $\gamma_{k+1} < 0$, onda je take $\gamma_{k+1} = 1$.
7: Postaviti $k := k+1$, preći na korak 2.
8: Izlaz x_{k+1} i $f(x_{k+1})$.

U analizi konvergencije konstruisanog *SM* metoda, autori najpre razmatraju skup uniformno konveksnih funkcija. Sledećom lemom određuje se veličina opadanja vrednosti objektne funkcije u svakoj iteraciji a Teorema 3.2 potvrđuje linearnu konvergenciju *SM* algoritma.

Lema 3.2. *Za dva puta neprekidno diferencijabilnu i uniformno konveksnu funkciju f na $\mathbb{R}^n$ i niz $\{x_k\}$ generisan Algoritmom (3.5.2) važi sledeća nejednakost*

$$f(x_k) - f(x_{k+1}) \geq \mu \|g_k\|^2, \tag{3.50}$$

pri čemu je

$$\mu = \min\left\{\frac{\sigma}{M}, \frac{\sigma(1-\sigma)}{L}\beta\right\}. \tag{3.51}$$

Teorema 3.2. *Neka je f dva puta neprekidno diferencijabilna i uniformno konveksna funkcija na $\mathbb{R}^n$ i neka je niz $\{x_k\}$ generisan Algoritmom (3.5.2). Tada važi*

$$\lim_{k\to\infty} \|g_k\| = 0, \tag{3.52}$$

i niz $\{x_k\}$ konvergira ka x^* bar linearno.

Autori u [1] su zatim analizirali konvergentna svojstva *SM* metoda na podskupu striktno konveksnih kvadratnih funkcija na kome su funkcije date u sledećem opštem obliku

$$f(x) = \frac{1}{2}x^T A x - b^T x. \tag{3.53}$$

U prethodnoj relaciji se pretpostavlja da je kvadratna matrica A reda n realna, simetrična i pozitivno definitna a da je vektor $b \in \mathbb{R}^n$. U analizi konvergecije na ovom skupu funkcija korišćena je sledeća pretpostavka Moline i Rajdana iz [16] o odnosu najmanje, λ_1, i najveće, λ_n, sopstvene vrednosti matrice A

$$\lambda_n < 2\lambda_1.$$

Lema 3.3. *Neka je na striktno konveksnu kvadratnu funkciju f definisanu relacijom* (3.53) *primenjen gradijentni metod* (3.44) *u kome su parametri γ_k i t_k definisani redom relacijom* (3.48) *i Algoritmom* (3.5.2). *Tada važe sledeće nejednakosti*

$$\lambda_1 \leq \frac{\gamma_{k+1}}{t_{k+1}} \leq \frac{2\lambda_n}{\sigma}, \quad k \in \mathbb{N}, \tag{3.54}$$

pri čemu su λ_1 i λ_n najmanja i najveća sopstvena vrednost matrice A respektivno.

Teorema 3.3. *Neka je gradijentni metod* (3.44) *primenjen na striktno konveksne kvadratne funkcije definisane izrazom* (3.53). *Neka pri tom najmanja i najveća sopstvena vrednost matrice A ispunjavaju uslov $\lambda_n < 2\lambda_1$. Tada važi nejednakost*

$$\left(d_i^{k+1}\right)^2 \leq \delta^2 \left(d_i^k\right)^2 \tag{3.55}$$

pri čemu je

$$\delta = \max\left\{1 - \frac{\sigma\lambda_1}{2\lambda_n}, \frac{\lambda_n}{\lambda_1} - 1\right\}, \tag{3.56}$$

i takođe važi

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{3.57}$$

Dokazi navedih tvrđenja se mogu naći u [1].

Numerička testiranja predstavljena u radu [1], baziraju se na poređenju konstrusisanog *SM* algoritma, Andrejevog *AGD* metoda prezentovanog u [3] i klasičnog gradijentnog metoda pada po gradijentu, *GD* metoda. Ipak, osnovni cilj prikazanih testranja je potvrda boljih performansi *SM* metoda u odnosu na *AGD* metod pošto je prestižnost *AGD* iteracije u odnosu na *GD* već pokazana u [3]. Odabrani skup broja promenljivih odlikuje se većim vrednostima od onih koji je koristio Andrej u [3]. Za razliku od Andrejevog izbora $n = 100, 200, \ldots, 1,000$, u [1] je korišćen sledeći izbor broja promenljivih: 100, 500, 1000, 2000, 3000, 5000, 7000, 8000, 10000 i 15000. Korišćena su dva kriterijuma završetka testiranja za sva tri testirana metoda

$$\|g_k\| \leq 10^{-6} \quad \text{i} \quad \frac{|f(x_{k+1}) - f(x_k)|}{1 + |f(x_k)|} \leq 10^{-16}.$$

U Algoritmu (3.5.1) backtracking procedure korišćene su vrednosti $\sigma = 0.0001$ i $\beta = 0.8$ Prćena su tri svojstva svakog od testiranih metoda: broj iterativnih koraka, CPU vreme u sekundama i broj evaluacija testirane funkcije. Testovi urađeni sa 30 test funkcija predloženih u [34] pokazali su da *SM* daje bolje vrednosti u 20 slučaja dok *AGD* u 10 po pitanju broja iteracija. U pogledu vremenskog trajanja testiranja i broja evaluacija funkcije, *SM* je brži od *AGD* u 23 slučaja. Ove rezultate potvrđuje naredna Tabela 3.8

Tabela 3.8. *Prikaz numeričkih rezultata* SM, AGD *i* GD *metoda kod* 30 *test funkcija.*

Test function	Br. iteracija			CPU vreme			Br. evaluacija funkc.		
	SM	AGD	GD	SM	AGD	GD	SM	AGD	GD
Extended Rosenbrock	1735	86574	205737	15.1	3161.5	5942.0	14183	2323901	6003899
Extended Penalty	547	276	1196	5.1	10.0	58.6	2872	7643	44104
Perturbed quadratic	79429	351987	384579	681.1	17132.7	29539.9	445587	13882174	16610440
Raydan-1	15117	21718	54183	50.2	190.1	895.0	82001	432311	1471691
Raydan-2	90	40	60	0.2	0.2	0.2	190	90	130
Diagonal 1	7557	38454	35025	29.2	881.3	866.5	40502	1319921	1327459
Diagonal 3	8664	107944	55385	56.3	4485.0	2679.3	47165	3855094	2083028
Hager	729	2410	3055	3.2	31.4	39.4	3474	36465	54374
Generalized Tridiagonal - 1	278	570	662	1.6	10.9	15.2	952	7835	10948
Extended Tridiagonal - 1	4320	3322	1250746	40.4	18.7	6910.9	38446	13318	6372398
Extended Three Exponï£¡	141	398	1874	0.6	2.4	12.0	700	3324	19117
Diagonal 4	80	100	8892	0.4	0.8	117.6	530	1110	173914
Diagonal 5	60	30	40	0.3	0.2	0.2	130	70	90
Extended Himmelblau	164	335	1355	1.0	7.9	32.9	558	5835	24705
Quadr. Diag. Perturbed	31157	2005445	891082	238.5	63203.8	31549.2	316264	72179199	35601182
Quadratic QF1	43601	261485	211981	287.3	47169.1	33466.7	245060	9213885	7997891
Extended Quad. Penalty QP1	235	191	567	5.9	4.4	18.2	2398	2305	10748
Extended Quad. Penalty QP2	2232	247742	134484	19.7	12493.2	3281.2	16179	6172728	3878140
Quadratic QF2	62988	99547	309459	796.9	7164.1	31029.3	347920	3976538	13972221
Extended EP1	63	40	501	0.8	1.0	15.1	584	816	13776
Extended Tridiagonal - 2	438	1433	1153	2.3	5.2	5.7	2429	7058	9504
Arwhead	256	684	43224	7.9	16.0	5646.4	4325	17997	1920203

Test function	Br. iteracija			CPU vreme			Br. evaluacija funkc.		
	SM	AGD	GD	SM	AGD	GD	SM	AGD	GD
Almost Perturbed Quadratic	43229	253483	200957	291.4	49128.1	36337.1	244132	9689916	8201237
Engval1	319	701	573	6.0	13.1	15.6	2601	6300	8724
Quartc	244	144	478799	0.6	0.5	1306.1	538	338	957648
Diagonal 6	90	40	60	0.2	0.1	0.1	216	90	130
Generalized quartic	157	158	1377	1.0	1.7	42.1	423	715	17683
Diagonal 7	90	60	544	0.3	0.4	3.6	223	284	3313
Diagonal 8	98	50	584	0.4	0.4	4.7	303	256	3840
Diagonal 9	12556	312344	211607	28.0	4814.4	1716.6	77830	12519797	9152384

Kako bi potvrdili dominantnost *SM* šeme u odnosu na druga dva poredbena algoritma, autori su izvršili testiranja nad dodatnih pet funkcija sa izborom manjih vrednosti broja promenljivih:$100, 200, \ldots, 1000$. Izabrana je manja dimenzija ovih problema iz razloga što metodi *AGD* a posebno *GD* proizvode dosta veliki broj iteraticija a samim tim je vreme potrebno za njihovo testiranje dosta duže. Rezulati dopunskih testiranja su ilustrovani u Tabeli 3.9.

Tabela 3.9. *5 dodatnih numeričkih ekperimenata*

Test function	Br. iteracija			CPU vreme			Br. evaluacija funkc.		
	SM	AGD	GD	SM	AGD	GD	SM	AGD	GD
Extended White & Holst	2962	621538	574171	3.1	1953.6	1918.3	26321	18638690	18808384
Tridia	56309	447900	346994	24.0	1629.1	1381.0	264738	17073495	14102900
LIARWHD	3841	64754	166334	7.1	423.6	1485.6	28819	2139319	6193310
DIXON3DQ	1123659	6543592	6493407	656.2	4032.8	5019.3	6544136	39261562	53196428
BIGGSB1	1015243	6001285	5959373	582.8	3659.1	4833.6	5917974	36007720	48821414

Konačnu potvrdu daleko boljih rezultata *SM* metoda po pitanju sva tri analizirana svojstva, u poređenju sa *AGD* i *GD* metodima ilustruje tabela prosečnih vrednosti, Tabela 3.10. Iz ovog prikaza se može utvrditi da *SM* algoritam ima oko 7 puta manje iteracija u odnosu na *AGD* i *GD* i čak 16 puta manji broj evaluacija funkcije. Što se tiče prosečnog vremena potrebnog za predložena testiranja, zaključuje se da je *SM* metod skoro 60 puta brži od druga dva.

Tabela 3.10. *Prosečne vrednosti numeričkih rezultata za* 35 *u ukupno* 350 *testiranja.*

Prosečne vrednosti	SM	AGD	GD
Broj iteracija	7196.22	49933.64	51514.34
CPU vreme (sec)	10.99	633.28	589.10
Broj evaluacija funkcije	42059.15	710851.71	734478.16

Glava 4

Dvokoračni i dvosmerni gradijentni metodi sa faktorom ubrzanja

PREDSTOJEĆA glava sadrži ciljne rezultate ostvarene urađenim istraživanjima a u svrsi izrade ove doktorske disertacije. Koristeći poznate osobine i karakteristike gradijentnih metoda, hronološki opisanih u prethodnim glavama, shodno definisanoj temi ovog rada postojeća znanja su primenjena u konstruisanju novih gradijentnih metoda specifičnih formi. Jedna od osnovnih osobenosti novo postavljenih metoda, opisanih u ovoj glavi, jeste ta što se u njihovim formulacijama javljaju ili dva vektora pravca traženja ili dve različite vrednosti iterativnog koraka. Ideja za definisanje dvosmernih i dvokoračnih iterativnih modela za bezuslovnu optimizaciju javila se prirodno obzirom na krucijalnu bitnost veličine vektora pravca i veličine iterativnog koraka kod svakog gradijentnog iterativnog metoda.

U konstruisanju dvosmernih i dvokoračnih metoda u cilju određivanja veličine iterativnog koraka korišćena je tehnika netačnog linijskog traženja, tačnije Armijeva backtracking procedura. Ovaj izbor traženja primenjen je kako kod dvosmernog tako i kod dvokoračnog modela.

Prikazana istraživanja zasnovana su na pristupu ubrzanja gradijentnih metoda, upotrebom dobro definisanog multiplikativnog parametra ubrzanja od kojih su neki predloženi u [3, 1, 4, 5]. Iz tog razloga definisani metodi su naznačeni kao *ubrzani* gradijentni metodi i mogu se svrstati u klasu ubrzanih gradijentnih metoda detektovanu u [1].

U prvom delu istraživnaja fokus je stavljen na pravac vektora traženja ili, preciznije rečeno, na pravce vektora pretraživanaja. Ispitivano je ponašanje iterativnog gradijentnog metoda bezuslovne optimizacije koji je definisan na vrednostima dva različita vektora traženja. Jedan od predloženih vektora pretraživanja ima pravac kolinearan pravcu gradijenta i smer jednak smeru negativnog gradijenta. Skalar kolinearnosti je pri tom određen tako da predstavlja parametar ubrzanja ovog metoda. Drugi vektor traženja je definisan modifikacijom ideje za izbor drugog vektora traženja iterativnog metoda bezuslovne optimizacije opisane u [2]. Najvažniji pomak u pogledu poboljšanja performansi gradijentnih metoda koji je postignut konstruisanjem dvosmerne šeme ogleda se u značajnom smanjenju broja iteracija.

Naredni fokus je stavljen na definisanje iteracije sa dve različite veličine iterativnog koraka. U tu svrhu korišćene su dve backtracking procedure sa različitim početnim parametrima. U dvokoračnoj šemi figuriše samo jedan vektor traženja u pravcu negativnog gradijenta. Slično dvosmernom algoritmu, i kod dvokoračnog je određen parametar ubrzanja koji je korišćen kao multiplikativni faktor odnosno faktor kolinearnosti gradijentnog vektora. Definisani dvokoračni metod pokazao je višestruko poboljšanje svih praćenih karakteristika: broj iteracija, CPU vreme i broj određivanja vrednosti funkcija.

Dosta se pitanja nametnulo prilikom istraživanja opisanih u narednim poglavljima. Samo neka su: kako izbor dva različita vektora traženja u jednom gradijentnom iterativnom metodu, i analogno dva različita iterativna koraka, utiče na karaketristike tako definisanih metoda; da li su ovakvi izbori poboljšali performanse gradijentnih iterativnih metoda bezuslovne optimzacije; uporedivši ovakve dve koncepcije, dvosmerne i dvokoračne, koja od ta dva pristupa daje generalno bolje rezultate. Odgovori na ova i mnoga druga pitanja vezanih za konstruisane dvosmerne i dvokoračne metode dati su upravo u ovoj glavi disertacije.

U Poglavlju 4.1 je najpre dat kraći opis metoda predstavljneog u [2]. Na osnovu predloženih ideja, koje se odnose generalno na nediferencijabilne slučajeve, postavljena je diferencijabilna varijanta na kojoj se temelji definisanje dvosmernog gradijentnog metoda iz [6] a koji je predstavljen u Poglavlju 4.2. Dvokoračni gradijentni algoritam iz [7] opisan je u Poglavlju 4.3. U Poglavlju 4.4 ove glave opisana je transformacija dvokoračnog metoda pod određenim uslovom postavljenim za veličine iterativnih koraka. Na samom kraju poslednje glave data su zaključna razmatranja i predložene nove ideje za dalja istraživanja vezana za dvosmerne i dvokoračne iteracije.

4.1 Višekoračni algoritam pretraživanja po krivoj u nelinearnoj optimizaciji -nediferencijabilan konveksan slučaj

U ovom poglavlju je dat jedan kraći opis algoritma nelinearne optimizacije opisanog u [2]. Autori u istraživanjima iz [2] razmatraju iteraciju oblika

$$x_{k+1} = x_k + \alpha_k s_k + \alpha_k^2 d_k, \tag{4.1}$$

za rešavanje minimizacije nelinearnih problema

$$\min f(x), \ \ x \in \mathbb{R}^n, \tag{4.2}$$

u opštem slučaju nediferencijabilne konveksne funkcije $f : \mathbb{R}^n \to \mathbb{R} \bigcup \{+\infty\}$.

U pomenutom radu korišćena je regularizacija Moro-Josida (*Moreau-Yosida regularization*) kojom se od date funkcije f dobija nova funkcija sa istim skupom minimuma koja pri tom ima Lipšic neprekidan gradijent i u slučaju kada je polazna funkcija f nediferencijabilna. U cilju dobrog definisanja algoritma nelinearne optimizacije za slučaj nediferencijabilne konveksne funkcije upotrebljen je takođe Dinijev drugi izvod u pravcu. Kako je predmet ove disertacije vezan za gradijentne metode a samim tim i za diferencijabilne funkcije, u daljem se navode samo osnovne definicije regularizacije Moro-Josida i Dinijevog izvoda u pravcu drugog reda koji su neophodni za postavljanje algoritma predstavljenog u [2]. Dokazi relevatnih tvrđenja se mogu naći u [2, 58, 59, 60, 61]. Takođe će biti formulisana važnija tvrđenja koja govore o konvergenciji ovog algoritma.

Definicija 4.1. *Neka je $f : \mathbb{R}^n \to \mathbb{R} \bigcup \{+\infty\}$ zatvorena konveksna funkcija. Moro-Josida regularizacija date funkcije f u nekoj M-metrici, označenoj sa F, definiše se na sledeći način:*

$$F(x) := \min_{y \in \mathbb{R}^n} \{ f(y) + \frac{1}{2} \|y - x\|_M^2 \} \tag{4.3}$$

Definicija 4.2. *Minimum funkcije F, u oznaci $p(x)$*

$$p(x) := \arg \min_{y \in \mathbb{R}^n} \{ f(y) + \frac{1}{2} \|y - x\|_M^2 \} \tag{4.4}$$

naziva se proksimalna tačka tačke x.

Propozicija 4.1. *Funkcija F definisana izrazom* (4.3) *je uvek diferencijabilna.*

Teorema 4.1. *Sledeća tvrđenja su ekvivalentna:*

- *x minimizuje f;*
- *$p(x) = x$;*
- *$\nabla F(x) = 0$;*
- *x minimizuje F;*
- *$f(p(x)) = f(x)$;*
- *$F(x) = f(x)$.*

Definicija 4.3. *Dinijev gornji izvod drugog reda funkcije $f \in LC^1$ u tački $x \in \mathbb{R}^n$ u pravcu $d \in \mathbb{R}^n$ je definisan sledećom jednakošću*

$$f_D''(x,d) = \limsup_{\alpha \downarrow 0} \frac{[\nabla f(x+\alpha d) - \nabla f(x)]^T \cdot d}{\alpha}. \tag{4.5}$$

Obzirom da je regularizacija Moro-Josida zatvorene konveksne funkcije f funkcija iz prostora LC^1, može se razmatrati Dinijev gornji izvod drugog reda u tački $x \in \mathbb{R}^n$ u pravcu $d \in \mathbb{R}^n$.

Lema 4.1. *Neka je $f : \mathbb{R}^n \to \mathbb{R}$ zatvorena konveksna funkcija i neka je F njena regularizacija Moro-Josida. Onda važi:*

- *$F_D''(x_k, kd) = k^2 F_D''(x_k, d)$;*
- *$F_D''(x_k, d_1 + d_2) \leq 2\left(F_D''(x_k, d_1) + F_D''(x_k, d_2)\right)$*
- *$|F_D''(x_k, d)| \leq K \cdot \|d\|^2$, gde je K neka konstanta.*

Na postavljenim teorijskim osnovama koje su navedene u skraćenom obliku, autori u [2] dalje predstavljaju algoritam za rešavanje problema (4.2) u slučaju nediferencijabilne konveksne funkcije f. Pri tome se pretpostavlja da je u svakoj tački $x \in \mathbb{R}^n$ moguće izračunati vrednosti

$f(x)$, $F(x)$, $\nabla F(x)$ i $f''_D(x,d)$ za dati pravac $d \in \mathbb{R}^n$. Da bi se odredile sve potrebne veličine u cilju formulisanja algoritma, u svakoj iteraciji posmatra se sledeći minimazacioni problem:

$$\min_{d\in\mathbb{R}^n} \Phi_k(d), \qquad \Phi_k(d) = \nabla F(x_k)^T d + \frac{1}{2} f''_D(x_k,d), \tag{4.6}$$

gde je sa $f''_D(x_k,d)$ označen Dinijev gornji izvod drugog reda u tački x_k u pravci d. Ovako definisanu pomoćnu funkciju $\Phi_k(d)$ autori su nazvali iteraciona funkcija. Dalje autori definišu tročlani skup $B_k = \{(x_i, f(x_i), g_i) | i \in I_k\}$ pri čemu je $I_k = \{1,2,\dots,k\}$ skup indeksa. Svaka tročlana vrednost iz B_k definiše jednu linearizaciju $f_i(x)$ objektne funkcije

$$f_i(x) = f(x_i) + g_i^T (x - x_i), \quad i \in I_k.$$

Kako za konveksnu funkciju f važi da je $f(x) = \max_{z\in\mathbb{R}^n}\{f(z) + g_i^T(x-z)\}$, za $g \in \partial f(z)$, može se smatrati da je funkcija

$$\hat{f}_k(x) = \max_{0\le i\le k} f_i(x) = \max_{0\le i\le k}\{f(x_i) + g_i^T (x - x_i)\}$$

dobra aproksimacija objektne funkcije f. Time su određene sve potrebne veličine za opis algoritam definisanog u [2]:

Algoritam 4.1.1 Algoritam iz [2]

Require: Objektna funkcija $f(x)$ odabrana inicijalna tačka $x_1 \in \mathbb{R}^n$ brojevi $0 < \rho < 1$, $0 < \sigma < 1$, ε i μ dovoljno mali realni pozitivni brojevi. Postaviti $k = 1$, $I_0 = \emptyset$, $B_0 = \emptyset$.

1: Za dato x_k izračunati $f_k = f(x_k)$ i $g_k = g(x_k)$. Postaviti $I_k = \{k\} \bigcup I_{k-1} \setminus S_k$, gde je $S_k = \{i \in I_{k-1} | \|x_i - x_k\| \ge \mu\}$. Postaviti $B_k = \{(x_i, f(x_i), g_i) | i \in I_k\}$.

2: Ako je $\|g_k\| \le \varepsilon$ onda zaustaviti algoritam; inače rešiti problem $\min \|\sum_{i\in\hat{I}_k} \lambda_i g_i\|$ tako da je $\sum_{i\in\hat{I}_k} \lambda_i = 1$, $\lambda_i \ge 0$; pri tom je $\hat{I}_k = \{i \in I_k | \hat{f}_k(x) = f_i(x)\}$, $\hat{f}_k(x) = \max_{0\le i\le k}\{f(x_i) + g_i^T (x - x_i)\}$ i $g \in \partial f(x_i)$, $i \in \hat{I}_k$. Označiti sa $\lambda_i^{(k)}$ rešenje prethodnog problema. Ako je $\|\sum_{i\in\hat{I}_k} \lambda_i^{(k)} g_i\| \le \varepsilon$ onda zustaviti algoritam. Inače preći na korak 3.

3: Postaviti $x_{k+1} = x_k + \alpha_k s_k(\alpha_k) + \alpha_k^2 d_k(\alpha_k)$, gde je α_k određen pravilom traženja po krivoj, $s_k(\alpha_k)$ i $d_k(\alpha_k)$ su definisani pravilima za određivanje pravaca vektora traženja s_k i d_k. Nadalje su veličine $s_k(\alpha_k)$, $d_k(\alpha_k)$ i $g(x_k)$ radi jednostavnijeg zapisa označene sa s_k, d_k i g_k respektivno.

4: Postaviti k:=k+1 i preći na korak 1.

Pravilo pretraživanja po krivoj: Odabrati $\alpha_k = q^{i(k)}, \quad 0 < q < 1$, pri čemu je $i(k)$ najmanji ceo broj iz skupa $\{0,1,2,\ldots,\}$ takav da važi

$$F(x_k) - F(x_k + q^{i(k)} s_k + q^{2i(k)} d_k) \geq \sigma \left(-q^{i(k)} g_k^T s_k + \frac{1}{2} q^{4i(k)} F_D''(x_k, d_k) \right). \tag{4.7}$$

Pravilo određivanja pravca vektora s_k:

$$s_k(\alpha) = \begin{cases} s_k^*, & \text{ako je } k \leq m-1 \\ \\ -\left[\left(1 - \sum_{i=2}^m \alpha^{i-1} p_k^i \right) g_k + \sum_{i=2}^m \alpha^{i-1} p_k^i s_{k-i+1} \right], & \text{ako je } k \geq m, \end{cases}$$

gde je $m = cardI_k$, $m > 1$,

$$p_k^i = \frac{\rho \|g_k\|^2}{(m-1)[\|g_k\|^2 + |g_k^T s_{k-i+1}|]}, \quad i = 2,3,\ldots,m$$

pri čemu je $s_k^* \neq 0$, $k \leq m-1$ je bilo koji vektor koji zadovoljava uslov pada $g_k^T s_k^* \leq 0$.

Pravilo određivanja pravca vektora d_k:

$$d_k(\alpha) = \begin{cases} d_k^*, & \text{if } k \leq m-1 \\ \\ \sum_{i=2}^m \alpha^{i-1} d_{k-i+1}^*, & \text{ako je } k \geq m, \end{cases}$$

gde je sa d_k^* označeno rešenje problema 4.6.

Za analizu konvergencije postavljenog algoritma potrebne su sledeće dve pretpostavke:

Uslov 1 Pretpostavlja se da postoje konstante $c_1 \geq c_2 > 0$ takve da važi $c_1 \|d\|^2 \leq F_D''(x_k, d) \leq c_2$ za svako $d \in \mathbb{R}^n$.

Uslov 2 $\|d\| = 1$ i $\|s_k\| = 1, \quad k = 0,1,\ldots$

Propozicija 4.2. *Ukoliko regularizacija Moro-Josida $F(\cdot)$ zatvorene konveksne funkcije zadovoljava Uslov 1, onda važi:*

1. *funkcija $F(\cdot)$ je uniformna i striktno konveksna;*

2. *skup* $L(x_0) = \{x \in \mathbb{R}^n | F(x) \leq F(x_0)\}$ *je kompaktan konveksan skup;*

3. *postoji jedinstvena tačka* x^* *za koju je* $F(x^*) = \min_{x \in L(x_0)} F(x)$.

Lema 4.2. *Važi ekvivalencija sledećih tvrđenja*

- $d = 0$ *je globalni optimum problema* (4.6)*;*
- 0 *je optimum funkcije iz* (4.6)*;*
- *za odgovarajuće* x_k *važi* $0 \in \partial f(x_k)$.

Lema 4.3. *Za* $\alpha \in [0,1]$ *i za sve* $k \geq m$ *ispunjeno je* $g_k^T s_k(\alpha) \leq -(1-\rho)\|g_k\|^2$.

Naredna teorema potvrđuje konvergenciju definisanog algoritma.

Teorema 4.2. *Neka je* f *zatvorena konveksna funkcija i* F *njena regularizacija Moro-Josida. Neka važe pretpostavke označene sa Uslov 1 i Uslov 2. Tada za bilo koju inicijalnu tačku* $x_0 \in \mathbb{R}^n$, $x_k \to \bar{x}$, $k \to \infty$, *pri čemu je* $\bar{x}$ *je jedinstvena minimalna tačka.*

Red konvergencije definiše sledeća teorema.

Teorema 4.3. *Pod pretpostavkama prethodne teoreme za niz* $\{x_k\}$ *generisan algoritmom 4.1.1 važi sledeća ocena*

$$F(x_n) - F(\bar{x}) \leq \mu_0 \left[1 + \frac{\mu_0}{\eta^2} \sum_{k=0}^{n-1} \frac{F(x_k) - F(x_{k+1})}{\|\nabla F(x_k)\|^2} \right]^{-1}, \quad n = 1, 2, \ldots$$

gde je $\mu_0 = F(x_0) - F(\bar{x})$ *i* $diam L(x_0) = \eta < \infty$.

Napomena 4.1. *Činjenica da je* $diam L(x_0) = \eta < \infty$ *direktno sledi iz Propozicije 4.2 obzirom da se ovom propozicijom potvrđuje ograničenost skupa* $L(x_0)$.

4.2 Dvosmerni ubrzani gradijentni metod

Izstraživanja opisana u radovima [2] i [1] su svakako bila od najvećeg motivacijskog značenja u konstruisanju dvosmerne šeme sa ubrzanim karakteristikama opisanim u [6]. Kako je predmet istraživanja ove doktorske disertacije u vezi sa gradijentnim metodima prvog reda javila se ideja o prilagođavanju zanimljivog nediferencijabilnog modela pretraživanja po krivoj predstavljenog u [2] diferencijabilnom slučaju. Pri tom se takođe pretpostavilo da bi uvođenje multiplikativnog parametra, koji generalno pospešuje ubrzana svojstva gradijentnih iteracija shodno istraživanjima iz [3, 1], dodatno poboljšalo karakteristike modela iz [2] modifikovanog za diferencijabilne funkcije. Sledeći ove pretpostavke, konstruisan je ubrzani dvosmerni gradijentni metod bezuslovne optimizacije koji je originalno nazvan ***A**ccelerated **D**ouble **D**irection method* ili u skraćenoj formi *ADD* metod. Definisana itertativna šema je implementirana. Numerička istraživanja pokazala su evidentan pomak u pogledu smanjenja broja iteracija u odnosu na relevantne gradijentne metode bezuslovne optimizacije.

Polazeći od iteracije (4.1) definisane u [2] koju radi preglednosti ponovo navodimo i u ovom poglavlju

$$x_{k+1} = x_k + \alpha_k s_k + \alpha_k^2 d_k,$$

i jasno definisanih algoritama za određivanje vektora pretraživanja s_k i d_k kao i veličine iterativnog koraka α_k, autori su u radu [6] izvršili modifikaciju pomenutih algoritama i prilagodili ih diferencijabilnim problemima.

Za izbor vektora s_k odabran je jednostavno vektor u pravcu negativnog gradijenta sa koeficijentom kolinearnosti $\gamma_k > 0$, odnosno

$$s_k = -\gamma_k^{-1} g_k. \tag{4.8}$$

Time je određivanje vektora s_k svedeno na izračunavanje pozitivnog realnog broja γ_k a polazna iteracija (4.1) postaje

$$x_{k+1} = x_k - \alpha_k \gamma_k^{-1} g_k + \alpha_k^2 d_k \tag{4.9}$$

U cilju definisanja vektora d_k za diferencijabilnu ciljnu funkciju po analogiji sa vektorom d_k, koji je određen posebnim pravilom navedenim u prethodnom poglavlju, bilo je potrebno prilagoditi dati problem diferencijabilnim uslovima. Podsetimo se najpre da funkcije iz pomoćnog minimizacionog problema 4.6:

$$\min_{d \in R^n} \Phi_k(d), \qquad \Phi_k(d) = \nabla F(x_k)^T d + \frac{1}{2} f_D^{''}(x_k, d),$$

$F(x_k)$ i $f_D^{''}(x_k, d)$, predstavljaju respektivno Moro-Josida regularizaciju nediferencijabilne objektne funkcije f i Dinijev gornji izvod drugog reda u tački x_k u pravcu d. Ove funkcije su zamenjene odgovarajućim veličinama prilagođenim diferencijabilnom slučaju: objektnom funkcijom f i aproksimacijom njenog Hesijana. Izbor aproksimacije Hesijana sličan je izboru u [1]

$$G_k^{-1} \approx S_k = -\gamma_k I, \quad \gamma_k > 0. \tag{4.10}$$

U prethodnoj relaciji sa S_k je označena apraksimacija inverza Hesijana u modifikovanoj Njutnovoj iteraciji opšte formulacije

$$x_{k+1} = x_k - t_k S_k g_k, \tag{4.11}$$

pri čemu je S_k pozitvno definitna matrica koja ne mora nužno zadovoljavati kvazi-Njutnovu jednačinu.

Pod postavljenim pretpostavkama, minimizacioni problem (4.6) koji se rešava u svakoj iteraciji dobija formulaciju

$$\min_{d \in R^n} \Phi_k(d), \qquad \Phi_k(d) = g_k^T d + \frac{1}{2} \gamma_{k+1} I.$$

Kako su određene sve veličine potrebne za definisanje vektora d_k, algoritam za njegovo izračunavanje je dat sledećim opisom:

Algoritam 4.2.1 Izračunavanje vektora pravca d_k.

Require: Iterativni korak α_k.

1: $\alpha = \alpha_k$.

2:

$$d_k(\alpha) = \begin{cases} d_k^*, & k \le m-1 \\ \sum_{i=2}^{m} \alpha^{i-1} d_{k-i+1}^*, & k \ge m, \end{cases}$$

gde je d_k^* rešenje problema $\min_{x\in\mathbb{R}} \Phi_k(d)$, pri čemu je $\Phi_k(d)$ definisano na sledeći način

$$\Phi_k(d) = g_k^T d + \frac{1}{2}\gamma_{k+1} I.$$

Računanje vrednosti iterativnog koraka α_k u šemi (4.9) suštinski se razlikuje od pravlila za određivanje veličine itrativnog koraka polazne iteracije (4.1) koje je ilustrovano u prethodnom poglavlju. Umesto njega, kod novodefinisane dvosmerne gradijentne iteracije odabrana je Armijeva backtracking procedura linijskog traženja sa početnim korakom $\alpha = 1$.

Algoritam 4.2.2 Izračunavanje iterativnog koraka α_k transformisanjem pravila pretraživanja po krivoj iz [2] na linijsko traženje definisano backtracing procedurom sa početnim $\alpha = 1$.

Require: Objektna funkcija $f(x)$, vektor traženja d_k u tački x_k i brojevi $0 < \sigma < 0.5$ i $\beta \in (\sigma, 1)$.

1: Postaviti $\alpha = 1$;

2: Dok važi $f(x_k + \alpha d_k) > f(x_k) + \sigma\alpha g_k^T d_k$ postaviti $\alpha := \alpha\beta$;

3: Izlazna veličina $\alpha_k = \alpha$.

Pre definisanja algoritamske šeme koja opisuje *ADD* metod, potrebno je odrediti realni pozitivni broj γ_k u iteraciiji (4.9). *ADD* metod je baziran na modifikovanom Njutnovom metodu opšte formulacije (4.11). Kako je realan broj γ_k^{-1} skalar u dijagonalnoj matrici S_k iz (4.10), koja predstavlja aproksimaciju inverza Hesijana objektne funkcije, ideja je zameniti vrednost Hesijana u Tejlorovom razvoju drugog reda ovom skalarnom aproksimacijom. Iz dobijenog izraza jednostavno se određuje veličina skalara γ_k u svakoj iteraciji. Dakle, kako je Tejlorov razvoj objektne funkcije f definisane izrazom (4.9) u tački x_{k+1}

$$f(x_{k+1}) \approx f(x_k) + g_k^T \left(\alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k\right) + \frac{1}{2}\left(\alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k\right)^T \nabla^2 f(\xi) \left(\alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k\right), \tag{4.12}$$

nadalje matricu Hesijana date funkcije $\nabla^2 f(\xi)$ zamenjujemo njenom skalrnom aproksimacijom $\gamma_{k+1} I$. U prethodnoj relaciji parametar ξ zadovoljava uslov

$$\xi \in [x_k, x_{k+1}], \quad \xi = x_k + \delta(x_{k+1} - x_k) = x_k + \delta\left(\alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k\right), \quad 0 \leq \delta \leq 1.$$

Nakon ove zamene Tejlorov razvoj (4.12) postaje

$$f(x_{k+1}) = f(x_k) + \alpha_k g_k^T \left(\alpha_k d_k - \gamma_k^{-1} g_k\right) + \frac{1}{2}\gamma_{k+1}\left(\alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k\right)^T \left(\alpha_k^2 d_k - \alpha_k \gamma_k^{-1} g_k\right). \tag{4.13}$$

Sada je iz (4.13) jednostavno odrediti parametar γ_{k+1} :

$$\gamma_{k+1} = 2\frac{f(x_{k+1}) - f(x_k) - \alpha_k g_k^T \left(\alpha_k d_k - \gamma_k^{-1} g_k\right)}{\left(\alpha_k d_k - \gamma_k^{-1} g_k\right)^T \left(\alpha_k d_k - \gamma_k^{-1} g_k\right)}. \tag{4.14}$$

Već je naglašena pretostavka $\gamma_{k+1} > 0$. Ovaj uslov zapravo garantuje ispunjenje neophodnog i potrebnog uslova optimalnosti drugog reda koji u protivnom, kada je $\gamma_{k+1} \leq 0$, nisu ispunjeni. U slučaju da se dogodi da vrednost parametra γ_{k+1} bude negativna uzima se vrednost $\gamma_{k+1} = 1$. Tada će naredna iterativna tačka x_{k+2} da se računa na sledeći način:

$$x_{k+2} = x_{k+1} - \alpha_{k+1} g_{k+1} + \alpha_{k+1}^2 d_{k+1} = x_{k+1} - \alpha_{k+1}(g_{k+1} - \alpha_{k+1} d_{k+1}).$$

Kako su definisani svi potrebni parametri, ubrzani dvosmerni metod je opisan narednim algoritmom.

Algoritam 4.2.3 Ubrzani dvosmerni *ADD* metod.

Require: $0 < \rho < 1, \quad 0 < \tau < 1, x_0$.

1: Postaviti $k = 0$, izračunati $f(x_0)$, g_0 i postaviti $\gamma_0 = 1$.
2: Ako je $\|g_k\| < \varepsilon$, preći na korak 9; inače nastaviti sa sledećim korakom;
3: Odrediti veličinu iterativnog koraka α_k primenom Algoritma 4.2.2.
4: Izračunati d_k pomoću Algoritma 4.2.1.
5: Izračunati x_{k+1} definisano relacijom (4.9), $f(x_{k+1})$ i g_{k+1}.
6: Odrediti skalar γ_{k+1} iz (4.14).
7: Ako je $\gamma_{k+1} < 0$, postaviti $\gamma_{k+1} = 1$.
8: Postaviti $k := k+1$, preći na korak 2.
9: Izlazne veličine x_{k+1} i $f(x_{k+1})$.

Može se napomenuti da je iteracija (4.9) uporediva sa uopštenom iterativnom šemom iz [22]. Naime, pravac traženja kod metoda predstavljenog u [22] je definisan kao linearna kombinacija vektora g_{k+1} i $x_{k+1} - x_k$. Slično, pravac pretraživanja u iteraciji (4.9) se definiše kao linearna kombinacija vektora $-g_k$ i vektora d_k određenog Algoritmom 4.2.1.

Sledi analiza konvergencije konstruisanog *ADD* metoda. Konvergencija je najpre utvrđena na skupu uniformno konveksnih funkcija a potom, uz određena ograničenja, i na podskupu striktno konveksnih funkcija. Najpre su navedene propozicija i lema iz [20, 62] koje govore o donjem ograničenju objektne funkcije.

Propozicija 4.3. [20, 62] *Neka je funkcija $f : \mathbb{R}^n \to \mathbb{R}$ dva puta neprekidno diferencijabilna i uniformno konveksna na $\mathbb{R}^n$. Tada važi:*

1. *funkcija f donju granicu na skupu $L_0 = \{x \in \mathbb{R}^n |\ f(x) \leq f(x_0)\}$, gde je $x_0 \in \mathbb{R}^n$ dopustiva tačka;*

2. *gradient g je Lipšic neprekidan na otvorenom konveksnom skupu B koji sadrži L_0, odnosno postoji konstanta $L > 0$ takva da je*

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \forall x, y \in B. \tag{4.15}$$

Lema 4.4. [20, 62] *Pod pretpostavkama Propozicije 4.3 postoje realni brojevi m i M za koje je ispunjeno*

$$0 < m \leq 1 \leq M, \tag{4.16}$$

tako da funkcija $f(x)$ ima jedinstvenu tačku minimuma x^* i pri tom važi:

$$m\|y\|^2 \leq y^T \nabla^2 f(x) y \leq M\|y\|^2, \quad \forall x, y \in \mathbb{R}^n; \tag{4.17}$$

$$\frac{1}{2}m\|x - x^*\|^2 \leq f(x) - f(x^*) \leq \frac{1}{2}M\|x - x^*\|^2, \quad \forall x \in \mathbb{R}^n; \tag{4.18}$$

$$m\|x - y\|^2 \leq (g(x) - g(y))^T (x - y) \leq M\|x - y\|^2, \quad \forall x, y \in \mathbb{R}^n. \tag{4.19}$$

Lema koja sledi govori o oceni vrednosti opadanja date uniformno konveksne funkcije prilikom svake iteracije. Dokaz leme 4.5 se može naći u [1] .

Lema 4.5. [1] *Za dva puta neprekidno diferencijabilnu i uniformno konveksnu funkciju f definisanu na $\mathbb{R}^n$, i za niz $\{x_k\}$ generisan Algoritmom 4.2.3 važi sledeća ocena*

$$f(x_k) - f(x_{k+1}) \geq \mu \|g_k\|^2, \tag{4.20}$$

gde je

$$\mu = \left\{ \frac{\sigma}{M}, \frac{\sigma(1-\sigma)}{L}\beta \right\}. \tag{4.21}$$

Dokaz. Postavljanjem $s_k = -\gamma_k^{-1} g_k$ umesto d_k u dokazu leme 4.2 u [1] direktno se potvrđuje nejednakost. □

Linearna konvergencija ubrzanog dvosmernog metoda za uniformno konveksne i striktno konveksne funkcije utvrđena je narednom teoremom. Kako je dokaz ove teoreme isti kao i dokaz Teoreme 4.1 u [1], ovde je izostavljen.

Teorema 4.4. [1] *Ako je funkcija f dva puta neprekidno diferencijabilna i uniformno konveksna na $\mathbb{R}^n$ a niz $\{x_k\}$ generisan Algoritmom 4.2.3, tada*

$$\lim_{k\to\infty} \|g_k\| = 0 \tag{4.22}$$

i niz $\{x_k\}$ konvergira ka x^* najmanje linearno.

Konvergenica *ADD* modela ispitana je i na skupu striktno konveksnih kvadratnih funkcija, opšteg oblika

$$f(x) = \frac{1}{2}x^T Ax - b^T x, \tag{4.23}$$

gde je sa A označena realna $n \times n$ simetrična pozitivno definitna matrica a vektor $b \in \mathbb{R}^n$. Generalno je dokazivanje konvergencije gradijentnih metoda dosta složeno i nestandardno. Iz tog razloga je analiziran i ovaj zaseban skup striktno konveksnih kvadratnih funkcija. Za potvrdu konvergencije u ovom izdvojenom slučaju bilo je potrebno postaviti dodatan uslov koji se odnosi na najmanju i najveću sopstvenu vrednost funkcije ovog skupa. U predstojećoj analizi biće korišćene neke poznate pretpostavke iz [5, 29, 16]. Preciznije, ako se sa $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ označe redom, počev od najmanje pa sve do najveće, sopstvene vrednosti date funkcije (4.23), tada se za najmanju i najveću sopstvenu vrednost uzima da važi dodatan uslov

$$\lambda_n < 2\lambda_1. \tag{4.24}$$

U radu [16] autori su pod pretpostavkom (4.24) ustanovili Q linearnu konvergenciju preuslovljenog *BB* metoda, tzv *PBB* metoda.

Lema 4.6. *Neka je f striktno konveksna kvadratna funkcija određena relacijom* (4.23) *pri čemu je $A \in \mathbb{R}^{n\times n}$ simetrična pozitivno definitna matrica čije su najmanja i najveća sopstvene vrednosti označene redom sa λ_1 i λ_n . Neka je za funkciju f primenjen metod 4.9 i parametri γ_k, α_k i d_k određeni redom relacijom* (4.14)*, Algoritmima 4.2.2 i 4.2.1. Tada važi:*

$$\lambda_1 \le \frac{\gamma_{k+1}}{\alpha_{k+1}} \le \frac{2\lambda_n}{\sigma}, \quad k \in \mathbb{N}. \tag{4.25}$$

Dokaz. Razlika vrednosti objektne striktno konveksne kvadratne funkcije određene relacijom (4.23) u tekućoj i prethodnoj iterativnoj tački iznosi

$$f(x_{k+1}) - f(x_k) = \frac{1}{2}x_{k+1}^T A x_{k+1} - b^T x_{k+1} - \frac{1}{2}x_k^T A x_k + b^T x_k. \tag{4.26}$$

Kako je $x_{k+1} = x_k - \alpha_k \gamma_k^{-1} g_k + \alpha_k^2 d_k$, sledi da je:

$$\begin{aligned} f(x_{k+1}) - f(x_k) &= \frac{1}{2}(x_k - \alpha_k\gamma_k^{-1}g_k + \alpha_k^2 d_k)^T A(x_k - \alpha_k\gamma_k^{-1}g_k + \alpha_k^2 d_k) - b^T(x_k - \alpha_k\gamma_k^{-1}g_k + \alpha_k^2 d_k) \\ &\quad - \frac{1}{2}x_k^T A x_k + b^T x_k \\ &= \frac{1}{2}x_k^T A x_k - \frac{1}{2}\alpha_k\gamma_k^{-1}g_k^T A x_k + \frac{1}{2}\alpha_k^2 d_k^T A x_k \quad - \frac{1}{2}\alpha_k\gamma_k^{-1}x_k^T A g_k + \frac{1}{2}\alpha_k^2\gamma_k^{-2}g_k^T A g_k \\ &\quad - \frac{1}{2}\alpha_k^3\gamma_k^{-1}d_k^T A g_k + \frac{1}{2}\alpha_k^2 x_k^T A d_k - \frac{1}{2}\alpha_k^3\gamma_k^{-1}g_k^T A d_k + \frac{1}{2}\alpha_k^4 d_k^T A d_k - b^T x_k \\ &\quad + \alpha_k\gamma_k^{-1}b^T g_k - \alpha_k^2 b^T d_k - \frac{1}{2}x_k^T A x_k + b^T x_k. \end{aligned}$$

Zamenom jednakosti $g_k = Ax_k - b$, prethodni izraz svodimo na

$$\begin{aligned} f(x_{k+1}) - f(x_k) &= -\alpha_k\gamma_k^{-1}x_k^T A g_k + \alpha_k^2 x_k^T A g_k - \alpha_k^3\gamma_k^{-1}d_k^T A g_k + \frac{1}{2}\alpha_k^2\gamma_k^{-2}g_k^T A g_k + \frac{1}{2}\alpha_k^4 d_k^T A d_k \\ &\quad + \alpha_k\gamma_k^{-1}b^T g_k - \alpha_k^2 b^T d_k. \end{aligned}$$

Daljim grupisanjem dobija se jednakost:

$$
\begin{aligned}
f(x_{k+1})-f(x_k)=\alpha_k\gamma_k^{-1}(b^Tg_k-x_k^TAg_k)+\alpha_k^2(x_k^TAd_k-b^Td_k)-\alpha_k^3\gamma_k^{-1}d_k^TAg_k\\
+\frac{1}{2}\alpha_k^2\gamma_k^{-2}g_k^TAg_k+\frac{1}{2}\alpha_k^4d_k^TAd_k
\end{aligned}
$$

odakle direktno sledi da je

$$
f(x_{k+1})-f(x_k)=-\alpha_k\gamma_k^{-1}g_k^Tg_k+\alpha_k^2g_k^Td_k-\alpha_k^3\gamma_k^{-1}d_k^TAg_k+\frac{1}{2}\alpha_k^2\gamma_k^{-2}g_k^TAg_k+\frac{1}{2}\alpha_k^4d_k^TAd_k.
$$

Iskoristivši svojstvo simetrije matrice A dobijamo

$$
\begin{aligned}
f(x_{k+1})-f(x_k)&=\alpha_kg_k^T(\alpha_kd_k-\gamma_k^{-1}g_k)-\frac{1}{2}\alpha_k^2\gamma_k^{-1}g_k^TA(\alpha_kd_k-\gamma_k^{-1}g_k)+\frac{1}{2}\alpha_k^3d_k^TA(\alpha_kd_k-\gamma_k^{-1}g_k)\\
&=\left(\alpha_kg_k^T-\frac{1}{2}\alpha_k^2\gamma_k^{-1}g_k^TA+\frac{1}{2}\alpha_k^3d_k^TA\right)\left(\alpha_kd_k-\gamma_k^{-1}g_k\right).
\end{aligned}
$$

Dobijenu jednakost koja definiše razliku vrednosti objektne funkcije u tekućoj i prethodnoj iterativnoj tački zamenimo u relaciji (4.14). Tada se za γ_{k+1} dobija

$$
\gamma_{k+1}=2\left[\frac{\left(\alpha_kg_k^T-\frac{1}{2}\alpha_k^2\gamma_k^{-1}g_k^TA+\frac{1}{2}\alpha_k^3d_k^TA\right)\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}{\alpha_k^2\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)^T\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}-\frac{\alpha_kg_k^T\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}{\alpha_k^2\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)^T\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}\right]
$$

iz čega sledi da je

$$
\gamma_{k+1}=2\frac{\left(\alpha_kg_k^T-\frac{1}{2}\alpha_k^2\gamma_k^{-1}g_k^TA+\frac{1}{2}\alpha_k^3d_k^TA-\alpha_kg_k^T\right)\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}{\alpha_k^2\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)^T\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}.
$$

Konačno,

$$
\gamma_{k+1}=2\cdot\frac{1}{2}\frac{\alpha_k^2\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)^TA\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)}{\alpha_k^2\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)^T\left(\alpha_kd_k-\gamma_k^{-1}g_k\right)},
$$

odnosno:

$$\gamma_{k+1} = \frac{\left(\alpha_k d_k - \gamma_k^{-1} g_k\right)^T A \left(\alpha_k d_k - \gamma_k^{-1} g_k\right)}{\left(\alpha_k d_k - \gamma_k^{-1} g_k\right)^T \left(\alpha_k d_k - \gamma_k^{-1} g_k\right)}. \tag{4.27}$$

Kako je A realna, simetrična i pozitivno definitna matrica i kako prethodna relacija za γ_{k+1} predstavlja Rejlijev količnik realne simetrične matrice A i vektora $\alpha_k d_k - \gamma_k^{-1} g_k$, zaključujemo da je

$$\lambda_1 \leq \gamma_{k+1} \leq \lambda_n, \quad k \in \mathbb{N}. \tag{4.28}$$

Iz činjenice da je $0 \leq \alpha_{k+1} \leq 1$ proizilazi leva strana nejednakosti (4.25). Desna strana iste nejednakosti sledi iz ocene

$$\alpha_k > \frac{\beta(1-\sigma)\gamma_k}{L}, \tag{4.29}$$

koja je dokazana u [1] , u lemi 4.2. Direktno se iz prethodne relacije dobija

$$\frac{\gamma_{k+1}}{\alpha_{k+1}} < \frac{L}{\beta(1-\sigma)}. \tag{4.30}$$

Uzevši u obzir simetričnost matrice A i činjenice da je $g(x) = A(x) - b$, zaključuje se da je

$$\|g(x) - g(y)\| = \|Ax - Ay\| = \|A(x-y)\| \leq \|A\|\|x-y\| = \lambda_n\|x-y\|. \tag{4.31}$$

Poslednja jednakost zapravo potvrđuje da najveća sopstvena vrednost matrice A, λ_n, ima svojstva Lipšicove konstante L. Obzirom da parametri backtracking procedure σ i β imaju sledeća ograničenja: $0 < \sigma < 0.5$ i $\beta \in (\sigma, 1)$ sledi da je

$$\frac{\gamma_{k+1}}{\alpha_{k+1}} < \frac{L}{\beta(1-\sigma)} = \frac{\lambda_n}{\beta(1-\sigma)} < \frac{2\lambda_n}{\sigma}$$

što završava dokaz postavljenog tvrđenja. $\square$

Sledeća teorema pod dodatnim uslovom (4.24) dokazuje konvergenciju *ADD* iteracije u slučaju striktno konveksnih kvadratnih funkcija.

Teorema 4.5. *Za striktno konveksnu kvadratnu funkciju f oblika* (4.23) *i gradijentni metod 4.9, pod dadatnom pretpostavkom* (4.24) *postavljenom nad najmanjom i najvećom sopstvenom vrednošću matrice A, važi da je*

$$\left(p_i^{k+1}\right)^2 \leq \delta^2 \left(p_i^k\right)^2 \quad i \quad \left(q_i^{k+1}\right)^2 \leq \lambda_n^2 \left(q_i^k\right)^2 \tag{4.32}$$

gde je

$$\delta = \max\left\{1 - \frac{\sigma\lambda_1}{2\lambda_n}, \frac{\lambda_n}{\lambda_1} - 1\right\}, \tag{4.33}$$

kao i da je

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{4.34}$$

Dokaz. Označimo sa $\{v_1, v_2, \ldots, v_n\}$ ortonormirane sopstvene vektore simetrične pozitivno definitne matrice A. Neka je$\{x_k\}$ niz vrednosti dobijenih primenom Algoritma 4.2.3. Za neko k i vrednost x_k važi $g_k = Ax_k - b$. Takođe važi i

$$g_k = \sum_{i=1}^{n} p_i^k v_i,\ d_k = \sum_{i=1}^{n} q_i^k v_i \tag{4.35}$$

za neke konstante $p_1^k, p_2^k, \ldots, p_n^k$ i $q_1^k, q_2^k, \ldots, q_n^k$.

Kako iz iteracije (4.9) sledi da je

$$g_{k+1} = A(x_k - \alpha_k\gamma_k^{-1} g_k + \alpha_k^2 d_k) - b = (I - \alpha_k\gamma_k^{-1}A)g_k + \alpha_k^2 A d_k, \tag{4.36}$$

primenivši reprezentaciju (4.35) dobijamo

$$\begin{aligned} g_{k+1} &= \sum_{i=1}^{n} p_i^{k+1} v_i = \sum_{i=1}^{n} \left((1 - \alpha_k\gamma_k^{-1}\lambda_i)p_i^k + \alpha_k^2\lambda_i q_i^k\right) v_i \\ &= \sum_{i=1}^{n} \left(1 - \alpha_k\gamma_k^{-1}\lambda_i\right) p_i^k v_i + \alpha_k^2 \sum_{i=1}^{n} \lambda_i q_i^k v_i. \end{aligned} \tag{4.37}$$

Da bi dokazali (4.32), dovoljno je pokazati da $\left|1 - \frac{\lambda_i}{\gamma_k\alpha_k^{-1}}\right| \leq \delta$ zato što je svakako $|\lambda_i| \leq \lambda_n$ ispunjeno za svako $i \in \{1, 2, \ldots, n\}$. Razmatramo dva slučaja. U prvom slučaju pretostavimo da je $\lambda_i \leq \frac{\gamma_k}{\alpha_k}$. Primenivši (4.25) zaključujemo:

$$1 > \frac{\lambda_i}{\gamma_k\alpha_k^{-1}} \geq \frac{\sigma\lambda_1}{2\lambda_n} \Longrightarrow 1 - \frac{\lambda_i}{\gamma_k\alpha_k^{-1}} \leq 1 - \frac{\sigma\lambda_1}{2\lambda_n} \leq \delta. \tag{4.38}$$

Razmotrimo sada drugi slučaj: $\frac{\gamma_k}{\alpha_k} < \lambda_i$. Kako važi

$$1 < \frac{\lambda_i}{\gamma_k\alpha_k^{-1}} \leq \frac{\lambda_n}{\lambda_1}, \tag{4.39}$$

zaključujemo da je

$$\left|1-\frac{\lambda_i}{\gamma_k\alpha_k^{-1}}\right| \leq \frac{\lambda_n}{\lambda_1}-1 \leq \delta. \tag{4.40}$$

Relaciju (4.34), dokazujemo pomoću reprezentacije (4.35):

$$\|g_k\|^2 = \sum_{i=1}^{n}(p_i^k)^2. \tag{4.41}$$

Imajući u vidu da je $0<\delta<1$ kao i dokazanu nejednakost (4.32) zaključujemo da važi jednakost (4.34). □

U nastavku sledi opis i prikaz numeričkih rezultata dobijenih testiranjem konstruisanog *ADD* metoda, *SM* metoda iz [1] i neubrzavajuće varijante *ADD* metoda - takozvanog *NADD* metoda. Kodovi korišćeni u testiranjima su napisani u C++ programskom jeziku a testiranja su sprovedena na računaru Workstation Intel Celeron 2.2 GHz. Dobijeni rezultati generalno pokazuju poboljšanje po pitanju smanjenja broja iteracija u odnosu na *SM* algoritam koji je opisan u Poglavlju 3.5. Sa ciljem potvrde ubrzanih karakteristika *ADD* metoda uslovljene dobrom aproksimacijom inverza Hesijana koja je određena parametrom γ_k, konstruisan je neubrzani pandam *ADD* algoritma bez pomenutog parametra i označen sa *NADD*.

Testiranja su izvršena nad 25 test funkcija iz [34]. Analiziran je broj potrebnih iteracija kod svih testiranih metoda. Nad svakom funkcijom urađeno je 10 numeričkih eksperimenata za različite vrednosti broja promenljivih. Odabran je skup velikih vrednosti broja promenljivih: 1000, 2000, 3000, 5000, 7000, 8000, 10000, 15000, 20000 i 30000. Na taj način postignuta je veća generalizacija u odnosu na rezultate istraživanja predstavljenih u [1] gde je za izbor broja promenljivih korišćen skup 100, 500, 1000, 2000, 3000, 5000, 7000, 8000, 10000 i 15000. U [1] numerički je argumentovano poboljšanje po pitanju smanjenja broja iteracija primenom *SM* algoritma u poređenju sa *AGD* i *GD* metodima. Kako je jedan od ciljeva u istraživanju iz [6] smanjenje broja iteracija, bilo je dovoljno za poredbeni metod odabrati *SM* metod.

Korišćen je isti kriterijum završetka kao u [1]

$$\|g_k\| \leq 10^{-6} \quad \texttt{i} \quad \frac{|f(x_{k+1})-f(x_k)|}{1+|f(x_k)|} \leq 10^{-16}.$$

Opšti rezultati pokazuju izuzetan napredak po pitanju smanjenja broja iteracija upotrebom *ADD* šeme. U poređenju sa *SM* algoritmom, *ADD* metodom je ostvaren manji broj iterativnih koraka kod 20 od 25 test funkcija i može se zaključiti da je *ADD* šema nesumnjivo efektivnija po pitanju pomenutnog svojstva.

Tabela 4.1. *Broj iterativnih koraka potrebnih primenom SM, ADD i NADD metoda u testiranju* 25 *test funkcija.*

Test funkcije	**SM**	**ADD**	**NADD**
Extended penalty	589	70	$> t_e$
Perturbed quadratic	111972	80	$> t_e$
Raydan 1	21125	86	$> t_e$
Diagonal 1	10417	81	$> t_e$
Diagonal 3	10574	80	$> t_e$
Generalized tridiagonal 1	278	110	$> t_e$
Extended tridiagonal 1	3560	120	$> t_e$
Extended three expon. Terms	164	100	3551
Diagonal 4	80	100	$> t_e$
Extended Himmelblau	168	100	$> t_e$
Quadr. Diag. Perturbed	53133	90	$> t_e$
Quadratic QF1	114510	80	$> t_e$
Extended quad. penalty QP1	224	90	$> t_e$
Extended quad. penalty QP2	162	80	$> t_e$
Quadratic QF2	118801	86	$> t_e$
Extended EP1	68	100	$> t_e$
Extended tridiagonal 2	584	120	$> t_e$
Arwhead	10	102	$> t_e$
Almost perturbed quadratic	110121	80	$> t_e$
Engval1	185	100	$> t_e$
Quartc	190	10	10
Generalized Quartic	156	100	3258
Diagonal 7	90	3287	$> t_e$
Diagonal 8	96	3225	$> t_e$
Diagonal 9	11235	80	$> t_e$

Shodno rezultatima iz Tabele 4.1 za 25 analiziranih funkcija i svih 250 urađenih testova, naredna Tabela 4.2 zapravo ilustruje činjenicu da je primenom *ADD* metoda potrebno oko 66 puta manje iteracija nego primenom *SM* metoda.

Tabela 4.2. *Prosečne vrednosti rezultata za* 25 *testiranih funkcija pri* 250 *numeričkih eksperimenata za svaki metod, SM i ADD, generisanih za velike vrednosti broja promenljivih:* $1000, 2000, 3000, 5000, 7000, 8000, 10000, 15000, 20000$ *i* 30000.

Prosečna vrednost	SM	ADD
Broj iteracija	22739.68	342.28

Značajno istraživanje opisano u [6] odnosi se na potvrdu ubrzanih karakteristika *ADD* metoda. Sa tim u vezi, dokazana su ubrzavajuća svojstva koja proističu od parametra γ_k. Dobijeni rezultati prikazani u Tabeli 4.3 opravdavaju izdvajanje klase ubrzanih metoda koja je detektovana u [1] . Pod klasom ubrzanih gradijentnih metoda autori u [1] podrazumevaju one gradijentne metoda koji u svojim formulacijama sadrže multiplikativni faktor, u [6] nazvan γ_k, koji je određen korišćenjem Tejlorovog razvoja. U cilju potvrde ubrzanih svojstava parametra γ_k konstruisana je neubrzavajuća varijanta *ADD* metoda, *NADD* metod. U formulaciji neubrzavajućeg pandama *ADD* metoda izostavljena je veličina γ_k, preciznije u svakoj iteraciji se uzima $\gamma_k = 1$:

$$x_{k+1} - x_k = \alpha_k^2 d_k - \alpha_k g_k. \tag{4.42}$$

Rezultati numeričkih eksperimenata opradvdali su atribut 'ubrzani' u nazivu parametra γ_k i istovremeno potvrdili da je aproksimacija Hesijana definisana relacijom

$$S_k = \gamma_k^{-1} I, \quad \gamma_k > 0$$

dobro postavljena. Kako su testiranja *NADD* metodom trajala neprihvatljivo dugo, uvedena je konstanta t_e kao ograničenje vremenskog trajanja svakog eksperimenta. Vrednost konstante t_e određena je na sledeći način: testiranje *ADD* metodom koje najduže traje je testiranje Diagonal 7 funkcije i iznosi 3287 sekundi. Za vrednost t_e uzeta je aproksimacija ove udvostručene vrednosti:

$$2 \cdot 3287 = 6574 = 1h49min34sec \approx 2h \equiv t_e.$$

U cilju podrobnije komparacije *ADD* i *NADD* modela urađeni su dodatni testovi svih funkcija iz tabele 4.1 za 100 puta manje vrednosti broja promenljivih: 10, 20, 30, 50, 70, 80 100, 150, 200 i 300. Ovakav izbor broja promenljivih je odabran pre svega zbog sporosti *NADD* metoda. U testiranjima sa većim brojem promenljivih, čiji su rezultati prikazani u Tabeli 4.1, sa postavljenim vremenskim ograničenjem moguće je bilo istestirati samo 3 od 25 test funkcija *NADD* metodom. Naredna Tabela 4.3 ilustruje rezultate testiranja odabranih test funkcija za pomenute 100 puta manje vrednosti broja promenljivih. Iz prikazanog se vidi da je kod svega 9 od 25 test funkcija bilo moguće testirati *NADD* sa postavljenom ograničavajućom vremenskom konstantom t_e.

Tabela 4.3. *Numerički rezultati dodatnih* 90 *testiranja* ADD *i* NADD *metoda nad* 9 *test funkcija sa manjim brojem promenljivih.*

Test funkcije	Broj iteracija	
	ADD	NADD
Perturbed Quadratic	80	7917
Extended Tridiagonal 1	120	901248
Extended Three Exponential Terms	100	3438
Diagonal 4	100	9036
Quadratic Diagonal Perturbed	102	224031
Quadratic QF1	92	8709
Almost Perturbed Quadratic	90	7939
QUARTC	10	10
Diagonal 9	80	339

Napomena 4.2. *U toku testiranja, funkciju Generalized Quartc Function je bilo moguće testirati za veće vrednosti broja promenljivih (1000, 2000, 3000, 5000, 7000, 8000, 10000, 15000, 20000 i 30000). Međutim, prilikom testiranja NADD iteracije za 100 puta manji broj promenljivih iste funkcije (posebno za n=30 i n=100), parametar vremenskog ograničenja je uzimao vrednosti* $t_e \gg 2h$. *Iz ovog razloga rezultati testiranja ove funkcije nisu prikazani u Tabeli 4.3.*

Na osnovu dodatnih rezultata dobijenih testiranjem *ADD* i *NADD* metoda koji su ilustrovani u Tabeli 4.3, izračunate su prosečne vrednsti istih analiza. Dobijeni prosečni rezultati odražavaju jasno poređenje *ADD* metoda sa njegovom neubrzanom varijantom i potvrđuju nesumnjiv značaj faktora ubrzanja γ_k. Ove vrednosti su prikazane u narednoj Tabeli 4.4. Konstruisanom *ADD* šemom ostvaruju se 1502 puta bolji rezultati po pitanju smanjena broja iteracija u poređenju sa *NADD*.

Tabela 4.4. *Prosečne vrednosti* 9 *testiranih funkcija u* 90 *numeričkih eksperimenata za NADD i ADD iteracije u slučaju izbora manjeg broja promenljivih:* $10, 20, 30, 50, 70, 80, 100, 150, 200$ *i* 300.

Metodi	NADD	ADD
Prosečne vrednosti broja iteracija	129185.22	86

Predstavljena dvosmerna gradijentna ubrzana šema za rešavanje problema bezuslovne optimizacije širokog spektra pripada klasi ubrzanih gradijentnih metoda koju su definisali autori u [1]. Potvrda toga je činjenica da je *ADD* algoritam baziran na Njutnovom metodu sa linijskim traženjem. Takođe, aproksimacijom inverza Hesijana koja je definisana odgovarajućom diagonalnom matricom ostvareno je određivanje osnovnog elementa te aproksimacione matrice -skalara γ_k, od koga potiču ubrzane karakteristike *ADD* metoda. U prikazanom istrživanju urađen je i novi pristup numeričke potvrde činjenice da ubrzana svojstva definisanog gradijentnog metoda potiču od ovako definisanog multiplikativnog parametra γ_k. Sa tim u vezi konstruisana je *neubrazana ADD* šema, *NADD*, u kojoj je izostavljen ovaj parametar, odnosno dodeljena mu je vrednost $\gamma_k = 1$. Numerička testiranja su nesumnjivo potvrdila pretpostavku da su efikasnost i veća brzina metoda prouzrokovana ovim parametrom koji je opravdano nazvan faktorom ubrzanja. Kako je značaj faktora ubrzanja izuzetno veliki, u cilju daljeg poboljšanja karaketristika gradijentnih metoda uopšte, otvara se pitanje novih ideja i načina za njegovo definisanje.

Opravdano je i očekivanje o smanjenju broja potrebnih iteracija primenom *ADD* metoda u poređenju sa *SM* metodom čija je prestižnost nad *GD* i *AGD* iteracijama potvrđena u [1] . Ipak, po pitanju nekih drugih karakteristika kao sto su CPU vreme i broj određivanja vrednosti funkcija, odnosno kako se drugačije označava - broj evaluacija date funkcije, *ADD* model nije presežan u odnosu na *SM*.

Može se zaključiti da je diferencijabilna modifikacija nediferencijabilnog modela iz [2], predstavljena konstruisanim *ADD* metodm dobro definisana. Kako u radu [2] nedostaju numerički eksperimenti koji bi potvrdili efikasnost prezentovanog metoda, ostvareni rezultati iz rada [6] praktično upotpunjuju rezultate prezentovane u [2] po ovom pitanju za slučaj diferencijabilnih funkcija.

4.3 Dvokoračni ubrzani gradijentni metod

Istraživanja opisana u prethodnom poglavlju pokazala su karakteristike jednog ubrazanog gradijentnog metoda sa dva pravca traženja od kojih je jedan gradijentni a drugi je definisan posebnom procedurom. Dobijeni rezultati potvrđuju smanjenje broja iterativnih koraka u odnosu na ubrzani gradijentni *SM* metod. Uopšte, konstrukcijom i analizom *ADD* metoda fokus istraživanja je stavljen na jednu od dve ključne komponente iterativnih metoda bezuslovne optimizacije - na vektor traženja odnosno definisanje dve ovakve veličine u sastavu jedne iteracije. Pitanje koje je prirodno usledilo odnosilo se na drugi esencijalni faktor bezuslovnih iteracija - veličinu iterativnog koraka i, analogno konstrukciji dvosmernog *ADD* modela, i mogućnosti definisanja gradijentnog metoda sa dva parametra iterativnog koraka. Da li bi pri tom tako definisan dvokoračni metod imao bar približno iste performanse po pitanju broja iterativnoih koraka? Može li se očekivati napredak i u pogledu ostala dva svojstva metoda koja u [6] nisu ispitivana: broja određivanja vrednosti objektne funkcije i potrebnog CPU vremena? Istraživanja objavljena u [7] pokazala su da su odgovori na postavljena pitanja potvrdna. U ovom poglavlju dat je pregled i analiza rezultata iz [7] koji opisuju, potvrđuju konvergentna svojstva i prikazuju rezultate numeričkih eksperimenata *dvokoračnog ubrzanog gradijentnog metoda*, skraćeno označenog kao *ADSS* metod.

U prethodnom poglavlju smo se upoznali sa *ADD* iteracijom:

$$x_{k+1} = x_k + \alpha_k s_k + \alpha_k^2 d_k \tag{4.43}$$

u kojoj figurišu dva vektora pravca traženja s_k i d_k. Vektor s_k uzima vrednost negativnog gradijenta objektne funkcije pomnoženog inverzom faktora ubrzanja, odnosno

$$s_k = -\gamma_k^{-1} g_k, \tag{4.44}$$

dok je vektor d_k definisan posebnom procedurom koja se sastoji u minimizaciji pomoćnog problema.

Neka su u (4.43) postavljene sledeće zamene: veličina α_k^2 je zamenjena novim iterativnim korakom označenim sa $\beta_k, \quad \beta_k \neq \alpha_k$, a za pravac vektora traženja d_k pretpostavimo da uzima vrednost negativnog gradijenta, odnosno neka važi

$$d_k = -g_k. \tag{4.45}$$

Pri tom se podrazumeva da su veličine iterativnih koraka β_k i α_k dobijene putem dve različite backtracking procedure.

Predloženim transformacijama se dva pravca pretraživanja iz *ADD* iteracije redukuju u jedan koji je definisan gradijentom. Takođe ovim putem je izbegnuto dodatno vreme potrebno za rešavanje pomoćnog minimizacionog problema koje definiše vektor d_k. Samim tim su i očekivanja da će novodefinisani dvokoračni metod biti brži od *ADD* algoritma opravdana.

Što se tiče faktora ubrzanja, analizom iz [1] pokazano je da uslov $\gamma_k > 0$ treba da važi. Imajući u vidu ovu pretpostavku, vektori s_k i d_k definisani redom relacijama (4.44) i (4.45) jesu smerovi pada u oba slučaja. Zamenom (4.44) i (4.45) u (4.43) dobijamo izraz dvokoračne ubrzane gradijentne *ADSS* iteracije:

$$x_{k+1} = x_k - \alpha_k \gamma_k^{-1} g_k - \beta_k g_k = x_k - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) g_k. \tag{4.46}$$

Kako je

$$x_{k+1} - x_k = -\left(\alpha_k \gamma_k^{-1} g_k + \beta_k g_k\right),$$

Faktor ubrzanja γ_k određen je kao i u [6, 1] iz Telorovog razvoja drugog reda:

$$f(x_{k+1}) \approx f(x_k) - g_k^T \left(\alpha_k \gamma_k^{-1} g_k + \beta_k g_k\right) + \frac{1}{2}\left(\alpha_k \gamma_k^{-1} g_k + \beta_k g_k\right)^T \nabla^2 f(\xi) \left(\alpha_k \gamma_k^{-1} g_k + \beta_k g_k\right), \tag{4.47}$$

pri čemu parametar ξ zadovoljava sledeći uslov:

$$\xi \in [x_k, x_{k+1}], \quad \xi = x_k + \delta(x_{k+1} - x_k) = x_k - \delta\left(\alpha_k \gamma_k^{-1} g_k + \beta_k g_k\right), \quad 0 \leq \delta \leq 1.$$

Dalje je Hesijan $\bigtriangledown^2 f(\xi)$ iz (4.47) zamenjen skalarnom matricom $\gamma_{k+1} I$. Ovom zamenom relacija (4.47) postaje:

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) \|\mathbf{g}_k\|^2 + \frac{1}{2}\left(\alpha_k \gamma_k^{-1} + \beta_k\right)^2 \gamma_{k+1} \|\mathbf{g}_k\|^2, \tag{4.48}$$

odakle izračunavamo γ_{k+1}:

$$\begin{aligned}
\gamma_{k+1} &= 2\frac{f(x_{k+1}) - f(x_k) + g_k^T\left(\alpha_k \gamma_k{}^{-1} g_k + \beta_k g_k\right)}{\left(\alpha_k \gamma_k{}^{-1} g_k + \beta_k g_k\right)^T\left(\alpha_k \gamma_k{}^{-1} g_k + \beta_k g_k\right)} \\
&= 2\frac{f(x_{k+1}) - f(x_k) + g_k^T\left(\alpha_k \gamma_k{}^{-1} + \beta_k\right) g_k}{\left(\alpha_k \gamma_k{}^{-1} + \beta_k\right) g_k^T\left(\alpha_k \gamma_k{}^{-1} + \beta_k\right) g_k} \\
&= 2\frac{f(x_{k+1}) - f(x_k) + \left(\alpha_k \gamma_k{}^{-1} + \beta_k\right) g_k^T g_k}{\left(\alpha_k \gamma_k{}^{-1} + \beta_k\right)^2 g_k^T g_k}.
\end{aligned}$$

Konačno, veličina γ_{k+1} je data sa

$$\gamma_{k+1} = 2\frac{f(x_{k+1}) - f(x_k) + \left(\alpha_k \gamma_k{}^{-1} + \beta_k\right) \|g_k\|^2}{\left(\alpha_k \gamma_k{}^{-1} + \beta_k\right)^2 \|g_k\|^2}. \tag{4.49}$$

Kao i u [6, 1] pretpostavljamo da je $\gamma_{k+1} > 0$ inače neće biti zadovoljeni neophodni i potrebni uslovi drugog reda. Ukoliko se u nekoj iteraciji dogodi da je $\gamma_{k+1} < 0$ uzima se vrednost $\gamma_{k+1} = 1$. Ovakav izbor je opravdan obzirom na činjenicu da kada G_k nije pozitivno definitna matrica vrednost $\gamma_{k+1} = 1$ definiše vektor traženja $s_k = -g_k$ što i jeste vektor opadanja. U tom slučaju narednu iterativnu tačku x_{k+2} računamo kao

$$x_{k+2} = x_{k+1} - (\alpha_{k+1} + \beta_{k+1})\, g_{k+1},$$

što dokazuje da je očuvana forma iteracije pada po gradijentu.

Kako bi smo odredili narednu iterativnu tačku, nakon određivanja parametra $\gamma_{k+1} > 0$, sledi izračunavanje veličina iterativnih koraka α_{k+1} i β_{k+1}

$$x_{k+2} = x_{k+1} - \left(\alpha_{k+1} \gamma_{k+1}^{-1} + \beta_{k+1}\right) g_{k+1}.$$

Sa ciljem dobrog definisanja parametara α_{k+1} i β_{k+1} analizirajmo funkciju

$$\Phi_{k+1}(\alpha, \beta) = f(x_{k+1}) - \left(\alpha \gamma_{k+1}^{-1} + \beta\right) \|g_{k+1}\|^2 + \frac{1}{2}\left(\alpha \gamma_{k+1}^{-1} + \beta\right)^2 \gamma_{k+1} \|g_{k+1}\|^2.$$

U prethodnom izrazu funkcija $\Phi_{k+1}(\alpha, \beta)$ je generisana pod pretpostavkom da je $\gamma_{k+2} = \gamma_{k+1}$. Jasno je da je $\Phi_{k+1}(\alpha, \beta)$ konveksna funkcija kada je $\gamma_{k+1} > 0$.

Gardijent funkcije $\Phi_{k+1}(\alpha,\beta)$ iznosi:

$$\nabla(\Phi_{k+1}(\alpha,\beta)) = \left\{\Phi_{k+1}(\alpha,\beta)'_{\alpha}, \Phi_{k+1}(\alpha,\beta)'_{\beta}\right\},$$

pri čemu je

$$\begin{aligned} (\Phi_{k+1})'_{\alpha} &= \left((\alpha-1)\gamma_{k+1}^{-1}+\beta\right)\|\mathbf{g}_{k+1}\|^2, \\ (\Phi_{k+1})'_{\beta} &= \gamma_{k+1}\cdot(\Phi_{k+1})'_{\alpha}. \end{aligned} \tag{4.50}$$

Dakle, važi

$$\Phi_{k+1}(0,0) = f(x_{k+1}).$$

Funkcija $\Phi_{k+1}(\alpha,\beta)$ opada u slučaju kada je $(\Phi_{k+1})'_{\alpha} < 0, (\Phi_{k+1})'_{\beta} < 0$. A ove nejednakosti važe za

$$\alpha \in (0,1), \quad \beta < (1-\alpha)\gamma_{k+1}^{-1}.$$

Takođe važi

$$\nabla(\Phi_{k+1}(\alpha,\beta)) = \{0,0\} \Longleftrightarrow \beta = (1-\alpha)\gamma_{k+1}^{-1}. \tag{4.51}$$

Kako je

$$\alpha_k\gamma_k^{-1}+\beta_k = \gamma_k^{-1} > 0,$$

prema (4.46), objektna funkcija f opada. Minimum funkcije Φ_{k+1} biće postignut zadovoljenjem relacije (4.51).

Iterativni korak α_{k+1} određujemo procedurom linijskog traženja backtracking 1 i pri tom je $\alpha_{k+1} \in (0,1)$. Slično, iterativni korak β_{k+1} računamo procedurom backtracking 2 gde je takođe $\beta_{k+1} \in (0,1)$.

Sledi prikaz backtracking procedura za izračunavanje iterativnih koraka α_{k+1} i β_{k+1}. Kako se za početnu vrednost oba iterativna koraka uzima jedinica, važi da je $\alpha_{k+1},\beta_{k+1} \in (0,1)$.

Algoritam 4.3.1 Backtracking procedura linijskog traženja sa početnim $\alpha = 1$. Izračunavanje iterativnog koraka α_k.

Require: Objektna funkcija $f(x)$, vektor pravca d_k u iterativnoj tački x_k i brojevi $0 < \sigma_\alpha < 0.5$ i $\eta_1 \in (\sigma_\alpha, 1)$.

1: Postaviti $\alpha = 1$.
2: Dok je ispunjen uslov $f(x_k+\alpha d_k) > f(x_k)+\sigma_\alpha \alpha g_k^T d_k$ postaviti $\alpha := \eta_1\alpha$.
3: Izlazna veličina $\alpha_k = \alpha$.

Algoritam 4.3.2 Backtracking procedura linijskog traženja sa početnim $\beta = 1$. Izračunavanje iterativnog koraka β_k.

Require: Objektna funkcija $f(x)$, vektor pravca d_k u iterativnoj tački x_k i brojevi $0 < \sigma_\beta < 0.5$ i $\eta_2 \in (\sigma_\beta, 1)$.

1: Postaviti $\beta = 1$.

2: Dok je ispunjen uslov $f(x_k + \beta d_k) > f(x_k) + \sigma_\beta \beta g_k^T d_k$ postaviti $\beta := \eta_2 \beta$.

3: Izlazna veličina $\beta_k = \beta$.

Korišćenjem prethodne dve Armijeve procedure za određivanje veličina iterativnih koraka, definišemo *ADSS* algoritam:

Algoritam 4.3.3 Ubrzani dvokoračni *ADSS* metod

Require: Objektna funkcija $f(x)$, inicijalna tačka x_0, $\gamma_0 = 1$.

1: Postaviti $k = 0$, izračunati $f(x_0)$, g_0 i uzeti $\gamma_0 = 1$.

2: Ako je $\|g_k\| < \varepsilon$, preći na korak 9, inače nastaviti sa sledećim korakom.

3: Naći veličinu iterativnog koraka α_k primenom Algoritma 4.3.1.

4: Naći veličinu iterativnog koraka β_k primenom Algoritma 4.3.2.

5: Izračunati x_{k+1} pomoću jednakosti (4.46).

6: Odrediti skalar γ_{k+1} iz relacije (4.49).

7: Ukoliko je $\gamma_{k+1} < 0$, uzeti $\gamma_{k+1} = 1$.

8: Postaviti $k := k + 1$, preći na korak 2.

9: Izlazne veličine x_{k+1} and $f(x_{k+1})$.

Koncepcija analize konvergencije konstruisane dvokoračne iteracije je slična analizi konvergencije dvosmernog ubrzanog *ADD* metoda iz [6] i ubrzanog gradijentnog *SM* metoda iz [1]. Dakle, prvo je sagledan skup uniformno konveksnih funkcija a zatim i podskup striktno konveksnih kvadratnih funkcija. Uzima se u obzir da vaze Propozicija 4.3 i lema 4.4, navedene u Poglavlju 4.2, koje važe na skupu uniformno konveksnih funkcija.

U sledećoj lemi ocenjena je vrednost opadanja objektne funkcije u svakoj iteraciji primenom modela (4.46).

Lema 4.7. *Neka je f dva puta neprekidno diferencijabilna i uniformno konveksna funkcija definisana na $\mathbb{R}^n$, i neka je niz $\{x_k\}$ generisan Algoritmom 4.3.3. Tada je zadovoljena sledeća nejednakost*

$$f(x_k) - f(x_{k+1}) \geq \mu \|g_k\|^2, \tag{4.52}$$

gde je

$$\mu = \min\left\{\frac{\sigma_\alpha}{M}, \frac{\sigma_\alpha\left(1-\sigma_\alpha\right)}{L}\eta_1, \sigma_\beta, \frac{\sigma_\beta\left(1-\sigma_\beta\right)}{L}\eta_2\right\}. \tag{4.53}$$

Dokaz. Neka je $L > 0$ Lipšicova konstanta i $M \geq 1$ broj definisan u Propoziciji 4.3. Kako je γ_k k-ta aproksimacija Hesijana objektne funkcije f, iz (4.17) i (4.46) sledi da je $\gamma_k < M$. Naredne dve nejednakosti su dobijene na isti način kao u dokazu leme 4.2 u [1] zamenjujući redom sa α_k, σ_α i η_1 promenljive t_k, σ i β respektivno

$$f\left(x_k\right) - f\left(x_{k+1}\right) > \frac{\sigma_\alpha\left(1-\sigma_\alpha\right)\eta_1}{L}\|g_k\|^2 \tag{4.54}$$

$$f\left(x_k\right) - f\left(x_{k+1}\right) > \frac{\sigma_\alpha}{M}\|g_k\|^2. \tag{4.55}$$

Preostale dve nejednakosti su ocenjene sledećom analizom. Kako parametar β_k zadovoljava izlazni kriterijum backtracking procedure iz Algoritma 4.3.2, važi

$$f\left(x_k\right) - f\left(x_{k+1}\right) \geq -\sigma_\beta \beta_k g_k^T d_k, \quad \forall k \in \mathbb{N}. \tag{4.56}$$

Kako je $d_k = -g_k$, prethodna nejednakost postaje

$$f\left(x_k\right) - f\left(x_{k+1}\right) \geq -\sigma_\beta \beta_k g_k^T\left(-g_k\right), \quad \forall k \in \mathbb{N}. \tag{4.57}$$

Sada razmatramo dva slučaja: $\beta_k < 1$ i $\beta_k = 1$.
Ukoliko je $\beta_k < 1$, slično kao u [32] zaključujemo da je

$$\beta_k > -\frac{\eta_2\left(1-\sigma_\beta\right)}{L}\frac{g_k^T\left(-g_k\right)}{\|g_k\|^2}, \tag{4.58}$$

odakle sledi

$$\beta_k > \frac{\eta_2\left(1-\sigma_\beta\right)}{L}. \tag{4.59}$$

Zamenom prethodne nejednakosti u (4.56) dobijamo:

$$\begin{aligned} f\left(x_k\right) - f\left(x_{k+1}\right) &\geq -\sigma_\beta \beta_k g_k^T\left(-g_k\right) \\ &> \frac{-\sigma_\beta\left(1-\sigma_\beta\right)\eta_2}{L} g_k^T\left(-g_k\right) \\ &> \frac{\sigma_\beta\left(1-\sigma_\beta\right)\eta_2}{L}\|g_k\|^2. \end{aligned}$$

U slučaju kada je $\beta_k = 1$ jasno je da važi

$$f(x_k) - f(x_{k+1}) \geq -\sigma_\beta g_k^T d_k \geq \sigma_\beta \|g_k\|^2. \tag{4.60}$$

Poslednje dve nejednakosti zajedno sa relacijama (4.54) i (4.55) dovode do konačnog zaključka

$$f(x_k) - f(x_{k+1}) \geq \min\left\{\frac{\sigma_\alpha}{M}, \frac{\sigma_\alpha(1-\sigma_\alpha)\eta_1}{L}, \sigma_\beta, \frac{\sigma_\beta(1-\sigma_\beta)\eta_2}{L}\right\} \tag{4.61}$$

koji smo i želeli da dokažemo. □

Teorema koja sledi potvrđuje linearnu konvergenciju definisanog *ADSS* metoda. Dokaz neće biti naveden obzirom da je isti kao i dokaz analogne teoreme (4.1) iz [1].

Teorema 4.6. *[1] Za objektnu funkciju f koja je dva puta neprekidno diferencijabilna i uniformno konveksna na $\mathbb{R}^n$, i niz $\{x_k\}$ genersan Algoritmom 4.3.3 važi*

$$\lim_{k\to\infty} \|g_k\| = 0 \tag{4.62}$$

i pri tom niz $\{x_k\}$ konvergira ka x^* najmanje linearno.

U nastavku je razmatran slučaj strikno konveksnih kvadratnih funkcija definisanih relacijom

$$f(x) = \frac{1}{2}x^T A x - b^T x \tag{4.63}$$

U izrazu (4.63) A predstavlja realnu $n \times n$ simetričnu pozitivno definitnu matricu dok je vektor $b \in \mathbb{R}^n$. Iz razloga koji je već naveden u [6] i u [7] zbog teškoća u nestandarnom dokazivanju konvergencije gradijentnih metoda pribeglo se parcijalnom utvrđivanju
konvergencije definisanog metoda na skupu striktno konveksnih kvadratnih funkcija. Pri tome se može javiti potreba za postavljanjem dodatnih uslova koji su obično vezani za sopstvene vrednosti matrice A iz (4.63). Tako će i u lemi i teoremi koje slede biti korišćeni pandami nekih poznatih pretpostavki predloženih u [5, 29, 16] prilagođeni iteraciji (4.46). Sa tim u vezi, označimo sopstvene vrednosti simetrične, pozitivno definitne matrice A sa $\lambda_1, \lambda_2, \ldots \lambda_n$ i pri tom neka važi $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

Lema 4.8. *Neka za strikno konveksnu kvadratnu funkciju f definisanu relacijom* (4.63) *u kojoj je A simetrična pozitivno definitna matrica, $A \in \mathbb{R}^{n\times n}$, primenjen gradijentni metod* (4.46)*, pri* *čemu su parametri γ_k, α_k i β_k određeni relacijom* (4.49)*, Algoritmom 4.3.1 i Algoritmom 4.3.2. Tada važe ocene:*

$$\frac{1}{\lambda_1}+1 \geq \alpha_{k+1}\gamma_{k+1}^{-1}+\beta_{k+1} \geq \frac{1}{\lambda_n}\left(\sigma_\alpha^2+\sigma_\beta^2\right) \tag{4.64}$$

gde su sa λ_1 i λ_n označene najmanja i naveća sopstvena vrednost matrice A respektivno.

Dokaz. Koristeći izraz (4.63), potražimo razliku vrednosti objektne funkcje f u tekućoj i prethodnoj tački:

$$f(x_{k+1})-f(x_k)=\frac{1}{2}x_{k+1}^T A x_{k+1}-b^T x_{k+1}-\frac{1}{2}x_k^T A x_k+b^T x_k. \tag{4.65}$$

Znajući da je

$$x_{k+1}=x_k-\alpha_k\gamma_k{}^{-1}g_k-\beta_k g_k=x_k-\left(\alpha_k\gamma_k^{-1}+\beta_k\right)g_k \tag{4.66}$$

dobijamo

$$\begin{aligned}
f(x_{k+1})-f(x_k)=&\frac{1}{2}\left(x_k-\alpha_k\gamma_k{}^{-1}g_k-\beta_k g_k\right)^T A\left(x_k-\alpha_k\gamma_k{}^{-1}g_k-\beta_k g_k\right)-b^T\left(x_k-\alpha_k\gamma_k{}^{-1}g_k-\beta_k g_k\right)\\
&-\frac{1}{2}x_k^T A x_k+b^T x_k\\
=&\frac{1}{2}x_k^T A x_k-\frac{1}{2}\alpha_k\gamma_k^{-1}x_k^T A g_k-\frac{1}{2}\beta_k x_k^T A g_k\quad-\frac{1}{2}\alpha_k\gamma_k^{-1}g_k^T A x_k+\frac{1}{2}\alpha_k^2\gamma_k^{-2}g_k^T A g_k\\
&+\frac{1}{2}\alpha_k\gamma_k^{-1}\beta_k g_k^T A g_k-\frac{1}{2}\beta_k g_k^T A x_k+\frac{1}{2}\beta_k\alpha_k\gamma_k^{-1}g_k^T A g_k+\frac{1}{2}\beta_k^2 g_k^T A g_k-b^T x_k\\
&+\alpha_k\gamma_k^{-1}b^T g_k+\beta_k b^T g_k-\frac{1}{2}x_k^T A x_k+b^T x_k.
\end{aligned}$$

Gradijent funkcije (4.63) iznosi $g_k=Ax_k-b$ a takođe važi i jednakost $g_k^T A g_k=g_k A g_k^T$. Iz ove dve činjenice zaključujemo

$$\begin{aligned}
f(x_{k+1})-f(x_k)=&-\alpha_k\gamma_k^{-1}x_k^T A g_k-\beta_k x_k^T A g_k+\beta_k\alpha_k\gamma_k^{-1}g_k^T A g_k+\frac{1}{2}\alpha_k^2\gamma_k^{-2}g_k^T A g_k+\frac{1}{2}\beta_k^2 g_k^T A g_k\\
&+\alpha_k\gamma_k^{-1}b^T g_k+\beta_k b^T g_k\\
=&-\alpha_k\gamma_k^{-1}\left(x_k^T A-b^T\right)g_k-\beta_k\left(x_k^T A-b^T\right)g_k+\frac{1}{2}\left(\alpha_k^2\gamma_k^{-2}+2\alpha_k\gamma_k^{-1}\beta_k+\beta_k^2\right)g_k^T A g_k\\
=&-\alpha_k\gamma_k^{-1}g_k^T g_k-\beta_k g_k^T g_k+\frac{1}{2}\left(\alpha_k\gamma_k^{-1}+\beta_k\right)^2 g_k^T A g_k\\
=&-\left(\alpha_k\gamma_k^{-1}+\beta_k\right)g_k^T g_k+\frac{1}{2}\left(\alpha_k\gamma_k^{-1}+\beta_k\right)^2 g_k^T A g_k.
\end{aligned}$$

Zamenom dobijenog rezultata u izraz koji definiše parametar γ_{k+1}, (4.49), dobija se da je

$$\begin{aligned}\gamma_{k+1} &= 2\frac{\frac{1}{2}\left(\alpha_k\gamma_k^{-1}+\beta_k\right)^2 g_k^T A g_k - \left(\alpha_k\gamma_k^{-1}+\beta_k\right) g_k^T g_k + \left(\alpha_k\gamma_k^{-1}+\beta_k\right) g_k^T g_k}{\left(\alpha_k\gamma_k^{-1}+\beta_k\right)^2 g_k^T g_k} \\ &= \frac{g_k^T A g_k}{g_k^T g_k}.\end{aligned}$$

Odavde zaključujemo da je γ_{k+1} Rejlijev količnik realne simetrične matrice A i vektora g_k, a iz ove činjenice sledi ocena

$$\lambda_1 \le \gamma_{k+1} \le \lambda_n, \quad k \in \mathbb{N}. \tag{4.67}$$

Kako važe ocene $0 \le \alpha_{k+1} \le 1$ i $0 \le \beta_{k+1} \le 1$ uzevši u obzir (4.67) proizilazi leva strana nejednakosti (4.64):

$$\frac{\alpha_{k+1}}{\gamma_{k+1}} + \beta_{k+1} \le \frac{1}{\lambda_1} + 1.$$

Desna strana iste nejednakosti sledi iz (4.59) i iz ocene

$$t_k > \frac{\beta\,(1-\sigma)\,\gamma_k}{L}, \tag{4.68}$$

koja je dokazana u [1] u Lemi(4.2). Upotrebom oznaka definisanih u [7] prethodni izraz postaje:

$$\alpha_k > \frac{\eta_1\,(1-\sigma_\alpha)\,\gamma_k}{L}. \tag{4.69}$$

Odavde direktno sledi:

$$\frac{\gamma_{k+1}}{\alpha_{k+1}} < \frac{L}{\eta_1\,(1-\sigma_\alpha)}. \tag{4.70}$$

Najveća sopstvena vrednost matrice A, λ_n ima svojstvo Lipšicove konstante L u (4.70). Dokaz ove činjenice proizilazi iz sledeće analize: A je simetrična a $g(x) = A(x) - b$. Sledi da je

$$\|g(x)-g(y)\| = \|Ax-Ay\| = \|A(x-y)\| \le \|A\|\|x-y\| = \lambda_n\|x-y\|. \tag{4.71}$$

U algoritmima backtracking procedura (4.3.1) i (4.3.2) parametri σ_α, η_1, σ_β i η_2 su odabrani tako da je ispunjeno: $0 < \sigma_\alpha < 0.5$ i $\eta_1 \in (\sigma_\alpha, 1)$. Respektivno, $0 < \sigma_\beta < 0.5$ i $\eta_2 \in (\sigma_\beta, 1)$.

Imajući u vidu ova ograničenja, zaključujemo:

$$\begin{aligned}\alpha_{k+1}\gamma_{k+1}^{-1} + \beta_{k+1} &> \frac{1}{L}\big(\eta_1(1-\sigma_\alpha) + \eta_2(1-\sigma_\beta)\big)\\ &> \frac{1}{\lambda_n}\left(\sigma_\alpha\sigma_\alpha + \sigma_\beta\sigma_\beta\right)\\ &= \frac{1}{\lambda_n}\left(\sigma_\alpha^2 + \sigma_\beta^2\right)\end{aligned}$$

što dokazuje desnu stranu u nejednakostima (4.64). □

Teorema 4.7. *Za striktno konveksnu kvadratnu funkciju f datu izrazom* (4.63) *i metod opadanja po gradijentu* (4.46)*, pod dodatnom pretpostavkom koju zadovoljavaju sopstvene vrednosti matrice A*

$$\lambda_n < \frac{2\lambda_1}{1+\lambda_1}, \tag{4.72}$$

važi

$$\left(d_i^{k+1}\right)^2 \le \delta^2\left(d_i^k\right)^2 \tag{4.73}$$

pri čemu je

$$\delta = \max\left\{1 - \frac{\lambda_1}{\lambda_n}\left(\sigma_\alpha^2 + \sigma_\beta^2\right), \lambda_n\left(\frac{1}{\lambda_1}+1\right) - 1\right\}, \tag{4.74}$$

kao i

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{4.75}$$

Dokaz. Neka su $\{v_1, v_2, \ldots, v_n\}$ otonormirani sopstveni vektori simetrične matrice A i neka je $\{x_k\}$ niz vrednosti generisan Algoritmom 4.3.3. Za neko k i neko x_k važi $g_k = Ax_k - b$. Uz to važi

$$g_k = \sum_{i=1}^{n} d_i^k v_i, \tag{4.76}$$

gde su $d_1^k, d_2^k, \ldots, d_n^k$ realne konstante.

Iz (4.46) sledi

$$g_{k+1} = A(x_k - \alpha_k\gamma_k^{-1}g_k - \beta_k g_k) - b = (I - \alpha_k\gamma_k^{-1}A - \beta_k A)g_k, \tag{4.77}$$

što primenjeno na (4.76) daje

$$g_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i = \sum_{i=1}^{n} \left(1 - \alpha_k \gamma_k^{-1} \lambda_i - \beta_k \lambda_i\right) d_i^k v_i. \tag{4.78}$$

Da bi dokazali (4.74), dovoljno je pokazati da je $|1 - \lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right)| \leq \delta$. Razlikuju se dva slučaja. Pretpostavimo prvo da je $\lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right) \leq 1$. Primenom (4.64) dobijamo:

$$\begin{aligned} 1 \geq \lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right) &\geq \frac{\lambda_1}{\lambda_n} \left(\sigma_\alpha^2 + \sigma_\beta^2\right) \\ \Longrightarrow 1 - \lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right) &\leq 1 - \frac{\lambda_1}{\lambda_n} \left(\sigma_\alpha^2 + \sigma_\beta^2\right) \leq \delta \end{aligned} \tag{4.79}$$

Sada pretpostavimo da je $\lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right) > 1$. Zaključak izvodimo iz sledećih ocena

$$\begin{aligned} 1 < \lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right) &\leq \lambda_n \left(\frac{1}{\lambda_1} + 1\right) \\ \Longrightarrow |1 - \lambda_i \left(\alpha_k \gamma_k^{-1} + \beta_k\right)| &\leq \lambda_n \left(\frac{1}{\lambda_1} + 1\right) - 1 < \delta. \end{aligned} \tag{4.80}$$

Ostaje još da se proveri da li je pod zadatim pretpostavkama parametar $\delta \in (0,1)$. Kako su sopstvene vrednosti pozitivno definitne matrice pozitivne, zaključujemo:

$$\begin{aligned} 0 < \frac{\lambda_1}{\lambda_n} \left(\sigma_\alpha^2 + \sigma_\beta^2\right) < 1 &\Longrightarrow 0 < 1 - \frac{\lambda_1}{\lambda_n} \left(\sigma_\alpha^2 + \sigma_\beta^2\right) < 1 \\ \lambda_n \left(\frac{1}{\lambda_1} + 1\right) > \frac{\lambda_1}{\lambda_1} + \lambda_1 > 1 &\Longrightarrow 0 < \lambda_n \left(\frac{1}{\lambda_1} + 1\right) - 1 < \frac{2\lambda_1}{1 + \lambda_1} \frac{1 + \lambda_1}{\lambda_1} - 1 = 1 \end{aligned} \tag{4.81}$$

Dokaz jednakosti (4.75) je isti kao dokaz analogne jednakosti iz [1]. Iz reprezentacije (4.76):

$$\|g_k\|^2 = \sum_{i=1}^{n} (d_i^k)^2 \tag{4.82}$$

i činjenice da je u nejednakost (4.73) parametar $0 < \delta < 1$, direktno sledi jednakost (4.75). □

Napomena 4.3. *Dodatni uslov* (4.72) *je neophodna pretpostavka potrebna u dokazivanju konvergencije konstruisanog ADSS metoda specijalno na skupu striktno konveksnih kvadratnih* *funkcija. Ovaj uslov je formulisan kao*

$$\lambda_n < 2\lambda_1 \tag{4.83}$$

u [16]. U pomenutom radu Molina i Rajdan su potvrdili Q-linearnu konvergenciju preuslovljenog BB metoda, označenog sa PBB, pod pretpostavkom (4.83). *Kasnije, u [6, 1], autori su koristili isti uslov da bi dokazali konvergenciju SM metoda za striktno konveksne kvadratne funkcije.*

Numerički eksperimenti urađeni su za 31 test funkciju iz [34]. Prvobitno su testovi obavljeni za skup testiranih funkcija u [6] pri čemu su primenjeni metodi *ADSS*, *ADD* i *SM*. Potom je radi dodatne potvrde bolje efikasnosti *ADSS* modela u odnosu na druge poredbene metode testirano još 6 funkcija iz [34]. Odabrani izbor broja promenljivih je kao i u [6], dakle više vrednosti: 1000, 2000, 3000, 5000, 7000, 8000, 10000, 15000, 20000 i 30000, čime je ostvarena veća generalizacija prezentovanog numeričkog istraživanja.

Korišćen je isti kriterijum završetka kao u [6] i [1] . Parametri backtracking procedura uzimaju sledeće vrednosti: $\sigma_1 = 0.0001, \eta_1 = 0.8$ u Algoritmu 4.3.1 and $\sigma_1 = 0.0002, \eta_1 = 0.9$ u Algoritmu 4.3.2. Za razliku od numeričkog istraživanja predstavljenog u [6], u radu [7] su praćene i analizirane tri osnovne karakteristike metoda: broj iteracija, CPU vreme i broj određivanja vrednosti funkcija. Podsetimo se da su u [1] urađena poređenja između *SM*, *GD* i *AGD* metoda i da je pri tom potvrđena značajna dominacija *SM* algoritma u odnosu na druga dva. Upravo iz tog razloga, numerički eksperimenti prezentovani u [7] prikazuju poređenje konstruisanog *ADSS* modela i *SM* metoda iz [1] . Kao poredbeni metod, pridodat je i *ADD* metod iz [6]. Efikasnost *ADD* šeme po pitanju smanjenja broja iteracija u odnosu na *SM* metod je numerički potvrđena u [6], ali u pogledu druge dve karakteristike poređenje ova dva metoda nije sprovedeno u [6]. Numerička analiza iz [7] upotpunila je istraživanja i rezultatima vezanim za ove dve karakteristike kod sva tri testirana metoda.

Dobijene vrednosti su ilustrovane u 4 tabele. U Tabeli 4.5 prikazani su numerička izračunavanja broja iteracija, CPU vremena datog u sekundama i broja određivanja vrednosti objektne funkcije za 25 test funkcija testiranih u [6].

Tabela 4.5. *Prikaz numeričkih rezultata* SM, ADD *i* ADSS *metoda tesiranih u slučaju* 25 *test funkcija.*

Test funkcija	Broj iteracija			CPU vreme			Broj evaluacija		
	SM	ADD	ADSS	SM	ADD	ADSS	SM	ADD	ADSS
Extended Penalty	589	70	50	5	2225	4	2987	220356	1780
Perturbed quadratic	111972	80	397	1868	542	0	632724	144886	1714
Raydan-1	21125	86	34	178	785	4	116757	139268	4844
Diagonal 1	10417	81	37	116	973	0	56135	136143	1448
Diagonal 3	10574	80	49	209	1441	1	59425	132375	1048
Generalized Tridiagonal - 1	278	110	77	2	982	0	989	176274	719
Extended Tridiagonal - 1	3560	120	70	35	812	0	30686	185863	420
Extended Three Exponï£¡	164	100	40	1	889	0	1224	104690	350
Diagonal 4	80	100	780	0	1685	0	530	223240	2590
Extended Himmelblau	168	100	70	0	2213	0	566	206110	480
Quadr. Diag. Perturbed	53133	90	393	1193	1937	1	547850	250354	1719
Quadratic QF1	114510	80	425	2127	440	0	649643	131998	1755
Extended Quad. Penalty QP1	224	90	60	12	3207	2	2566	254293	841
Extended Quad. Penalty QP2	162	80	60	7	1815	4	2057	151534	843
Quadratic QF2	118801	86	60	2544	1960	1	662486	209232	836
Extended EP1	68	100	40	1	2927	0	764	278890	487
Extended Tridiagonal - 2	584	120	80	0	591	0	2144	133922	420
Arwhead	10	102	64	0	3176	3	30	253919	1082
Almost Perturbed Quadratic	110121	80	397	2148	499	0	627287	144276	1712
Engval1	185	100	70	7	1084	0	2177	130390	460
Quartc	190	10	70	0	7	0	430	40	390
Generalized quartic	156	100	70	0	822	0	423	107188	614
Diagonal 7	90	3287	2201	0	3382	33	220	11092	6633
Diagonal 8	96	3225	2213	0	3333	40	594	11025	6709
Diagonal 9	11235	80	43	118	1010	0	60566	139826	3646

Prikazani numerički rezultati potvrđuju prestiž *ADSS* metoda u odnosu na *SM* i *ADD* metode po pitanju sve tri praćene karakteristike. U pogledu potrebnog broja iteracija, vidimo da *ADSS* daje bolje rezultate kod 21 od 25 testirnih funkcija u poređenju sa *SM* metodom, a od *ADD* metoda je bolji u 18 od 25 slučajeva. Po pitanju brzine testiranja, odnosno potrebnog CPU vremena, *ADSS* je brži od *SM* metoda kod 17 od 22 funkcije a u slučaju 3 test funkcije imaju jednako vreme izvršenja. Analiza i treće praćene karakteristike - broja određivanja vrednosti objektne funkcije, takođe potvrđuje dominaciju rezultata ostvarenih testiranjem *ADSS* modelom. Preciznije, u 20 od 25 slučajeva *ADSS* ima manji broj određivanja vrednosti objektne funkcije, a u odnosu na *ADD* metod čak 24 od 25. Više je nego očigledno da je broj slučajeva kod kojih *SM* i *ADD* metodi daju bolje rezultate od *ADSS* modela neuporedivo manji u odnosu na obrnut slučaj.

Prosečne vrednosti rezultata iz Tabele 4.5, sortirane su u sledećoj Tabeli 4.6. Rezulati iz ove tabele potvrđuju prednost *ADSS* algoritma i sa aspekta prosečnih vrednosti.

Tabela 4.6. *Prosečne vrednosti numeričkih rezultata za* 25 *test funkcija za* 10 *numeričkih eksperimenata u svakoj iteraciji.*

Prosečne vrednosti	ADSS	SM	ADD
Broj iteracija	314	22739.68	342.28
CPU vreme (u sekundama)	3.72	422.84	1549.48
Broj evaluacija obj. funkcije	1741.6	138450.4	155087.36

U Tabeli 4.6 ilustrovane su prosečne vrednosti rezultata 250 testiranja za svaki od tri analizirana algoritma. Prikazani rezultati argumentuju superiornost *ADSS* šeme po pitanju sva tri testirana svojstva: broja iteracija, CPU vremena i broja određivanja vrednosti objektne funkcije. Evidentno je da se *ADSS* i *ADD* metodi ponašaju slično u pogledu potrebnog broja iteracija uz blagu prednost u korist *ADSS* algoritma. Po istom pitanju, *ADSS* je oko 70 puta bolji u odnosu na *SM* metod. Testiranja *ADSS* metodom traju oko 113 puta kraće nego li *SM* iteracijom dok upotrebom *ADD* šeme testiranja traju oko 416 puta duže u odnosu na *ADSS*. Što se tiče broja određivanja vrednosti date funkcije, *ADSS* algoritmom se ovaj broj ostvaruje oko 80 puta manje u odnosu na *SM* a oko 90 puta manje u odnosu na *ADD* algoritam.

Prednost *ADSS* modela u odnosu na druga dva poredbena modela potvrđena je testitarenjem dodatnih 6 funkcija iz [34] koje se ne nalaze u skupu testiranih funkcija u [6]. I ovi testovi su takođe urađeni za isti skup promenljivih većih vrednosti: 1000, 2000, 3000, 5000, 7000, 8000, 10000, 15000, 20000, 30000. U ovom delu numeričkih istraživanja praćeni su i slučajevi u kojima su testiranja trajala dosta dugo. Sa tim u vezi parametar definisan u [6] t_e, tzv. parametar vremenskog ograničenja (*execution time limiter parameter*), je usvojen kao dopunski kriterijum završetka. Korišćena je ista vrednost ovog parametra kao i u [6]: $t_e \approx 2h$. Iz naredne Tabele

4.7 vidi se da su testiranja *ADSS* metodom najbolja u 5 slučajeva u pogledu broja iteracija, pri tome u jednom od tih pet slučajeva sva tri metoda imaju jednak broj iteracija. Testiranja *ADSS* modelom traju najkraće kod 4 test funkcije dok je *SM* metod najbrži u slučaju jedne funkcije. Slično je i sa brojem evaluacija: *ADSS* daje najmanji broj kod 4 funkcije a *SM* kod jedne. Generalno, u 3 od 6 slučajeva testiranja *SM* i *ADD* modelima traju duže od predviđenog vremenskog ograničenja t_e, dok je prilikom testiranja *ADSS* metodom ovaj kriterijum završetka bio ispunjen samo u slučaju jedne funkcije.

Tabela 4.7. *Numerički rezultati za 6 dodatnih test funkcija*

Test funkcija	Broj iteracija			CPU vreme			Broj evaluacija		
	SM	ADD	ADSS	SM	ADD	ADSS	SM	ADD	ADSS
DIXON3DQ	112899	120	10	1908	220	0	639957	184810	40
NONSCOMP	10	10	10	0	0	0	30	40	40
HIMMELH	30	$> t_e$	5079	0	$> t_e$	14	80	$> t_e$	15267
Power(Cute)	$> t_e$	80	195	$> t_e$	1875	0	$> t_e$	252225	1510
Indef	$> t_e$	$> t_e$	20	$> t_e$	$> t_e$	0	$> t_e$	$> t_e$	70
Sine	$> t_e$	$> t_e$	$> t_e$	$> t_e$	$> t_e$	$> t_e$	$> t_e$	$> t_e$	$> t_e$

Poslednja Tabela 4.8 prikazuje generalne performanse numerički analiziranih metoda za sve tri karakteristike u ukupno 310 urađenih eksperimenata. Opšti je zaključak da su *ADSS* modelom ostvareni najbolji rezultati po pitanju sva tri aspekta u najvećem broju slučajeva.

Tabela 4.8. *Performanse ADSS, SM i ADD algoritama kod 310 test problema.*

Kriterijum performansi	Broj problema
Minimalni broj iteracija ostvaren ADSS modelom	190
Minimalni broj iteracija ostvaren SM modelom	60
Minimalni broj iteracija ostvaren ADD modelom	70
Jednak broj iteracija ostvaren ADSS i SM modelima	10
Jednak broj iteracija ostvaren ADSS i ADD modelima	10

Jednak broj iteracija ostvaren ADSS, SM i ADD modelima	10
Minimalno CPU vreme ostvareno ADSS modelom	260
Minimalno CPU vreme ostvareno SM modelom	100
Minimalno CPU vreme ostvareno ADD modelom	10
Jednako CPU vreme ostvareno ADSS i SM modelima	60
Jednako CPU vreme ostvareno ADSS i ADD modelima	10
Jednako CPU vreme ostvareno ADSS, SM i ADD modelima	10
Minimalni broj evaluacija funkcija ostvaren ADSS modelom	220
Minimalni broj evaluacija funkcija ostvaren SM modelom	70
Minimalni broj evaluacija funkcija ostvaren ADD modelom	10
Jednak broj evaluacije funkcija ostvarenih ADSS i SM modelima	0
Jednak broj evaluacije funkcija ostvarenih ADSS i ADD modelima	10
Jednak broj evaluacije funkcija ostvarenih ADSS, SM i ADD modelima	0

Kodovi metoda za potrebne numeričke analize napisani su u programskom jeziku visal C++ a testiranja su izvršena na računaru Workstation Intel 2.2 GHz.

Sudeći po urađenim numeričkim eksperimentima iz [7], vidimo da *ADSS* šema po svim praćenim karakteristikama a to su broj iteracija, CPU vreme i broj određivanja vrednosti objektne funkcije, višestruko prevazilazi *SM* i *ADD* algoritme. Izuzetno dobri numerički rezultati iz [7] bili su podstrek za dalje proučavanje dvosmernih a pre svega dvokoračnih modela definisanih po sličnom principu. Šta više, opravdano se nametnula ideja za izadvajanje klase dvosmernih i dvokoračnih modela za različite izbore smerova traženja i veličina iterativnih koraka. Premda su još u nedovoljno istraženoj fazi, motivi za nastavak proučavanja dvosmernih i dvokoračnih metoda bezuslovne optimizacije su predloženi u poslednjem poglavlju disertacije. U narednom poglavlju je predstavljena jedna modifikacija *ADSS* modela kojom je nastavljeno dalje izučavanje karakteristika dvokoračnih metoda.

4.4 Jedna transformacija dvokoračnog ubrzanog gradijentnog metoda

Izuzetni numerički rezultati prikazani u [7] bili su dovoljna motivacija za nastavak istraživanja *ADSS* modela. Ideja prikazana u radu [8] sastoji se u modifikovanju *ADSS* dvokoračne šeme redukovanjem na jednokoračnu. Ova redukcija se ostvaruje postavljanjem uslova nad parametrima koji određuju veličine iterativnih koraka u *ADSS* iteraciji, α_k i β_k. Pomenuti uslov je definisan relacijom:

$$\beta_k = 1 - \alpha_k \tag{4.84}$$

Podsetimo se da je *ADSS* iteracija definisana izrazom:

$$x_{k+1} = x_k - \left(\alpha_k \gamma_k^{-1} + \beta_k\right) g_k. \tag{4.85}$$

Zamenom (4.84) u (4.85), *ADSS* iteraciju transformišemuo u jednokoračni model

$$x_{k+1} = x_k - [\alpha_k(\gamma_k^{-1} - 1) + 1] g_k. \tag{4.86}$$

Relacija (4.86) predstavlja redukciju dvokoračnog *ADSS* modela na jednokoračni pod uslovom (4.84). Zato je iteracija (4.86) nazvana *Transformisana ADSS* iteracija ili skraćeno *TADSS* metod. Na ovaj način definisana iteracija (4.86) predstavlja takođe i izvesnu modifikaciju jednokoračnog *SM* metoda pri čemu je proizvod iterativnog koraka i inverza faktora ubrzanja iz iteracije *SM*

$$t_k \gamma_k^{-1}$$

modifikovan izrazom

$$\alpha_k(\gamma_k^{-1} - 1) + 1.$$

Dobijeni rezultati urađenih eksperimenata koji su u nastavku priloženi pokazuju da je definisanje ubrzanog jednokoračnog gradijentnog modela datog relacijom (4.86) daleko efikasniji u numeričkom smislu od klasičnog ubrzanog modela predloženog u [1].

Korišćenjem Tejlorovog razvoja drugog reda, slično procedurama opisanim u [1, 6, 7] određujemo vrednost faktora ubrzanja γ_{k+1} u $k+1$ iterativnom koraku.

Kako je

$$x_{k+1} - x_k = -\left(\alpha_k \gamma_k{}^{-1} g_k + \beta_k g_k\right),$$

Faktor ubrzanja γ_k određen je kao i u [1, 6, 7] iz Telorovog razvoja drugog reda:

$$f(x_{k+1}) \approx f(x_k) - g_k^T \left(\alpha_k(\gamma_k^{-1} - 1)g_k + g_k\right) + \frac{1}{2}\left(\alpha_k(\gamma_k^{-1} - 1)g_k + g_k\right)^T \nabla^2 f(\xi)\left(\alpha_k(\gamma_k^{-1} - 1)g_k + g_k\right), \tag{4.87}$$

pri čemu parametar ξ zadovoljava sledeći uslov:

$$\xi \in [x_k, x_{k+1}], \quad \xi = x_k + \delta(x_{k+1} - x_k) = x_k - \delta\left(\alpha_k(\gamma_k^{-1} - 1)g_k + g_k\right), \quad 0 \le \delta \le 1.$$

U izrazu (4.87) je Hesijan $\bigtriangledown^2 f(\xi)$ zamenjen njegovom skalarnom aproksimacijom $\gamma_{k+1} I$. Time (4.87) postaje:

$$f(\mathbf{x}_{k+1}) = f(\mathbf{x}_k) - \left(\alpha_k(\gamma_k^{-1} - 1) + 1\right)\|\mathbf{g}_k\|^2 + \frac{1}{2}\left(\alpha_k(\gamma_k^{-1} - 1) + 1\right)^2 \gamma_{k+1}\|\mathbf{g}_k\|^2. \tag{4.88}$$

Iz prethodne jednakosti direktno se izračunava vrednost veličine γ_{k+1}:

$$\gamma_{k+1} = 2\frac{f(x_{k+1}) - f(x_k) + \left(\alpha_k(\gamma_k^{-1} - 1) + 1\right)\|g_k\|^2}{\left(\alpha_k(\gamma_k^{-1} - 1) + 1\right)^2 \|g_k\|^2}. \tag{4.89}$$

Da bi neophodni i dovoljni uslovi optimalnosti bili zadovoljeni, nadalje se pretpostavlja da je parametar $\gamma_{k+1} > 0$. U slučaju da je u nekoj iteraticiji izračunata veličina γ_{k+1} negativna, dodeljuje joj se vrednost $\gamma_{k+1} = 1$. Tada se naredna iterativna tačka računa

$$x_{k+2} = x_{k+1} - (\alpha_k(1-1) + 1)\, g_{k+1} = x_{k+1} - g_{k+1}$$

a ovo jeste klasičan metod pada po gradijentu.

Vratimo se sada određivanju naredne iterativne tačke u slučaju kada je parametar $\gamma_{k+1} > 0$. U tu svrhu odredimo vrednost iterativnog koraka u $k+1$ iteraciji

$$x_{k+2} = x_{k+1} - [\alpha_{k+1}(\gamma_{k+1}^{-1} - 1) + 1]g_{k+1}.$$

Posmatrajmo pomoćnu funkciju $\Phi_{k+1}(\alpha)$

$$\Phi_{k+1}(\alpha) = f(x_{k+1}) - \left(\alpha(\gamma_{k+1}^{-1} - 1) + 1\right) \|g_{k+1}\|^2 + \frac{1}{2}\left(\alpha(\gamma_{k+1}^{-1} - 1) + 1\right)^2 \gamma_{k+1}\|g_{k+1}\|^2.$$

Napomenimo da je funkcija $\Phi_{k+1}(\alpha)$ definisana pod pretpostavkom da je $\gamma_{k+2} = \gamma_{k+1}$. Funkcija $\Phi_{k+1}(\alpha)$ je konveksna u slučaju kada je $\gamma_{k+1} > 0$, a njen gradijent $\nabla(\Phi_{k+1}(\alpha)) = \{\Phi_{k+1}(\alpha)'_\alpha\}$, iznosi

$$\begin{aligned}
(\Phi_{k+1})'_\alpha &= -\left(\gamma_{k+1}^{-1} - 1\right) \|\mathbf{g}_{k+1}\|^2 + \left(\alpha(\gamma_{k+1}^{-1} - 1) + 1\right) \gamma_{k+1}\|\mathbf{g}_{k+1}\|^2 \left(\gamma_{k+1}^{-1} - 1\right) \\
&= \left(\gamma_{k+1}^{-1} - 1\right)\left(-1 + (\alpha(\gamma_{k+1}^{-1} - 1) + 1)\gamma_{k+1}\right) \|\mathbf{g}_{k+1}\|^2 \\
&= \left(\gamma_{k+1}^{-1} - 1\right)(-1 + \alpha - \alpha\gamma_{k+1} + \gamma_{k+1}) \|\mathbf{g}_{k+1}\|^2 \\
&= \left(\gamma_{k+1}^{-1} - 1\right)(\alpha - 1)(1 - \gamma_{k+1}) \|\mathbf{g}_{k+1}\|^2 \\
&= \gamma_{k+1}^{-1}(1 - \gamma_{k+1})^2 (\alpha - 1) \|\mathbf{g}_{k+1}\|^2.
\end{aligned}$$

Kako funkcija $\Phi_{k+1}(\alpha)$ opada onda kada je ispunjeno $(\Phi_{k+1})'_\alpha < 0$, za $\gamma_{k+1} > 0$ imamo da važi

$$(\Phi_{k+1})'_\alpha < 0 \Leftrightarrow \alpha < 1.$$

Pri tome je

$$\nabla(\Phi_{k+1}(\alpha)) = \{0\} \Longleftrightarrow \alpha = 1, \tag{4.90}$$

Iz dokazanog dalje sledi da je

$$\alpha_k(\gamma_k^{-1} - 1) + 1 = \gamma_k^{-1} - 1 + 1 = \gamma_k^{-1} > 0,$$

pa prema (4.86) zaključujemo da objektna funkcija f opada.

Iterativni korak α_{k+1} određujemo procedurom linijskog traženja backtracking 1 uzimajući za početnu veličinu iterativnog koraka, shodno prethodnom razmatranju, $\alpha = 1$ i pri tom za svaki naredni iteratini korak važi $\alpha_{k+1} \in (0,1)$. Sledi prikaz *TADSS* algoritma.

Algoritam 4.4.1 Backtracking procedura za izračunavanje veličine iterativnog koraka α_k sa početnim $\alpha = 1$.

Require: Objektna funkcija $f(x)$, vektor traženja d_k u tački x_k, brojevi $0 < \sigma < 0.5$ i $\eta \in (\sigma, 1)$.

1: Postaviti $\alpha = 1$.
2: Dok važi $f(x_k + \alpha d_k) > f(x_k) + \sigma \alpha g_k^T d_k$ postaviti $\alpha := \eta \alpha$.
3: Izlaz $\alpha_k = \alpha$.

Algoritam 4.4.2 Transformisani ubrzani dvokoračni algoritam, (*TADSS* metod).

Require: $0 < \rho < 1, \quad 0 < \tau < 1, x_0, \gamma_0 = 1$.

1: Postaviti $k = 0$, izračunati $f(x_0)$, g_0 i postaviti $\gamma_0 = 1$.
2: Ukoliko je $\|g_k\| < \varepsilon$, preći na korak 9, inače preći na naredni korak.
3: Izračunati veličinu iterativnog koraka α_k primenom Algoritma 4.4.1.
4: Izračunati sledeću iterativnu tačku x_{k+1} iz relacije (4.86).
5: Odrediti skalar γ_{k+1} pomoću jednakosti (4.89).
6: Ako je $\gamma_{k+1} < 0$, uzeti $\gamma_{k+1} = 1$.
7: Postaviti $k := k+1$, i preći na korak 2.
8: Izlaz x_{k+1} i $f(x_{k+1})$.

Dokazivanje konvergenicije *TADSS* algoritma prati klasičan koncept predložen u [1, 6, 7]. Dakle, najpre je razmatran skup uniformno konveksnih funkcija, a potom i skup striktno konveksnih kvadratnih funkcija. U dokazima definisanih tvrđenja podrazumeva se važenje Propozicije 4.3, Leme 4.4 i 4.5 navedenih u Poglavlju 4.2.

Narednom teoremom se potvrđuje linearna konvergencija *TADSS* iteracije. Dokaz se može naći u [1].

Teorema 4.8. *Ako je f dva puta neprekidno diferencijabilna i uniformno konveksna funkcija na $\mathbb{R}^n$ i niz $\{x_k\}$ generisan Algoritmom* (4.4.2) *tada je*

$$\lim_{k\to\infty} \|g_k\| = 0, \tag{4.91}$$

a niz $\{x_k\}$ konvergira ka x^* bar linearno.

Sledećom lemom i teoremom se opisuje konvergencija *TADSS* modela na skupu striktno konveksnih kvadratnih funkcija koje su definisane jednakošću

$$f(x) = \frac{1}{2}x^T Ax - b^T x, \tag{4.92}$$

Iz analize konvergencije *SM*, *ADD* i *ADSS* metoda na skupu striktno konveksnih kvardatnih funkcija, znamo da je matrica A iz prethodne relacije realna $n \times n$ simetrična pozitivno definitna matrica dok je $b \in \mathbb{R}^n$. I ovde pretpostavljamo da za n sopstvenih vrednosti matrice A označenih sa $\lambda_1, \lambda_2, \cdots, \lambda_n$ važi uređenje: $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. U dokazu konvergencije na ovom skupu funkcija, slično kao kod analize konvergencije kod opisanih ubrzanih gradijentnih metoda iz prethodnih poglavlja, potrebno je postaviti dodatni uslov nad najmanjom i najvećom sopstvenom vrednošću.

Lema 4.9. *Primenom metoda pada po gradijentu definisanog relacijom* (4.86) *u kojoj su parametri* γ_k *i* α_k *određeni jednakošću* (4.89) *i Algoritmom* (4.4.1) *na striktno konveksne kvadratne funkcije definisane izrazom* (4.92) *u kome je* $A \in \mathbb{R}^{n\times n}$ *simetrična pozitivno definitna matrica, važe sledeće nejednakosti*

$$\frac{1}{2\lambda_n} \leq \alpha_{k+1}(\gamma_{k+1}^{-1} - 1) + 1 \leq \frac{1}{\lambda_1} \tag{4.93}$$

gde su λ_1 *i* λ_n *respektivno najmanja i najveća sopstvena vrednost matrice* A.

Dokaz. Prema relaciji (4.92), razlika vrednosti objektne funkcije u prethodnoj i tekućoj iterativnoj tački je

$$f(x_{k+1}) - f(x_k) = \frac{1}{2}x_{k+1}^T A x_{k+1} - b^T x_{k+1} - \frac{1}{2}x_k^T A x_k + b^T x_k. \tag{4.94}$$

Primenom izraza (4.86) koji definiše *TADSS* iteraciju dobijamo

$$x_{k+1} = x_k - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k. \tag{4.95}$$

Dalje je

$$\begin{aligned}
f(x_{k+1}) - f(x_k) &= \frac{1}{2}\left(x_k - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k\right)^T A \left(x_k - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k\right) \\
&\quad - b^T \left(x_k - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k\right) - \frac{1}{2}x_k^T A x_k + b^T x_k \\
&= -\frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1]g_k^T A x_k - \frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1]x_k^T A g_k \\
&\quad + \frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1]^2 g_k^T A g_k + [\alpha_k(\gamma_k^{-1} - 1) + 1]b^T g_k.
\end{aligned}$$

Kako gradijent striktno konveksne kvadratne funkcije (4.92) iznosi $g_k = Ax_k - b$ i kako važi jednakost $g_k^T A g_k = g_k A g_k^T$ sledi da je:

$$
\begin{aligned}
f(x_{k+1}) - f(x_k) &= -\frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1](g_k^T A x_k + x_k^T A g_k - [\alpha_k(\gamma_k^{-1} - 1) + 1] g_k^T A g_k - 2b^T g_k) \\
&= -\frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1](g_k^T (A x_k - b^T) + g_k^T (A x_k - b^T) - [\alpha_k(\gamma_k^{-1} - 1) + 1] g_k^T A g_k) \\
&= -[\alpha_k(\gamma_k^{-1} - 1) + 1] g_k^T g_k + -\frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1]^2 g_k^T A g_k.
\end{aligned}
$$

Zamenom dobijene relacije u izraz koji određuje γ_{k+1} (4.89), imamo da važi

$$
\begin{aligned}
\gamma_{k+1} &= 2\frac{-[\alpha_k(\gamma_k^{-1} - 1) + 1] g_k^T g_k + \frac{1}{2}[\alpha_k(\gamma_k^{-1} - 1) + 1]^2 g_k^T A g_k + [\alpha_k(\gamma_k^{-1} - 1) + 1] g_k^T g_k}{[\alpha_k(\gamma_k^{-1} - 1) + 1]^2 g_k^T g_k} \\
&= \frac{g_k^T A g_k}{g_k^T g_k}.
\end{aligned}
$$

Poslednja relacija potvrđuje činjenicu da je γ_{k+1} Rejlijev količnik realne simetrične matrice A u vektoru g_k, pa iz tog razloga važi naredna ocena:

$$
\lambda_1 \leq \gamma_{k+1} \leq \lambda_n, \quad k \in \mathbb{N}. \tag{4.96}
$$

Uzevši u obzir da je $0 \leq \alpha_{k+1} \leq 1$ dokaz desne strane nejednakosti u (4.93) direktno sledi:

$$
\alpha_{k+1}(\gamma_{k+1}^{-1} - 1) + 1 \leq \gamma_{k+1}^{-1} - 1 + 1 = \frac{1}{\gamma_{k+1}} \leq \frac{1}{\lambda_1}.
$$

Da bi smo dokazali levu stranu nejednakosti (4.93), razmatramo ocenu

$$
\alpha_k > \frac{\beta\,(1-\sigma)\,\gamma_k}{L} \tag{4.97}
$$

koja je dokazana u [1]. Prema usvojenim oznakama iz ovog poglavalja prethodna nejednakost (4.97) postaje:

$$
\alpha_k > \frac{\eta\,(1-\sigma)\,\gamma_k}{L}. \tag{4.98}
$$

Iz nejednakosti (4.98) i činjenice da je $\eta \in (\sigma, 1), \sigma \in (0, 0.5), \alpha_{k+1} \in (0, 1)$ izvodimo sledeći zaključak:

$$\alpha_{k+1}(\gamma_{k+1}^{-1}-1)+1=\frac{\alpha_{k+1}}{\gamma_{k+1}}-\alpha_{k+1}+1>\frac{\alpha_{k+1}}{\gamma_{k+1}}-\alpha_{k+1}\geq\frac{\alpha_{k+1}}{\gamma_{k+1}}\geq\frac{\eta\,(1-\sigma)}{L}\geq\frac{1}{2L} \tag{4.99}$$

Lipšicovu konstantu L iz poslednje nejednakosti možemo zameniti vrednošću najveće sopstvene vrednosti matrice A, λ_n. Da λ_n ima svojstva Lipšicove konstante L pokazujemo narednom analizom. Kako je matrica A simetrična i $g(x)=A(x)-b$, važi da je:

$$\|g(x)-g(y)\|=\|Ax-Ay\|=\|A(x-y)\|\leq\|A\|\|x-y\|=\lambda_n\|x-y\|. \tag{4.100}$$

Zamenom konstante L vrednošću λ_n, nejednakost (4.99) postaje:

$$\frac{1}{2\lambda_n}\leq\alpha_{k+1}(\gamma_{k+1}^{-1}-1)+1.$$

a ovo dokazuje levu stranu nekjednakosti (4.93). □

Napomena 4.4. *Upoređujući rezultate ocena dobijenih iz slično formulisanih lema u radovima [1, 6, 7] , može se primetiti da ocena iz prethodne Leme 4.93 koja se odnosi na TADSS iteraciju zavisi samo od vrednosti najmanje i najveće sopstvene vrednosti matrice A, λ_1 i λ_n, ali ne i od vrednosti parametra σ iz backtracking procedure.*

Teorema 4.9. *Za striktno konveksnu kvadratnu funkciju f definisanu relacijom* (4.92) *i metod pada po gradijentu* (4.86), *pod datim uslovom koji važi za najmanju i najveću sopstvenu vrednost matrice A*

$$\lambda_n<2\lambda_1, \tag{4.101}$$

važi da je

$$\left(d_i^{k+1}\right)^2\leq\delta^2\left(d_i^k\right)^2 \tag{4.102}$$

pri čemu je

$$\delta=\max\left\{1-\frac{\lambda_1}{2\lambda_n},\frac{\lambda_n}{\lambda_1}-1\right\}, \tag{4.103}$$

i pri tom je

$$\lim_{k\to\infty}\|g_k\|=0. \tag{4.104}$$

Dokaz. Neka su $\{v_1,v_2,\ldots,v_n\}$ ortonormirani vektori simetrične pozitivno definitne matrice A. Pretpostavimo da je niz $\{x_k\}$ niz vrednosti dobijenih primenom Algoritma (4.4.2). Gradijent striktno konveksne kvadratne funkcije (4.92) je $g_k = Ax_k - b$ i može se prikazati na sledeći način:

$$g_k = \sum_{i=1}^{n} d_i^k v_i, \tag{4.105}$$

za neke realne konstante $d_1^k, d_2^k, \ldots, d_n^k$ i neko k. Primenom (4.86)

$$g_{k+1} = A(x_k - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k) - b = g_k - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k = I - [\alpha_k(\gamma_k^{-1} - 1) + 1]g_k, \tag{4.106}$$

na reprezentaciju (4.105) dobijamo da važi

$$g_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i = \sum_{i=1}^{n} \left(1 - [\alpha_k(\gamma_k^{-1} - 1) + 1]\lambda_i\right) d_i^k v_i. \tag{4.107}$$

Da bi smo dokazali da je ispunjen uslov (4.102), dovoljno je pokazati da je $|1 - [\alpha_k(\gamma_k^{-1} - 1) + 1]\lambda_i| \leq \delta$. Prethodnu nejednakost dokazujemo u dva slučaja. U prvom pretpostavimo da je $\lambda_i[\alpha_k(\gamma_k^{-1} - 1) + 1] \leq 1$. Uzevši u obzir (4.93) dobijamo:

$$1 > \lambda_i[\alpha_k(\gamma_k^{-1} - 1) + 1] \geq \frac{\lambda_1}{2\lambda_n} \quad \Longrightarrow \quad 1 - \lambda_i[\alpha_k(\gamma_k^{-1} - 1) + 1] \leq 1 - \frac{\lambda_1}{2\lambda_n} \leq \delta \tag{4.108}$$

U drugom slučaju, pretpostavka je da je $\lambda_i[\alpha_k(\gamma_k^{-1} - 1) + 1] \geq 1$. Iz datog uslova sledi da je:

$$1 < \lambda_i[\alpha_k(\gamma_k^{-1} - 1) + 1] \leq \lambda_n \frac{1}{\lambda_1} \quad \Longrightarrow \quad |\lambda_i[\alpha_k(\gamma_k^{-1} - 1) + 1] - 1| \leq \frac{\lambda_n}{\lambda_1} - 1 \leq \delta. \tag{4.109}$$

Iz (4.105) sledi:

$$\|g_k\|^2 = \sum_{i=1}^{n} (d_i^k)^2. \tag{4.110}$$

Kako parametar δ iz (4.102) pod dodatnom pretpostvakom (4.101) zadovoljava uslov $0 < \delta < 1$, kao posledicu imamo da važi (4.104). $\square$

U numeričkim istraživanjima testostovi su sprovedeni nad 22 test funkcije predloženih u [34]. Kao i u [1, 7, 6] eksperimenti su vršeni sa izborom većeg broja promenljivih: 1000, 2000, 3000, 5000, 7000, 8000, 10000, 15000, 20000 i 30000. Korišćen je isti kriterijum završetka kao u [6, 7, 1]. U backtracking proceduri uzeti su parametri sa vrednostima $\sigma = 0.0001, \eta = 0.8$.

Kako je *TADSS* šema nastala redukcijom *ADSS* metoda na jedan korak, kao i u [7] analizrane su sve tri karakterisike koje jasno određuju svojstva definisanog algoritma. Te karakteristike su broj iteracija, CPU vreme i broj određivanja vrednosti funkcije. *TADSS* metod kao specijalan slučaj *ADSS* šeme upoređen je sa *ADSS* metodom. Podsetimo se da je u [1] potvrđena dominacija *SM* metoda u odnosu na *GD* i *AGD* algoritme. Numerički rezultati iz [7] pokazali su da *ADSS* model pokazuje bolje svojstva u odnosu na poredbene algoritme: *ADD* i *SM*. Iz tog razloga prilikom numeričkih istraživanja vezanih za *TADSS* metod bilo je bitno uporediti novokonstrusani model sa njegovim prethodnikom - *ADSS* modelom. Pri tom, važno je naglasiti da je jedna od osnovnih svrha numeričkog istraživanja bilo ispitivanje održivosti dobrih performansi jednokoračne modifikacije *ADSS* i eventualno njihovo dalje poboljšanje.

Dobijeni numerički rezultati, koji se odnose na broj iteracija, CPU vreme i broj određivanja vrednosti objektne funkcije, prikazani su u Tabeli 4.9.

Tabela 4.9. *Prikaz numeričkih rezultata testiranja* TADSS *i* ADSS *za* 22 *test funkcije.*

Test funkcije	Broj iteracija		CPU vreme		Broj evaluacija funkcije	
	TADSS	ADSS	TADSS	ADSS	TADSS	ADSS
Extended Penalty	40	50	2	4	1280	1780
Raydan-1	466	34	11	4	8504	4844
Diagonal 1	20	37	0	0	335	1448
Diagonal 3	21	49	0	1	417	1048
Generalized Tridiagonal - 1	61	77	0	0	431	719
Extended Tridiagonal - 1	60	70	0	0	250	420
Extended Three Exponï£¡	40	40	0	0	400	350
Diagonal 4	40	780	0	0	270	2590
Extended Himmelblau	60	70	0	0	300	480
Quadratic QF1	4953	425	15	0	13738	1755
Extended Quad. Penalty QP1	50	60	0	2	571	841
Extended Quad. Penalty QP2	50	60	0	4	569	843
Quadratic QF2	50	60	0	1	583	836
Extended EP1	186	40	1	0	758	487
Extended Tridiagonal - 2	638	80	0	0	2052	420
Arwhead	50	64	0	3	601	1082
Engval1	60	70	0	0	290	460
Quartc	10	70	0	0	30	390
Generalized quartic	60	70	0	0	250	614
Diagonal 7	189	2201	0	33	566	6633
Diagonal 8	160	2213	1	40	586	6709
Diagonal 9	20	43	0	0	352	3646

Prikazani rezultati, pokazuju izvesno poboljšanje u korist *TADSS* metoda po pitanju sva tri svojstva. Preciznije, u pogledu broja iteracija, *TADSS* daje bolje rezultate u slučaju 17 funkcija od ukupno 22, dok *ADSS* nadmašuje *TADSS* u slučaju 4 test funkcije. Prilikom testiranja Extended Three Exponential Terms funkcije, oba metoda su proizvela jednak broj iteracija.

Rezultati koji se odnose na vreme izvršenja potvrđuju da su oba modela izuzetno brza. U 9 od 22 slučaja *TADSS* je brži od *ADSS*,u 2 od 22 testiranja *ADSS* je brži od *TADSS* a čak u polovini ukupnog broja testiranja CPU vreme za obe iteracije je jednako nuli.

Po pitanju broja određivanja vrednosti objektne funkcije, *TADSS* algoritam daje manje vrednosti od *ADSS* algoritma u 17 od 22 testiranja a u obrnutom slučaju je 5 od 22.

Naredna Tabela 4.10 ilustruje prosečne vrednosti ostvarenih rezultata.

Tabela 4.10. *Prosečne vrednosti numeričkih rezultata TADSS i ADSS za* 220 *testova za svaki od testiranih modela kod* 22 *test funkcije za* 10 *numeričkih eksperimenata u svakoj iteraciji.*

Prosečne performanse	TADSS	ADSS
Broj iteracija	331.09	302.86
CPU vreme (u secundama)	1.36	4.18
Broj evaluacija funkcije	1506.05	1745.23

Iako rezultati iz Tabele 4.10 pokazuju bolje performanse *ADSS* u 17 od 22 testiranja po pitanju broja iteracija, analizom prosečnih vrednosti opšta zapažanja pokazuju blagu dominaciju *ADSS* metoda po ovom pitanju. Nasuprot ovoj analizi, analiza prosečnog broja određivanja vrednosti funkcije pokazuje boljitak u korist *TADSS* šeme. Konačno, prosečno vreme izvršavanja numeričkih eksperimenata pokazuje da je *TADSS* modifikacija oko tri puta brža od svog originala - *ADSS* metoda. Generalni zaključak je da jednokoračna verzija *ADSS* metoda, konstruisana *TADSS* šema, ima svojstva blago poboljšana u odnosu na originalnu *ADSS* iteraciju, posebno po pitanju brzine izvršenja.

Urađena su i dodatna numerička istraživanja kojima se prikazuje poređenje *TADSS* i *SM* iteracija. Potreba za ovim dodatnim testiranjima javila se iz razloga koji je već napomenut a to je da su oba od pomenutih modela jednokoračna ubrzana i imaju pad po gradijentu. Težnja je bila da se numeričkim putem potvrdi da jednokoračni ubrzani gradijentni *TADSS* metod, izveden iz dvokoračnog *ADSS* modela, pokazuje daleko bolje performanse u pogledu svih analiziranih aspekata u odnosu na klasičnim putem definisan ubrzani gradijentni *SM* metod. Ova pretpostavka je potvrđena a tome svedoče prikazani numerički rezultati iz Tabele 4.11 za 30 test funkcija.

Tabela 4.11. *Prikaz numeričkih rezultata testiranja* TADSS *i* SM *za* 30 *test functions.*

Test funkcije	Broj iteracija		CPU vreme		Broj evaluacija funkcije	
	TADSS	SM	TADSS	SM	TADSS	SM
Extended Penalty	40	589	2	5	1280	2987
Perturbed quadratic	4276	111972	13	1868	12017	632724
Raydan-1	466	21125	11	178	8504	116757
Diagonal 1	20	10417	0	116	335	56135
Diagonal 3	21	10574	0	209	417	59425
Generalized Tridiagonal - 1	61	278	0	2	431	989
Extended Tridiagonal - 1	60	3560	0	35	250	30686
Extended Three Exponï£¡	40	164	0	1	400	1224
Diagonal 4	40	80	0	0	270	530
Extended Himmelblau	60	168	0	0	300	566
Quadr.Diag.Perturbed	11542	53133	58	1193	56359	547850
Quadratic QF1	4953	114510	15	2127	13738	649643
Extended Quad. Penalty QP1	50	224	0	12	571	2566
Extended Quad. Penalty QP2	50	162	0	7	569	2057
Quadratic QF2	50	118801	0	2544	583	662486
Extended EP1	186	68	1	1	758	764
Extended Tridiagonal - 2	638	584	0	0	2052	2144
Arwhead	50	10	0	0	601	30
Almost Perturbed Quadratic	4202	110121	15	2148	14974	627287
Engval1	60	185	0	7	290	2177
Quartc	10	190	0	0	30	430
Generalized quartic	60	156	0	0	250	423
Diagonal 7	189	90	0	0	566	220
Diagonal 8	160	96	1	0	586	594
Diagonal 9	20	11235	0	118	352	60566
DIXON3DQ	10	112899	0	1908	30	639957
NONSCOMP	10	10	0	0	30	30
HIMMELH	10	30	0	0	40	80
Power Cute	1455	$> t_e$	4	$> t_e$	4567	$> t_e$
Indef	20	$> t_e$	0	$> t_e$	50	$> t_e$

Iz priloženog se može zaključiti da po pitanju broja iteracija *TADSS* metod daje bolje rezultate od *SM* metoda u 24 od 30 slučaja, dok je *SM* bolji samo u 5 od 30 testiranih funkcija. Za NONSCOMP test funkciju oba modela daju isti broj iteracija. U pogledu vremenskog trajanja izvršenja testiranja, oba metoda daju jednake rezultate kod 10 test funkcija. U slučaju 19 test funkcija *TADSS* metod je brži od *SM* metoda dok obrnut slučaj važi samo po pitanju jedne test funkcije. Impozantni su rezltati u korist *TADSS* modela koji prikazuji broj određivanja vrednosti objektne funkcije. Po ovom pitanju, *TADSS* daje bolje vrednosti kod 27 od 30 test funkcija, dok obrnuti slučaj važi samo za dve test funkcije. Prilikom testiranja NONSCOMP test funkcije oba modela daju jednak broj određivanja vrednosti. Kao što se vidi iz Tabele 4.11, za dve od predloženih 30 test funkcija, testiranja *SM* metodom traju duže od konstante vremenskog ograničenja t_e definisane u [6], dok kod svih 30 ilustrovanih test funkcija, testiranja *TADSS* metodom traju daleko kraće od t_e.

Još generalniju sliku boljih performansi *TADSS* metoda u odnosu na *SM* metod prikazuju rezulati prosečnih vrednosti 28 test funkcija iz Tabele 4.11 za koje je bilo moguće testirati *SM* metod shodno definisanoj t_e konstanti. Ovi rezultati su prikazani u naradnoj Tabeli 4.12.

Tabela 4.12. *Prosečne vrednosti numeričkih rezultata TADSS i SM metoda za* 280 *testova za svaki od testiranih modela kod* 28 *test funkcija za* 10 *numeričkih eksperimenata u svakoj iteraciji.*

Prosečne performanse	TADSS	SM
Broj iteracija	976.21	24336.82
CPU vreme (u secundama)	4.14	445.68
Broj evaluacija funkcije	4163.68	146475.96

Rezultati iz poslednje tabele pokazuju da je primenom *TADSS* metoda potrebno u proseku skoro 25 puta manje iteracija nego primenom *SM* metoda i čak se prosečno 35 puta manje izvrši računanje vrednosti objektne funkcije primenom *TADSS* šeme. Konačno, rezultati iz Tabele 4.12 potvrđuju da testiranja *TADSS* metodom traju u proseku čak 107 puta kraće nego ista testiranja urađena primenom *SM* algoritma.

Kodovi upotrebljeni za numeričke eksperimente su napisani u Visual C++ programskom jeziku, a testiranja su sprovedena na računaru Workstation Intel 2.2 GHz.

4.5 Zaključna razmatranja. Sistematizacija dvosmernih i dvokoračnih metoda i predlozi za dalja istraživanja

Istraživanaja predstavljena u ovoj disertaciji donela su značajni pomak u poboljšanju svojstava gradijentnih metoda bezuslovne optimizacije. Oba koncepta, kako dvosmerni tako i dvokoračni doprinela su opštem unapređenju gradijentnih metoda. Parametar ubrznja, označen sa γ_k, uveden kod svih predstavljenih modela jeste jedan od najvažnijih elemenata svih novodefinsanih algoritama. Otuda je u nazivu svakog od njih opravdan pridev 'ubrzani'. U Poglavlju 3.5 istaknuto je da su autori u [1] detektovali klasu ubrzanih gradijentnih metoda, originalno nazvanom *class of accelerated gradient methods*. Metodi ove klase generalno imaju osobinu da u svojim formulacijama sadrže parametar ubrzanja kao multiplikativni faktor vrednosti iterativnog koraka. Određivanje ovog parametara, kako je moglo da se primeti iz predstavljenih analiza, uglavnom je bazirano na Tejlorovom razvoju drugog reda i adekvatnoj aproksimaciji inverza Hesijana. Neki od načina definisanja ovog značajnog faktora opisani su u Poglavljima 3.4,3.5,4.2,4.3 i 4.4.

U originalnim radovima ove disertacije, predstavljen ubrzani dvosmerni *ADD* metod doprineo je smanjenju broja iteracije. Ubrzani dvokoračni *ADSS* model upospešio je sve tri analizirane karakteristike jednog gradijentnog metoda: broj iteracija, CPU vreme i broj određivanja vrednosti objektne funkcije. Izuzetni pomak u performansama *ADSS* iteracije bio je motivacija za dalje proučavanje ovog dvokoračnog algoritma. Postavljanjem dodatnog uslova koji povezuje dva iterativna koraka iz *ADSS* šeme, ubrzani dvokoračni *ADSS* metod je transformisan u jednokoračni i označen sa *TADSS*. *TADSS* iteracija takođe pripada klasi ubrzanih gradijentnih metoda. *TADSS* algoritam je održao i blago poboljšao već dobra svojstva svog dvokoračnog originala, *ADSS* metoda. Sa druge strane je u mnogome nadmašio po pitanju svih praćenih karakteristika, primarno jednokoračni *SM* metod koji sam pripada klasi ubrzanih gradijentnih metoda.

Rezultati svih dosadašnjih proučavanja dvosmernih a posebno dvokoračnih algoritama opravdala su svoju svrhu i našla primenu i kod metoda sa jednim smerom traženja i jednim iterativnim korakom. Pri tome su se otvorile mogućnosti za poboljšanje karakteristika postojećih jednosmernih i jednokoračnih algoritama a na temeljima novodefinisanih dvokoračunih iteracija.

Zbog značajnih dostignuća primenom koncepta dvosmernih i dvokoračnih metoda otvaraju se nove mogućnosti i ideje za dalje izučavanje sličnih modela. Uz alteranativno zadržavanje ubrzanog svojstva moguće je detektovati klasu dvosmernih i dvokoračnih metoda opšte formulacije

$$x_{k+1} - x_k = \beta_k[\theta_k]d_k - \alpha_k[\gamma_k{}^{-1}]g_k, \tag{4.111}$$

Parametri iz prethodne relacije dati u uglastim zagradama, θ_k i γ_k predstavljaju opcione variable ubrzanja. Imajući u vidu prethodnu relaciju zaključujemo da ukoliko u izrazu (4.111) postavimo $\theta_k = 1$, $\beta_k = \alpha_k^2$ dobijamo *ADD* metod. Za $d_k = -g_k$ i $\theta_k = 1$ relacija (4.111) predstavlja *ADSS* iteraciju. Ukoliko pri tom u (4.111) postavimo dodatan uslov nad vrednostima iterativnih koraka α_k i β_k, $\alpha_k + \beta_k = 1$ dobijamo *TADSS* iterativnu gardijentnu šemu.

Jasno je da ima dosta prostora i razloga za dalja proučavanja dvosmernih i/ili dvokoračnih modela opšteg oblika 4.111. Ostvareni rezultati predstavljaju dobru motivaciju i povod za nova istraživanja u vezi ove teme.

Literatura

[1] P. S. Stanimirović and M. B. Miladinović, "Accelerated gradient descent methods with line search," *Numer. Algor.*, vol. 54, pp. 503–520, 2010.

[2] N. I. Djuranović-Milicić and M. Gardasević-Filipović, "A multi-step curve search algorithm in nonlinear optimization: Nondifferentiable convex case," *Facta Universitatis, Ser. Math. Inform*, vol. 25, pp. 11–24, 2010.

[3] N. Andrei, "An acceleration of gradient descent algorithm with backtracking for unconstrained optimizatio," *Numerical Algorithms*, vol. 42, no. 1, pp. 63–73, 2006.

[4] P. S. M. Miladinović and S. Miljković, "Scalar correction method for solving large scale unconstrained minimization problems," *J Optim Theory Appl*, vol. 151, p. 304âĂŞ320, 2011.

[5] B. Barzilai and J. M. Borwein, "Two point step-size gradient method," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.

[6] M. J. Petrović and P. S. Stanimirović, "Accelerated double direction method for solving unconstrained optimization problems," *Mathematical Problems in Engineering*, vol. 2014, no. 8, April 2014.

[7] M. J. Petrović, "An accelerated double step size method in unconstrained optimization," *Applied Mathematics and Computation*, vol. 250, pp. 309–319, January 2015.

[8] P. S. Stanimirović, G. Milovanović, and M. J. Petrović, "Transformation of the accelerated double step size method," *under review*.

[9] J. E. Deniss and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Englewood Cliffs, New Jerssey, 1983.

[10] R. Fletcher, *Practical methods of optimization*. Wiley, New York, 198y.

[11] S. L. S. Jacoby, J. S. Kowalik, and J. T. Pizzo, *Iterative Methods for Nonlinear Optimization Problems*. Prentice-Hall, Inc, Englewood, New Jersey, 1977.

[12] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer-Verlag New York, Inc, 1999.

[13] M. J. D. Powel, "A survey of numerical methods for unconstrained optimization," *SIAM Rev.*, vol. 12, pp. 79–97, 1970.

[14] S. Stojanov, *Metodi i Algoritmi za Optimizacija*. Drzavno izdatelstvo, Tehnika, Sofija, 1990.

[15] S. Zlobec, *Nelinearno programiranje*. Naucna knjiga, Beograd,, 1989.

[16] B. Molina and M. Raydan, "Preconditioned barzilai-borwein method for the numerical solution of partial differential equations," *Numer. Algor.*, vol. 13, pp. 45–60, 1996.

[17] M. Raydan, "The barzilai and borwein gradient method for the large scale unconstrained minimization problem," *SIAM J. Optim.*, vol. 7, pp. 26–33, 1997.

[18] ——, "On the barzilai and borwein choice of steplength for the gradient method," *IMA J. Numer. Anal.*, vol. 13, pp. 321–326, 1993.

[19] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *Comput. J.*, vol. 7, pp. 149–154, 1964.

[20] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equation in Several Variables*. Academic, London, 1970.

[21] E. Polak and G. Ribiére, "Note sur la convergence de méthodes de directions conjuguées," *Revue Francaise Informat. Researche Opérationnelle, 3e Année*, vol. 16, pp. 35–43, 1969.

[22] D. G. Luenberg and Y. Ye, *Linear and nonlinear programming*. Springer Science+Business Media, LLC, New York, 2008.

[23] W. Sun and Y. X. Yuan, *Optimization Theory and Methods: Nonlinear Programming*. Springer, New York, 2006.

[24] Y. H. Dai and R. Fletcher, "On the asymptotic behaviour of some new gradient methods," *Numerical Analysis Report, NA/212, Dept. of Math. University of Dundee, Scotland, UK*, 2003.

[25] Y. H. Dai, J. Y. Yuan, and Y. Yuan, "Modified two-point step-size gradient methods for unconstrained optimization," *Computational Optimization and Applications*, vol. 22, pp. 103–109, 2002.

[26] Y. H. Dai and Y. Yuan, "Alternate minimization gradient method," *IMA Journal of Numerical Analysis*, vol. 23, pp. 377–393, 2003.

[27] ——, "Analysis of monotone gradient methods," *J. Industrial and Management Optimization*, vol. 1, pp. 181–192, 2005.

[28] Y. H. Dai and H. Zhang, "Adaptive two-point stepsize gradient algorithm," *Numer. Algor.*, vol. 27, pp. 377–385, 2001.

[29] Y. H. Dai and L. Z. Liao, "R-linear convergence of barzilai and borwein gradient method," *IMA J. Numer. Anal.*, vol. 22, no. 1, pp. 1–10, 2002.

[30] A. Friedlander, J. M. Martinez, B. Molina, and M. Raydan, "Gradient method with retards and generalizations," *SIAM J. Numer. Anal.*, vol. 36, pp. 275–289, 1999.

[31] M. J. D. Powell, "Some global convergence properties of a variable-metric algorithm for minimization without exact line search," *AIAM-AMS Proc., Philadelphia*, vol. 9, pp. 53–72, 1976.

[32] Z. J. Shi, "Convergence of line search methods for unconstrained optimization," *App. Math. Comput.*, vol. 157, pp. 393–405, 2004.

[33] N. Andrei, "An accelerated conjugate gradient algorithm with guaranteed descent and conjugacy conditions for unconstrained optimization," *Optimization Methods and Software*, vol. 27, no. 4-5, pp. 583–606, 2012.

[34] ——, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, pp. 147–161, 2008.

[35] C. Lemaréchal, "A view of line search," *In: Auslander, A., Oetti, W., Stoer J. (eds.) Optimization and Optimal Control, Springer, Berlin*, pp. 59–78, 1981.

[36] J. J. Moré and D. J. Thuente, *On line search algorithm with guaranteed sufficient decrease*. Mathematics and Computer Science Division Preprint MSC-P153-0590, Argone National Laboratory, Argone, 1990.

[37] F. A. Potra and Y. Shi, "Efficient line search algorithm for unconstrained optimization," *J. Optim.Theory Appl.*, vol. 85, pp. 677–704, 1995.

[38] J. E. Dennis and J. J. Moré, "Quasi-newton methods, motivation and theory," *SIAM Rev.*, vol. 19, pp. 46–89, 1977.

[39] Y. H. Dai, "On the nonmonotone line search," *J. Optim. Theory Appl.*, vol. 33, pp. 315–330, 2002.

[40] P. S. Stanimirović and M. B. Miladinović, *Nelinearno programranje*. PMF Nis, 2015.

[41] A. Cauchy, "Metode generale pour la resolution des systems d'equations simultanees," *Comp. Rend. Sci.*, vol. 25, pp. 46–89, 1847.

[42] L. Armijo, "Minimization of functions having lipschitz first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, pp. 1–3, 1966.

[43] A. I. Cohen, "Stepsize analysis for descent methods," *J. Optim. Theory Appl.*, vol. 33, pp. 187–205, 1981.

[44] A. A. Goldstein, "On steepest descent," *SIAM J. Control*, vol. 3, pp. 147–151, 1965.

[45] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Rev.*, vol. 11, pp. 226–235, 1968.

[46] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for newton's method," *SIAM J. Numer. ANAL.*, vol. 23, pp. 707–716, 1986.

[47] ——, "A class of nonmonotone stability methods in unconstrained optimization," *Numerische Mathematik*, vol. 62, pp. 779–805, 1991.

[48] ——, "A truncated newton method with non-monotone line search for unconstrained optimization," *J. Optim. Theory Appl.*, vol. 60, pp. 401–419, 1989.

[49] J. Y. H. W. Y. Sun and J. Sun, "Global convergence of nonmonotone descent methods for unconstrained optimization problems," *J. Comp. Appl. Math.*, vol. 146, pp. 89–98, 2002.

[50] P. L. Toint, "A nonmonotone trust region algorithm for nonlinear optimization subject to convex constraints," *Math. Prog.*, vol. 77, pp. 69–94, 1997.

[51] M. Raydan and B. F. Svaiter, "Relaxed steepest descent and cauchy-barzilai-borwein method," *Comput. Optim. Appl.*, vol. 21, pp. 155–167, 2001.

[52] N. Andrei, "A new gradient descent method for unconstrained optimization," *http://www.ici.ro/camo/neculai/newstep.pdf*, April 2004.

[53] M. N. Vrahatis, J. N. L. G. S. Androulakis, and G. D. Magoulas, "A class of gradient unconstrained minimization algorithms with adaptive step-size," *J. Comp. and Appl.Math.*, vol. 114, pp. 367–386, 2000.

[54] W. Glunt, T. L. Hayden, and M. Raydan, "Molecular conformations from distance matrices," *J. Comput. Chem.*, vol. 14, pp. 114–120, 1993.

[55] ——, "Preconditioners for distance matrix algorithms," *J. Comput. Chem.*, vol. 15, pp. 227–232, 1994.

[56] R. Fletcher, "On the barzilai borwein method," *Numerical Analysis Report NA*, vol. 207, 2001.

[57] C. Brezinski, "A classification of quasi-newton methods," *Numer. Algorithms*, vol. 33, pp. 123–135, 2003.

[58] F. H. Clarke, *Optimization and Nonsmooth Analysis*. Wiley, New York, 1983.

[59] N. I. Djuranović-Milicić and M. Gardasević-Filipović, "An algorithm for minimization of a nondifferentiable convex function," in *The 2009 International Conference of Applied and Engineering Mathematics, World Congress on Engineering 2009, Proceedings*, Beograd, Srbija, 2009, pp. 1241–1246.

[60] A. Fuduli, *Metodi Numerici per la Minimizzazione di Funzioni Convesse Nondifferenziabili, PhD Thesis*. Departimento di Electronica Informatica e Sistemistica, Univerzita della Calabria, 1998.

[61] L. Qi, *Second Order Analysis of the Moreau-Yoshida Regularization*. School of Mathematics, The University of New Wales Sydney, New South Wales 2052, Australia, 1998.

[62] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, Princeton, 1970.

BIOGRAFIJA

Milena Petrović je rođena u Nišu 24.12.1975. godine. Stalno mesto prebivališta je Zelengorska 6/32, 18000 Niš.

Gimnaziju ''Drakče Milovanović" prirodno-matematički smer, završila je 1994. godine kada i upisuje osnovne studije na Prirodno-matematičkom fakultetu Univerziteta u Beogradu, na Odseku za numeričku matematiku i optimizaciju.

Master studije završava 22.12 2006. godine na Univerzitetu u Lundu, Švedska, na katedri za numeričku analizu, pod mentorstvom profesora Akima Šrola (*Achim Schroll*), odbranom master teze pod nazivom : *A Truly Third Order Finite Volume Scheme On Quadrilateral Mesh.*

Doktorske studije je upisala 24.11. 2008. godine na Prirodno-matematičkom fakultetu u Nišu, na studijskom programu Matematika.

Svoj prvi radni odnos zasnovala je 2002. godine u osnovnoj školi "Ratko Vukićević" u Nišu kao nastavnik matematike. Na Prirodno-matematičkom fakultetu Univerziteta u Prištini sa privremenim sedištem u Kosovskoj Mitrovici, Odsek za matematiku, stekla je zvanje asistenta 30.09.2009 godine kada je i zasnovala radni odnos na istom fakultetu.

Do sada je objavila 8 naučnih radova, od toga dva u stranim časopisima međunarodnog značaja (M21) iz oblasti matematike. Rezultate svojih istraživanja saopštila je na četiri međunarodne naučne konferencije.

Majka Anastasije i Vasilija.

BIBLIOGRAFIJA

1. Spisak prihvaćenih i objavljenih naučnih radova

1.1. Radovi u časopisima međunarodnog značaja

1.1.1 (M21) **Milena J. Petrovic,** *Accelerated Double Stepsize Method in Unconstrained Optimization*, Applied Mthematics and Computation, pp 309-319, Volume 250, January 2015 (IF 2014=1.672)

1.1.2 (M21) **Milena J. Petrovic**, Predrag S. Stanimirovic, *Accelerated Double Direction Method For Solving Unconstrained Optimization Problems,* Mathematical Problems in Engineering, Volume 2014 , April 2014 (IF 2014=1.383)

1.1.3 (M24) Ivan Krstić, Negovan Stamenković, **Milena Petrović**, Vidosav Stojanović, *Binary to RNS encoder with Modulo* $2^n + 1$ *Channel in Diminished-1 Number System*, International Journal of Computational Engineering and Management, accepted

1.1.4 (M24) Srdjan Jovkovic, Dejan Milic, Danijel Djosic, **Milena Petrovic**, Stanislav Veljkovic, Caslav Stefanovic,” Level Crossing Rate of L-Branch SC Receiver over α-k-μ Fading Channel in the Presence α-k-μ Co-Channel Interference”, WSEAS Transactions on Communications, Volume 13, pp. 249-255, 2014

1.2. Radovi objavljeni u celosti u zbornicima sa međunarodnih konferencija

1.2.1 (M33) Petkovic D.,**Petrović, M**. *A Truly Third Order Finite Volume Scheme On The Quadrilateral Mesh,* Međunarodna konferencija"Matematičke i informacione tehnologije" - MIT 2009, Kopaonik 27. – 31. avgust.; Budva 31. avgust – 5. septembar 2009., Zbornik radova, pp. 293-302, ISBN 978-86-83237-90-6.

1.2.2 (M33) **Petrović,M.** *Local Double Logarithmic Technique,* Međunarodna konferencija "Matematičke i informacione tehnologije" - MIT 2011, Vrnjačka Banja 27.–31. avgust; Budva 31.avgust–5.septembar 2011., Zbornik radova, pp. 325-329, ISBN 978-86-83237-90-6.

1.2.3 D. Đošić, Č. Stefanović, P. Spalević, N. Stamenković, N. Kontrec, **M. Petrović**, *Second order statistics of MRC receiver over α-μ multipath fading channels*, XLVIII

International Scientific Conference on Information Communication and Energy Systems and Technologies- ICEST 2013, 26 - 29 June 2013, Ohrid, Proceedings of papers, vol. 2, pp.83-86, ISBN: 978-9989-786-89-1.

1.2.4 M. Bandjur, S. Jovkovic, D. Djošić, **M. Petrović**, P. Spalević, S. Maričić, *Second order statistics of MRC receiver over k-μ fading channels*, International Scientific Conference Unitech 2013, 22-23 Novembar 2013, Gabrovo.

2. Spisak saopštenja na međunarodnim naučnim skupovima

2.1 (M33) Petkovic D.,**Petrović, M**. *A Truly Third Order Finite Volume Scheme On The Quadrilateral Mesh,*Međunarodna konferencija"Matematičke i informacione tehnologije" - MIT 2009, Kopaonik 27. – 31. avgust.; Budva 31. avgust – 5. septembar 2009.,Zbornik radova, pp. 293-302, ISBN 978-86-83237-90-6.

2.2 (M33) **Petrović, M.** *Local Double Logarithmic Technique,* Međunarodna konferencija "Matematičke i informacione tehnologije"-MIT 2011, Vrnjačka Banja 27.–31. avgust.;Budva 31.avgust–5.septembar 2011., Zbornik radova, pp. 325-329, ISBN 978-86-83237-90-6.

2.3 D. Đošić, Č. Stefanović, P. Spalević, N. Stamenković, N. Kontrec, **M. Petrović**, *Second order statistics of MRC receiver over α-μ multipath fading channels*, XLVIII International Scientific Conference on Information Communication and Energy Systems and Technologies- ICEST 2013, 26 - 29 June 2013, Ohrid, Proceedings of papers, vol. 2,pp.83-86, ISBN: 978-9989-786-89-1.

2.4 M. Bandjur, S. Jovkovic, D. Djošić, **M. Petrović**, P. Spalević, S. Maričić, *Second order statistics of MRC receiver over k-μ fading channels*, International Scientific Conference Unitech 2013, 22-23 Novembar 2013, Gabrovo.

Прилог 1.

ИЗЈАВА О АУТОРСТВУ

Изјављујем да је докторска дисертација, под насловом

ДВОСМЕРНИ И ДВОКОРАЧНИ УБРЗАНИ МЕТОДИ ЗА БЕЗУСЛОВНУ ОПТИМИЗАЦИЈУ

- резултат сопственог истраживачког рада,
- да предложена дисертација, ни у целини, ни у деловима, није била предложена за добијање било које дипломе, према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права, нити злоупотребио/ла интелектуалну својину других лица.

У Нишу, _

Аутор дисертације:
Милена Петровић

Потпис докторанда:

Прилог 2.

ИЗЈАВА О ИСТОВЕТНОСТИ ШТАМПАНЕ И ЕЛЕКТРОНСКЕ ВЕРЗИЈЕ ДОКТОРСКЕ ДИСЕРТАЦИЈЕ

Име и презиме аутора: Милена Петровић

Студијски програм: математика

Наслов рада: ДВОСМЕРНИ И ДВОКОРАЧНИ УБРЗАНИ МЕТОДИ ЗА БЕЗУСЛОВНУ ОПТИМИЗАЦИЈУ

Ментор: ред.проф. Предраг Станимировић

Изјављујем да је штампана верзија моје докторске дисертације истоветна електронској верзији, коју сам предао/ла за уношење у Дигитални репозиторијум Универзитета у Нишу.

Дозвољавам да се објаве моји лични подаци, који су у вези са добијањем академског звања доктора наука, као што су име и презиме, година и место рођења и датум одбране рада, и то у каталогу Библиотеке, Дигиталном репозиторијуму Универзитета у Нишу, као и у публикацијама Универзитета у Нишу.

У Нишу,

Аутор дисертације:
Милена Петровић

Потпис докторанда:

Прилог 5.

ИЗЈАВА О КОРИШЋЕЊУ

Овлашћујем Универзитетску библиотеку „Никола Тесла“ да, у Дигитални репозиторијум Универзитета у Нишу, унесе моју докторску дисертацију, под насловом:

ДВОСМЕРНИ И ДВОКОРАЧНИ УБРЗАНИ МЕТОДИ ЗА БЕЗУСЛОВНУ ОПТИМИЗАЦИЈУ

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату, погодном за трајно архивирање.

Моју докторску дисертацију, унету у Дигитални репозиторијум Универзитета у Нишу, могу користити сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons), за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство – некомерцијално
3. Ауторство – некомерцијално – без прераде
4. Ауторство – некомерцијално – делити под истим условима
5. Ауторство – без прераде
6. Ауторство – делити под истим условима

(Молимо да подвучете само једну од шест понуђених лиценци; кратак опис лиценци је у наставку текста).

У Нишу,

Аутор дисертације:
Милена Петровић

Потпис докторанда:

ТИПОВИ ЛИЦЕНЦИ

1. Ауторство. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци (CC BY 3.0).

2. Ауторство – некомерцијално. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела (CC BY- NC 3.0).

3. Ауторство – некомерцијално – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе вашег дела у делима других аутора, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела (CC BY-NC-ND 3.0).

4. Ауторство – некомерцијално – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце, и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прераде (CC BY-NC-SA 3.0).

5. Ауторство – без прераде. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе вашег дела у делима других аутора, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела (CC BY-ND 3.0).

6. Ауторство – делити под истим условима. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора, на начин одређен од аутора или даваоца лиценце, и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода (CC BY-SA 3.0).