

Optimizacija višedimenzionalnog unimodalnog kriterija Southwellovim pretraživanjem

Seminarski rad iz predmeta Optimalno upravljanje

Student: Vedad Halimić

Mentor: Red. prof. dr Samim Konjicija

Univerzitet u Sarajevu

Odsjek za automatiku i elektroniku

Elektrotehnički fakultet Sarajevo

Univerzitet u Sarajevu

vhalimic1@etf.unsa.ba

Abstract—Optimizacija višedimenzionalnog kriterija podrazumijeva jednodimenzionalno pretraživanje po varijablama funkcije kriterija. Varijabla po kojoj će se vršiti pretraživanje u jednodimenzionalnom prostoru se može birati nasumično ili po nekom definiranom kriteriju. Southwell pretraživanje podrazumijeva da se za varijablu po kojoj će se provesti jednodimenzionalno pretraživanje bira ona varijabla po kojoj je prvi parcijalni izvod funkcije kriterija najveći po apsolutnoj vrijednosti u tački trenutnog rješenja. Procedura se zaustavlja onda kada svi parcijalni izvodi budu približno jednaki nuli. U ovom radu je predstavljena optimizacija višedimenzionalnog unimodalnog kriterija, pri čemu se odabir varijable vrši Southwellovim kriterijem, dok se za jednodimenzionalno pretraživanje koristi Newton-Raphsonov metod. Programsko rješenje je implementirano u programskom jeziku Python, a u radu su prikazani primjeri i rezultati korištenja algoritma.

Abstract—An n -dimensional criteria optimisation consists of adjusting criteria variables one at a time using one-dimensional search, maintaining other variables constant. Particular variable could be picked for one-dimensional search randomly or by using a particular rule. If we pick variable which yields the maximum rate of change of our criteria, it means that we are using the Southwell search rule. Procedure is carried on until all of criteria partial derivations are approximately zero for current problem solution. An n -dimensional unimodal criteria optimisation using Southwell search is presented in this paper. For one-dimensional search, the Newton-Raphson method is used. The algorithm is implemented in Python programming language. Examples and results of using the algorithm are also presented in this paper.

Index Terms—optimisation, Southwell, Newton-Raphson

I. UVOD

Višedimenzionalni problemi su u praksi više zastupljeni od jednodimenzionalnih. Da bi se takvi problemi riješili, moguće je koristiti jednodimenzionalno pretraživanje u n -dimenzionalnom prostoru. Ideja je poboljšavanje kriterija kroz nekoliko iteracija po svakoj od varijabli problemskog prostora pojedinačno sve dok se ne postigne najbolja vrijednost, pri čemu vrijednosti ostalih varijabli ostaju konstantne. Ovaj način pretraživanja se naziva univarijatno pretraživanje [1]. Odabir varijable po kojoj će se vršiti jednodimenzionalno pretraživanje može biti nasumičan ili se može koristiti neki

od kriterija, kao što su npr. *Southwell* [2] ili *Southwell-Synge* [3] kriteriji. Newton-Raphson metoda [4], metoda zlatnog reza ili Fibonaccijeva metoda su samo neki od mogućih algoritama pogodnih za jednodimenzionalno pretraživanje. Do konačnog rješenja se jednostavnije može doći ukoliko je kriterij unimodalan [5]. Pojedini primjeri nad kojima se vršilo testiranje implementirane metode su preuzeti sa internet stranice date u [6]. Metoda Southwell pretraživanja je pogodna za minimizaciju konveksnih funkcija, odnosno rješavanje problema vezanih za mašinsko učenje, kao što su: logistička regresija, SVM (*engl. Support Vector Machines*) i problemi koji se rješavaju metodom najmanjih kvadrata. Također, pogodna je u slučajevima kada funkcija kriterija pripada porodici kvadratnih funkcija, kao i za grafovski bazirane algoritme za djelimično nadgledano mašinsko učenje [7]. U ovom radu su najprije dati matematički opisi Southwell i Newton-Raphson metoda pretraživanja. Zatim je predstavljeno programsko rješenje u programskom jeziku *Python*, zajedno sa primjerima koji realiziraju navedene algoritme.

A. Unimodalnost kriterija

Definicija Funkcija $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ je striktno unimodalna na $[a, b]$ ako postoji $x_0 \in [a, b]$ takvo da je $f(x_0) = \min\{f(x) | x_0 \in [a, b]\}$ i ako za svako $a \leq x_1 < x_2 \leq b$ vrijedi:

$$\begin{cases} x_2 \leq x_0 \Rightarrow f(x_1) > f(x_2) \\ x_0 \leq x_1 \Rightarrow f(x_2) > f(x_1) \end{cases} \quad (1)$$

Dakle, svaka striktno unimodalna funkcija na intervalu $[a, b]$ ima jedinstveni minimum na $[a, b]$ [5]. Treba napomenuti da se data definicija odnosi na funkciju jedne promjenljive.

B. Southwell pretraživanje

Southwell metoda predstavlja tip univarijantnog pretraživanja koje podrazumijeva optimiziranje samo jedne od varijabli problemskog prostora u svakoj iteraciji, dok ostale varijable zadržavaju svoje prethodne vrijednosti. Neka je potrebno odrediti minimum funkcije kao:

$$x^* = \min_{x \in \Omega} f(\mathbf{x}), \quad (2)$$

pri čemu je funkcija kriterija funkcija n varijabli:

$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_n). \quad (3)$$

Da bi bilo moguće vršiti jednodimenzionalno pretraživanje u n-dimenzionalnom prostoru, potrebno je odrediti varijablu koja će se optimizirati. Southwell kriterij podrazumijeva biranje one varijable, po kojoj je parcijalni iznos u tački trenutnog rješenja po apsolutnoj vrijednosti najveći. Odabrana varijabla x_i zadovoljava uslov:

$$\left| \frac{\partial f}{\partial x_i} \right|_{\mathbf{x}=\mathbf{x}^k} \geq \left| \frac{\partial f}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}^k} \quad (4)$$

za svako $j = 1, 2, \dots, n$ [2]. Pretraživanje po pravcu x_i se može vršiti metodama kao što su Newton-Raphson, Fibonaccijeva ili metoda zlatnog reza.

C. Newton-Raphson metoda

U ovom radu će se za jednodimenzionalnu optimizaciju koristiti Newton-Raphson metoda [4], koja računa novo potencijalno rješenje na sljedeći način:

$$x_i^{k+1} = x_i^k - \alpha \frac{\partial f / \partial x_i}{\partial^2 f / \partial x_i^2} \Big|_{\mathbf{x}=\mathbf{x}^k} \quad (5)$$

i vrijedi da je:

$$x_j^{k+1} = x_j^k, \forall j \neq i. \quad (6)$$

Ovim postupkom se ažurira vrijednost jedne od varijabli problemskog prostora. Postupak se ponavlja sve dok svi parcijalni izvodi po svim varijablama u tački trenutnog rješenja imaju vrijednost koja je približno jednaka nuli [2], odnosno manja od neke unaprijed definirane tolerancije ε .

II. KORIŠTENI ALGORITAM

U cilju implementacije optimizacije višedimenzionalnog unimodalnog kriterija Southwellovim pretraživanjem u programskom jeziku Python kreirana je biblioteka SouthwellSearchLibVH, gdje je algoritam pretraživanja tačke ekstrema implementiran unutar funkcije Southwell_search. Definicija funkcije je data u nastavku.

```
Southwell_search(funkcija, pocetna_tacka,
max_iter=1e3, preciznost=1e-6, crtanje
=0, red_varijabli=[], ispis=0)
```

Biblioteka SouthwellSearchLibVH sadrži i sljedeće pomoćne funkcije:

- parcijalni_izvod
- crtanje_funkcije
- sortiraj_varijable
- skaliranje_opsega
- ispis_kraj

Funkcija Southwell_search koristi sve navedene funkcije za rješavanje problema pretrage ekstrema, ali se neke

od navedenih funkcija mogu koristiti i samostalno, ukoliko korisnik ima potrebu za tim. Prilikom učitavanja biblioteke SouthwellSearchLibVH, ispisuje se poruka koja obavještava korisnika o učitanim metodama iz standardnih Python biblioteka, koje su neophodne za pravilno izvršavanje algoritma.

```
Biblioteka SouthwellSearchLibVH učitana.
Biblioteka >>matplotlib.pyplot<< učitana
kao >>plt<<.
Učitana je funkcija 'linspace' iz >>numpy
<< biblioteke.
Učitane su funkcije 'symbols', 'diff' i '
lambdify' iz >>sympy<< biblioteke.
```

Detaljan opis svih kreiranih funkcija iz biblioteke SouthwellSearchLibVH¹ će biti dat u narednom poglavlju. Da bi korisnik na pravilan način mogao koristiti funkciju Southwell_search, najprije je potrebno definirati simboličke varijable pomoću metode symbols i simboličku funkciju. Zatim je potrebno definirati i početnu tačku pretraživanja u obliku n-torke (engl. *tuple*) ili kao numerički cjelobrojni (*int*) ili realni (*float*) tip. Broj koordinata početne tačke i broj varijabli u simboličkoj funkciji mora biti isti. Primjer:

```
x1, x2, x3 = lib.symbols('x1_x2_x3')
funkcija = x1**2 + x2**2 + x3**2
x0 = (1, 2, 3)
```

Korisniku je ponekad potrebno da definira vlastiti redoslijed varijabli/koordinata u funkciji. Redoslijed varijabli je u programskom rješenju izuzetno bitan radi izračunavanja tačne vrijednosti izvoda funkcije u nekoj tački. Primjer: Izračunati parcijalni izvod funkcije date u (7), pomoću metode parcijalni_izvod u tački $\mathbf{t}_0 = (x_0, y_0, m_0) = (1, 2, 3)$, po varijabli m .

$$f(x, y, m) = x^2 + y^2 + m^2 \quad (7)$$

Tražena vrijednost parcijalnog izvoda u tački se može odrediti pomoću sljedećeg koda.

```
x, y, m = lib.symbols('x_y_m')
f = m**2 + y**2 + x**2
t0 = (1, 2, 3)
df, f0 = lib.parcijalni_izvod(f, m, t0)
```

Funkciji parcijalni_izvod kao argument nije proslijeđen redoslijed varijabli. Ova funkcija interno koristi metodu sortiraj_varijable koja sortira varijable po ASCII vrijednostima. Rezultat izvršavanja koda nije ispravan.

```
Izvod funkcije: 2*m
Vrijednost izvoda u tacki: 2
```

Ukoliko korisnik želi definirati vlastiti redoslijed varijabli npr. takav da je x prva, y druga a m treća koordinata, potrebno je funkciji proslijediti listu stringova sa željenim redoslijedom. Lista mora biti usklađena sa uređenom n-torkom.

¹Biblioteka će u primjerima biti referencirana sa oznakom *lib*.

```
red_varijabli = ['x', 'y', 'm']
```

Dakle, redoslijed varijabli/koordinata u simboličkom izrazu nije bitan, ali redoslijed vrijednosti u uređenoj n-torki jeste. Sada izvršavanje koda daje u potpunosti tačan rezultat.

Izvod funkcije: 2*m

Vrijednost izvoda u tacki: 6

Pseudokod za algoritam pretraživanja korištenjem Southwell metode je dat u nastavku.

Algorithm 1 Southwell_search algoritam

```
sortiranje varijabli: sortiraj_varijable
for k < max_iter do
    računanje izvoda u tački trenutnog rješenja:
    parcijalni_izvod
    if max(izvodi) < preciznost then
        if crtanje then
            nacrtati funkciju jedne ili dvije promjenljive:
            crtanje_funkcije
        end if
        return tacka_ekstrema, vrijednost_funkcije,
        tacke_pretrazivanja, br_iteracija
    end if
    i ← indeks varijable sa po apsolutnoj vrijednosti na-
    jvećom vrijednošću parcijalnog izvoda (Southwell kri-
    terij)
    g1 ← parcijalni izvod funkcije po varijabli sa indeksom
    i u tački trenutnog rješenja
    g2 ← drugi parcijalni izvod funkcije po varijabli sa
    indeksom i u tački trenutnog rješenja
     $x_{k+1} \leftarrow x_k - \alpha \cdot g1/g2$  - ažuriranje vrijednosti varijable
    odabrane za jednodimenzionalno pretraživanje (Newton-
    Raphson metod)
end for
```

III. SIMULACIJSKI REZULTATI

U ovom poglavlju će biti prikazani primjeri traženja ekstrema funkcije korištenjem metode Southwell_search iz SouthwellSearchLibVH biblioteke. Na kraju poglavlja će biti dat vodič za korištenje metoda iz spomenute biblioteke.

A. Unimodalna funkcija dvije promjenljive u obliku paraboloide

Odrediti minimum funkcije:

$$f_1(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 9)^2 \quad (8)$$

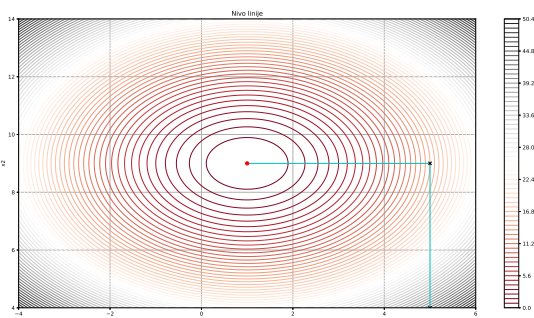
Posmatrajući datu funkciju, može se zaključiti da se minimum nalazi u tački $x_{min} = (1, 9)$ i da funkcija u toj tački ima vrijednost $f_1(1, 9) = 0$. Metoda Southwell_search nudi opciju ispisa koraka traženja ekstrema i crtanja funkcija jedne ili više promjenljivih. Te opcije su uključene slanjem odgovarajućih vrijednosti argumenata prilikom poziva metode. Opcija ispisa koraka je uključena s obzirom na to da su algoritmu bile potrebne samo dvije iteracije da odredi rješenje

problema, odnosno potvrda rješenja je došla u trećoj iteraciji. Opcija crtanja funkcije je uključena zato što je kriterij dvodimenzionalan. Za početnu tačku je odabran par $x_0 = (5, 4)$.

```
x1, x2 = lib.symbols('x1_x2')
f = (x1-1)**2 + (x2-9)**2
x0 = (5, 4)
lib.Southwell_search(f, x0, tacnost=1e-3,
    crtanje=1, ispisi=1)
```

Ispis koraka daje za svaku iteraciju opis procedure izbora varijable koja će se optimizirati jednodimenzionalnim pretraživanjem, kao i vrijednost rješenja nakon svake iteracije.

```
Southwell pretrazivanje:
Funkcija: (x1 - 1)**2 + (x2 - 9)**2
Variable [x1, x2]
*****
Iteracija br: 1
Parcijalni izvod funkcije po varijabli x1
    iznosi 2*x1 - 2
U tacki (5, 4) iznosi 8
Parcijalni izvod funkcije po varijabli x2
    iznosi 2*x2 - 18
U tacki (5, 4) iznosi -10
Vrijednosti izvoda funkcije po svim
    varijablama [x1, x2] u tacki iznosi
    [8, 10]
Za varijablu po kojoj se vrsi
    jednodimenzionalno pretrazivanje bira
    se: x2
Izvod funkcije po varijabli x2 iznosi 2*
    x2 - 18
Vrijednost izvoda iznosi: -10
Drugi izvod funkcije po varijabli x2
    iznosi 2
Vrijednost drugog izvoda iznosi: 2
Stara vrijednost rjesenja: [x1, x2] = (5,
    4)
Nova vrijednost rjesenja: [x1, x2] = (5,
    9.0)
*****
Iteracija br: 2
Parcijalni izvod funkcije po varijabli x1
    iznosi 2*x1 - 2
U tacki (5, 9.0) iznosi 8
Parcijalni izvod funkcije po varijabli x2
    iznosi 2*x2 - 18
U tacki (5, 9.0) iznosi 0.0
Vrijednosti izvoda funkcije po svim
    varijablama [x1, x2] u tacki iznosi
    [8, 0.0]
Za varijablu po kojoj se vrsi
    jednodimenzionalno pretrazivanje bira
    se: x1
Izvod funkcije po varijabli x1 iznosi 2*
    x1 - 2
```



Slika 1. Nivo linije funkcije f_1 sa označenim minimumom

```
Vrijednost izvoda iznosi: 8
Drugi izvod funkcije po varijabli x1
    iznosi 2
Vrijednost drugog izvoda iznosi: 2
Stara vrijednost rjesenja: [x1, x2] = (5,
    9.0)
Nova vrijednost rjesenja: [x1, x2] =
    (1.0, 9.0)
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
Iteracija br: 3
Parcijalni izvod funkcije po varijabli x1
    iznosi 2*x1 - 2
U tacki (1.0, 9.0) iznosi 0.0
Parcijalni izvod funkcije po varijabli x2
    iznosi 2*x2 - 18
U tacki (1.0, 9.0) iznosi 0.0
Vrijednosti izvoda funkcije po svim
    varijablama [x1, x2] u tacki iznosi
    [0.0, 0.0]
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

Slika 2. 3D prikaz funkcije f_1 sa označenim minimumom

$$f_2(x, y) = 0.26 \cdot (x^2 + y^2) - 0.48 \cdot x \cdot y; \quad (9)$$

Ideja za oblik funkcije je preuzeta sa [6]. Neka je početna tačka $(x_0, y_0) = (-11, -8)$. Za pronalazak rješenja, koje se nalazi u koordinatnom početku, algoritmu je bilo potrebno 169 iteracija. Vrijednost funkcije u tački minimuma je jednaka nuli. S obzirom na brojnost tačaka u kojima se vršilo pretraživanje, njihova lista nije data u nastavku. Međutim, na slici (3) se može vidjeti konvergencija rješenja optimizacionog problema. Na slici (4) je dat 3D prikaz funkcije kriterija, zajedno sa tačkom minimuma koja je označena plavim kružićem.

```
Rjesenje pronadjeno: DA
Broj iteracija: 169
Tacka (trenutnog) rjesenja: [-1.253e-05
-1.156e-05]
Vrijednost funkcije u tacki: 0.0
```

C. Unimodalna funkcija četiri promjenljive

Odrediti minimum funkcije:

$$f_3(\mathbf{x}) = (x_1 - 1)^2 + (x_2 + 5)^2 + x_3^2 + (x_4 + 4)^2 + 2 \quad (10)$$

Neka je početna tačka $\mathbf{x}_0 = (-1, 9, 9, 7)$. Algoritam je potvrdio rješenje problema u petoj iteraciji kao $\mathbf{x}_{min} = (x_1, x_2, x_3, x_4) = (1.0, -5.0, 0.0, -4.0)$ i vrijednost funkcije u toj tački iznosi $f_3(\mathbf{x}_{min}) = 2$.

Rjesenje pronadjeno: DA
Broj iteracija: 5

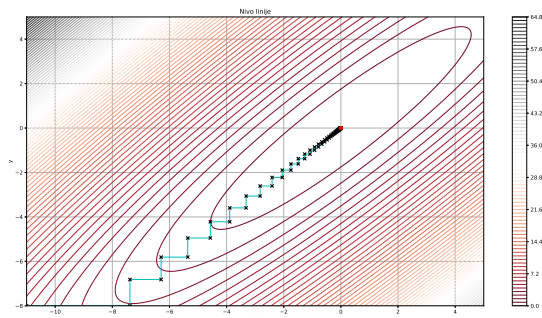
Kada algoritam završi svoj rad, bilo uspješno ili neuspješno, ispisuje se poruka o rezultatima simulacije koja sadrži informacije o tački ekstrema, vrijednosti funkcije u toj tački, broj iteracija potrebnih za pronalazak rješenja i niz tačaka u kojima se vršilo pretraživanje. Treba napomenuti da se ove informacije ispisuju i onda kada je opcija za ispis koraka isključena.

```
Rjesenje pronadjeno: DA
Broj iteracija: 3
Tacke u kojima se vrsilo pretrazivanje:
    [(5, 4), (5, 9.0), (1.0, 9.0)]
Tacka (trenutnog) rjesenja: (1.0, 9.0)
Vrijednost funkcije u tacki: 0.0
```

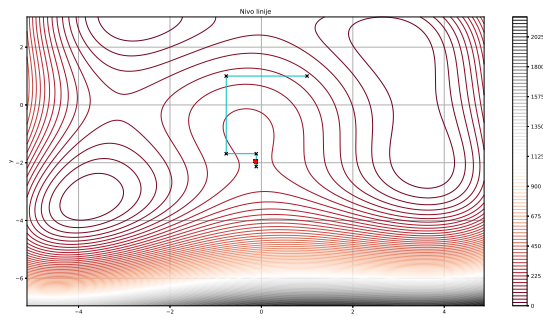
Na slici (1) su prikazane nivo linije funkcije kriterija. Tačke pretraživanja su označene crnom bojom i znakom \times . Tačka minimuma je označena crvenim kružićem. Na slici (2) je dat trodimenzionalni prikaz funkcije kriterija. Funkcija ima oblik paraboloida. Minimum je označen plavim kružićem.

B. Unimodalna funkcija dvije promjenljive

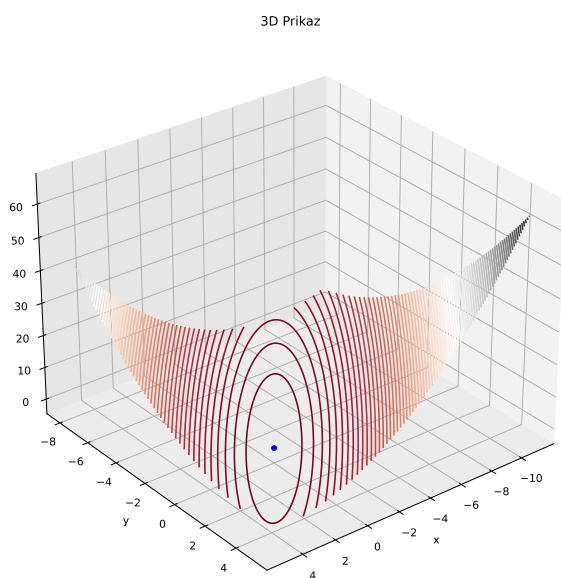
Odrediti minimum funkcije:



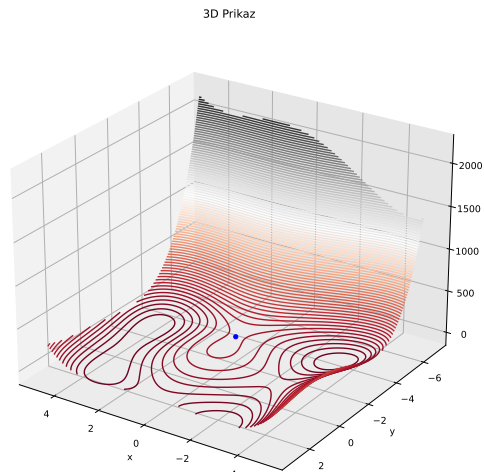
Slika 3. Nivo linije funkcije f_2 sa označenim minimumom



Slika 5. Nivo linije funkcije f_4 sa označenim minimumom



Slika 4. 3D prikaz funkcije f_2 sa označenim minimumom



Slika 6. 3D prikaz funkcije f_4 sa označenim minimumom

Tacke u kojima se vrsilo pretrazivanje:

$[(-1, 9, 9, 7),$
 $(-1, -5.0, 9, 7),$
 $(-1, -5.0, 9, -4.0),$
 $(-1, -5.0, 0.0, -4.0),$
 $(1.0, -5.0, 0.0, -4.0)]$

Tacka (trenutnog) rjesenja: $(1.0, -5.0,$
 $0.0, -4.0)$

Vrijednost funkcije u tacki: 2.0

D. Multimodalna funkcija dvije promjenljive

Odrediti ekstrem funkcije:

$$f_4(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2 \quad (11)$$

Ideja za oblik funkcije je preuzeta iz [6]. Ako se nad ovom funkcijom kriterija primjeni algoritam Southwell_search metode, može se primijetiti da funkcija ima nekoliko lokalnih ekstrema. Nivo linije kriterija su prikazani na slici (5), dok je na slici (6) dat 3D prikaz zadane funkcije. Iako je tema rada optimizacija unimodalnih problema, rješenje je testirano na multimodalnom problemu da bi se utvrdilo da li je algoritam osjetljiv na upadanje u lokalne optimume.

Za pronalazak rješenja ovog problema je bilo potrebno 18 iteracija. Tačka minimuma je $(x_{m1}, y_{m1}) \approx (-0.128, -1.954)$ a vrijednost funkcije u istoj iznosi $f_4(x_{m1}, y_{m1}) = 178.34$. Neka je početna tačka $(x_{01}, y_{01}) = (1, 1)$.

Rjesenje pronadjeno: DA

Broj iteracija: 18

Tacke u kojima se vrsilo pretrazivanje:

$\begin{bmatrix} 1. & 1. & \\ -0.76923077 & 1. & \end{bmatrix}$

```

[-0.76923077 -1.68745669]
[-0.11197446 -1.68745669]
[-0.11197446 -2.13221968]
[-0.11197446 -1.97719307]
[-0.11197446 -1.94787143]
[-0.12893291 -1.94787143]
[-0.12893291 -1.95418255]
[-0.12788284 -1.95418255]
[-0.12788284 -1.9536814 ]
[-0.12796698 -1.9536814 ]
[-0.12796698 -1.95371741]
[-0.12796094 -1.95371741]
[-0.12796094 -1.9537148 ]
[-0.12796138 -1.9537148 ]
[-0.12796138 -1.95371499]
[-0.12796134 -1.95371499]]

```

Tacka (trenutnog) rjesenja: [-0.12796134
-1.95371499]

Vrijednost funkcije u tacki: 178.3372392

Može se zaključiti da je algoritam osjetljiv na upadanje u lokalne optimume, te da u zavisnosti od odabrane početne tačke algoritam može dati različita rješenja kada su u pitanju multimodalni problemi. Primjerice, za početnu tačku $(x_{02}, y_{02}) = (-5, -2)$ se dobije rješenje $(x_{m1}, y_{m1}) \approx (-3.779, -3.283)$. Moguće rješenje za poboljšanje rada algoritma je upravo pretraživanje ekstrema za različite vrijednosti početnih tačaka, pretraga unutar definiranih granica za sve varijable problemskog prostora (Fibonaccijev ili metod zlatnog reza) ili korištenjem Southwell-Synge kriterija u kojem se na osnovu drugog izvoda funkcije cilja odlučuje koja će se varijabla koristiti za jednodimenzionalno pretraživanje. Kako se u algoritmu koristi Newton-Raphson metoda za pretraživanje u jednodimenzionalnom prostoru, poboljšanje konvergencije ka rješenju bi se moglo ostvariti povećanjem parametra α iz relacije (5), koji u prezentiranom rješenju ima vrijednost 1.

E. Vodiči za implementirane metode

1) *Southwell_search*: je najznačajnija metoda kreirana za potrebe izrade ovog rada. Ona implementira Southwell pretraživanje u n-dimenzionalnom prostoru.

```

Southwell_search(funkcija , pocetna_tacka ,
    max_iter=1e3 , preciznost=1e-6 , crtanje
    =0 , red_varijabli=[], ispis=0):
    , , ,

```

Southwell_search trazi optimum
prosljednene funkcije.

Argumenti

```

funkcija : simbolicka funkcija ciji
    ekstrem trazimo
pocetna_tacka : pocetna tacka za
    pretrazivanje ekstrema (float ili
    tuple)

```

```

max_iter : maksimalni broj iteracija (int
    ), ako nije specificirano , maksimalni
    broj je 1000
preciznost : algoritam se zaustavlja kada
    je maksimalna vrijednost izvoda po
    svim varijablama manja od ove
    vrijednosti , ako nije specificirano ,
    preciznost iznosi 1 mikron (float)
red_varijabli : lista stringova naziva
    varijabli , redosljed bitan zbog
    izracunavanja vrijednosti u tacki
ispis : ako je ovaj parametar razlicit od
    nule , ispisuju se koraci algoritma ,
    ako je ovaj parametar jednak nuli ili
    nije specificiran , koraci se ne
    ispisuju (bool)

```

Povratne vrijednosti

```

x_ex : tacka ekstremuma (float)
fx_ex : vrijednost funkcije u tacki
    ekstremuma (float)
k : broj iteracija koji je bio potreban
    za dostizanje rjesenja (int)
, , ,

```

2) *parcijalni_izvod*: je metoda koja simbolički i numerički (u tački) računa parcijalni izvod funkcije date kao argument. Značajna je za metodu *Southwell_search*, ali ju je moguće pozivati i samostalno.

```

parcijalni_izvod(funkcija , varijabla , tacka
    , n=1 , red_varijabli=[]):
    , , ,

```

Argumenti

```

funkcija : funkcija/izraz ciji izvod
    racunamo simbolicki i/ili numericki
varijabla : varijabla po kojoj simbolicki
    racunamo izvod
tacka : koordinate tacke u kojoj
    numericki racunamo izvod (float ili
    tuple)
n : redni broj izvoda , ako ovaj parametar
    nije specificiran onda je n=1 (int)
red_varijabli : redosljed varijabli ,
    vazno za racunanje izvoda u tacki , ako
    nije specificirano , varijable se
    sortiraju abecedno (lista stringova)

```

Povratne vrijednosti

```

izvod : simbolicki oblik n-tog izvoda
    funkcije
vrijednost : vrijednost izvoda funkcije u
    tacki (float) , ako broj nezavisnih
    varijabli i koordinata tacke nije isti

```

```
, funkcija umjesto vrijednosti vraca
None
,,,
```

3) *crtanje_funkcije*: je metoda koja crta simboličku funkciju pomoću nivo linija i u 3D prostoru, tačke potencijalnih rješenja i tačku optimuma. Koristi se unutar metode Southwell_search, ali se može pozivati i samostalno.

```
crtanje_funkcije(funkcija, varijable, tacka
, ekstrem, potencijalnaRjesenja=[],
, granica=5, brojTacaka=500):
,,,
```

Argumenti

funkcija : simbolicka funkcija jedne ili dvije promjenljive koju zelimo iscrtati

varijable : sortirana lista varijabli koje se nalaze u prosljedjenoj funkciji

tacka : tacka(float) ili uredjeni par u kojoj se nalazi ekstrem

ekstrem : vrijednost funkcije u tacki ekstrema (float)

potencijalnaRjesenja : lista tacaka/ uredjenih parova potencijalnih rjesenja

granica : odredjuje opseg za koji se crtaju funkcije (float), ako nije specificirano, vrijednost ovog argumenta iznosi 5

brojTacaka : broj tacaka u opsegu u kojem se crta funkcija, ako nije specificirano, vrijednost ovog argumenta iznosi 500

Povratne vrijednosti

0 – crtanje neuspjesno

1 – crtanje uspjesno

```
,,,
```

4) *skaliranje_opsega*: je metoda implementirana radi rješavanja problema prikaza svih potencijalnih rješenja na grafiku funkcije koja se crta pomoću metode *crtanje_funkcije*. Samostalno pozivanje ove funkcije nema neki veliki značaj za korisnika.

```
skaliranje_opsega(potencijalnaRjesenja,
, lijeva, desna):
,,,
```

Argumenti

potencijalnaRjesenja : float lista – potencijalnih rjesenja po odgovarajucoj varijabli

lijeva : lijeva granica opsega u kojem se crta funkcija (float)

desna : desna granica opsega u kojem se crta funkcija (float)

Ukoliko se neko od potencijalnih rjesenja nalazi izvan opsega u kojem se crta funkcija, opseg se prosiruje tako da se prikazu sve tacke pretrazivanja

Povratne vrijednosti

lijeva : (modifikovana) lijeva granica za crtanje funkcije

desna : (modifikovana) desna granica za crtanje funkcije

```
,,,
```

5) *ispis_kraj*: je metoda koja se poziva iz funkcije Southwell_search na kraju uspješnog ili neuspješnog pretraživanja ekstrema funkcije. Samostalno pozivanje ove metode od strane korisnika nema neki veliki značaj.

```
ispis_kraj(tacka, vrijednost, br_iter,
, tacke_pretrazivanja, nadjeno_rjesenje,
, preciznost):
,,,
```

Argumenti

tacka : float ili tuple – tacka trenutnog /krajnjeg rjesenja

vrijednost : float – vrijednost kriterija u trenutnom rjesenju

br_iter : int – trenutno dostignuti broj iteracija

tacke_pretrazivanja : float ili tuple lista tacaka u kojima se vrsilo pretrazivanje

nadjeno_rjesenje : string koji moze imati vrijednosti DA ili NE, u zavisnosti od toga da li je algoritam dostigao rjesenje

preciznost : prosljedjuje se kao argument radi prilagodjenog ispisa rezultata

Funkcija ne vraca nista

```
,,,
```

IV. ZAKLJUČAK

Da bi se efikasnije implementiralo jednodimenzionalno pretraživanje u n-dimenzionalnom prostoru, potrebno je koristiti unaprijed određeni kriterij kojim će se ubrzati konvergencija ka konačnom rješenju. U ovom radu je predstavljen Southwellov kriterij za odabir varijable po kojoj će se vršiti jednodimenzionalno pretraživanje. Cilj je bio implementirati spomenuti

kriterij u programskom jeziku *Python* i testirati njegov rad nad unimodalnim kriterijima. Može se reći da je u tom pogledu implementacija uspješna i da su rezultati zadovoljavajući. Testiranjem algoritma nad multimodalnim kriterijem, dolazi se do zaključka da postoji velika osjetljivost na upadanje u lokalne minimume. Za različite početne tačke, algoritam može dati različita rješenja, kada su u pitanju multimodalne funkcije kriterija. Kako bi se poboljšala konvergencija algoritma ka konačnom rješenju, moguće je nametnuti ograničenja na varijable problemskog prostora i umjesto Newton-Rapshon metode koristiti Fibonaccijevu ili metodu zlatnog reza. Ukoliko se ipak želi koristiti Newton-Rapshon metod, za bržu konvergenciju je moguće povećati parametar α .

LITERATURA

REFERENCES

- [1] Samim Konjicija, Predavanja na predmetu Optimalno upravljanje, Elektrotehnički fakultet, Univerzitet u Sarajevu, ak. 2020/2021. godina.
- [2] Pierre, Donald A. Optimization theory with applications. Courier Corporation, 1986.
- [3] Synge, J. L. "A geometrical interpretation of the relaxation method." Quarterly of Applied Mathematics 2.1 (1944): 87-89.
- [4] Ypma, Tjalling J. "Historical development of the Newton-Raphson method." SIAM review 37.4 (1995): 531-551.
- [5] Ortega, James M., and Werner C. Rheinboldt. Iterative solution of nonlinear equations in several variables. Society for Industrial and Applied Mathematics, 2000.
- [6] J. Brownlee. (2021.) "Two-Dimensional (2D) Test Functions for Function Optimization". [Online] Dostupno na: <https://machinelearningmastery.com/2d-test-functions-for-function-optimization/>.
- [7] Nutini, Julie, et al. "Coordinate descent converges faster with the gauss-southwell rule than random selection." International Conference on Machine Learning. PMLR, 2015.