

### 3. Laboratorijska Vježba

Ivan Škrabo

#### Opis implementirane funkcionalnosti

Funkcionalnost je ostvarena kroz tri python datoteke:

main.py – glavni pokretač vježbe i pozivatelj funkcija za svaki zadatak. Deklarira varijable, funkcije i poziv traženih algoritama računanja minimuma. Također ispisuje konačna rješenja.

functions.py – sadrži deklaracije traženih funkcionalnosti. Svaka funkcionalnost prima početnu točku, funkciju nad kojom tražimo minimum, traženu preciznost i po potrebi dodatne varijable poput gradijenta, hessian-a, da li koristimo Golden Ratio, implicitne i eksplicitne granice i drugo.

declarations.py – sadrži deklaracije za stvaranje objekata funkcije, gradijenta, hessian-a, funkcije zamjene za računanje Golden Ratio-a, deklaracije implicitnih i eksplicitnih ograničenja te stvaranje transformirane funkcije.

#### Upute za prevođenje i pokretanje, s parametrima

Funkcionalnosti su ostvarene u programskom jeziku Python, koristeći Visual Studio Code. Pokreće se main.py (gdje se po potrebi trebaju odkomentirati dijelovi za pojedini laboratorijski zadatak), koji ne prima argumente. Za promenu argumenata se trebaju mijenjati varijable unutar main.py. Argumenti su: x - deklarira početnu točku, fja – deklarira nad kojom funkcijom se računa minimum, grad – gradijent od funkcije, hessian – hessian matrica od funkcije, conditions – stvaranje objekata koji reprezentira implicitna i eksplicitna ograničenja te poziv njegove unutarnje funkcije za postavljanje tih ograničenja. Povratne vrijednosti od algoritama je xnew (i ponekad iters), koji predstavlja pronađeni minimum od funkcije. Također fja, grad i hess sadrže unutarnju varijablu *step* koja predstavlja broj poziva te funkcije.

#### Ispis rezultata programa i odgovori na pitanja iz lab. Vježbe

-----Zadatak 1-----

Rezultati:

Funkcija divergira. Zaustavljanje algoritma.

Rezultati bez korištenja golden ratio:

Algoritam pronalazi minimum u točki:

0.000 0.000

Broj iteracija je 101

Broj iteracija gradijent je 101

Rezultati uz pomoć algoritma golden ratio:

Algoritam pronalazi minimum u točki:

2.000 -3.000

Broj iteracija je 151

Broj iteracija gradijent je 3

Zaključak – Postupak gradijentnog spusta ne pronalazi konačni minimum u slučajevima kada funkcija divergira. To se može popraviti korištenjem algoritma zlatnog reza nad funkcijom u kojoj tražimo iznos optimalnog pomaka u smjeru negativnog gradijenta, dok se ne korištenjem algoritma zlatnog reza uvijek pomičemo za isti iznos u smjeru negativnog gradijenta.

-----Zadatak 2-----

Rezultati:

Rezultati uz pomoć algoritma golden ratio:

Algoritam pronalazi minimum u točki:

1.000 1.000

Broj iteracija je 288101

Broj iteracija gradijent je 3947

Algoritam pronalazi minimum u točki:

1.000 1.000

Broj iteracija je 1142

Broj iteracija gradijent je 16

Broj iteracija hessian je 16

Rezultati uz pomoć algoritma golden ratio:

Algoritam pronalazi minimum u točki:

4.000 2.000

Broj iteracija je 1972

Broj iteracija gradijent je 28

Algoritam pronalazi minimum u točki:

4.000 2.000

Broj iteracija je 151

Broj iteracija gradijent je 3

Broj iteracija hessian je 3

Zaključak – Newton-Raphson računa minimum puno brže naspram postupka gradijentnog spusta. Minimum druge funkcije pronalazi nakon samo jedne iteracije zato što je funkcija kvadratna, dok je prva funkcija nekvadratna i potrebno je više iteracija za pronalazak

minimumu. Funkcija završava kada je euklidska norma gradijenta manja od zadane preciznosti.

-----Zadatak 3-----

Rezultati:

Algoritam pronalazi minimum u točki:

0.010 0.010

Broj iteracija je 8196

Algoritam pronalazi minimum u točki:

2.000 2.000

Broj iteracija je 1316

Zaključak – Uz nametnuta ograničenja, optimum se mijenja s [1, 1] u [0.01, 0.01] za prvu funkciju i s [4, 2] u [2, 2] za drugu funkciju.

-----Zadatak 4-----

Rezultati:

Algoritam pronalazi minimum u točki:

0.010 0.010

Broj iteracija je 8

Algoritam pronalazi minimum u točki:

2.000 2.000

Broj iteracija je 12

Zaključak – Korišten je postupak Hooke-Jeeves. Uz zadanu točku nije moguće naći optimalno rješenje. Algoritam pronalazi optimum za početnu točku [1, 4] za prvu funkciju, dok za drugu funkciju ne uspijeva pronaći minimum [4, 2] jer ne ispunjava implicitno ograničenje ( $x_2 - x_1 \geq 0$ ), tj.  $2 - 4$  je manje od 0 što ne ispunjava ograničenje.

-----Zadatak 5-----

Rezultati:

Algoritam pronalazi minimum u točki:

2.000 1.000

Broj iteracija je 8

Zaključak – Također je korišten postupak Hooke-Jeeves. Funkcija nije pronašla minimum [3, 0] s početnom točkom [0, 0], te ne pronalazi minimum s početnom točkom [5, 5] jer ne zadovoljava implicitna ograničenja. Minimum [3, 0] ne može pronaći zbog ograničenja  $x_2 - 1 = 0$ , čime se postavlja  $x_2$  na vrijednost 1.