Univerzitet u Sarajevu
Elektrotehnički fakultet
Predmet: Optimizacija resursa
Akademska godina: 2020/2021
Student: Samra Mujčinović
Datum: 10.11.2020.

Izvještaj sa laboratorijske vježbe 3

Zadatak 1

Ideja za rješenje zadatka

Za realizaciju funkcije pronalaska lokalnog minimuma zadane funkcije, potrebno je prije svega pronaći sve okolne tačke za zadanu početnu tačku. Takvih tačaka ima osam. Zatim za svaku tačku iz skupa pronađenih susjednih tačaka provjeravamo da li pripada zadanom intervalu, tj da li je u zadanom opsegu, te se traži lokalni minimum u svakoj susjednoj tački. Ukoliko je novonađeni minimum manji od prethodno pronađenog, usvaja se da je novonađeni minimum zapravo trenutni lokalni minimum. Ovaj postupak se izvršava N puta ili dok razlika između prethodno nađenog minimuma i novonađenog minimuma ne postane dovoljno mala. Kao rezultat funkcije vraćamo lokalni minimum zadane funkcije.

Vodič kroz kôd/simulaciju

Jedina funkcija koja je definisana u ovom zadatku je LS funkcija čiji je prototip zadan u postavci. U njoj je sadržana cjelokupna logika pronalaženja lokalnog minimuma zadane funkcije.

Pored ove funkcije, definisane su četiri funkcije date u trećem zadatku radi provjere rezultata koje vraća funkcija LS.

Rezultati kôda/simulacije

Prilikom testiranja rada implamentirane funkcije LS, bile su korištene testne funkcije iz trećeg zadatka, te sljedeće vrijednosti nekih od argumenata funkcije LS:

N = 300

delta_x = 0.01

eps = 0.0001

Početna tačka, kao i opseg su različiti za svaku funkciju, s tim da su za početnu tačku uzete vrijednosti koje su približne globalnom minimumu, kako bismo dobili što približnije rezultate. Promatrani opseg za svaku funkciju je isti kao u trećem zadatku.

Za prvu, parabolid, funkciju početna tačka koju smo uzeli je (1, 1), te u kombinaciji sa gore navedenim vrijednostima ostalih argumenata dobijamo lokalni minimum (3, -1), te uviđamo da je ova funkcija pronašla lokalni minimum jednak globalnom.

Sljedeća funkcija, rastrigin, za početnu tačku (1, -1), vraća lokalni minimum u (0.99, -0.99). Ako bismo kao početnu tačku uzeli tačku (0, 0) dobili bismo (0, 0) što i jeste tačka globalnog minimuma.

Za funkciju Drop-Waves kao početnu tačku uzeli smo (-1, 1) te kao lokalni minimum dobili tačku (-1.2, 1.09). Da smo krenuli od tačke (0, 0) lokalni minimum bi bio (0, 0) što je ujedno i tačka globalnog minimuma.

I za posljednju, Holder Table, funkciju uzeli smo početnu tačku približnu globalnom minimumu, te dobili rezultat (8.01, 8.99) što je približno globalnom minimumu.

Zaključak

Na osnovu provedenih testiranja, uzimajući različite vrijednosti za argumente funkcije LS, dolazimo do zaključka da sa povećanjem broja iteracija N, možemo doći do tačnijih rezultata. Također, uzmemo li za početnu tačku neku koja je približna globalnom minimumu, veća je mogućnost da ćemo kao rezultat dobiti lokalni minimum koji je dosta blizak globalnom. Još jedan način kako bismo mogli povećati vjerovatnoću poklapanja globalnog i lokalnog minimuma jeste da povećamo delta_x, međutim to ne bi bilo u skladu sa idejom samog traženja minimuma na ovaj način.

Zadatak 2

Ideja za rješenje zadatka

Na početku kreiramo listu u kojoj će se nalaziti sve tačke koje pronađemo pretragom lokalnog minimuma koristeći funkciju definisanu u prethodnom zadatku(LS). Nakon toga u petlji koja se izvršava M puta pozivamo funkciju Perturbacija koja generiše sljedeću početnu tačku pretraživanja lokalnog minimuma. Definisana je i funkcija AzuriranjeMemorije koja dodaje rezultat funkcije LS (gdje je početna tačka ona koju je generisala funkcija Perturbacija) u prethodno definisanu listu. U istoj petlji se porede vrijednosti funkcije za prethodno pronađeni i novonađeni minimum. I konačno kao rezultat ova funkcija vraća lokalni minimum.

Vodič kroz kôd/simulaciju

U ovom zadatku definisane su tri funkcije:

- ILS – sadrži cjelokupnu logiku traženja lokalnog minimuma i unutar sebe poziva preostale dvije funkcije, kao i funkciju LS iz prethodnog zadatka
- AzuriranjeMemorije – traženu u postavci zadatka, koja pohranjuje nove tačke u listu
- Perturbacija – generise novu početnu tačku lokalnog pretraživanja

Rezultati kôda/simulacije

Prilikom testiranja rada implamentirane funkcije ILS, bile su korištene testne funkcije iz trećeg zadatka, te sljedeće vrijednosti nekih od argumenata funkcije ILS:

N = 300

delta_x = 0.01

eps = 0.001

M = 200

Početna tačka, kao i opseg su različiti za svaku funkciju. Promatrani opseg za svaku funkciju je isti kao u trećem zadatku.

Za prvu, parabolid, funkciju početna tačka koju smo uzeli je (1, 1), te u kombinaciji sa gore navedenim vrijednostima ostalih argumenata dobijamo lokalni minimum (2.99, -0.99), te uviđamo da je ova funkcija pronašla lokalni minimum skoro jednak globalnom.

Sljedeća funkcija, rastrigin, za početnu tačku (-1, 1), vraća lokalni minimum u (0.00047517, 0.00301883), što je vrlo blisko globalnom minimumu. Ako bismo kao početnu tačku uzeli tačku (0, 0) dobili bismo (0, 0) što i jeste tačka globalnog minimuma.

Za funkciju Drop-Waves kao početnu tačku uzeli smo (1, -1) te kao lokalni minimum dobili tačku (0.25064685, -0.45425858), što je vrlo blisko globalnom minimumu. Da smo krenuli od tačke (0, 0) lokalni minimum bi bio (0, 0) što je ujedno i tačka globalnog minimuma.

I za posljednju, Holder Table, funkciju uzeli smo početnu tačku (0,0), te dobili rezultat (-8.05425732, -9.66486863) što je jednako globalnom minimumu.

Zaključak

Na osnovu provedenih testiranja, možemo zaključiti da iterativno traženje lokalnog minimuma daje dosta bolje rezultate. Ovaj metod daje bolji konačni minimum funkcije nezavisno od početne tačke. Razlog zbog kojeg je ILS pretraga vjerodostojnija, je što se unutar nje nastavlja pretraga i proračun tražeći bolje opcije, čak i nakon što se nađe minimum. Približnost lokalnog minimuma nađenog ovom iterativnom metodom i globalnog minimuma, u ovoj metodi ponajviše zavisi od random odabira tačke u funkciji Perturbacija. No, bez obzira na random odabir tačke, rezultat će uvijek biti približniji globalnom minimumu nego što je to kod funkcije LS.

Zadatak 3

Ideja za rješenje zadatka

U ovom zadatku je potrebno samo grafički prikazati rezultate funkcija definisanih u prethodnim zadacima. To je urađeno po uzoru na primjer koji je dat u postavci vježbe.

Vodič kroz kôd/simulaciju

U kodu su za svaku datu funkciju iz postavke zadatka definisane funkcije koje crtaju grafike za svaku od njih, te obilježavaju nađene minimume zadanih funkcija. Na kraju su samo pozvane ove funkcije.

Rezultati kôda/simulacije

Za potrebe ovog zadatka iskorištene su sljedeće vrijednosti za argumente funkcija LS i ILS:

N = 300

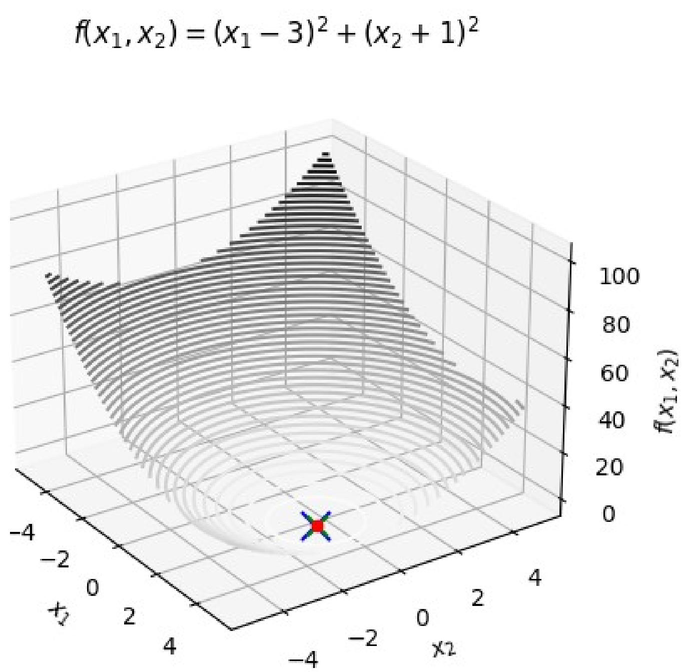
delta_x = 0.01

eps = 0.001

M = 200

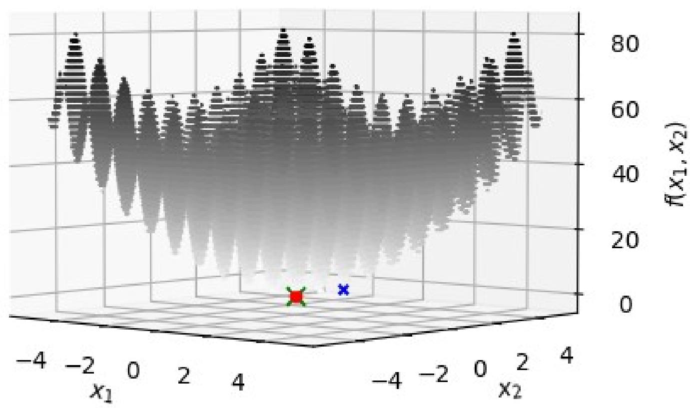
Početna tačka, kao i opseg su različiti za svaku funkciju. Promatrani opseg za svaku funkciju je isti kao u trećem zadatku.

Za prvu, parabolid, funkciju početna tačka koju smo uzeli je (0, 0):



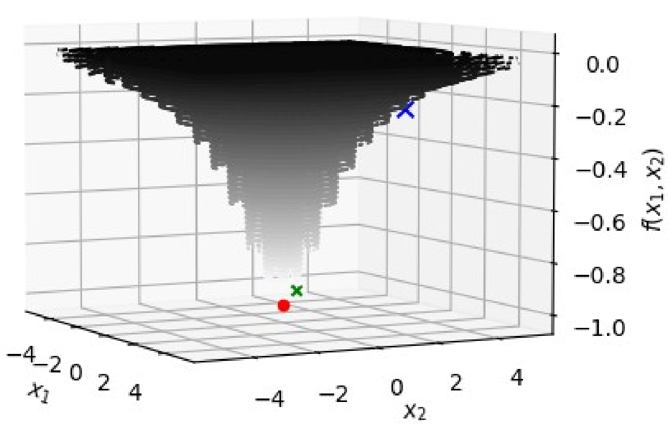
Sljedeća funkcija, rastrigin, za početnu tačku (1, 1):

$$f(x_1, x_2) = 20 + (x1^2 - 10cos(2\pi * x1) + x2^2 - 10cos(2\pi * x2))$$

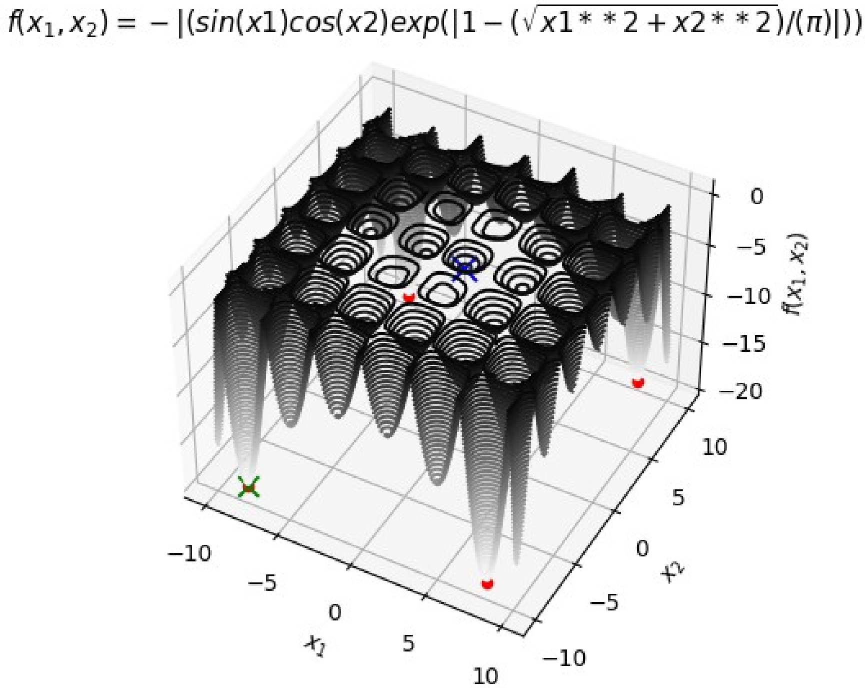


Za funkciju Drop-Waves kao početnu tačku uzeli smo (2, 3):

$$f(x_1, x_2) = -((1 + cos(12\sqrt{x1^2 + x2^2}))/(0.5(x1^2 + x2^2) + 2))$$



I za posljednju, Holder Table, funkciju uzeli smo početnu tačku (0,0):

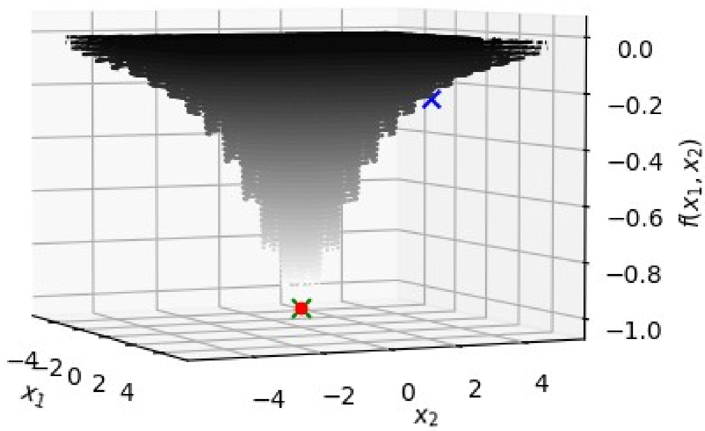


Poređenje nađenih minimuma pomoću LS i ILS funkcija:

```
LS za paraboloid funkciju:  [ 2.96 -1.  ]
ILS za paraboloid funkciju:  [ 2.98131549 -0.98939193]
LS za rastrigin funkciju:  [0.99 0.99]
ILS za rastrigin funkciju:  [-0.00310068 -0.00134626]
LS za Drop-Wave funkciju:  [2.05 3.03]
ILS za Drop-Wave funkciju:  [0.48729588 0.18402004]
LS za Holder Table funkciju:  [1.22 0.  ]
ILS za Holder Table funkciju:  [-8.05427643 -9.66554571]
```

Pri ponovnom pokretanju programa, za iste vrijednosti svih parametara kao u prethodnom pokretanju, za treću funkciju dobili smo približniji lokalni minimum koristeći ILS što je prikazano na sljedećoj slici:

$$f(x_1, x_2) = -((1 + cos(12\sqrt{x1^2 + x2^2}))/(0.5(x1^2 + x2^2) + 2))$$



Za ostale funkcije su ostali približno isti rezultati što možemo vidjeti na sljedećem ispisu:

```
LS za paraboloid funkciju:  [ 2.96 -1.  ]
ILS za paraboloid funkciju:  [ 3.02358031 -0.99546325]
LS za rastrigin funkciju:  [0.99 0.99]
ILS za rastrigin funkciju:  [-0.00277365 -0.00184424]
LS za Drop-Wave funkciju:  [2.05 3.03]
ILS za Drop-Wave funkciju:  [-0.00395725 -0.00225344]
LS za Holder Table funkciju:  [1.22 0.  ]
ILS za Holder Table funkciju:  [ 8.05572048 -9.66617755]
```

Zaključak

Na prethodnim slikama moguće je bolje uočiti razlike između rezultata LS i ILS funkcija nego što je to bilo moguće u prethodnim zadacima. Zaključujemo da je u svakom slučaju ILS funkcija dala bolje rezultate u odnosu na LS funkciju, pri istim vrijednostima parametara.