SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU ELEKTROTEHNIČKI FAKULTET

Diplomski studij računarstva

Laboratorijska vježba 3

Rješavanje problema pronalaska minimuma N-dimenzionalne funkcije koristeći PSO i GA algoritme

Ivan Budoš, DRB

Osijek, 2022.

SADRŽAJ

UVOD

U trećoj laboratorijskoj vježbi potrebno je riješiti problem pronalaska minimuma n-dimenzionalne funkcije korištenjem genetskog algoritma i algoritma roja čestica. Potrebno je usporediti performanse GA i PSO algoritma prilikom pronalaska globalnog minimuma. Genetski algoritam za parametre ima faktor mutacije, broj elitnih jedinki i maksimalnu vrijednost mutacije, a PSO algoritam za parametre ima veličinu populacije, mjeru inercije, mjeru individualnog faktora i mjeru socijalnog faktora.

PROBLEM N-DIMENZIONALNE RASTRIGINOVE FUNKCIJE

Rastriginova funkcija je ne konveksna funkcija korištena za ispitivanje problema performansi algoritama optimizacije. Tipičan je primjer nelinearne multimodalne funkcije. Prvi put je predložena 1974. godine od strane Rastrigina kao dvodimenzionalna funkcija i generalizirana je od strane Rudolpha. Pronalaženje minimuma ove funkcije je prilično težak problem zbog velikog prostora za pretraživanje i velikog broja lokalnih minimuma.

Opis rješenja klasičnim algoritmom

Ovaj problem ima veliko područje pretrage te jako veliki broj lokalnih minimuma. Zbog toga traženje globalnog minimuma ovog problema korištenjem klasičnog algoritma zahtjeva puno vremena i resursa.

Genetski algoritam

Genetski algoritam je meta heuristička metoda optimiranja koja imitira prirodni evolucijski proces. Evolucija je robustan proces pretraživanja prostora rješenja. Po načinu djelovanja ubrajaju se u metode usmjerenog slučajnog pretraživanja prostora rješenja (guided random search techniques) u potrazi za globalnim optimumom.

Populacija je skup jedinki odnosno rješenja u i-tom koraku rada algoritma. Kromosom je jedna jedinka rješenja odnosno jedno moguće rješenje zadanog problema. Dok gen predstavlja jediničnu informaciju, odnosno nositelj je jedne informacije iz rješenja. Geni se mogu kodirati na razne načine koje odgovaraju pojedinim tipovima problema. Najčešći tipovi kodiranja su: binarni, vrijednosni, permutacijski i stablasti.

Binarno kodiranje – gen može poprimiti samo dvije vrijednosti: 0 ili 1.
Vrijednosno kodiranje – gen može poprimiti cjelobrojne/realne vrijednosti iz zadanog intervala.
Permutacijsko kodiranje – gen može poprimiti cjelobrojne vrijednosti tako da kromosom uvijek sadrži sve brojeve ali različiti raspored.
Stablasto kodiranje – gen je čvor stabla.

Tijekom rada genetski algoritam koristi genetske operatore za stvaranje novih populacija odnosno novih rješenja. Koriste se sljedeći genetski operatori:

REKOMBINACIJA (križanje)- Kombiniranje gena dva roditelja u svrhu stvaranja novih i boljih potomaka. Najčešća rekombinacija može biti:

Rekombinacija u jednoj točki- Slučajnim odabirom točke rekombinacije kod roditelja vrši se zamjena gena od te točke na dalje. 3

Rekombinacija u dvije (više) točaka- Točke rekombinacije se biraju slučajno i nakon njih se naizmjence vrši zamjena gena.

Uniformna rekombinacija- Slučajno se odabire svaki gen da li će biti odabran iz jednog roditelja ili drugog. Odnosno svaki gen ima vjerojatnost odabira od jednog roditelja ili drugog.

MUTACIJA mijenja vrijednost nasumično odabranog gena ili više gena i na taj način unosi nove informacije u populaciju i omogućuje izlazak iz lokalnog minimuma. Najčešće se baziraju na vjerojatnosti mutacije jednog gena. Postoji više tipova:

Jednostavna mutacija- slučajno se odabire gen unutar kromosona i mijenja ga se

Potpuna mutacija- slučajno se bira kromosom i zatim se ispremješta gene u njemu.

Ovisno o kojem kodiranju gena se radi, mutacija drugačije radi. Kod binarno kodiranih invertira bitove, kod vrijednosnih kodiranja mijenja vrijednost gena za određeni interval, kod permutacijskog kodiranja odabire dva gena i mijenja im mjesta, dok kod stablastih mijenja mjesto na stablu.

Da bi genetski algoritam provodio ove prethodno navedene genetske operatore prvo mora odabrati određene „dobre“ roditelje za stvaranje nove populacije. Odabir roditelja se vrši pomoću metoda selekcije. Stoga je svrha selekcije čuvanje i prenošenje dobrih svojstava na slijedeću generaciju jedinki. Genetske algoritme s obzirom na vrstu selekcije dijelimo na :

Generacijske – Generacijski genetski algoritam u jednoj iteraciji raspolaže sa dvije populacije (što je ujedno i nedostatak generacijskog genetskog algoritma), jer se odabiru dobre jedinke iz stare populacije koje cine novu populaciju i nakon selekcije sudjeluju u procesu reprodukcije. Vrijeme preživljavanja jedinki je točno jedna generacija.

Eliminacijske – za razliku od generacijske selekcije, eliminacijske selekcije ne bira dobre kromosone za sljedeću populaciju, već loše koje treba eliminirati i reprodukcijom ih zamijeniti sa novima. Dakle, loši kromosoni umiru a njih nadomještaju djeca nastala reprodukcijom roditelja, tj. preživjelih kromosona. Nema stroge granice među generacijama, vrijeme preživljavanja jedinke ovisi o njenoj kvaliteti.

Koristeći bilo koje od ove dvije metode selekcije postoji opasnost da se dobro rješenje dobiveno nakon puno iteracija izgubi ukoliko ga genetski operatori izmjene. Stoga se javlja potreba za mehanizmom zaštite najbolje jedinke od bilo kakve izmjene ili eliminacije tijekom evolucijskog procesa. Takav mehanizam se naziva elitizam. Genetski algoritam s ugrađenim elitizmom iz generacije u generaciju, asimptotski teži ka globalnom optimumu, odnosno rješenju problema.

Funkcija dobrote ili funkcija ocjene kvalitete jedinke se u literaturi još naziva fitness funkcija, funkcija sposobnosti, funkcija cilja ili eval funkcija i u najjednostavnijoj interpretaciji ekvivalent je funkciji f koju treba optimizirati odnosno ona predstavlja naš problem koji pokušavamo riješiti. Što je dobrota jedinke veća, jedinka ima veću vjerojatnost preživljavanja i križanja. Funkcija dobrote je ključ za proces selekcije.

Algoritam roja čestica

Algoritam roja čestica (engl. Particle Swarm Optimization, PSO) je biološki inspiriran metaheuristički algoritam za optimizaciju. Biološka inspiracija je uglavnom navođena socijalnim ponašanjem ptica u jatima i riba u plovovima. PSO algoritmi su originalno osmišljeni od strane Kenedy-a i Eberhart-a 90-ih godina prošlog stoljeća u svrhu proučavanja kretanja ptica, gdje su primijetili da novostvoreni algoritmi omogućavaju pretragu velikog područja mogućih rješenja nekog problema uz zadanu kvalitetu pojedinog rješenja, odnosno da provode optimizaciju.

PSO algoritmi jednako kao i genetski algoritmi (engl. Genetic Algorithms, GA) posjeduju populaciju sačinjenu od niza pojedinih mogućih rješenja koji se ovdje nazivaju čestice. Pošto PSO algoritmi nemaju mogućnost izravnog križanja pojedinih čestica kao što to mogu GA algoritmi putem operatora rekombinacije, čestice se ovdje ne dijele na pojedine nositelje informacije kao što su to geni kod genetskih algoritama. Jednako kao i genetski algoritmi, PSO algoritmi također zahtijevaju neku mjeru određivanja kvalitete pojedinog rješenja, fitnes funkciju tj. funkciju dobrote.

Čestice PSO algoritma se gibaju kroz područje pretraživanja koristeći informacije o vlastitom položaju u prostoru pretraživanja i brzini, te položaju trenutno najbolje čestice u roju. Pri tome u svom radu svaka čestica pamti slijedeće podatke:

- Svoje do sada najbolje pronađeno rješenje problema
- Svoje trenutno rješenje problema
- Trenutno najbolje rješenje u roju kojemu pripada

Na temelju ta tri podatka svaka čestica proračunava novu vlastitu brzinu koju dodaje trenutnom položaju i definira novi položaj promatrane čestice. Dakle, svaka čestica mijenja svoj položaj temeljem vlastitog iskustva, te iskustva bliskih susjeda (na taj se način modelira socijalna interakcija između čestica). Prethodno navedeni podaci se opisuju kao vektori n-dimenzionalnog prostora kojeg se pretražuje:

x – opisuje trenutni položaj čestice u prostoru pretraživanja,

p – opisuje položaj najboljeg rješenja pronađenog od strane promatrane čestice,

v – opisuje smjer (gradijent, brzina) kojem će čestica gibati ako je neometana.

Također su definirane dvije fitnes vrijednosti:

x_{FIT} – mjera kvalitete vektora x

p_{FIT} – mjera kvalitete vektora p

Na razini cijelog roja su poznate vrijednosti:

g – položaj najbolje jedinke u roju

g_{FIT} – mjera kvalitete najbolje jedinke

Čestica prelazi is jednog položaja u drugi na slijedeći način:

$$x_{k+1} = x_k + v_{k+1}$$

gdje je x_{k+1} novi položaj čestice, x_k prošli položaj čestice, a v_{k+1} je nova brzina čestice. Prilikom formiranja novog smjera gibanja odnosno nove brzine čestice uzimaju se u obzir trenutna brzina čestice koja je otežana s konstantom c_0 , smjer gibanja prema nekom prošlom najboljem položaju trenutno razmatrane čestice otežan s c_1 , te smjer gibanja prema najbolje rangiranoj čestici u roju otežan s c_2 . Tada dobivamo sljedeći izraz iz izračunavanje vektora brzine:

$$v_{k+1} = c_0 * v_k + c_1 * \text{rand}() * (p_k - x_k) + c_2 * \text{rand}() * (g_k - x_k)$$

gdje je p_k najbolja postignuta pozicija za razmatranu česticu, a g_k je trenutno najbolja pozicija u roju, „rand()“ funkcija daje nasumični broj u intervalu 0-1, dok konstante c_0 , c_1 i c_2 zadaje korisnik i one definiraju sljedeće:

c_0 – mjera inercije – opisuje bitnost trenutnog smjera;

c_1 – mjera individualnog faktora - opisuje mjeru individualnosti jedinke, potiče istraživanje prostora oko nekog prethodnog najboljeg rješenja promatrane čestice;

c_2 – mjera socijalnog faktora - opisuje mjeru socijalnog utjecaja, potiče detaljnije istraživanje okoline trenutnog najboljeg rješenja pronađenog od svih čestica.

Programski kod

Za GA i PSO se koristi implementacija fitness funkcije prikazana na slici.

```
def evaluateInd(individual):
    temp_rastrigin_val = 0
    # Implement Rastrigin function
    for x in individual:
        temp_rastrigin_val += (x ** 2 - A * math.cos(w * x))
    fit_val = A * NO_DIMS + temp_rastrigin_val
    return fit_val,
```

Slika 1. Fitness funkcija

```
for dim in dims_vec:
    NO_DIMS = dim

    print("# Starting dimension {} #".format(NO_DIMS))
    logger.write("# Starting dimension {} #\n".format(NO_DIMS))

    for inertia in inertia_vec:
        PSO_INERTIA = inertia

        full_dirpath = "PSO/" + str(NO_DIMS) + "/inertia"
        full_filepath = full_dirpath + "/inertia_" + str(inertia) + ".csv"

        iteration_individuals.clear()
        iteration_fitnesses_all.clear()
        iteration_fitnesses_min.clear()

        print("Finding for => inert: {}, pers: {}, soc: {}".format(PSO_INERTIA, PSO_PERSONAL, PSO_SOCIAL))
        logger.write(
            "Finding for => inert: {}, pers: {}, soc: {}\n".format(PSO_INERTIA, PSO_PERSONAL, PSO_SOCIAL))

        while len(iteration_individuals) < 3:
            ui.btnStartPSO_Click()

        median_index = findIndexOfMedian(iteration_fitnesses_min)
        best_index = findIndexOfBest(iteration_fitnesses_min)

        saveToCSV(iteration_fitnesses_all[median_index], full_filepath, full_dirpath)
```

Slika 2. Pronalaženje rješenja fitness funkcije PSO algoritma za parametar inercije

Algoritam je realiziran kroz dvije for petlje i jednu while petlju. Prva for petlja je zadužena za prolazak kroz polje dims_vec koje predstavlja vektor u koji su upisane vrijednosti dimenzija odnosno u našem slučaju 5 i 10. Druga for petlja je zadužena za prolazak kroz polje inertia_vec koje predstavlja vektor u kojem se nalaze vrijednosti parametara inercije koje su u našem slučaju 0.0, 0.37 i 0.74. While služi za pronalazak 5 rješenja svakog određenog parametra. Rješenja se spremaju u csv datoteku. Ostali parametri GA i PSO algoritma se izmjenjuju na isti način.

REZULTATI

Potrebno je realizirati 5-dimenzionalnu i 10-dimenzionalnu Rastriginovu funkciju i pronaći njen minimum koristeći PSO i GA optimizacijske algoritme. Potrebno je usporediti performanse GA i PSO algoritma prilikom pronalaska globalnog minimuma. Rastriginova funkcija za n-dimenzionalni slučaj je sljedeća:

$$f(x) = A*n + \sum (x_i^2 - A*\cos(\omega * x_i))$$

gdje je u našem slučaju A = 10, n = 5 ili 10 (broj dimenzija) i $\omega = 2 \cdot \pi$. Potrebno je ispitati utjecaj promjene parametara genetskog algoritma (faktor mutacije, broj elitnih jedinki, maksimalna vrijednost mutacije), te parametara PSO algoritma (veličina populacije, mjera inercije, mjera individualnog faktora i mjera socijalnog faktora) na kvalitetu dobivenog rješenja. Zbog stohastičke naravi GA i PSO algoritama svaki eksperiment potrebno je ponoviti najmanje pet puta i zabilježiti najbolji i prosječan (medijan) rezultat. Medijan prikazati grafički. Broj generacija/iteracija za sve eksperimente je 5000.

Parametri genetskog algoritma mijenjaju se na sljedeće vrijednosti:

- Populacija: 100
- Mutacija: 5%, 10%, 20%
- Broj elitnih članova: 4, 8, 16
- Najveća apsolutna vrijednost mutacije realnog gena: 0.1, 0.4, 0.8

Parametri PSO algoritma mijenjaju na sljedeće vrijednosti:

- Populacija: 100
- Mjera inercije: 0.0, 0.37, 0.74
- Mjera individualnog faktora: 0.5, 1.0, 1.5
- Mjera socijalnog faktora: 0.5, 1.0, 1.5

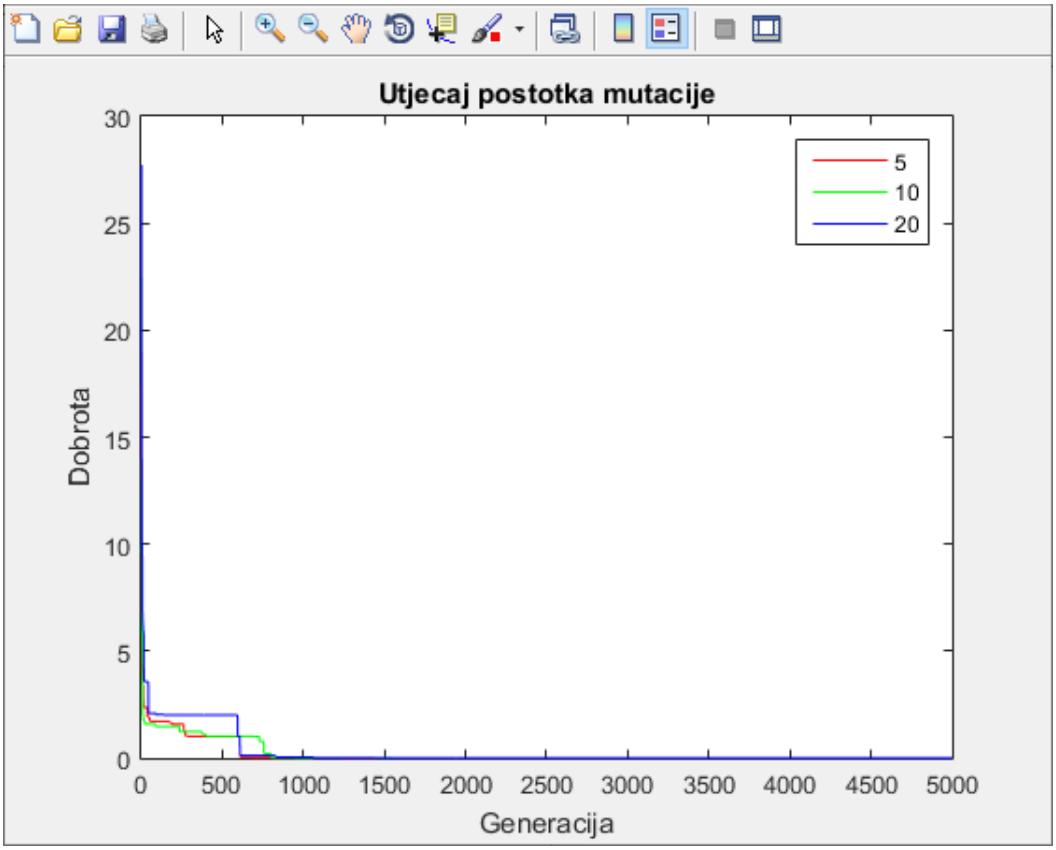
GENETSKI ALGORITAM

N = 5

Utjecaj postotka mutacije

Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Broj elitnih članova	4		
Mutacija	5%	10%	20%
Rezultati	0.000177, 0.000608, 0.000440, 0.00432, 0.000684	0.000986, 0.00276, 0.000982, 0.000676, 0.000761	0.0004182, 0.00208, 6.807e-05, 0.000704, 0.000529
Najbolji rezultat	0.000177	0.000676	6.807-05
Medijan vrijednosti	0.000608	0.000982	0.000529

Tablica 1. Utjecaj postotka mutacije

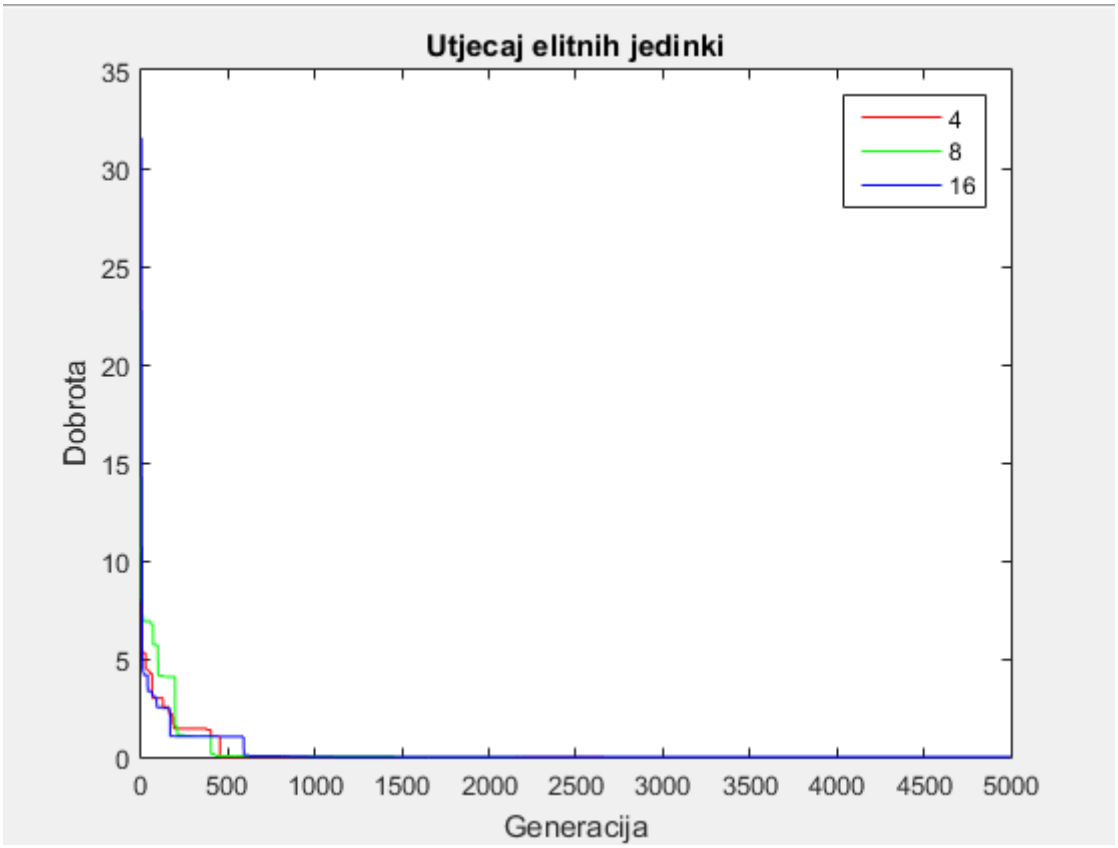


Slika 3. Prikaz fitness funkcije u ovisnosti o postotku mutacije

Utjecaj broja elitnih jedinki

Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Mutacija	5%		
Broj elitnih jedinki	4	8	16
Rezultati	0.00115, 0.000629, 0.00179, 0.000816, 0.000645	0.000286, 0.00141, 0.000394, 0.00114, 0.000380	0.00085, 0.000206, 0.000626, 0.00098, 0.000796
Najbolji rezultat	0.000629	0.000286	0.000206
Medijan vrijednosti	0.000816	0.000394	0.000796

Tablica 2. Utjecaj broja elitnih jedinki

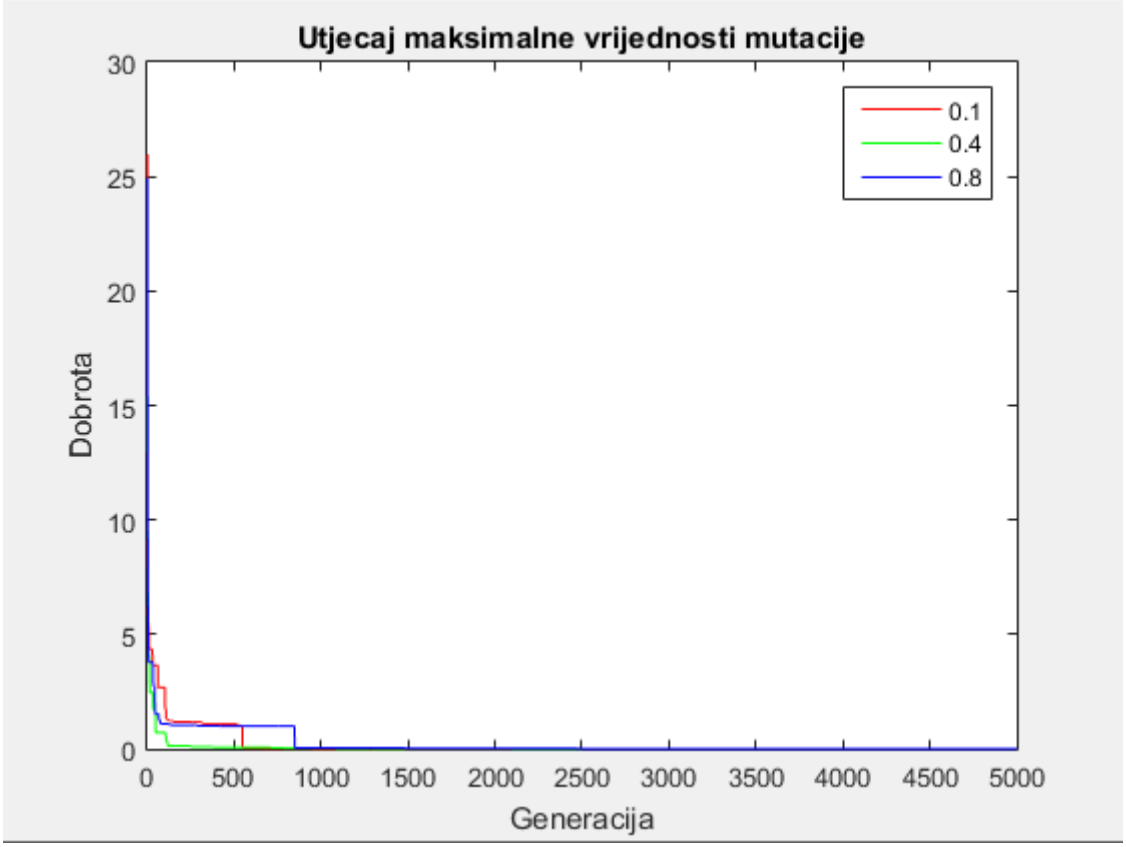


Slika 4. Prikaz fitness funkcije u ovisnosti o broju elitnih jedinki

Utjecaj maksimalne vrijednosti mutacije

Broj elitnih jedinki	4		
Mutacija	5%		
Najveća apsolutna vrijednost mutacije realnog gena	0.1	0.4	0.8
Rezultati	0.000157, 0.00107, 0.000449, 0.00155, 7.691e-05	0.000153, 0.000927, 0.00297, 0.000811, 0.000106	0.000113, 9.203e-05, 0.000463, 0.00145, 0.000217
Najbolji rezultat	7.691e-05	0.000106	9.203e-05
Medijan vrijednosti	0.000449	0.000811	0.000217

Tablica 3. Utjecaj maksimalne vrijednosti mutacije



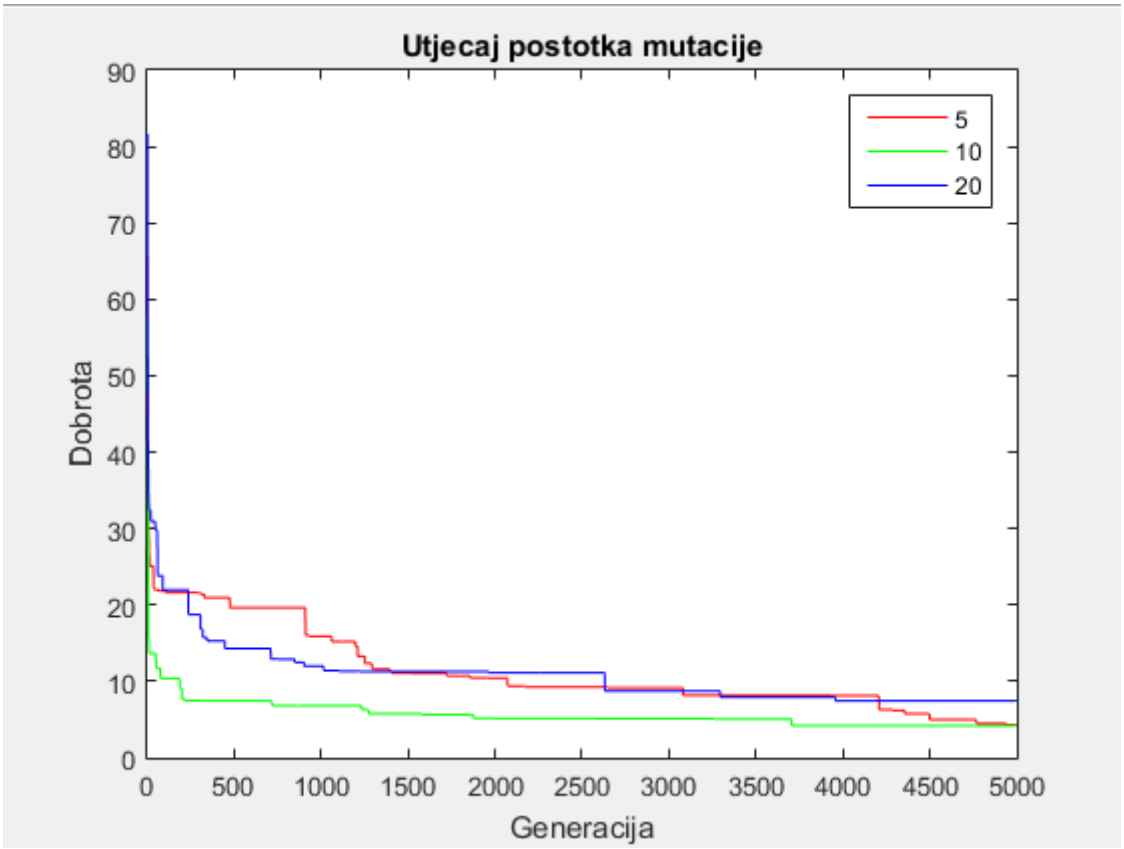
Slika 5. Prikaz fitness funkcije u ovisnosti o maksimalnoj vrijednosti mutacije

N = 10

Utjecaj postotka mutacije

Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Broj elitnih članova	4		
Mutacija	5%	10%	20%
Rezultati	1.0658, 6.261, 5.396, 2.253, 4.319	2.400, 2.391, 5.205, 5.221, 4.182	7.152, 4.288, 2.131, 2.322, 7.452
Najbolji rezultat	1.0658	2.391	2.131
Medijan vrijednosti	4.319	4.182	4.288

Tablica 4. Utjecaj postotka mutacije

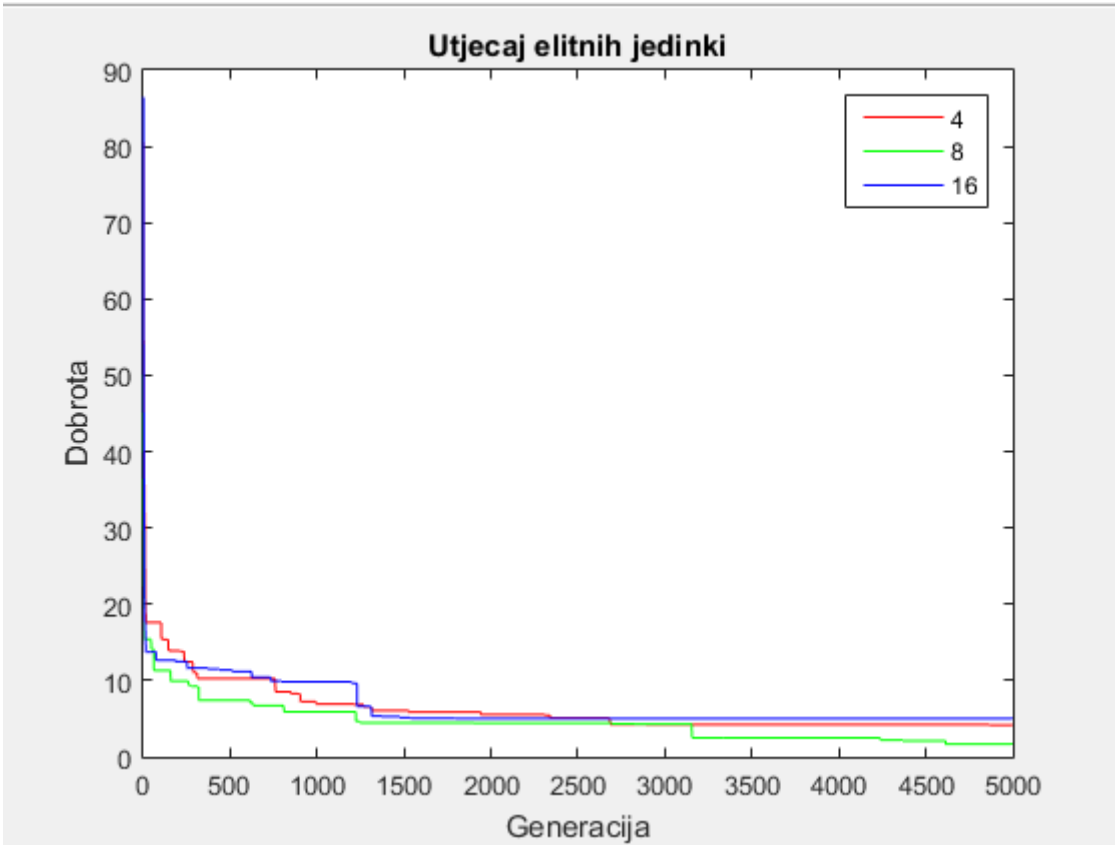


Slika 6. Prikaz fitness funkcije u ovisnosti o postotku mutacije

Utjecaj broja elitnih jedinki

Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Mutacija	5%		
Broj elitnih jedinki	4	8	16
Rezultati	3.238, 1.080, 3.243, 3.216, 4.181	3.243, 3.142, 2.184, 4.195, 1.656	4.242, 3.437, 3.162, 6.522, 5.011
Najbolji rezultat	1.080	1.656	3.162
Medijan vrijednosti	3.238	3.142	4.242

Tablica 5. Utjecaj broja elitnih jedinki

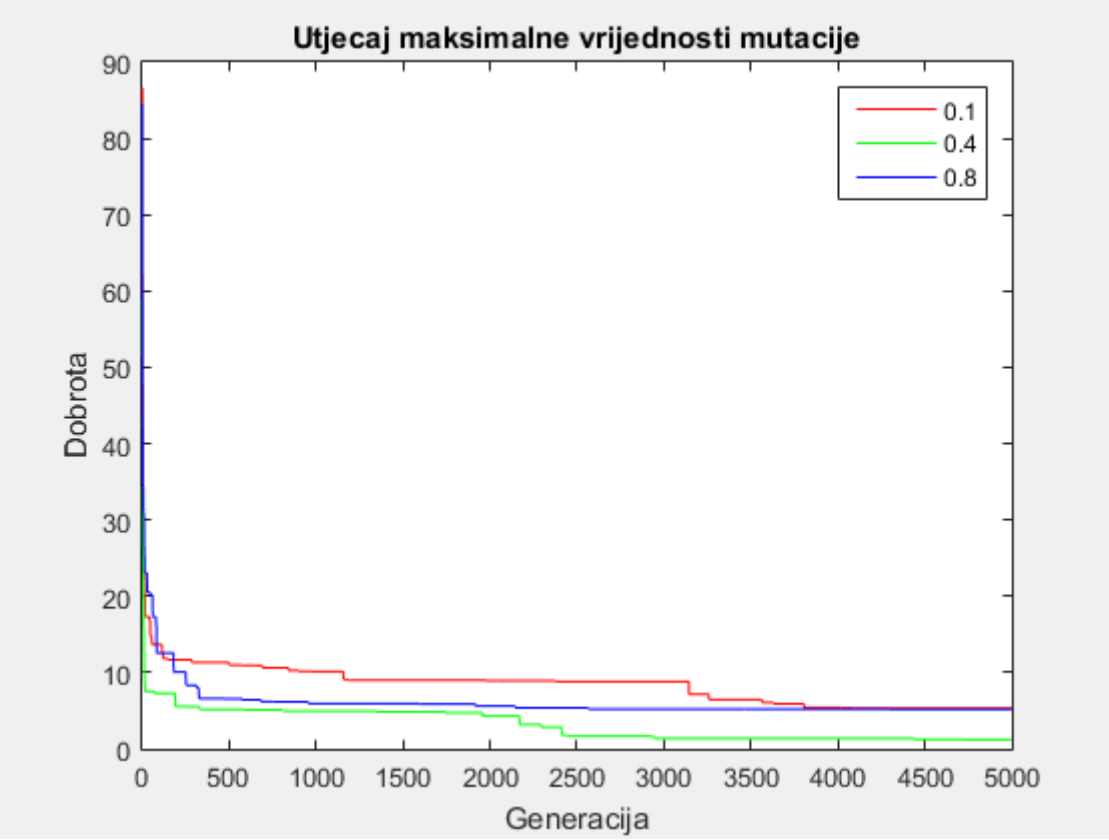


Slika 7. Prikaz fitness funkcije u ovisnosti o broju elitnih jedinki

Utjecaj maksimalne vrijednosti mutacije

Broj elitnih jedinki	4		
Mutacija	5%		
Najveća apsolutna vrijednost mutacije realnog gena	0.1	0.4	0.8
Rezultati	3.2, 5.039, 2.142, 1.089, 5.343	4.218, 2.196, 3.213, 1.761, 1.214	5.147, 4.097, 4.163, 1.074, 5.176
Najbolji rezultat	1.089	1.214	1.074
Medijan vrijednosti	3.2	2.196	4.163

Tablica 6. Utjecaj maksimalne vrijednosti mutacije



Slika 8. Prikaz fitness funkcije u ovisnosti o maksimalnoj vrijednosti mutacije

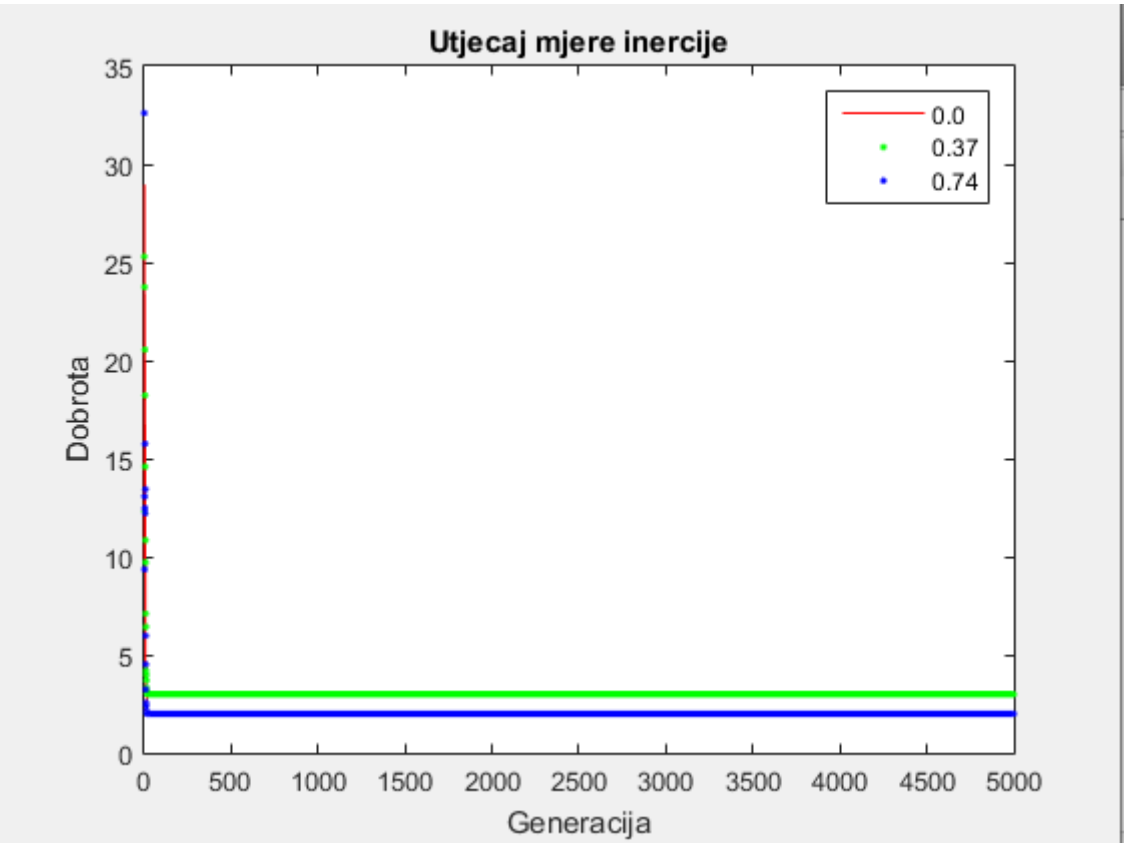
PSO

N = 5

Utjecaj mjere inercije

Mjera socijalnog faktora	1.0		
Mjera individualnog faktora	1.5		
Mjera inercije	0.0	0.37	0.74
Rezultati	3.979, 1.989, 1.989	1.989, 5.969, 2.984	5.969, 1.989, 1.989
Najbolji rezultat	1.989	1.989	1.989
Medijan vrijednosti	3.979	5.969	5.969

Tablica 7. Utjecaj mjere inercije

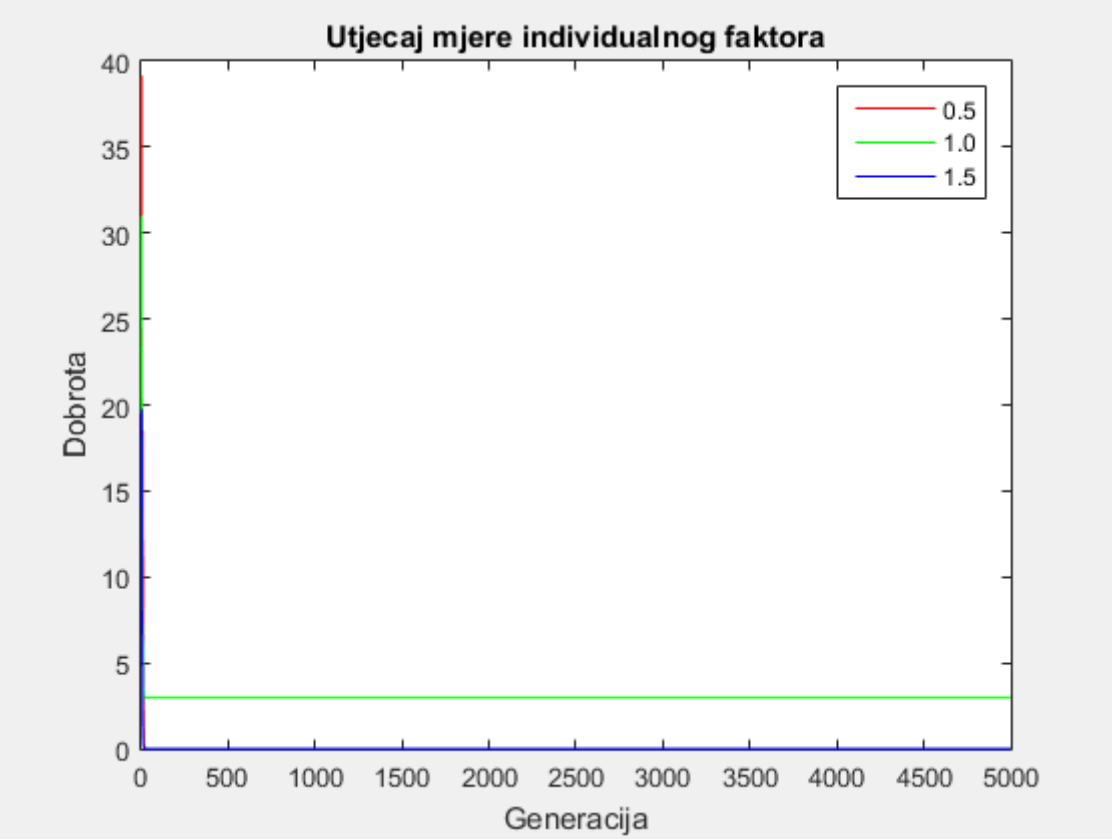


Slika 9. Prikaz fitness funkcije u ovisnosti o mjeri inercije

Utjecaj mjere individualnog faktora

Mjera socijalnog faktora	1.0		
Mjera inercije	0.37		
Mjera individualnog faktora	0.5	1.0	1.5
Rezultati	4.974, 6.964, 3.979, 8.954, 0.0	7.959, 0.994, 1.989, 6.964, 2.984	4.974, 3.979, 4.974, 4.974, 0.0
Najbolji rezultat	0.0	0.994	0.0
Medijan vrijednosti	4.974	2.984	4.974

Tablica 8. Utjecaj mjere individualnog faktora

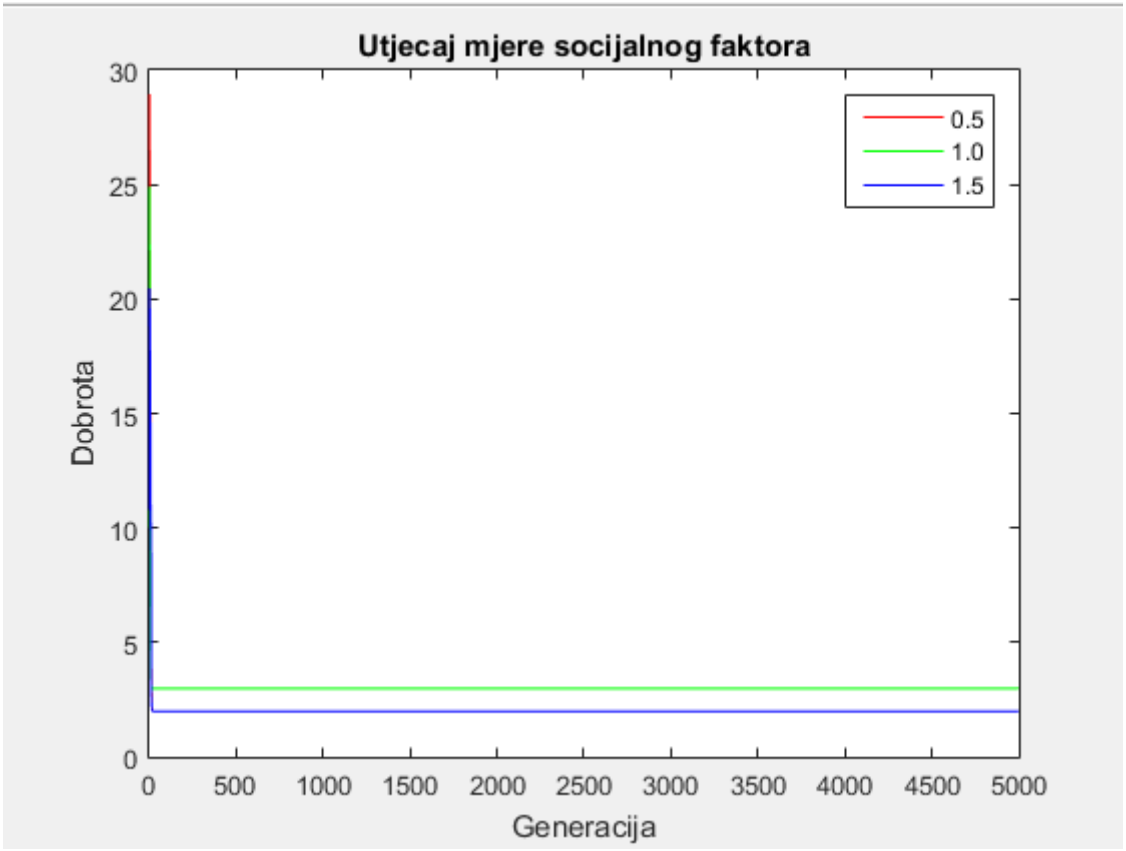


Slika 10. Utjecaj mjere individualnog faktora

Utjecaj mjere socijalnog faktora

Mjere individualnog faktora	1.5		
Mjera inercije	0.37		
Mjera socijalnog faktora	0.5	1.0	1.5
Rezultati	4.974, 6.964, 2.984, 0.994, 1.989	2.984, 0.994, 4.974, 2.984, 2.984	1.989, 2.984, 1.989, 0.994, 1.989
Najbolji rezultat	0.994	0.994	0.994
Medijan vrijednosti	2.984	2.984	1.989

Tablica 9. Utjecaj mjere socijalnog faktora



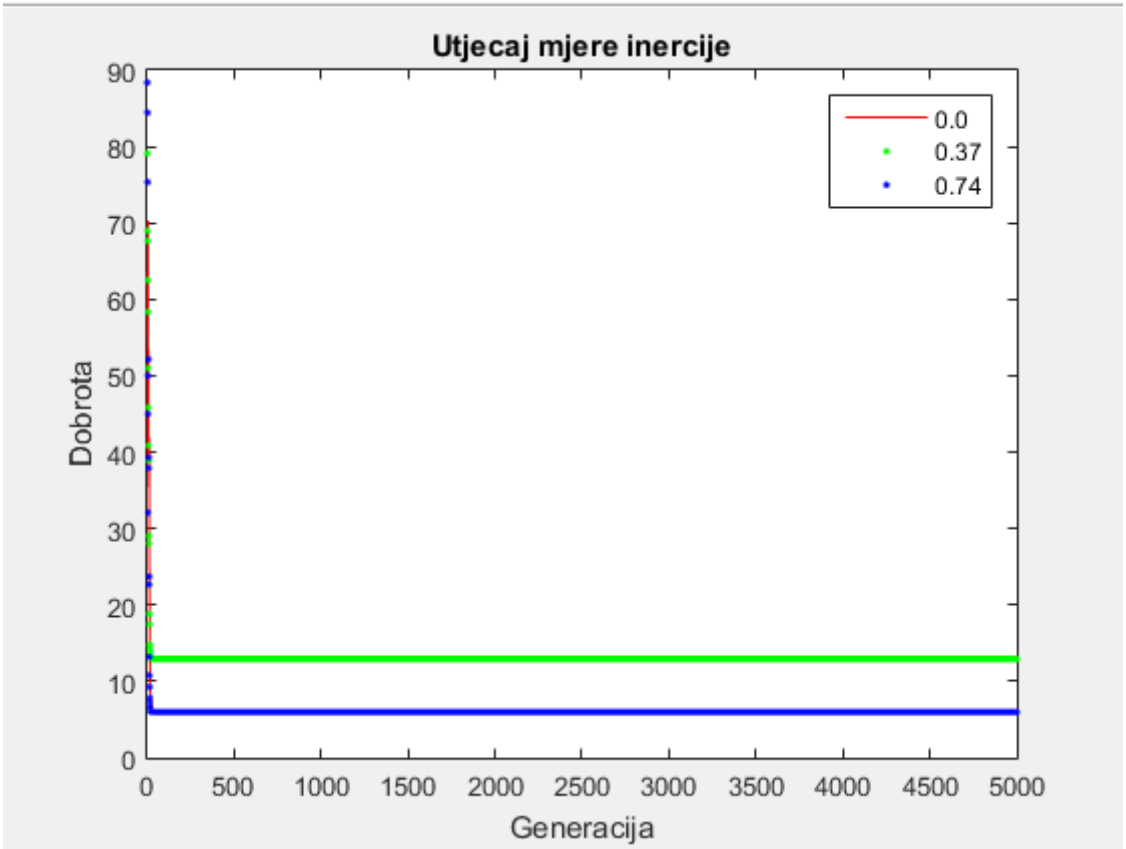
Slika 11. Prikaz fitness funkcije u ovisnosti o mjeri socijalnog faktora

N = 10

Utjecaj mjere inercije

Mjera socijalnog faktora	1.0		
Mjera individualnog faktora	1.5		
Mjera inercije	0.0	0.37	0.74
Rezultati	14.924, 15.1, 6.085	11.12, 19.037, 12.934	25.872, 24.873, 5.97
Najbolji rezultat	6.085	11.12	5.97
Medijan vrijednosti	15.1	19.037	25.872

Tablica 10. Utjecaj mjere inercije

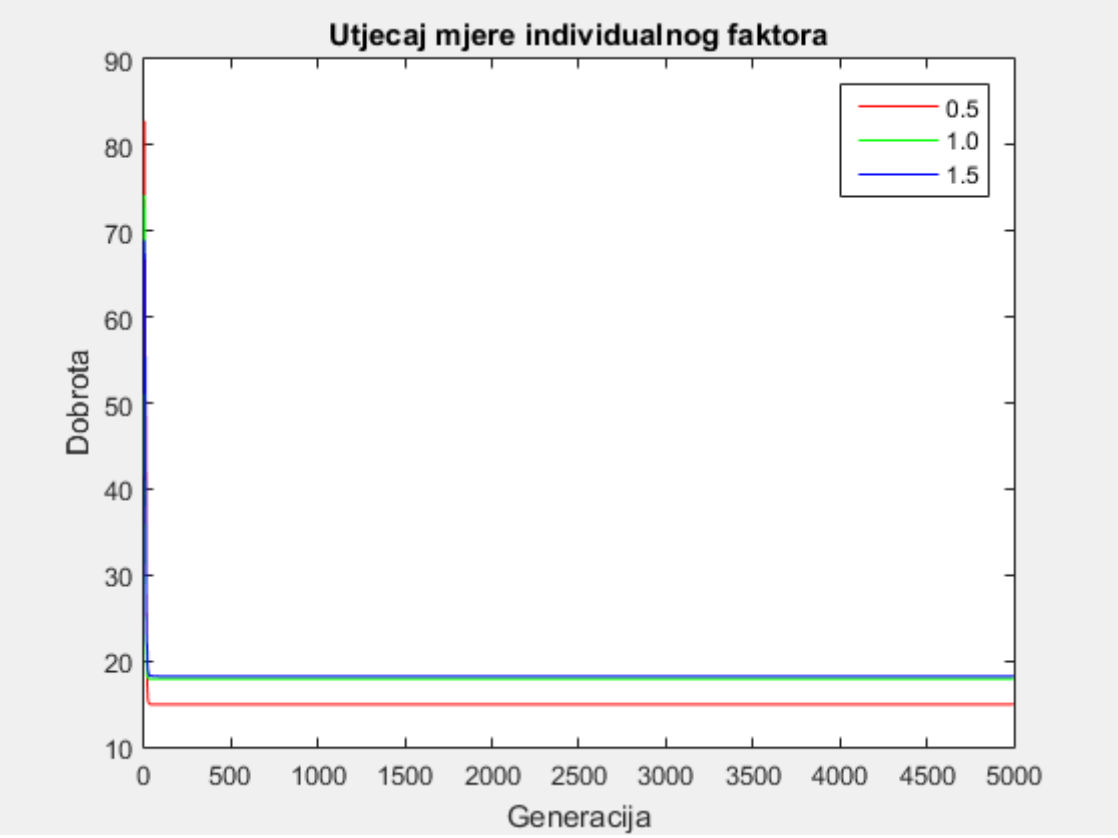


Slika 12. Prikaz fitness funkcije u ovisnosti o mjeri inercije

Utjecaj mjere individualnog faktora

Mjera socijalnog faktora	1.0		
Mjera inercije	0.37		
Mjera individualnog faktora	0.5	1.0	1.5
Rezultati	8.958, 15.983, 20.894, 8.03, 14.92	16.914, 11.942, 14.968, 9.949, 17.9	13.931, 4.979, 8.963, 4.067, 18.205
Najbolji rezultat	8.03	9.949	4.067
Medijan vrijednosti	14.927	14.968	8.963

Tablica 11. Utjecaj mjere individualnog faktora

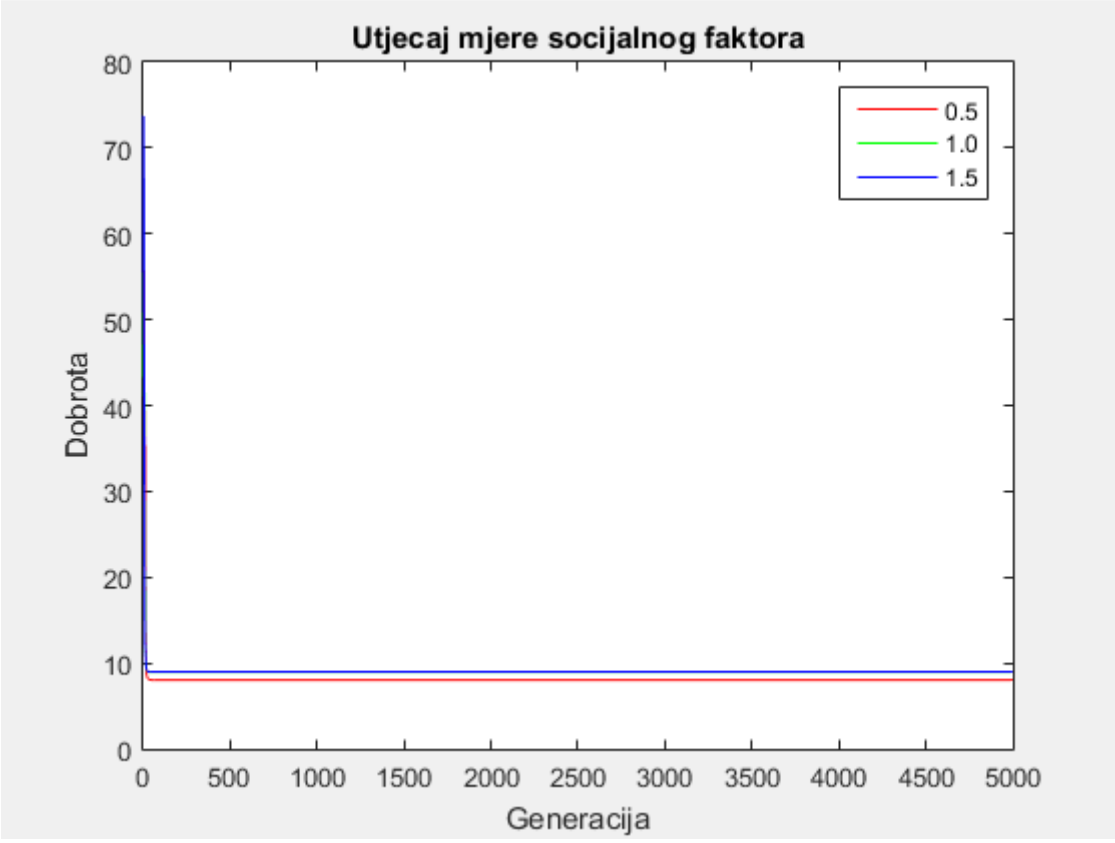


Slika 13. Utjecaj mjere individualnog faktora

Utjecaj mjere socijalnog faktora

Mjere individualnog faktora	1.5		
Mjera inercije	0.37		
Mjera socijalnog faktora	0.5	1.0	1.5
Rezultati	7.0676, 13.929, 12.0814, 10.944, 8.019	9.951, 14.924, 18.908, 15.006, 8.96	9.963, 13.933, 12.935, 7.963, 8.97
Najbolji rezultat	7.0676	8.96	7.963
Medijan vrijednosti	10.944	14.924	9.963

Tablica 12. Utjecaj mjere socijalnog faktora



Slika 14. Prikaz fitness funkcije u ovisnosti o mjeri socijalnog faktora

ZAKLJUČAK

Nakon izvedene vježbe odnosno pronalaženja globalnog minimuma 5-dimenzionalne i 10-dimenzionalne Rastriginove funkcije vidi se da genetski algoritam daje točnije rješenje. Iako je pronađen globalni minimum i to preko PSO algoritma, ipak genetski algoritam daje točnija rješenja. Korištenjem genetskog algoritma za traženje globalnog minimuma Rastringinove funkcije najbliže rješenje se dobilo parametrima: broj elitnih jedinki = 4, mutacija = 5%, najveća apsolutna vrijednost mutacije realnog gena = 0.4. Korištenjem tih parametara dobili smo rješenje 0.000177. Korištenjem PSO algoritma pronađen je globalni minimum i to sa parametrima mjera socijalnog faktora=1.0,mjera inercije= 0.37 i mjera individualnog faktora = 0.5 i 1.5. Povećanjem postotka mutacije kod genetskog algoritma dobivena su bolja rješenja. Iako povećanjem mutacije može doći i do lošijih rješenja zato što s većim postotkom mutacije postoje šanse i da se dobre jedinke izmjene. Što se tiče utjecaja broja elitnih članova, bolja rješenja su dobivena sa vrijednostima 8 i 16 za N = 5, dok je za vrijednost 4 dobiveno lošije rješenje. Teoretski ukoliko je broj elitnih članova premalen može doći do gubitka dobre jedinke, dok prevelik broj elitnih članova može prenijeti dosta loših jedinki, što kraju utječe na točnost rezultata i na vrijeme izvođenja programa.

Što je faktor inercije manji to su pomaci u prostoru rješenja manja i postoji veća vjerojatnost da algoritam zaglavi u lokalnom minimumu. Za veće vrijednosti mjere inercije čestice bi radile velike pomake u prostoru pa postoji velika vjerojatnost da će čestica preletjeti preko potencijalno dobrih rješenja. Povećavanjem mjere inercije za N = 10 kod PSO algoritma rješenja su postajala lošija. Najbolja su bila sa mjerom inercije 0.0.

Veća vrijednost individualne komponente će potencirati eksploraciju, dok će veće vrijednosti kognitivne i socijalne komponente potencirati eksplorativno ponašanje. Razlog tome je što zanemarivanjem kognitivne i socijalne komponente čestica u manjoj mjeri odstupa od svoje (početne) putanje, a zanemarivanjem individualne komponente, snažno je privučena lokalnim optimumima te se time smanjuje vjerojatnost nailaženja na druga kvalitetna područja u prostoru rješenja.

Vrijeme izvođenja genetskog algoritma: 41 minuta

Vrijeme izvođenja PSO algoritma: 12 minuta