

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

**Rješavanje Problema Pronalaska Minimuma N -
Dimenzionalne Funkcije Koristeći PSO i GA Algoritme**

Meko računarstvo

Laboratorijska vježba 3

Ivan Gudelj

Diplomski studij računarstva, DRB

Osijek, 2022.

UVOD	3
OPIS PROBLEMA I RJEŠENJA	3
Genetski algoritam	3
Algoritam roja čestica (PSO)	5
Rješavanje problema i fitness funkcija	7
Funkcija dobrote (fitness function)	8
REZULTATI ZA TRAŽENI PROBLEM - GENETSKI ALGORITAM (GA)	9
Rezultati za n = 5	9
Utjecaj promjene postotka mutacije	9
Utjecaj promjene broja elitnih članova	10
Utjecaj promjene apsolutne vrijednosti mutacije realnog gena	12
Rezultati za n = 10	13
Utjecaj promjene postotka mutacije	13
Utjecaj promjene broja elitnih članova	15
Utjecaj promjene apsolutne vrijednosti mutacije realnog gena	16
REZULTATI ZA TRAŽENI PROBLEM - ALGORITAM ROJA ČESTICA (PSO)	17
Rezultati za n = 5	17
Utjecaj promjene mjere inercije	17
Utjecaj promjene individualnog faktora	19
Utjecaj promjene mjere socijalnog faktora	20
Rezultati za n = 10	21
Utjecaj promjene mjere inercije	21
Utjecaj promjene individualnog faktora	23
Utjecaj promjene mjere socijalnog faktora	24
ZAKLJUČAK	26

1. UVOD

Cilj treće laboratorijske vježbe bio je proučiti osnove i način rada genetskog algoritma te algoritma roja čestica kroz problem pronalaska minimuma n -dimenzionalne funkcije (Rastriginova funkcija). Za rješavanje ovog problema korišten je Python programski jezik. U vježbi će biti uspoređen rad genetskog algoritma i algoritma roja čestica na ovom primjeru. Kod genetskog algoritma parametri će biti populacija, mutacija, broj elitnih članova te najveća apsolutna vrijednost mutacije realnog gena. Kod PSO algoritma parametri će biti populacija, mjera inercije, mjera individualnog faktora te mjera socijalnog faktora. Zbog stohastičke naravi GA i PSO algoritama svaki eksperiment potrebno je ponoviti najmanje pet puta i zabilježiti najbolji i prosječan (medijan) rezultat.

2. OPIS PROBLEMA I RJEŠENJA

2.1. Genetski algoritam

Prvo ćemo opisati osnove genetskog algoritma. Genetski algoritam je metaheuristička metoda optimiranja koja se temelji na ideji biološke evolucije vrsta, tj. preživljavanje najposposobnijih vrsta. Evolucija je prirodni proces traženja najbolje i najprilagodljivije jedinke u okolini i uvjetima u prirodi. Jedinka koja je najbolje prilagođena uvjetima i okolini u kojoj živi ima najveću vjerojatnost preživljavanja i parenja, a time i prenošenja svojega genetskog materijala na svoje potomke. Genetički podaci (parametri) koji obilježavaju jedinku zapisani su u kromosomima.

U populaciji jedne vrste, nova se jedinka stvara selekcijom (reprodukcijom) i rekombinacijom (križanjem) genetičkih materijala (gena) obaju roditelja. Time se dobiva različitost među jedinkama iste vrste, ali i sličnosti s roditeljima jedinke. Na gen jedinke (djeteta) može djelovati i mutacija. Riječ je o slučajnom mijenjanju genetskog materijala koji nastaje pod djelovanjem vanjskih uzroka. Za dobivanje dobrog rješenja genetskog algoritma dovoljno je kodirati problem i kvalitetno definirati funkciju cilja (definirati što je dobro rješenje).

Odabrani roditelji dobrih svojstava imaju šansu dati potomka koji će imati bolja svojstva od svakog pojedinog roditelja. Roditelji dobrih svojstava imaju veću šansu dati potomke i prenijeti svoja svojstva (gene) u iduću generaciju. Svaka sljedeća generacija imat će sve više dobrih svojstava.

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.

Osnovna struktura genetskog algoritma podjeljena je na 6 koraka:

- Generiraj početnu populaciju mogućih rješenja (kromosomi)
- Odredi sposobnost svakog kromosoma u populaciji
- Generiraj nove kromosome koristeći genetičke operatore
- Odbaci nepoželjne jedinke (kromosome) populacije
- Uključi nove kromosome u populaciju da se stvori nova populacija
- Korake 2. – 6. nastavi sve dok nije zadovoljen unaprijed određeni uvjet

Genetski algoritam 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 i eliminacijske. Način kodiranja ili prikaz rješenja može bitno utjecati na učinkovitost genetičkog algoritma, pa je stoga izbor prikaza izuzetno značajan. Neki od tipova genetskog kodiranja su: binarni, vrijednosni, permutacijski i stablasti.

Kod binarnog kodiranja gen može poprimiti samo dvije vrijednosti: 0 ili 1 (primjer punjenje spremnika, određuje se da li je određeni predmet u spremniku). Kod vrijednosnog kodiranja gen može poprimiti cjelobrojne/realne vrijednosti iz zadanog intervala (primjeri traženja maksimuma f-je više varijabli). Kod permutacijskog kodiranja gen može poprimiti cjelobrojne vrijednosti tako da kromosom uvijek sadrži sve brojeve 1 ... N u različitom redoslijedu (primjer: problem trgovačkog putnika, pronaći najkraći put obilaska N gradova

tako da se svaki grad posjeti točno jednom). Kod stablastog kodiranja gen je čvor stabla (primjer: pronalaženje analitičke funkcije iz skupa vrijednosti).

Veličina populacije N određuje se na samom početku algoritma. Najčešće se početna populacija kromosoma (ili potencijalnih rješenja) generira tako da se generira N slučajnih brojeva ili rješenja u intervalu $[d, g]$ te se prikazu u odgovarajućem obliku ovisno o načinu prikaza. Obično se uzima da je $N = \text{konst}$, odnosno veličina populacije se ne mijenja tijekom evolucije.

Najvažniji dio primjene genetskog algoritma je zapravo funkcija dobrote, tj. fitness funkcija. Ova funkcija govori koliko je određeno rješenje dobro i što je dobrota jedinke, tj. rješenja veća (odnosno manja, ovisno o problemu i načinu implementacije), jedinka ima veću vjerojatnost preživljavanja i križanja. Ova funkcija je ključ za proces selekcije. Funkcija dobrote može biti bilo koja nelinearna, prekidna, nederivabilna pozitivna funkcija jer je bitno samo odrediti dobrotu za svaki kromosom.

2.2. Algoritam roja čestica (PSO)

Algoritam roja čestica (engl. Particle Swarm Optimization, PSO) je također biološki inspiriran metaheuristički algoritam za optimizaciju. PSO algoritmi su originalno osmišljeni od strane Kenedy-a i Eberhart-a 90-ih godina prošlog stoljeća u svrhu proučavanja kretanja ptica.

PSO algoritmi jednako kao i genetski algoritmi (engl. Genetic Algorithms, GA) posjeduju populaciju sačinjenu od niza pojedinih mogućih rješenja koji se ovdje nazivaju čestice. PSO nema mogućnost izravnog križanja pojedinih čestica kao što to imaju GA algoritmi putem operatora rekombinacije. Zbog toga se čestice 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, *fitness 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.

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 fitness 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 iz jednog položaja u drugi na sljedeći način:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{v}_{k+1}$$

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 **c0**, smjer gibanja prema nekom prošlom najboljem položaju trenutno razmatrane čestice otežan s **c1**, te smjer gibanja prema najbolje rangiranoj čestici u roju otežan s **c2**. Tada dobivamo sljedeći izraz iz izračunavanja vektora brzine:

$$\mathbf{v}_{k+1} = \mathbf{c}_0 * \mathbf{v}_k + \mathbf{c}_1 * \text{rand}() * (\mathbf{p}_k - \mathbf{x}_k) + \mathbf{c}_2 * \text{rand}() * (\mathbf{g}_k - \mathbf{x}_k)$$

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 **c0**, **c1** i **c2** zadaje korisnik i one definiraju sljedeće:

c0—mjera inercije – opisuje bitnost trenutnog smjera;

c1—mjera individualnog faktora - opisuje mjeru individualnosti jedinke, potiče istraživanja prostora oko nekog prethodnog najboljeg rješenja promatrane čestice;

c2—mjera socijalnog faktora - opisuje mjeru socijalnog utjecaja, potiče detaljnije istraživanje okoline trenutnog najboljeg rješenja pronađenog od svih čestica.

2.3. Rješavanje problema i fitness funkcija

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.

—GA—

Parametre genetskog algoritma mijenjati 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

Broj generacija/iteracija za sve eksperimente je 15000. Rastriginova funkcija za n dimenzionalni slučaj je sljedeća:

$$f(x) = A \cdot n + \sum_{i=1}^n (x_i^2 - A \cdot \cos(\omega \cdot x_i))$$

A=10, n=5 ili 10 (broj dimenzija) i omega=2*pi.

—PSO—

PSO algoritma mijenjati 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

Broj generacija/iteracija za sve eksperimente je 15000. Rastriginova funkcija za n dimenzionalni slučaj je sljedeća:

$$f(x) = A \cdot n + \sum_{i=1}^n (x_i^2 - A \cdot \cos(\omega \cdot x_i))$$

$A=10$, $n=5$ ili 10 (broj dimenzija) i $\omega=2\pi$.

Funkcija dobrote (*fitness function*)

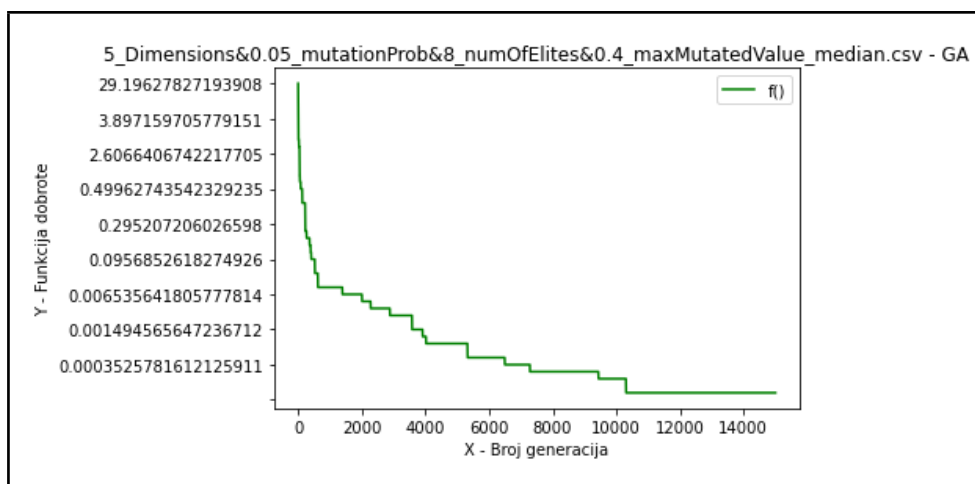
```
#Define evaluation (fitness) function for individual (cromosome or particle)
#Rastrigin function
A = 10
n=5 #ILI n=10
omega = 2*np.pi
def evaluateInd(individual):
    is_OPTIMIZED = False
    #Implement Rastrigin function
    fit_val = A * len(individual)
    counter = 0
    for newGene in individual:
        fit_val += newGene*newGene - A*math.cos(omega*newGene)
        if newGene == 0.0:
            counter+=1
    if counter == len(individual):
        is_OPTIMIZED = True
    return fit_val,
```

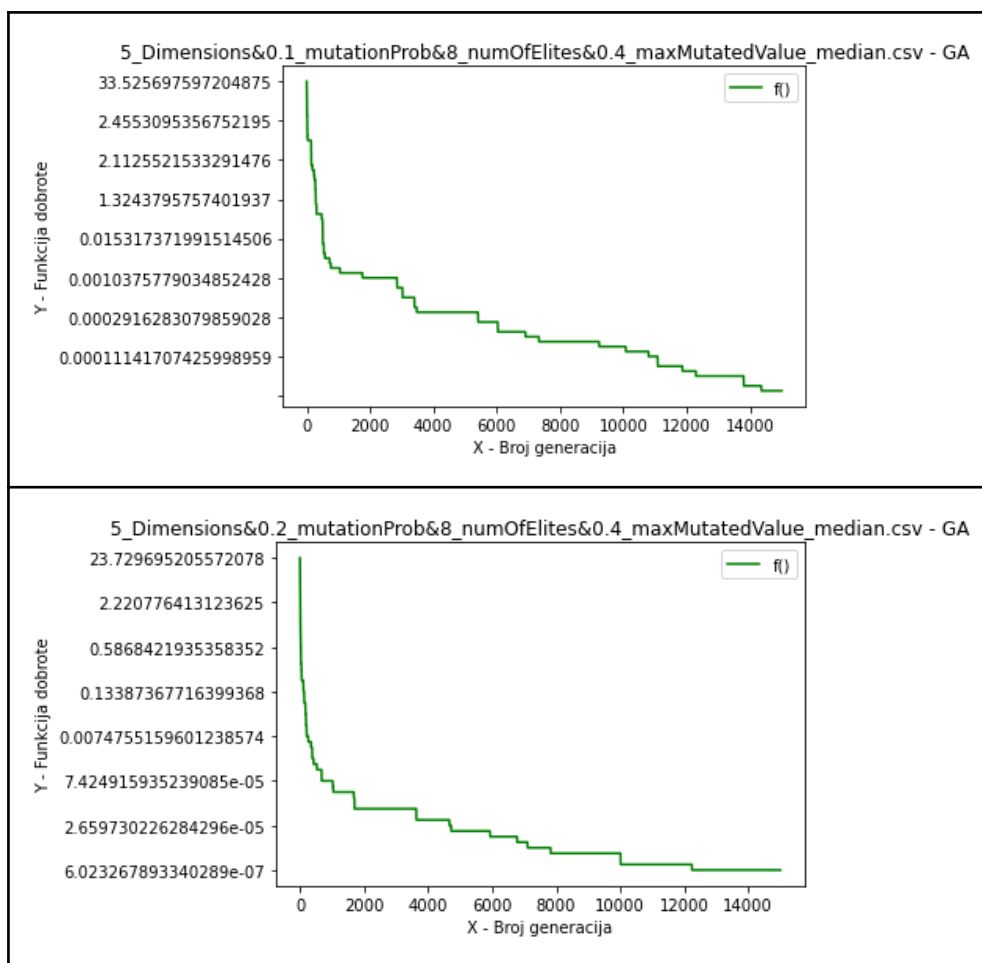

3. REZULTATI ZA TRAŽENI PROBLEM - GENETSKI ALGORITAM (GA)

3.1. Rezultati za n = 5

3.1.1. Utjecaj promjene postotka mutacije

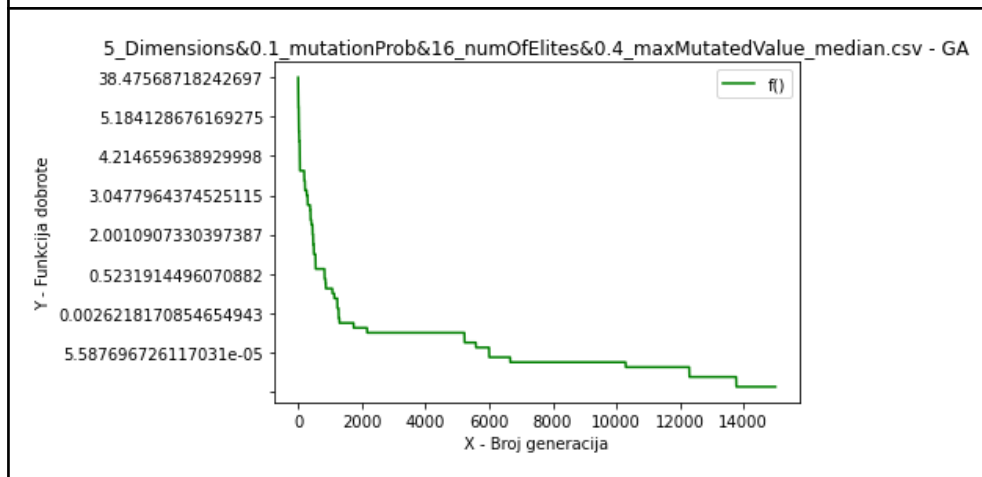
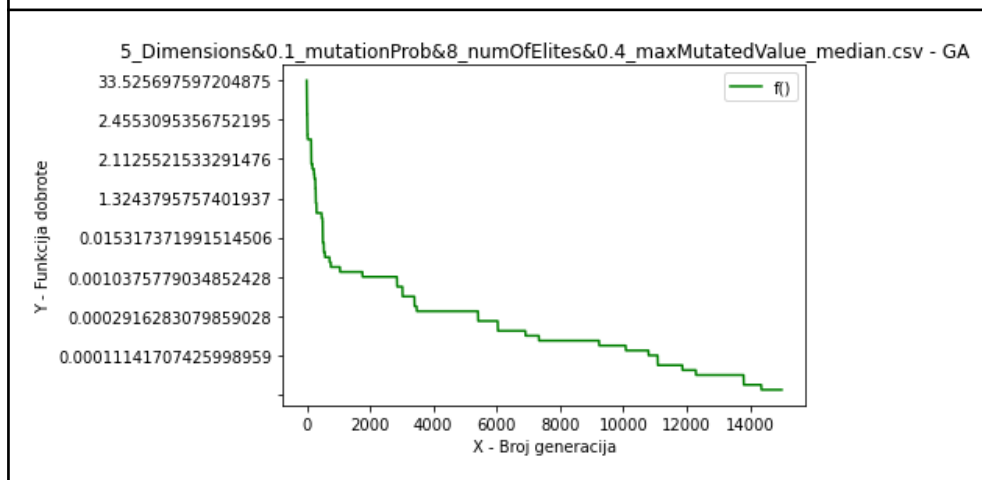
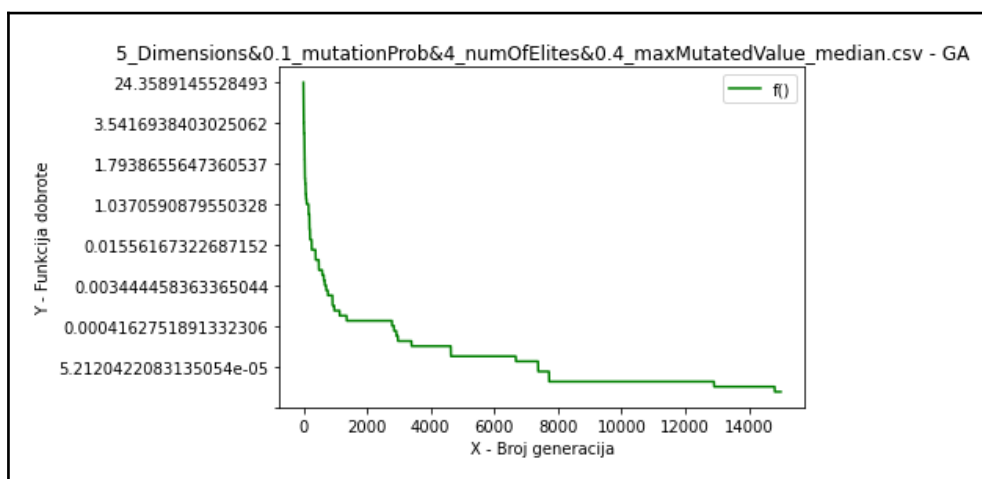
Broj elitnih jedinki	8		
Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Postotak mutacije	5%	10%	20%
Vrijednosti pronađenih rješenja	2.174e-05, 2.304e-05, 2.989e-05, 4.268e-05, 9.169e-05	6.998e-06, 9.638e-06, 1.312e-05, 1.376e-05, 3.839e-05	2.528e-07, 3.527e-07, 6.023e-07, 2.197e-06, 2.789e-06
Prosječna vrijednost	6.364e-05	1.638e-05	1.239e-05
Median vrijednost	2.989e-05	1.312e-05	6.023e-07





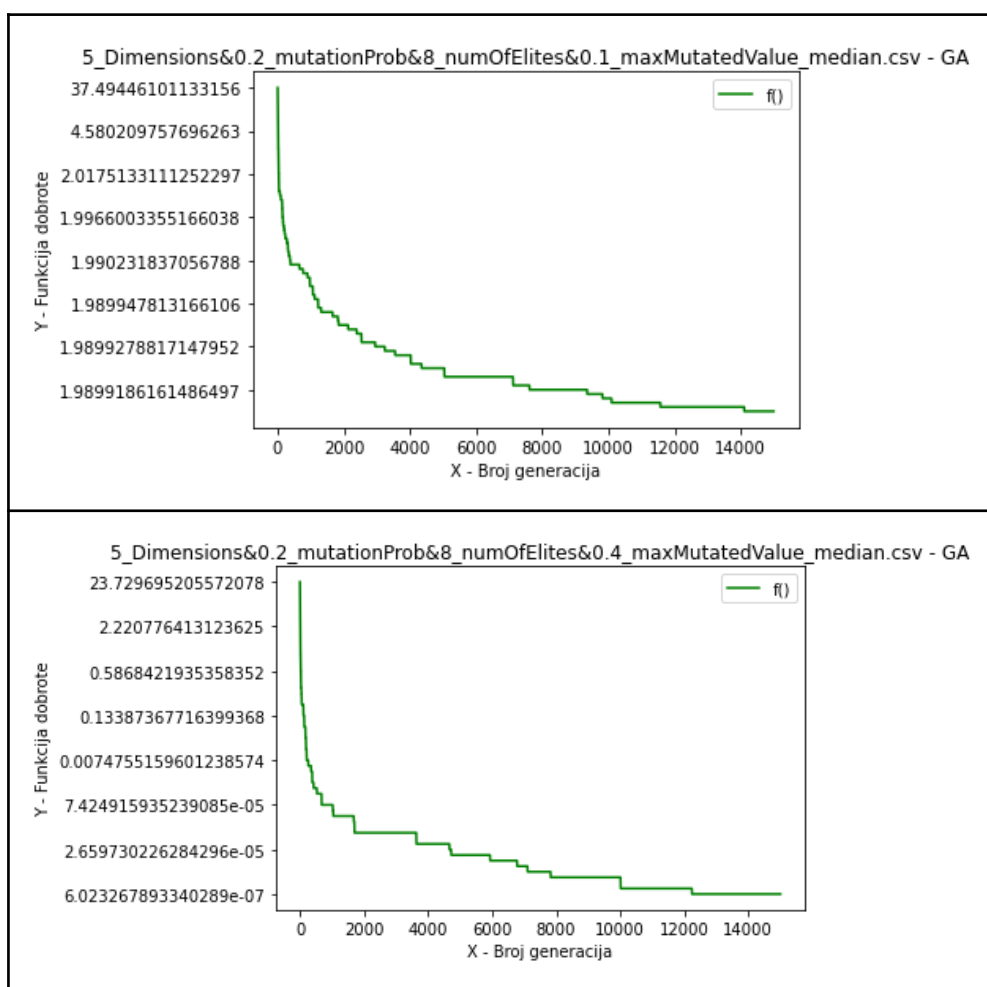
3.1.2. Utjecaj promjene broja elitnih članova

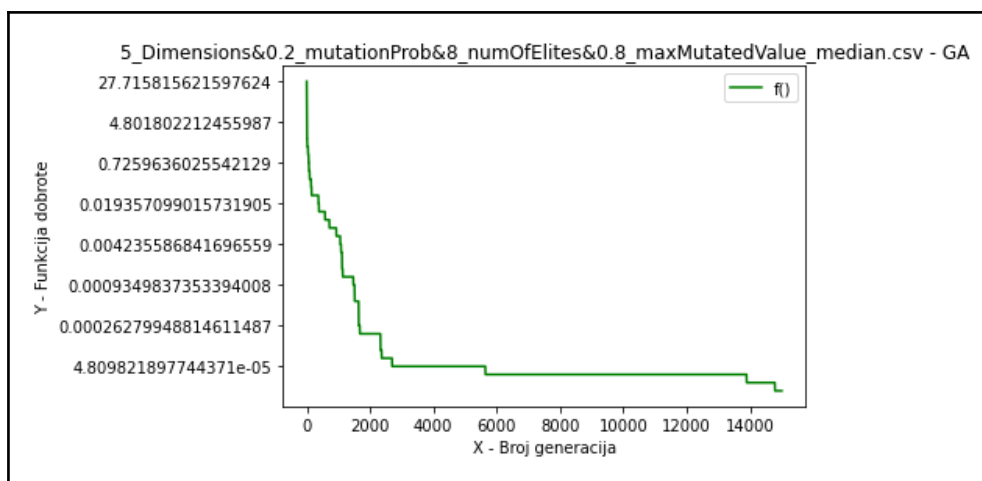
Postotak mutacije	10%		
Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Broj elitnih jedinki	4	8	16
Vrijednosti pronađenih rješenja	5.091e-06, 1.127e-05, 1.483e-05, 2.230e-05, 5.603e-05	6.998e-06, 9.638e-06, 1.312e-05, 1.376e-05, 3.839e-05	1.641e-05, 2.197e-05, 2.822e-05, 4.645e-05, 10.003e-05
Prosječna vrijednost	2.190e-05	1.638e-05	4.262e-05
Median vrijednost	1.483e-05	1.312e-05	2.822e-05



3.1.3. Utjecaj promjene apsolutne vrijednosti mutacije realnog gena

Broj elitnih jedinki	8		
Postotak mutacije	20%		
Najveća apsolutna vrijednost mutacije realnog gena	0.1	0.4	0.8
Vrijednosti pronađenih rješenja	0.994, 1.989, 1.989, 1.989, 2.984	2.528e-07, 3.527e-07, 6.023e-07, 2.197e-06, 2.789e-06	3.241e-06, 5.782e-06, 1.234e-05, 1.539e-05, 2.151e-05
Prosječna vrijednost	1.989	1.239e-06	1.165e-05
Median vrijednost	1.989	6.023e-07	1.234e-05

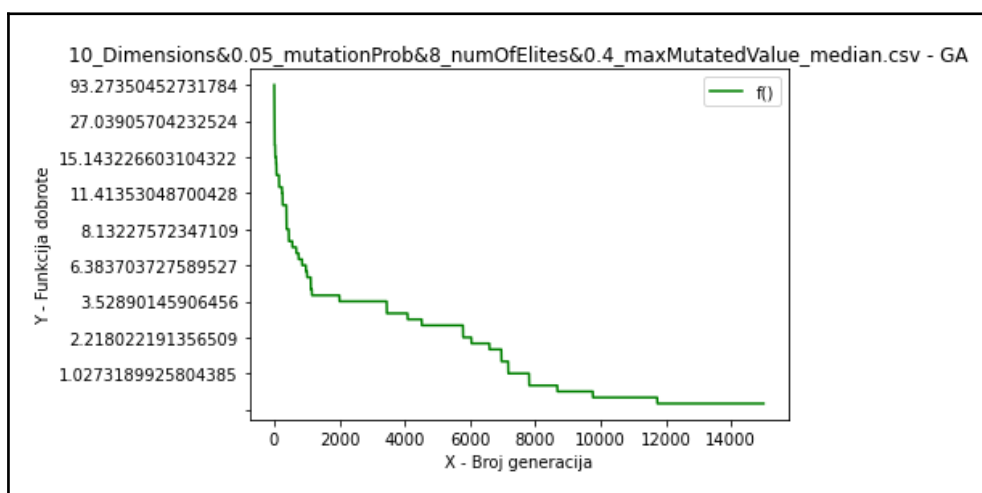


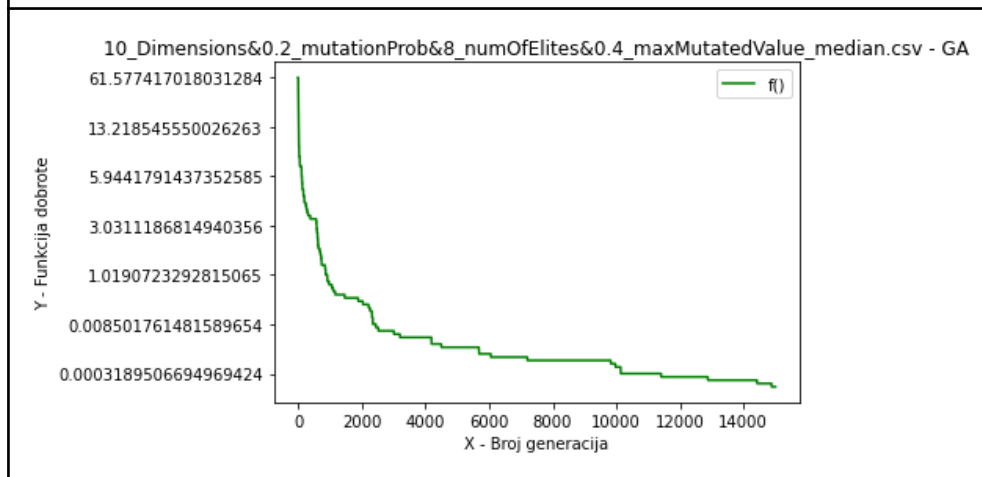
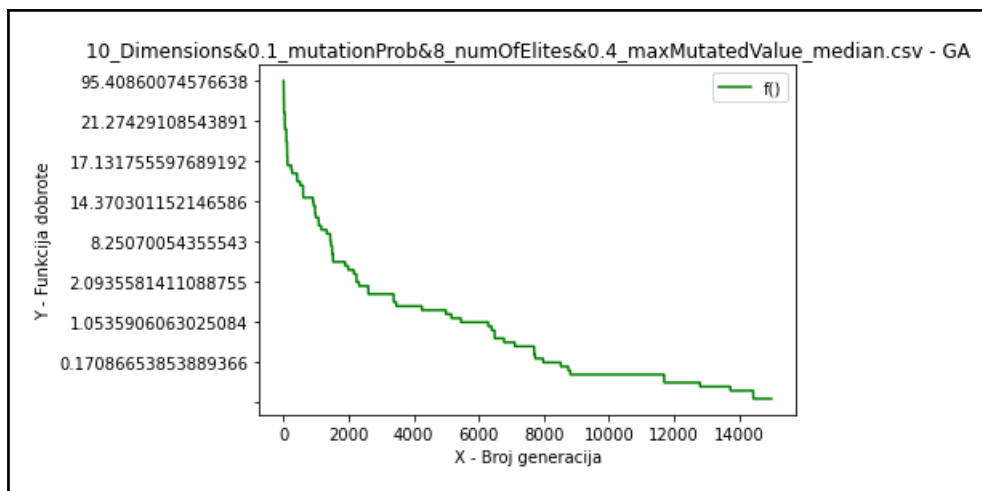


3.2. Rezultati za $n = 10$

3.2.1. Utjecaj promjene postotka mutacije

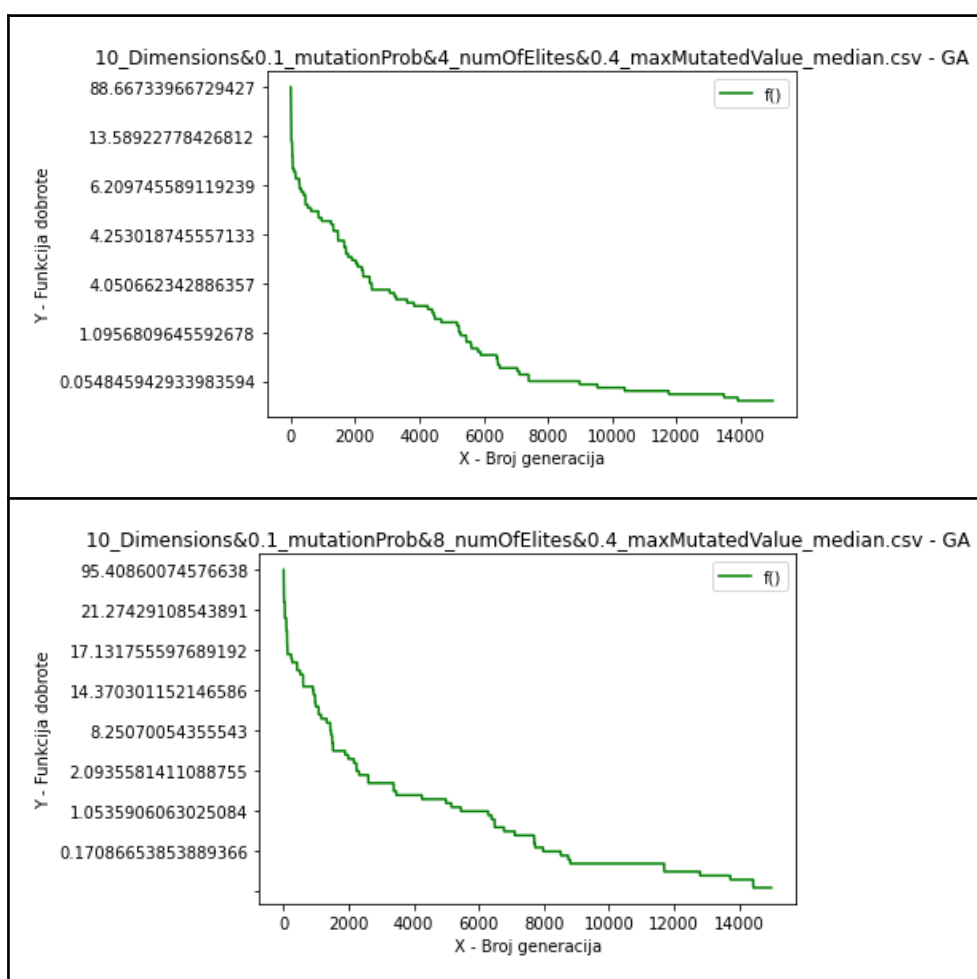
Broj elitnih jedinki	8		
Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Postotak mutacije	5%	10%	20%
Vrijednosti pronađenih rješenja	0.014, 0.128, 1.014, 1.156, 2.139	0.002, 0.003, 0.003, 0.005, 0.008	4.22e-05, 16.2e-05, 16.7e-05, 16.9e-05, 24.6e-05
Prosječna vrijednost	0.8902	0.0042	15.724e-05
Median vrijednost	1.014	0.003	16.7e-05

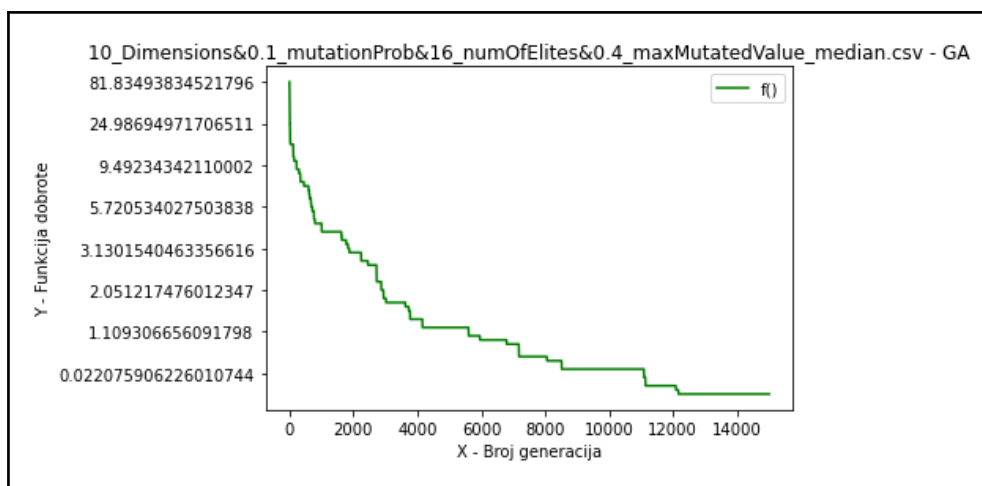




3.2.2. Utjecaj promjene broja elitnih članova

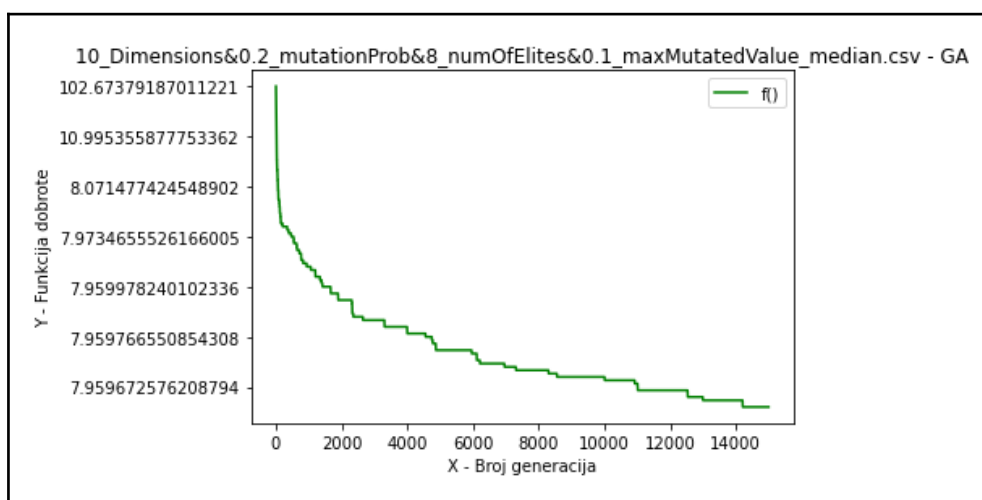
Postotak mutacije	10%		
Najveća apsolutna vrijednost mutacije realnog gena	0.4		
Broj elitnih jedinki	4	8	16
Vrijednosti pronađenih rješenja	0.00086, 0.00107, 0.00168, 0.00235, 0.00299	0.00241, 0.00346, 0.00351, 0.00557, 0.00823	0.0016, 0.0017, 0.0021, 0.0021, 0.0088
Prosječna vrijednost	0.00179	0.004636	0.00326
Median vrijednost	0.00168	0.00351	0.0021

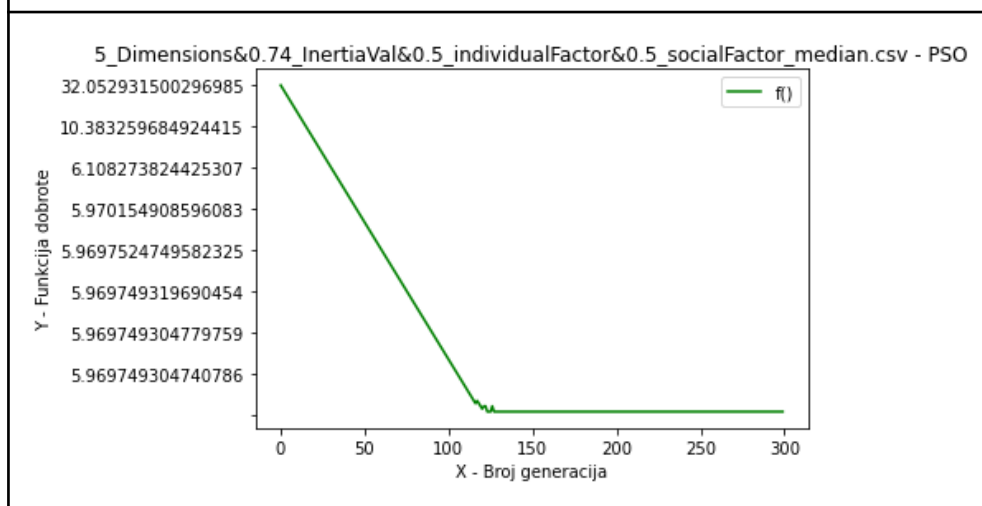
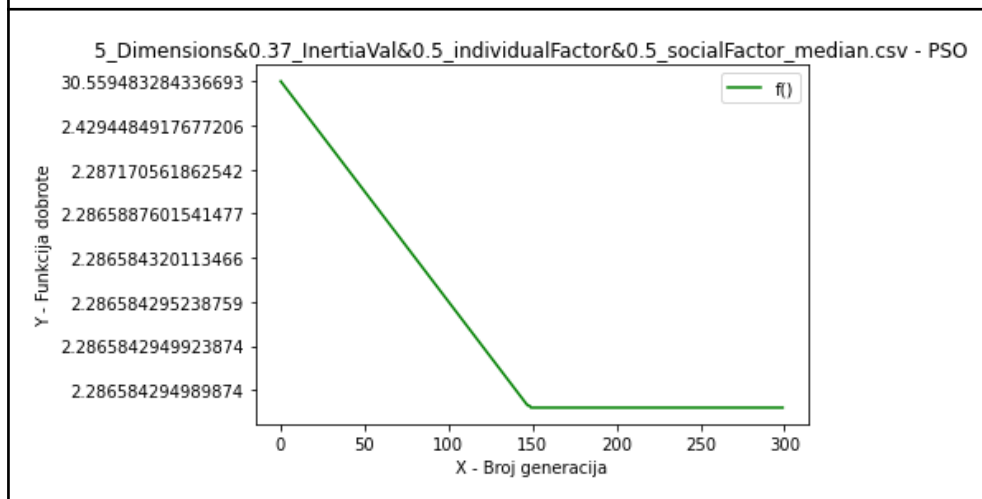
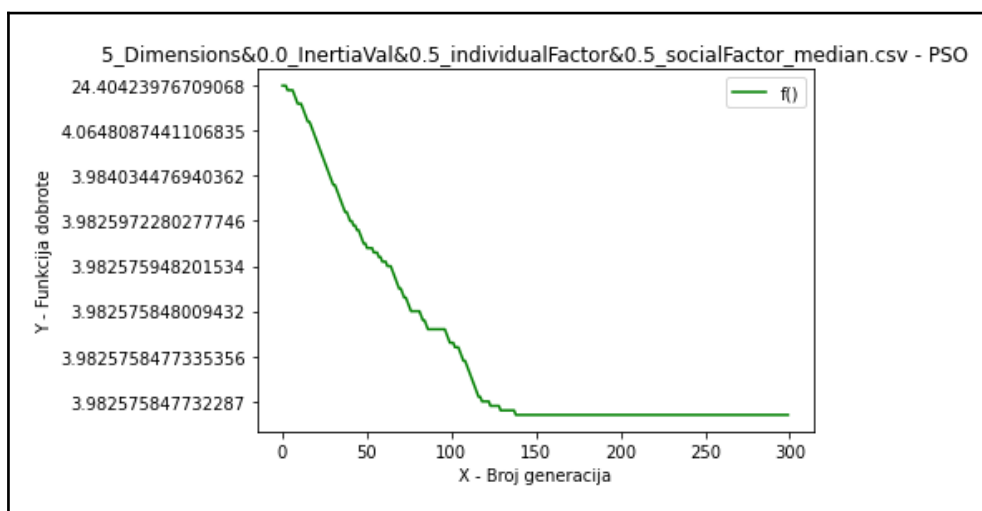




3.2.3. Utjecaj promjene apsolutne vrijednosti mutacije realnog gena

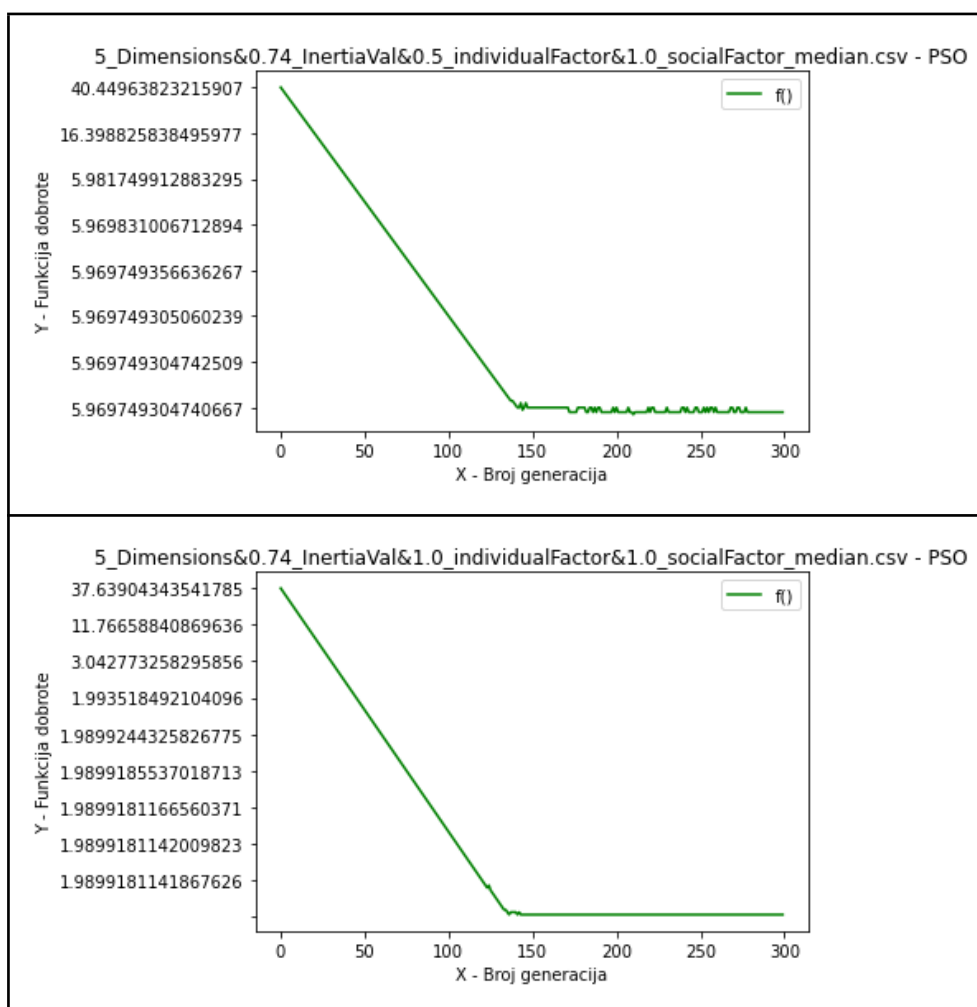
Broj elitnih jedinki	8		
Postotak mutacije	20%		
Najveća apsolutna vrijednost mutacije realnog gena	0.1	0.4	0.8
Vrijednosti pronađenih rješenja	4.974, 6.964, 7.959, 7.9592, 15.919	4.2233e-05, 16.229e-05, 16.704e-05, 16.945e-05, 24.217e-05	25.467e-05, 37.407e-05, 41.953e-05, 59.533e-05, 60.828e-05
Prosječna vrijednost	8.75504	12.663e-05	45.038e-05
Median vrijednost	7.959	16.704e-05	41.953e-05

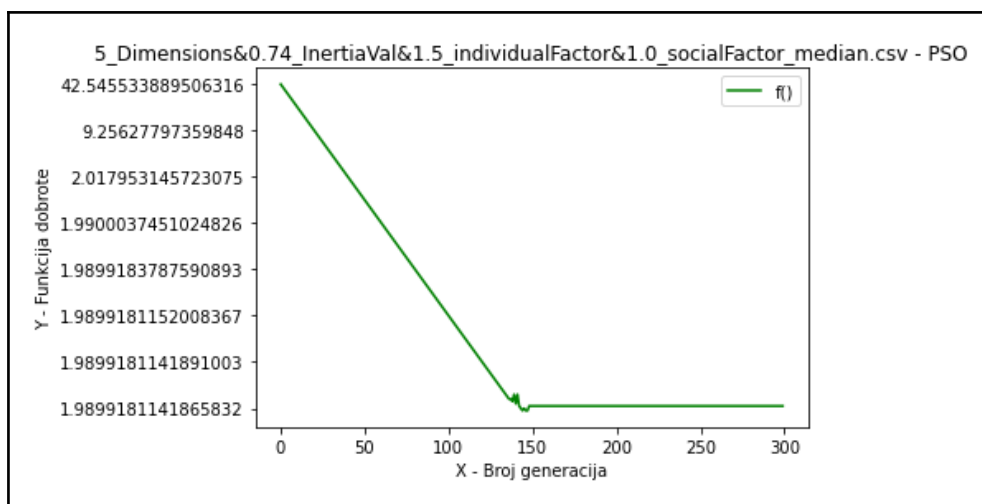




4.1.2. Utjecaj promjene individualnog faktora

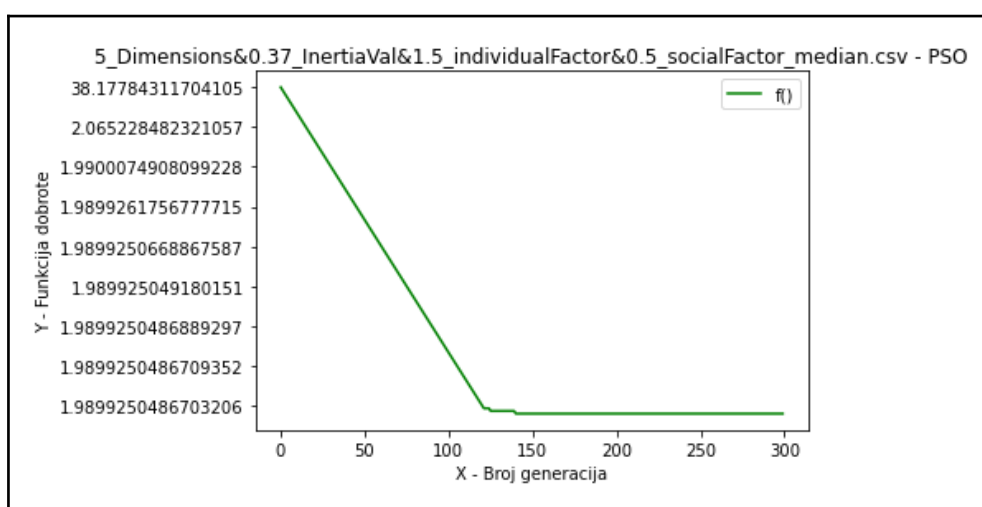
Mjera socijalnog faktora	1.0		
Mjera inercije	0.74		
Mjera individualnog faktora	0.5	1.0	1.5
Vrijednosti pronađenih rješenja	3.979, 3.979, 5.969, 9.949, 10.944	0.994, 1.989, 1.989, 2.984, 2.984	0.994, 0.994, 1.989, 2.984, 3.979
Prosječna vrijednost	6,9632	2,188	2,188
Median vrijednost	5,969	1,989	1,989

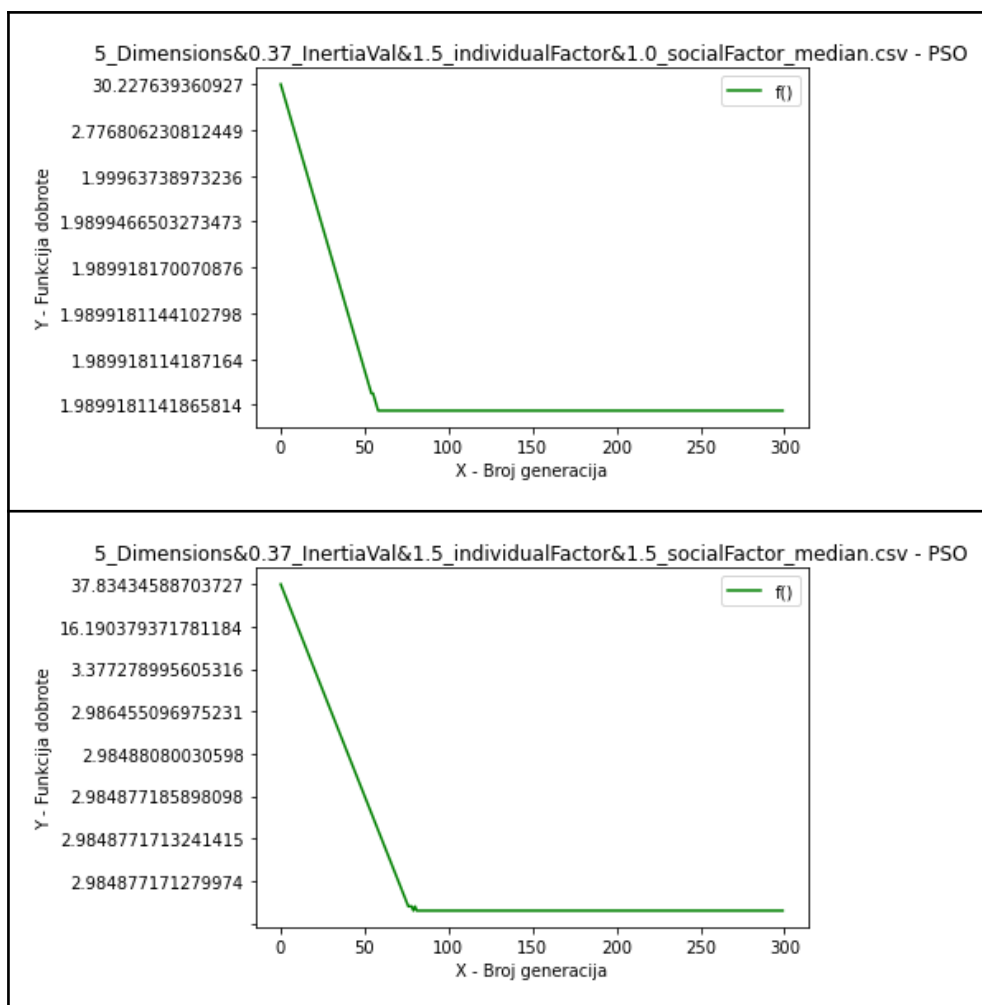




4.1.3. Utjecaj promjene mjere socijalnog faktora

Mjera individualnog faktora	1.5		
Mjera inercije	0.37		
Mjera socijalnog faktora	0.5	1.0	1.5
Vrijednosti pronađenih rješenja	0.994, 1.989, 1.989, 2.984, 2.984	0.994, 0.994, 1.989, 4.974, 5.969	0.0, 2.984, 2.984, 4.974, 5.969
Prosječna vrijednost	2,188	2,984	3,382
Median vrijednost	1,989	1,989	2,984





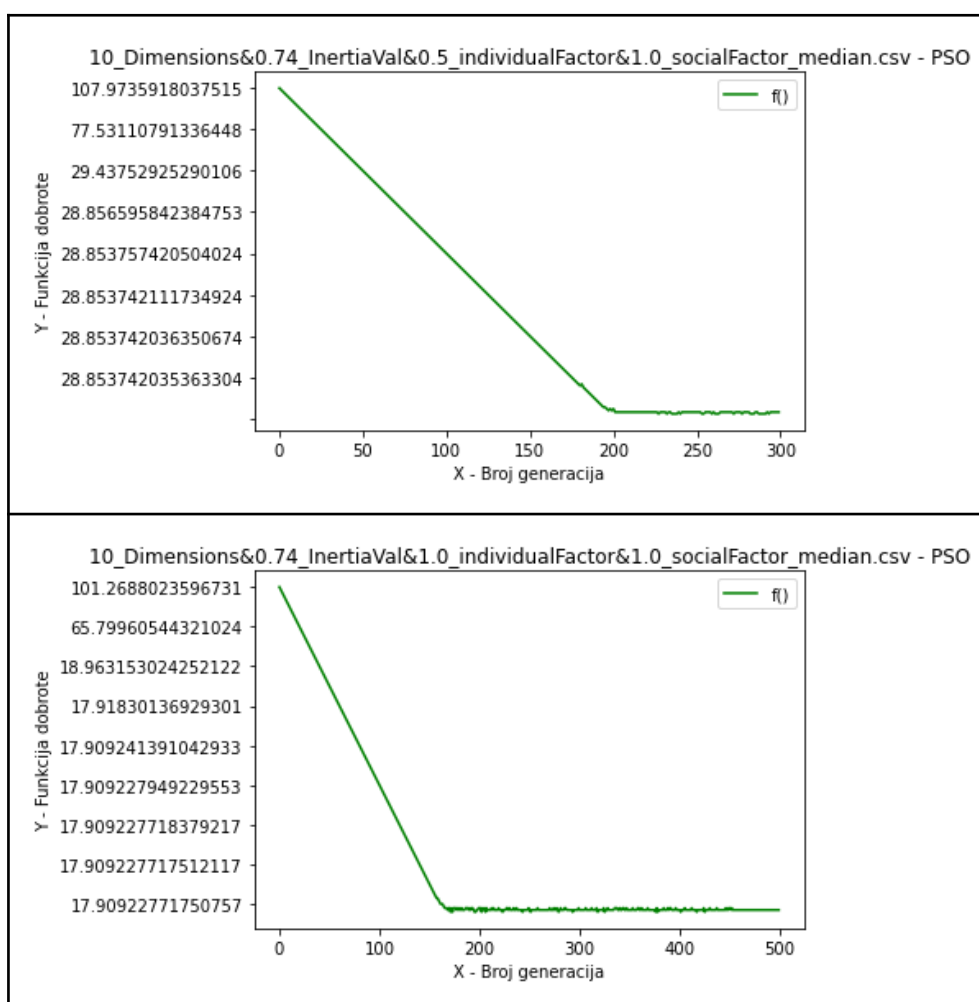
4.2. Rezultati za n = 10

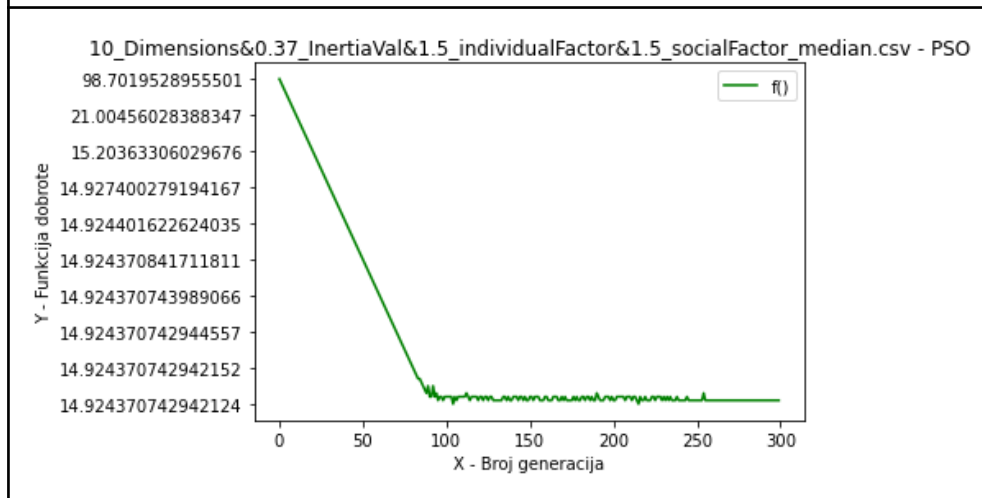
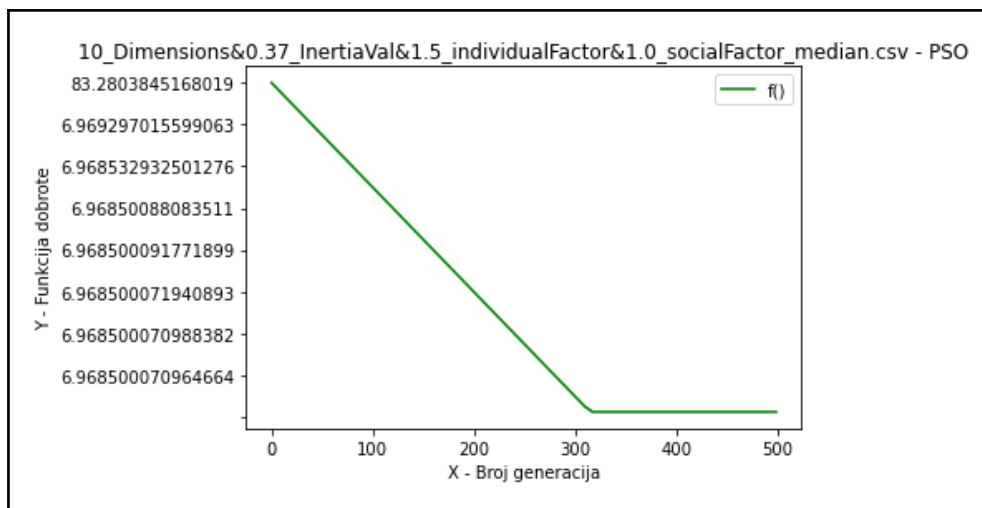
4.2.1. Utjecaj promjene mjere inercije

Mjera socijalnog faktora	0.5		
Mjera individualnog faktora	0.5		
Mjera inercije	0.0	0.37	0.74
Vrijednosti pronađenih rješenja	6.363, 11.103, 15.298, 18.727, 31.319	16.666, 17.961, 20.664, 21.986, 26.508	7.959, 9.949, 9.949, 15.919, 16.914
Prosječna vrijednost	16,562	20,757	12,138
Median vrijednost	15.298	20.664	9.949

4.2.2. Utjecaj promjene individualnog faktora

Mjera socijalnog faktora	1.0		
Mjera inercije	0.74		
Mjera individualnog faktora	0.5	1.0	1.5
Vrijednosti pronađenih rješenja	23.878, 25.868, 28.853, 28.853, 49.747	5.969, 13.929, 17.909, 28.853, 29.848	9.949, 10.944, 10.944, 13.929, 14.924
Prosječna vrijednost	31,4398	19,3016	12,138
Median vrijednost	28.853	17.909	10.944





5. ZAKLJUČAK

Zaključujemo da kod pronalaženja minimuma 5-dimenzionalne i 10-dimenzionalne Rastriginove funkcije, mnogo točnija rješenja dobijamo upotrebom genetskog algoritma. Naš globalni minimum funkcije je 0. Rješenje koje je najbliže tom globalnom minimumu je najtočnije. Uspješno smo došli do globalnog minimuma sa kombinacijom sljedećih parametara za PSO algoritam; Mjera individualnog faktora -1.5, Mjera inercije - 0.37, Mjera socijalnog faktora- 1.5 (Moguće je da je rješenje slučajno).

Povećanjem postotka mutacije kod genetskog algoritma za 5-dimenzionalnu funkciju, median se pogoršava (povećava mu se vrijednost), dok s druge strane za 10-dimenzionalnu funkciju, median se poboljšava odnosno smanjuje se vrijednost funkcije dobrote.

Povećanjem broja elitnih članova kod genetskog algoritma smanjuje se točnost rješenja odnosno divergiramo od globalnog minimuma. Najgora rješenja smo dobili ukoliko je broj elitnih članova 16 budući da se ti geni ne mjenjaju.

Promjenom najveće apsolutne vrijednosti mutacije realnog gena za obje funkcije (5 i 10 dim) dobijamo bolje rješenje te je najbolje rješenje kada je najveća apsolutna vrijednost mutacije realnog gena 0.8

Kod PSO algoritma, povećanjem mjere socijalnog faktora za obje funkcije (5 i 10 dim), dobijamo bolje vrijednosti. Također povećanjem mjere inercije za obje funkcije, algoritam daje znatno bolje rješenje. Iz grafova možemo vidjeti da i povećanjem mjere individualnog faktora dolazimo do boljih rješenja. Srednja vrijednost za svaku kombinaciju parametara je uzeta kao prosjek 5 rezultata pokretanja skripte s određenom kombinacijom parametara