

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU  
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I  
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

**Diplomski studij**

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

**Meko računarstvo  
Laboratorijska vježba 3**

**Andrej Bošnjak  
DRB**

**Osijek, 2023.**

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
<b>2. PROBLEM PRONALASKA MINIMUMA N-DIMENZIONALNE FUNKCIJE.....</b>	<b>2</b>
2.1. Genetski algoritam (GA) .....	2
2.2. Algoritam roja čestica (PSO).....	3
2.3. Opis problema i njegovo rješenje.....	5
<b>3. GENETSKI ALGORITAM REZULTATI.....</b>	<b>6</b>
3.1. N=5.....	6
3.2. N=10 .....	9
<b>4. ALGORITAM ROJA ČESTICA REZULTATI .....</b>	<b>12</b>
4.1. N=5 .....	12
4.2. N=10 .....	15
<b>5. Fitness funkcija.....</b>	<b>19</b>
<b>6. Zaključak .....</b>	<b>20</b>

## 1. UVOD

Na trećoj laboratorijskoj vježbi se koristeći postupak genetski algoritam i algoritam roja čestica rješava problem pronalaska minimuma  $n$ -dimenzionalne funkcije (Rastriginova funkcija). Cilj ove vježbe je usporediti dobivene rezultate kada se mijenjaju različiti parametri genetskog algoritma i algoritma roja čestica, te također usporediti rezultate dva korištena algoritma.

Budući da se za svaku konfiguraciju pronalazi pet rješenja, od njih se odabire generacija koja je najbliža srednjoj vrijednosti svih pet generacija (medijan), za koje se postiglo rješenje problema. Iteracije eksperimenta sa dobivenom medijan generacijom se potom prikazuju grafički i tablično.

## 2. PROBLEM PRONALASKA MINIMUMA N-DIMENZIONALNE FUNKCIJE

Problem minimuma n-dimenzionalne funkcije se rješava

### 2.1. Genetski algoritam (GA)

Genetski algoritam je 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 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:

- Binarni način kodiranja: 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 od 1 do N u različitom redoslijedu
- Stablasto kodiranje: gen je čvor stabla

Genetski algoritmi tijekom svog rada koriste genetske operator za stvaranje novih populacija. Koriste se slijedeći genetski operatori:

**Rekombinacija:** Kombiniranje gena dva roditelja u svrhu stvaranja novih i boljih potomaka. Najčešće rekombinacije koje se koriste su:

- Rekombinacija u jednoj točki
- Rekombinacija u dvije ili više točaka
- Uniformna rekombinacija

**Mutacija:** 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
- Potpuna mutacija

Uloga mutacije je i također i u obnavljanju izgubljenog genetskog materijala. Dogodi li se, npr. da sve jedinke populacije imaju isti gen na određenom mjestu u kromosomu, samo križanjem se taj gen nikad ne bi mogao promijeniti.

Genetski algoritam prvo treba odabrati određene dobre roditelje za stvaranje nove populacije. To se vrši metodom selekcije. Svrha selekcije je održavanje i prenošenje dobrih svojstava na slijedeću generaciju. Metodu selekcije dijelimo na:

- Generacijske: Generacijski genetski algoritam u jednoj iteraciji raspolaže s dvije populacije
- Eliminacijske: Za razliku od generacijske selekcije, eliminacijska selekcija ne bira dobre kromosome za slijedeću populaciju, već loše koje treba eliminirati i reprodukcijom ih zamijeniti novima.

U svrhu očuvanja dobrih rješenja (jedinke) nakon puno iteracija algoritma se uvodi i pojam elitizma. Elitizam je mehanizam koji čuva najbolju jedinku od promjena kroz neki od genetskih operatora.

Najvažniji dio genetskog algoritma jest određivanje funkcije dobrote (fitness funkcije) koja će nam govoriti koliko je neko rješenje dobro. Kroz generacije se uz svaki kromosom dodjeljuje i njegova pripadajuća fitness vrijednost, koja se algoritmom pokušava minimizirati ili maksimizirati, ovisno o zadanom problemu i definiciji same fitness funkcije.

## **2.2. Algoritam roja čestica (PSO)**

Algoritam roja čestica (engl. Particle Swarm Optimization, PSO) je biološki inspiriran metaheuristički algoritam za optimizaciju. PSO algoritmi su originalno osmišljeni od strane Kennedy-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 iz 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:

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

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

### 2.3. Opis problema i njegovo rješenje

Potrebno je realizirati 5- i 10-dimenzionalnu Rastriginovu funkciju i pronaći njen minimum koristeći GA i PSO optimizacijske algoritme.

Parametri GA algoritma su:

- 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 su:

- 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

Rastriginova funkcija je definirana sljedećom jednačbom:

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

Gdje je  $A=10$ ,  $\omega=2*\pi$  i  $n=5$  ili  $n=10$ .

Broj generacija za sve iteracije je 5000.

### 3. GENETSKI ALGORITAM REZULTATI

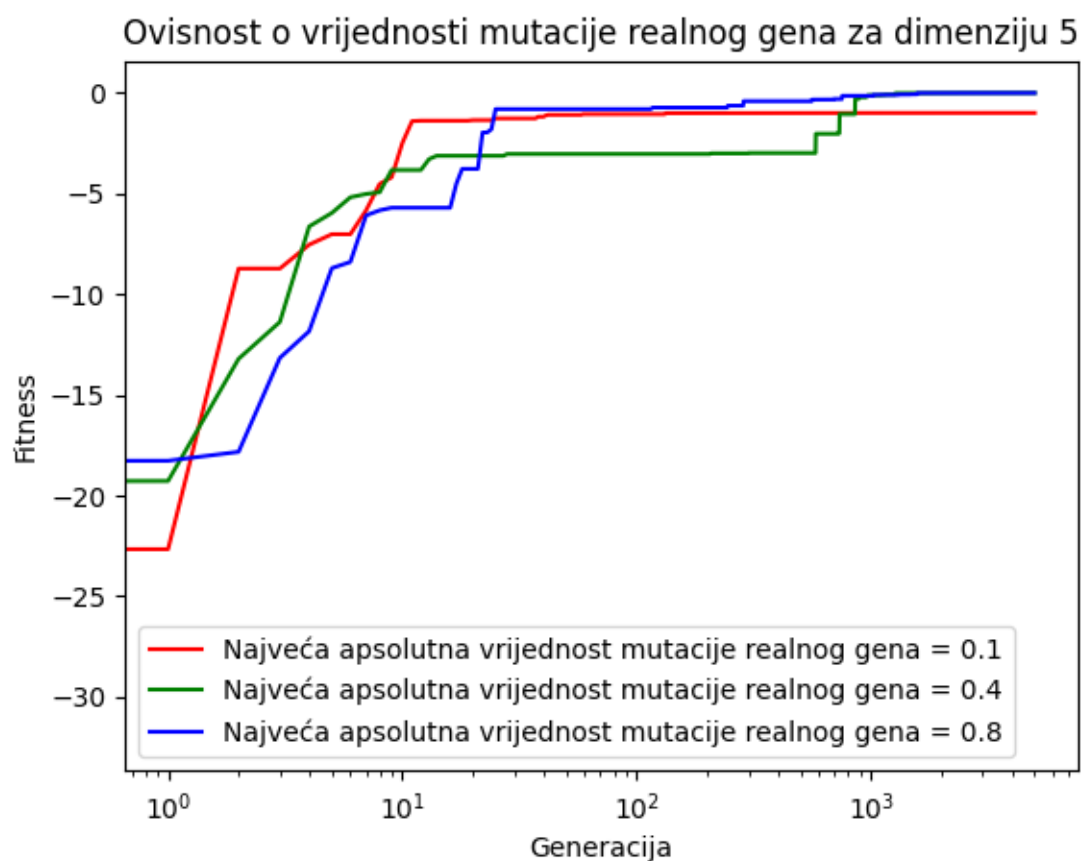
Svi rezultati su prikazani u idućim podnaslovima. X skala je logaritamska jer se rezultat na početku vrlo brzo mijenja te kasnije uspori.

#### 3.1. N=5

Populacija	100		
Broj elitnih članova	4		
Mutacija	5%		
Mutacija realnog gena	0.1	0.4	0.8
Dobivene vrijednosti fitness funkcije	1.0694e-05	0.0004743	0.0006186
	0.9949	0.0004763	0.0033047
	0.9949	0.0006478	0.0044877
	0.9950	0.0007641	0.0058979
	4.9747	0.0011220	0.0177628
Prosječna vrijednost fitness funkcije	0.994923526564545	0.0006478978312571826	0.004487730713890414
Prosječno rješenje	[-0. 0. -0. 1. 0.]	[ 0. 0. 0. 0. -0.]	[ 0. 0. -0. 0. 0.]

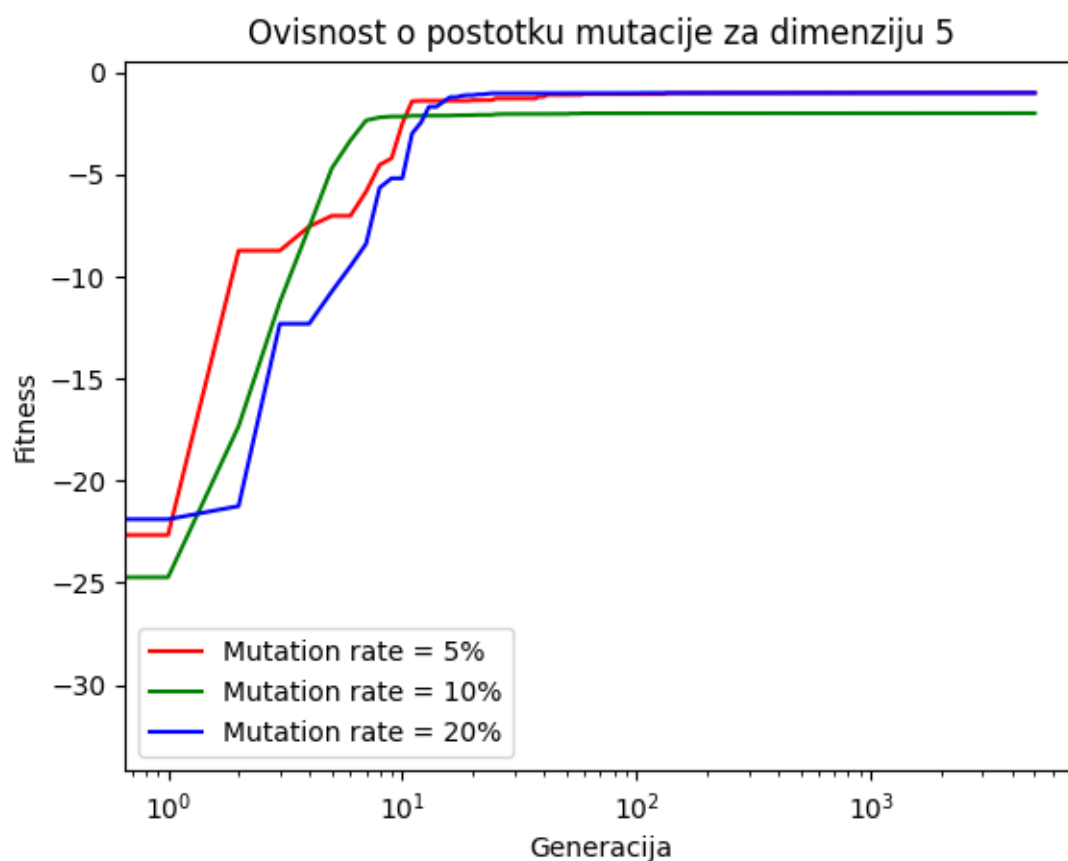
3.1 Ovisnost mutaciji realnog gena GA algoritma za N=5





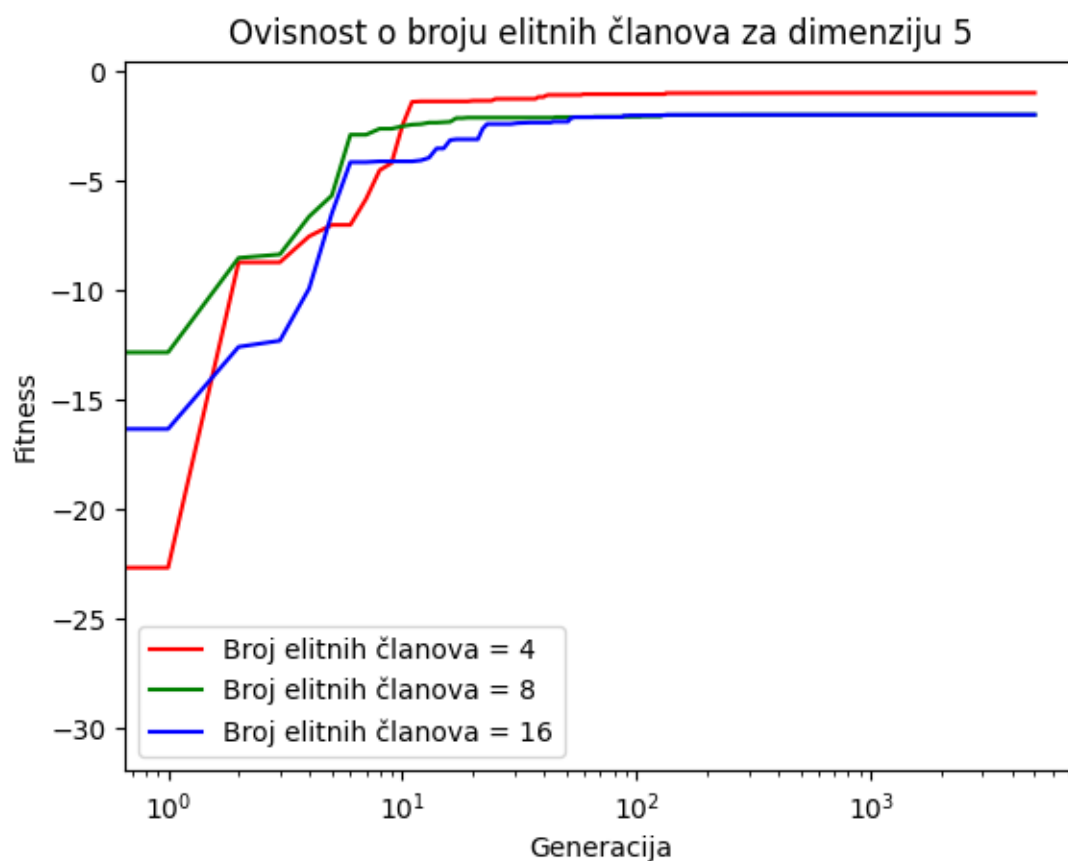
Populacija	100		
Broj elitnih članova	4		
Mutacija realnog gena	0.1		
Mutacija	5%	10%	15%
Dobivene vrijednosti fitness funkcije	1.0694e-05	4.10423e-06	0.994959
	0.9949	0.994944	0.994959
	0.9949	1.989915	0.994959
	0.9950	3.979812	2.984877
	4.9747	5.969795	2.984881
Prosječna vrijednost fitness funkcije	0.994923526564545	1.9899387413871015	0.9949598840446079
Prosječno rješenje	[-0. 0. -0. 1. 0.]	[ 0.99, 0, 0, -1, 0 ]	[-0, 0.99 -0, -0, -0. ]

3.2 Ovisnost mutaciji GA algoritma za N=5



Populacija	100		
Mutacija	5%		
Mutacija realnog gena	0.1		
Broj elitnih članova	4	8	16
Dobivene vrijednosti fitness funkcije	1.0694e-05	0.9949	1.77197e-05
	0.9949	0.9949	0.99493
	0.9949	1.9899	1.98997
	0.9950	1.9899	2.98488
	4.9747	2.9849	2.98512
Prosječna vrijednost fitness funkcije	0.994923526564545	1.9899398150399819	1.9899595091897897
Prosječno rješenje	[-0. 0. -0. 1. 0.]	[-0, 1, -0.99, -0, 0 ]	[-1, 0, 0, -0.99, 0 ]

3.3 Ovisnost o broju elitnih članova GA algoritma za N=5

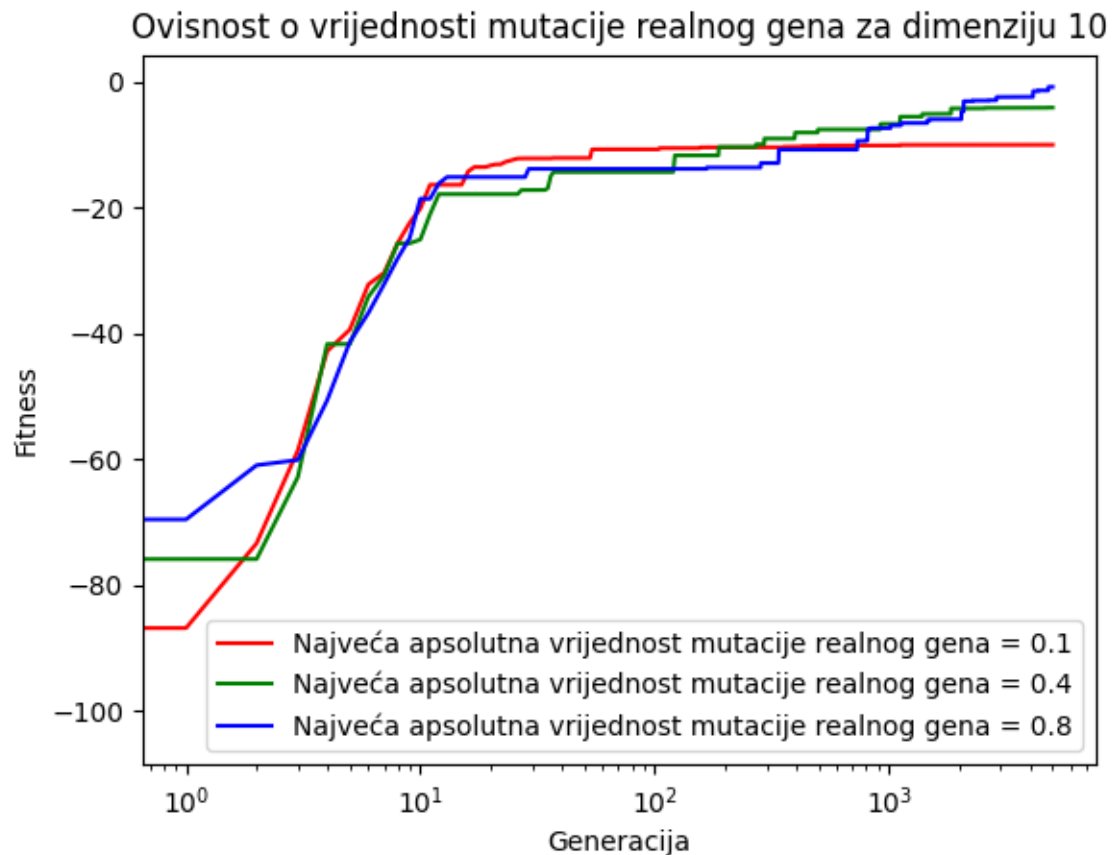


### 3.2. N=10

Populacija	100		
Broj elitnih članova	4		
Mutacija	5%		
Mutacija realnog gena	0.1	0.4	0.8
Dobivene vrijednosti fitness funkcije	6.96951543	0.2582326347	2.1685152429
	6.97141274	0.557797742	3.3843441346
	9.96493735	0.745366346	4.0399729134
	10.9535252	1.25273089	5.1885912354
	12.9379571	1.76387363	11.238682264
Prosječna vrijednost fitness funkcije	9.964948829843866	0.7453653743416346	4.039972913466633

Prosječno rješenje	[-0.99 1. 1. -0.99 - 2. 0. 0. -0.99 -1. 0. ]	[ 0.01 -0.01 0.02 0.01 0.02 -0.05 0.02 -0.01 -0.01 0. ]	[ 0.99 -0. 0.99 0.99 0. -0.99 -0. 0. - 0.01 0. ]
--------------------	--	---	--

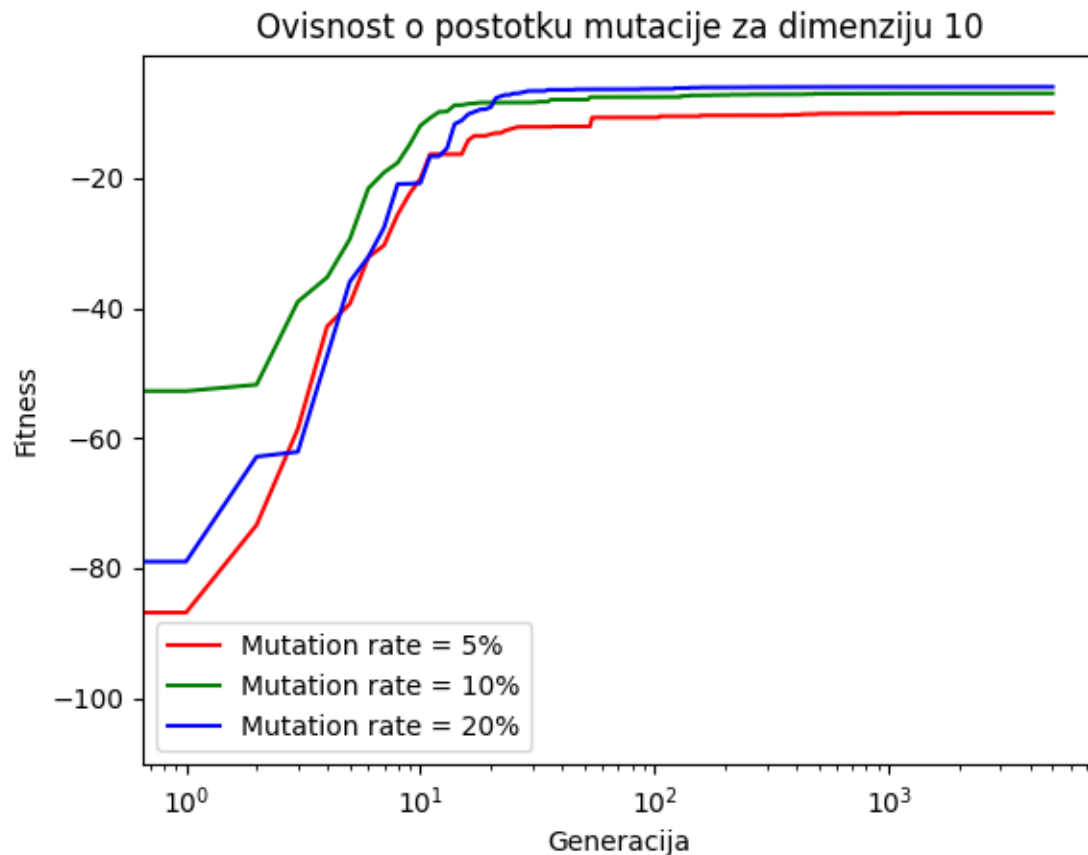
3.4 Ovisnost o mutaciji realnog gena GA algoritma za N=10



Populacija	100		
Broj elitnih članova	4		
Mutacija realnog gena	0.1		
Mutacija	5%	10%	15%
Dobivene vrijednosti fitness funkcije	6.96951543	2.9860478671188737	1.98998151131
	6.97141274	4.975220864786678	5.96977071931
	9.96493735	6.966013859428973	5.96977528534
	10.9535252	7.961183428893827	8.95464476073
	12.9379571	8.95591912912885	14.9244022055
Prosječna vrijednost fitness funkcije	9.964948829843866	6.966013859428973	5.969775285342557

Prosječno rješenje	[-0.99 1. 1. -0.99 - 2. 0. 0. -0.99 -1. 0. ]	[ 0.99 1. 0.99 -0.99 -0. -1. -1. 0.99 0. 0. ]	[ 0. -1. -0. 0.99 - 0.99 0.99 -0.99 -1. - 0. -0. ]
--------------------	--	---	--

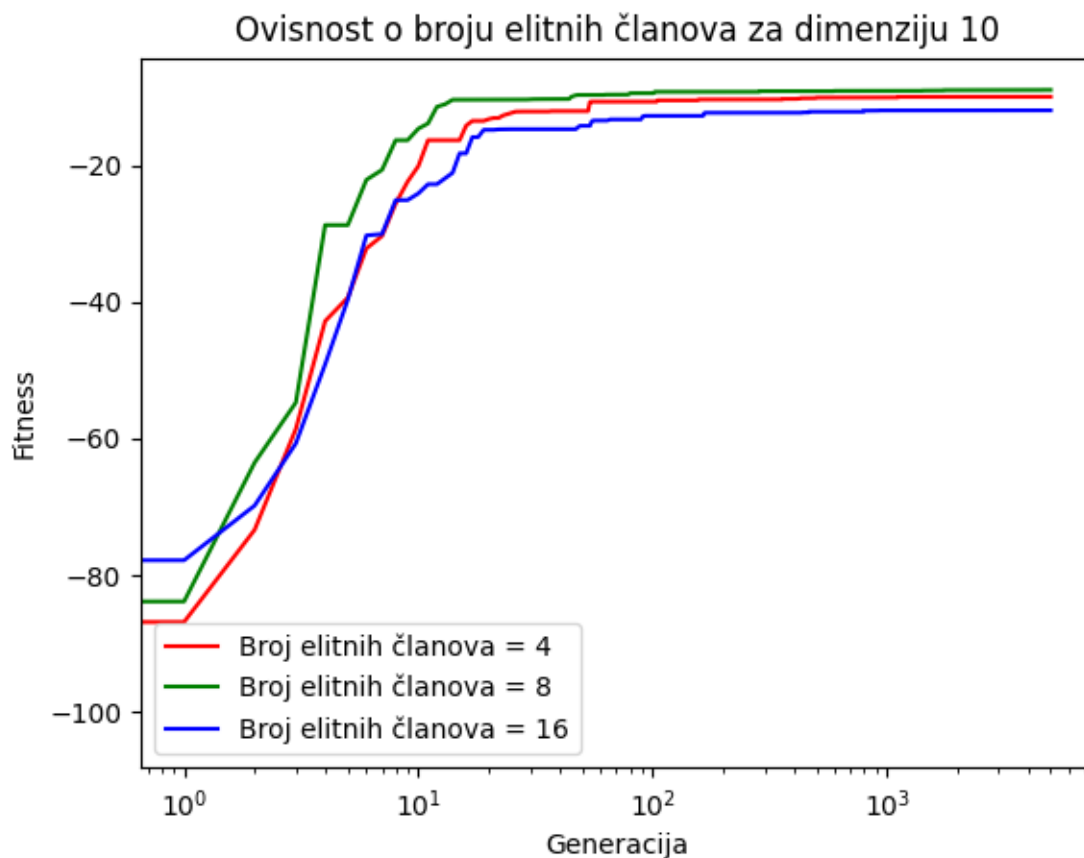
3.5 Ovisnost o mutaciji GA algoritma za N=10



Populacija	100		
Mutacija	5%		
Mutacija realnog gena	0.1		
Broj elitnih članova	4	8	16
Dobivene vrijednosti fitness funkcije	6.96951543	4.988973768967037	6.980283807760916
	6.97141274	7.0080320120226265	8.9896779156857
	9.96493735	8.956546343063255	11.952546849993157
	10.9535252	8.963432576863857	15.92457250645727
	12.9379571	16.92114744748224	30.846650364506925
Prosječna vrijednost fitness funkcije	9.964948829843866	8.956546343063255	11.952546849993157

Prosječno rješenje	[-0.99 1. 1. -0.99 - 2. 0. 0. -0.99 -1. 0. ]	[ 0. -0. 1.99 0. 0.99 0. -1.99 -0. - 0. -0. ]	[ 1.99 -1. -1.99 0. - 0. -0. 0. -1. -1. 0.99]
--------------------	--	---	---

3.6 Ovisnost o broju elitnih članova GA algoritma za N=10



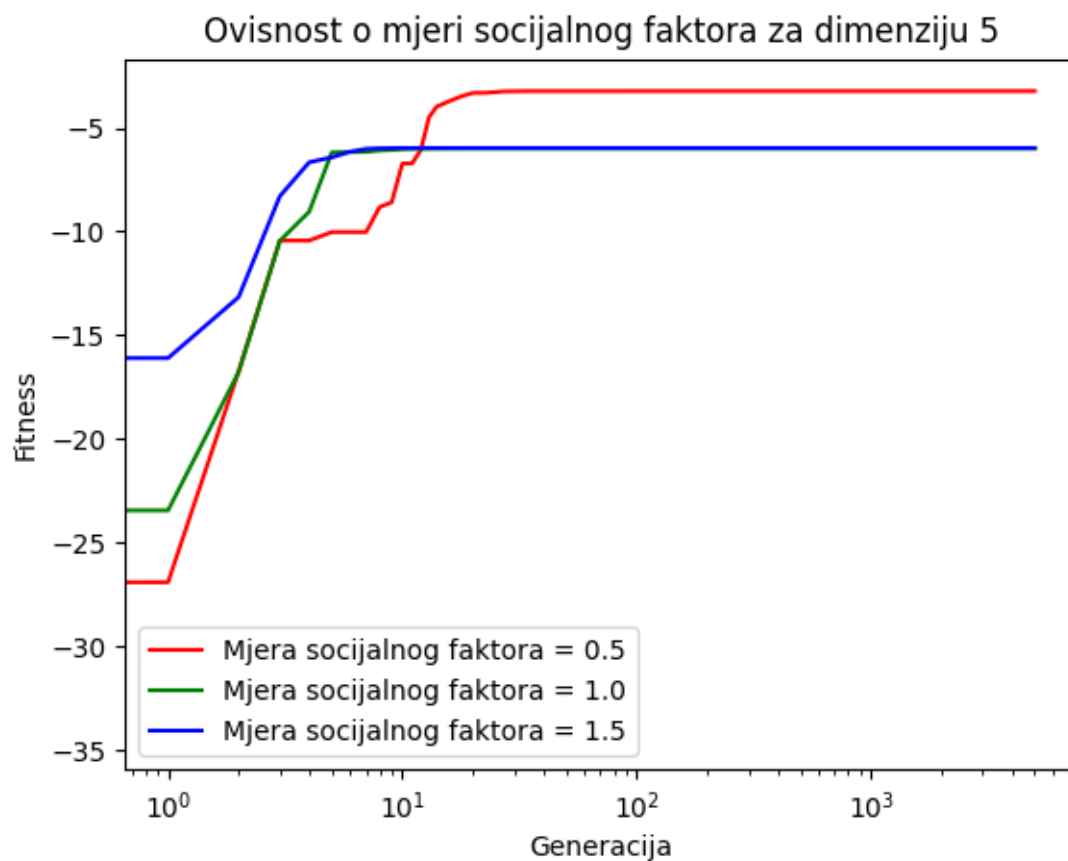
## 4. ALGORITAM ROJA ČESTICA REZULTATI

### 4.1. N=5

Populacija	100		
Mjera inercije	0.0		
Mjera individualnog faktora	0.5		
Mjera socijalnog faktora	0.5	1.0	1.5
Dobivene vrijednosti fitness funkcije	0.9950929933160744 3.0152694497040944	0.9950783813354906 5.761864475858621	2.9849049695255783 3.9798367597525335

	3.2247355570119662 4.979226506796367 7.219421468934387	5.993509640367595 6.023761764171825 6.969529570182006	5.969750134627516 7.965237002458245 9.94956029858033
Prosječna vrijednost fitness funkcije	3.2247355570119662	5.993509640367595	5.969750134627516
Prosječno rješenje	[ 0.97 -0.01 -1.01 - 0.01 -0.98]	[ 1.99 1. -1. 0.01 0. ]	[ 0. -0.99 0. 1.99 - 0.99]

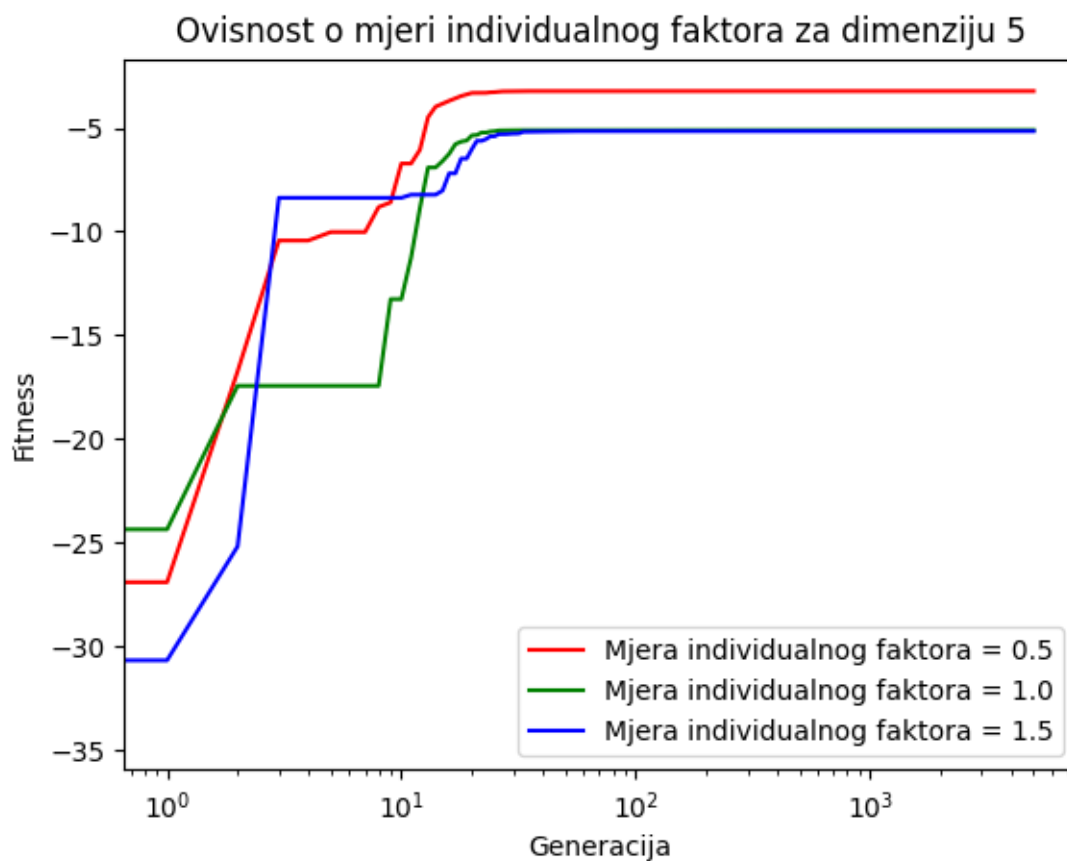
4.1 Ovisnost o mjeri socijalnog faktora PSO algoritma za N=5



Populacija	100		
Mjera inercije	0.0		
Mjera socijalnog faktora	0.5		
Mjera individualnog faktora	0.5	1.0	1.5
Dobivene vrijednosti fitness funkcije	0.9950929933160744 3.0152694497040944	0.9975132873388954 2.103779737514147	2.985141807269631 5.119168222826943

	3.2247355570119662 4.979226506796367 7.219421468934387	5.103446627130278 6.117780322912958 7.342548595302418	5.159358593664056 6.9647084145963625 7.023991451995219
Prosječna vrijednost fitness funkcije	3.2247355570119662	5.103446627130278	5.159358593664056
Prosječno rješenje	[ 0.97 -0.01 -1.01 - 0.01 -0.98]	[ 0.01 -0. 0.98 -1.97 0. ]	[ 0.02 0.01 1.97 - 0.01 0.99]

4.2 Ovisnost o mjeri individualnog faktora PSO algoritma za N=5

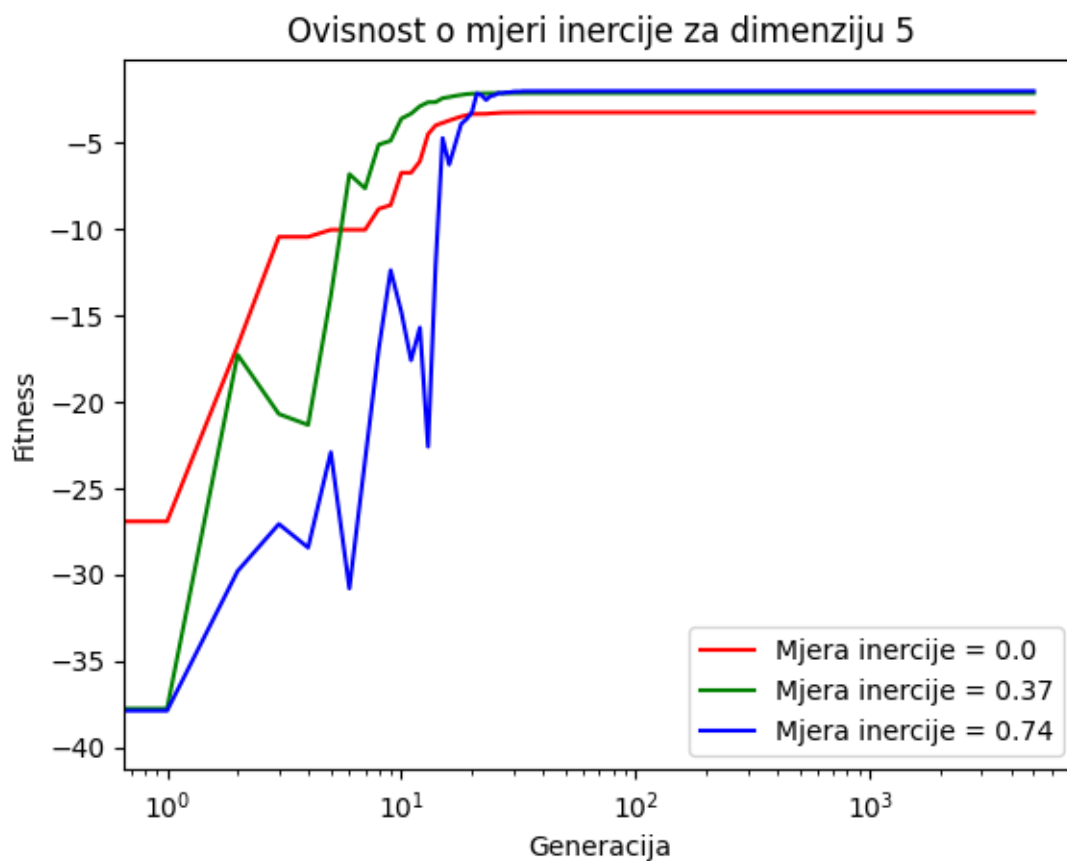


Populacija	100		
Mjera individualnog faktora	0.0		
Mjera socijalnog faktora	0.5		
Mjera inercije	0.0	0.37	0.74
Dobivene vrijednosti fitness funkcije	0.9950929933160744 3.0152694497040944	1.3597501101068108 1.9946065766909218	0.9949590570932898 0.9949590570932898



	3.2247355570119662 4.979226506796367 7.219421468934387	2.1068764879030084 4.974801702583916 4.981842938236843	1.9899181141865796 1.9899181141865796 6.9647083618339565
Prosječna vrijednost fitness funkcije	3.2247355570119662	2.1068764879030084	1.9899181141865796
Prosječno rješenje	[ 0.97 -0.01 -1.01 - 0.01 -0.98]	[ 0. -0.98 0. -0. - 1.01]	[-0.99 0. -0. 0. - 0.99]

4.3 Ovisnost o mjeri inercije PSO algoritma za N=5

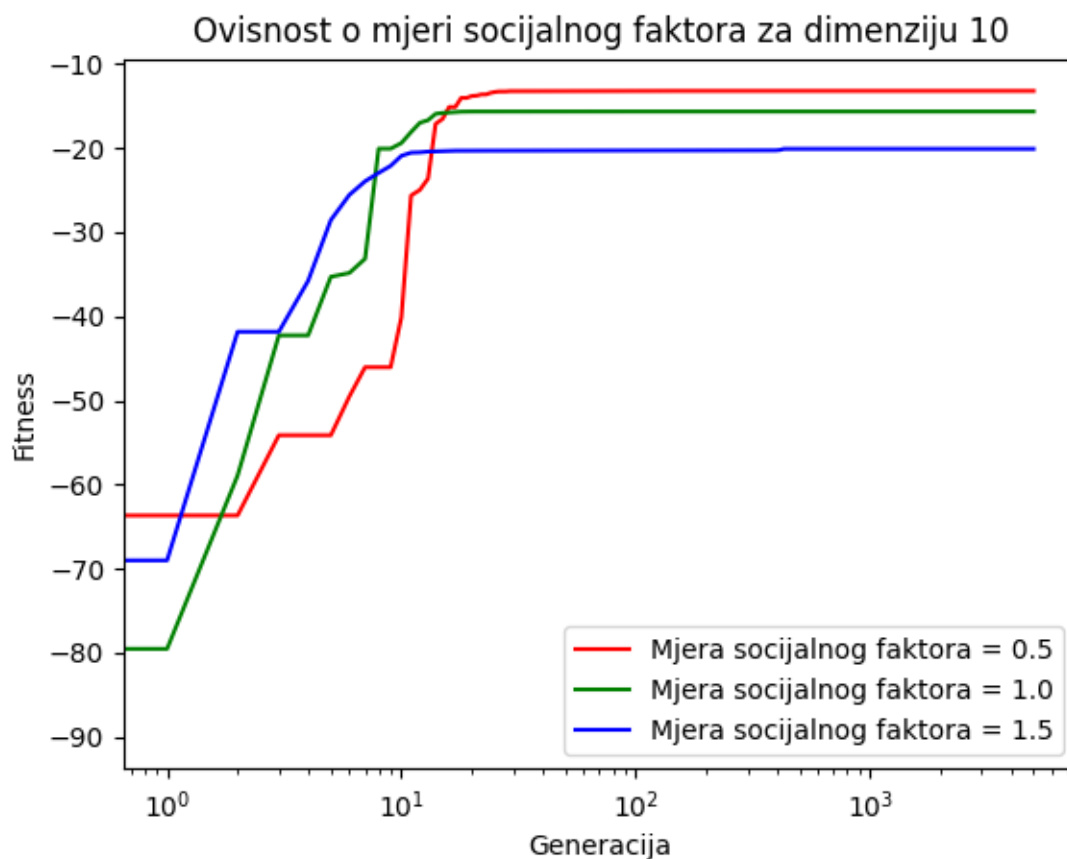


## 4.2. N=10

Populacija	100		
Mjera inercije	0.0		
Mjera individualnog faktora	0.5		
Mjera socijalnog faktora	0.5	1.0	1.5

Dobivene vrijednosti fitness funkcije	11.983292704601855 12.857487023264802 13.238910497015029 24.379602939796357 33.94871993596795	8.629509275938888 13.599728353044842 15.686187242034318 20.567205708285655 22.59011838019498	11.573895333104643 19.609122915091138 20.129453117737768 27.97117065201993 30.99670807568164
Prosječna vrijednost fitness funkcije	13.238910497015029	15.686187242034318	20.129453117737768
Prosječno rješenje	[-0.92 -0.04 -2.02 0.9 -0.94 -0.03 0.08 0.04 0.04 0.02]	[-0.05 0.98 0.07 - 0.08 -1.97 0.06 1.02 -0.05 -1.06 0.87]	[-1.99 0. -2.99 -0.99 1. -1.01 0.03 1.97 0. -0. ]

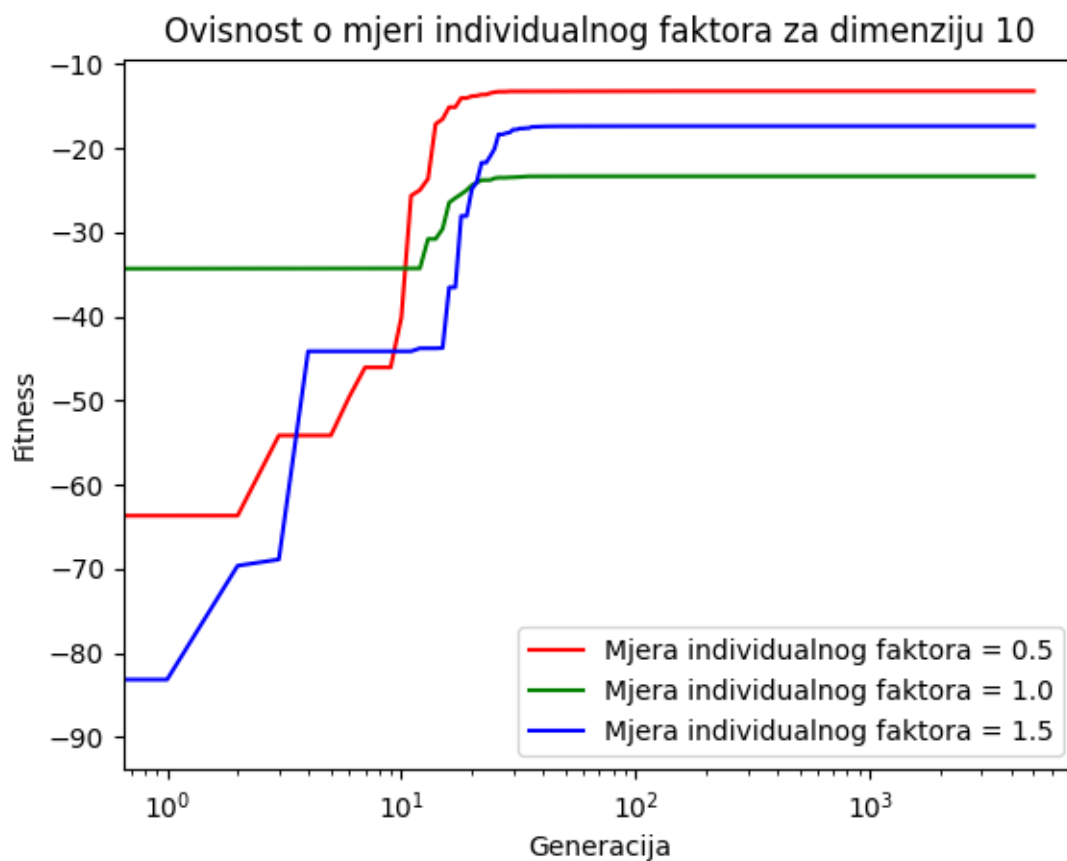
4.4 Ovisnost o mjeri socijalnog faktora PSO algoritma za N=10



Populacija	100
Mjera inercije	0.0
Mjera socijalnog faktora	0.5

Mjera individualnog faktora	0.5	1.0	1.5
Dobivene vrijednosti fitness funkcije	11.983292704601855 12.857487023264802 13.238910497015029 24.379602939796357 33.94871993596795	13.46146438720421 21.36396065684172 23.359571500712264 24.018790576999688 28.150544451831877	6.926285848116795 16.87831631813362 17.403153882678886 22.697403512355198 27.632376935911427
Prosječna vrijednost fitness funkcije	13.238910497015029	23.359571500712264	17.403153882678886
Prosječno rješenje	[-0.92 -0.04 -2.02 0.9 -0.94 -0.03 0.08 0.04 0.04 0.02]	[ 1.94 0.96 1.96 0.94 1.95 -0.03 -0.86 -0.01 0.09 0.93]	[ 0.02 0.92 -0.98 - 0.98 -1.01 1.02 0.09 1. 2.01 -0.86]

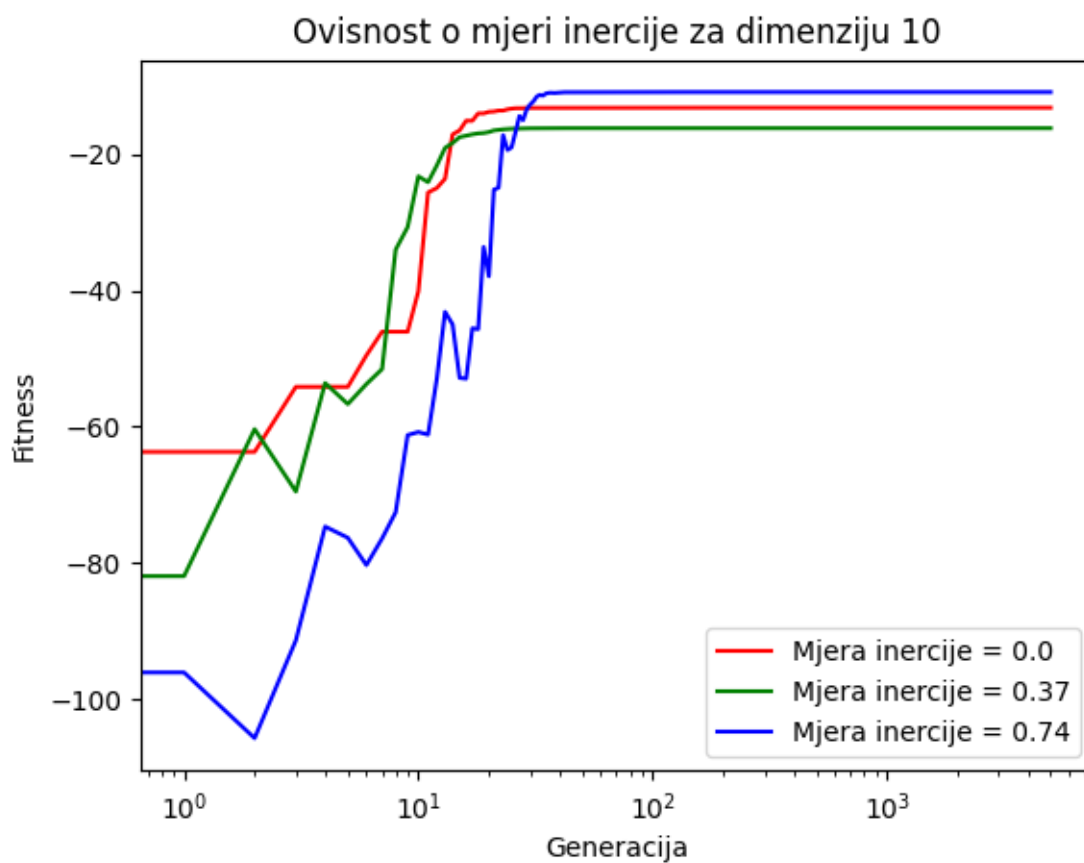
4.5 Ovisnost o mjeri individualnog faktora PSO algoritma za N=10



Populacija	100
Mjera individualnog faktora	0.5

Mjera socijalnog faktora	0.5		
Mjera inercije	0.0	0.37	0.74
Dobivene vrijednosti fitness funkcije	11.983292704601855 12.857487023264802 13.238910497015029 24.379602939796357 33.94871993596795	9.597372835426407 10.249844762651586 16.23749150637546 19.50362359902462 25.511908619677275	8.954626476020536 9.949585533113833 10.94454459020713 14.924355584046758 23.878971984429143
Prosječna vrijednost fitness funkcije	13.238910497015029	16.23749150637546	10.94454459020713
Prosječno rješenje	[-0.92 -0.04 -2.02 0.9 -0.94 -0.03 0.08 0.04 0.04 0.02]	[ 0.01 -1.97 -1.9 - 0.01 -0.98 0.94 -0.07 -0.97 1.02 1. ]	[-0.99 1.99 0.99 - 0.99 0. 0.99 0.99 0. 0.99 -0.99]

4.6 Ovisnost o mjeri inercije PSO algoritma za N=10



## 5. Fitness funkcija

Kod s kojim je definirana fitness funkcija dan je u nastavku:

```
66  #Rastrigin function
67  def evaluateInd(individual):
68      fit_val = 0
69      #Implement Rastrigin function
70      A=10
71      fit_val = A*NO_DIMS
72      for i in range(0,NO_DIMS):
73          fit_val += individual[i]**2 - A*math.cos(2*math.pi*individual[i])
74
75      return fit_val,
```

Implementirana je Rastriginova funkcija opisana jednadžbom u 2. poglavlju. Ista se koristi i za PSO i za GA algoritme.

## 6. Zaključak

Nakon izvršenih kombinacija parametara za navedeni algoritam računanja problema određivanje minimuma  $N$  dimenzionalne funkcije možemo zaključiti da promjenom bilo kojeg parametra direktno utječemo na rezultate rješenja. Ta ovisnost se najbolje može vidjeti na prikazanim grafovima.

Uočavamo bolje rezultate kod GA algoritma nego kod PSO algoritma. Za  $N=5$  dimenzija, GA algoritam često pronalazi i globalni minimum, dok za  $N=10$  je prosječna vrijednost fitness funkcije za sve kombinacije parametara oko 10. Najveće smanjenje fitness funkcije dobivamo uz povećanje vrijednosti mutacije.

PSO algoritam je dao općenito lošije rezultate, što uočavamo za  $N=5$  je prosječna vrijednost fitness funkcije za sve kombinacije oko 2, čime možemo zaključiti da vrlo rijetko pronalazi globalni minimum. Za  $N=10$  dobivamo i još veće vrijednosti fitness funkcije nego što je davao GA algoritam. Najveću promjenu uočavamo promjenom mjere inercije.

Također se može uočiti da, kako povećavamo mjeru inercije, u nekoliko generacija se fitness vrijednost i povećava, što nam govori da lokalni minimumi nekih čestica nisu globalna rješenja, a kako povećavamo mjeru inercije, tako će ti minimumi sve „jače“ povlačiti te čestice. Zato vidimo nagla povećanja fitness funkcije.