

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

**Rješavanje problema N-dama koristeći genetski algoritam**

Meko računarstvo

Laboratorijska vježba 1

Ivan Gudelj

Diplomski studij računarstva, DRB

Osijek, 2022.

<b>UVOD</b>	<b>3</b>
<b>OPIS PROBLEMA I RJEŠENJA</b>	<b>4</b>
Opis problema i genetski algoritam	4
Rješavanje problema NxN dama	6
<b>ANALIZA REZULTATA</b>	<b>7</b>
Ovisnost o postotku mutacije (bez permutacija)	7
Ovisnost o broju elitnih članova (bez permutacija)	10
Ovisnost o broju populacija (bez permutacija)	13
Ovisnost o postotku mutacije (s permutacijama)	16
Ovisnost o broju elitnih članova (s permutacijama)	19
Ovisnost o broju populacija (s permutacijama)	22
<b>ZAKLJUČAK</b>	<b>25</b>
Popis tablica	26

# 1. UVOD

Cilj prve laboratorijske vježbe je bio proučiti osnove i način rada genetskog algoritma kroz problem postavljanja  $N$  dama na šahovsku ploču dimenzija  $N \times N$  tako da se međusobno ne napadaju.

Program za rješavanje ovog problema napisan je u Python programskom jeziku. Zbog stohastičke naravi genetskog algoritma svaki eksperiment bilo je potrebno ponoviti najmanje 5 puta i zabilježiti srednji (medijan) rezultat. Pokusi će biti provedeni koristeći standardno cjelobrojno kodiranje, te permutacijsko kodiranje gena, a njihove rezultate ćemo usporediti.

## 2. OPIS PROBLEMA I RJEŠENJA

### 2.1. Opis problema i 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 podijeljena je na 6 koraka:

1. Generiraj početnu populaciju mogućih rješenja (kromosomi)
2. Odredi sposobnost svakog kromosoma u populaciji
3. Generiraj nove kromosome koristeći genetičke operatore
4. Odbaci nepoželjne jedinke (kromosome) populacije
5. Uključi nove kromosome u populaciju da se stvori nova populacija
6. 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  $1 \dots N$  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 sposobnosti može biti bilo koja nelinearna, prekidna, nederivabilna pozitivna funkcija jer je bitno samo odrediti sposobnost za svaki kromosom.

Problem N-dama potrebno je riješiti tako da se postavi N broj dama na šahovsku ploču dimenzija  $N \times N$  na način da se one međusobno ne napadaju. Rješavanje ovoga problema omogućuje genetski algoritam koji za određeni broj dama i određene parametre pronalazi rješenje postavljanja tih dama na šahovsku ploču na ranije opisani način. Koristi se standardno cjelobrojno kodiranje, te permutacijsko cjelobrojno kodiranje gena.

## 2.2. Rješavanje problema $N \times N$ dama

Potrebno je proučiti rješenje problema postavljanja N dama na šahovsku ploču dimenzija  $N \times N$  tako da se međusobno ne napadaju. Problem N-dama bilo je potrebno riješiti korištenjem cjelobrojnog te permutacijskog cjelobrojnog kodiranja. Za oba načina bilo je potrebno analizirati ponašanje algoritma za veličinu ploče  $N=12, 24$  i  $48$ . Također, bilo je potrebno ispitati i utjecaj promjene parametara algoritma (veličina populacije, faktor mutacije i broj elitnih članova) na brzinu konvergiranja prema rješenju. Svaki eksperiment bilo je potrebno ponoviti najmanje 5 puta i zabilježiti srednji (medijan) rezultat. Bilo je potrebno i pamtit koliko je puta bilo potrebno pokrenuti algoritam s nekom konfiguracijom kako bi se došlo do 5 točnih rješenja.

**\*\*Grafovi s funkcijama dobrote (*fitness* funkcije) prikazuju funkciju medijana 5 rješenja, za određene parametre algoritma.**

### 3. ANALIZA REZULTATA

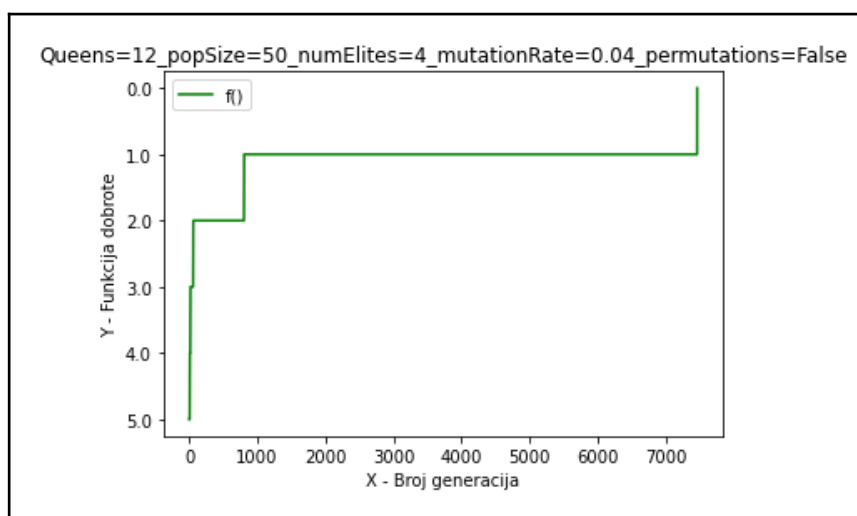
Parametre genetskog algoritma bilo je potrebno mijenjati na slijedeće vrijednosti:

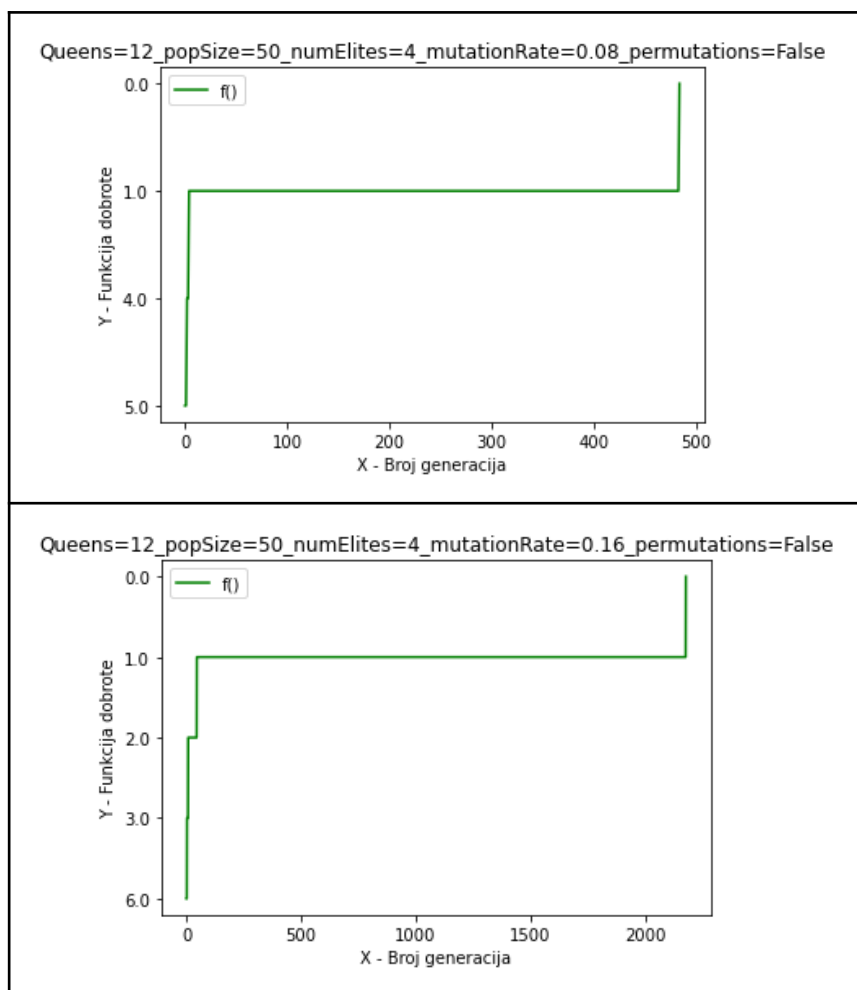
- populacija: 50, 100, 200
- mutacija: 4%, 8%, 16%
- broj elitnih članova: 4, 8, 16

#### 3.1. Ovisnost o postotku mutacije (bez permutacija)

Veličina ploče N	12		
Populacija	50		
Broj generacija	50000		
Broj elitnih članova	4		
Mutacija	4%	8%	16%
Broj generacija rješenja	1230,5846,7464,8963,17604	199,244,485,984,5103	427,603,2178,9087,21351
Prosječan broj generacija rješenja	8221.4	1403	6729.2
Broj pokušaja	12	9	5

Tablica 3.1.1. Utjecaj postotka mutacije za N=12



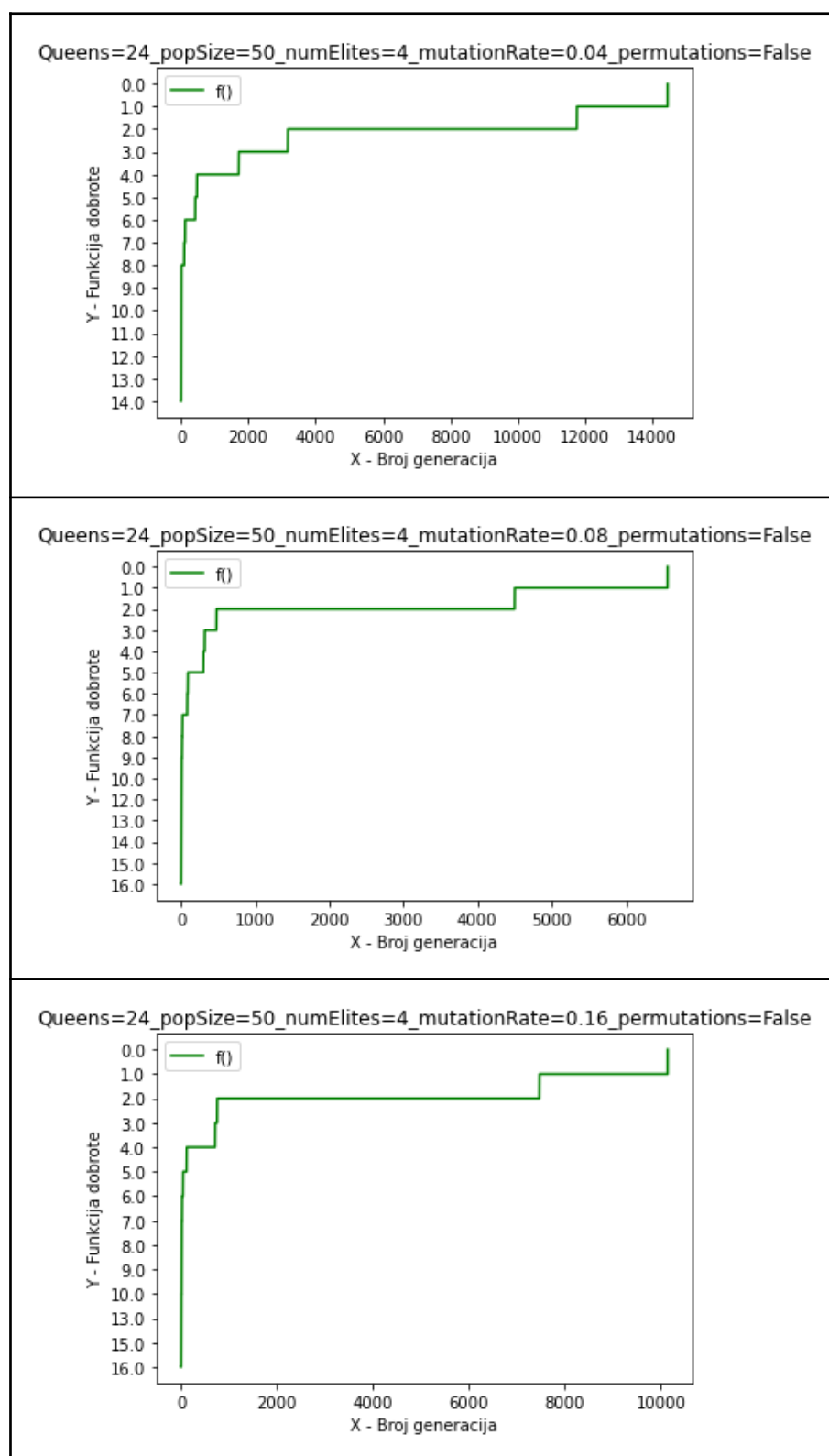


Tablica 3.1.2. Fitness funkcija u ovisnosti o postotku mutacije za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	50000		
Broj elitnih članova	4		
Mutacija	4%	8%	16%
Broj generacija rješenja	6953,11742,14457,16606,24763	3161,4299,6564,12260,18547	1221,7285,10143,19327,48540
Prosječan broj generacija rješenja	14904.2	8966.2	17303.2
Broj pokušaja	16	13	7

Tablica 3.1.3. Utjecaj postotka mutacije za N=24



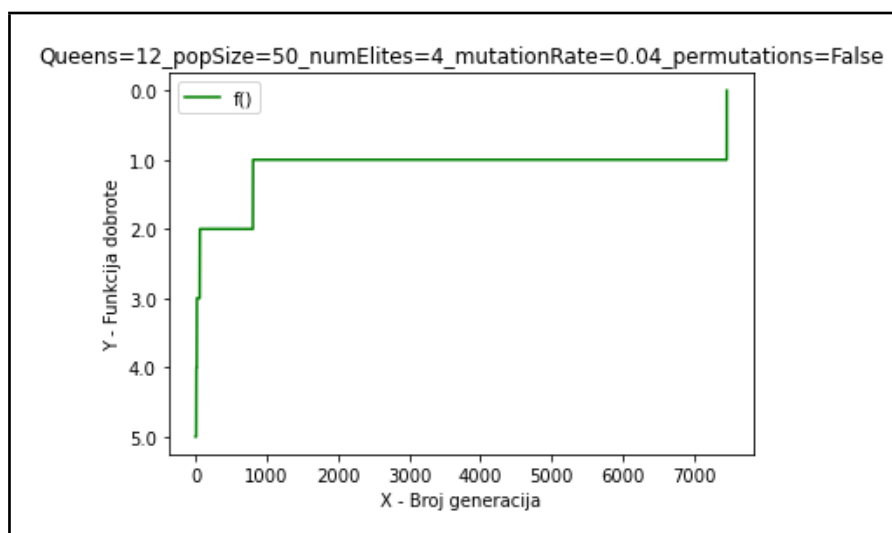


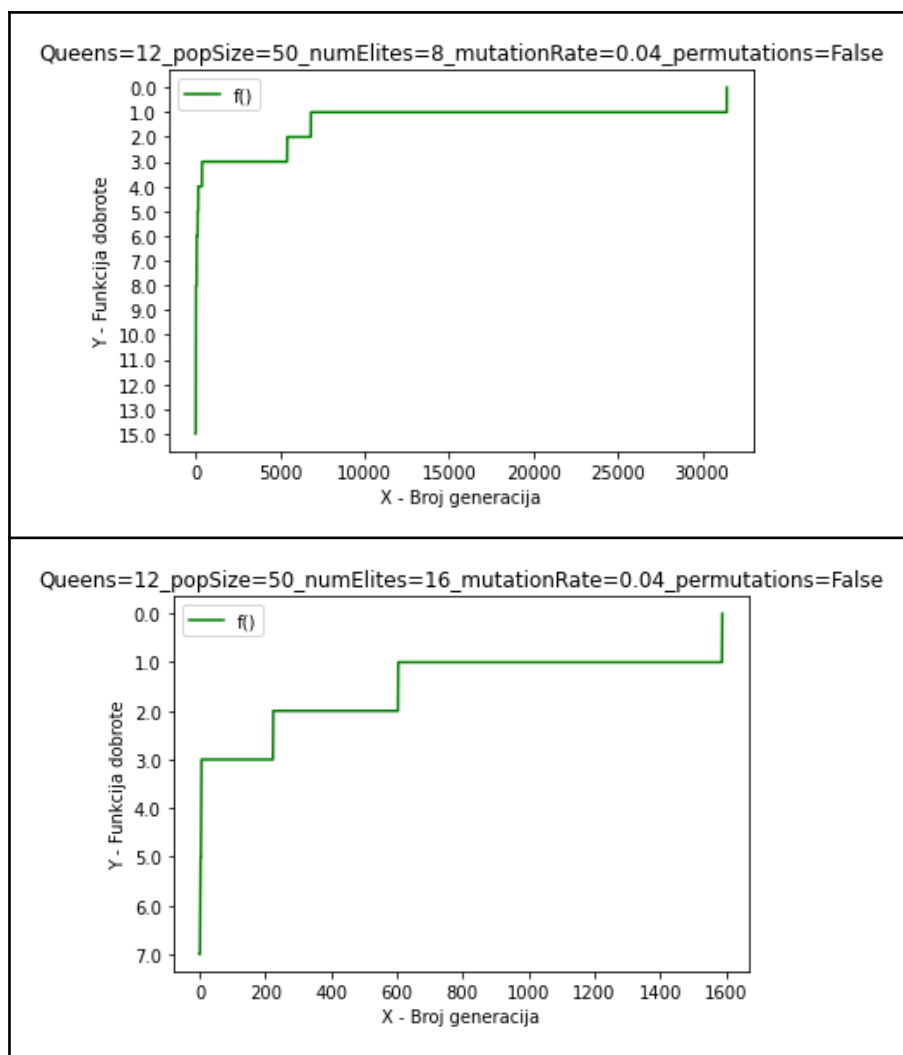
Tablica 3.1.4. Fitness funkcija u ovisnosti o postotku mutacije za N=24

### 3.2. Ovisnost o broju elitnih članova (bez permutacija)

Veličina ploče N	12		
Populacija	50		
Broj generacija	50000		
Mutacija	4%		
Broj elitnih članova	4	8	16
Broj generacija rješenja	1230,5846,7464,8963,17604	428,11007,15100,17807,38649	907,1083,1590,2920,4077
Prosječan broj generacija rješenja	8221.4	16598.2	2115.4
Broj pokušaja	12	7	8

Tablica 3.2.1. Utjecaj postotka broja elitnih članova za N=12

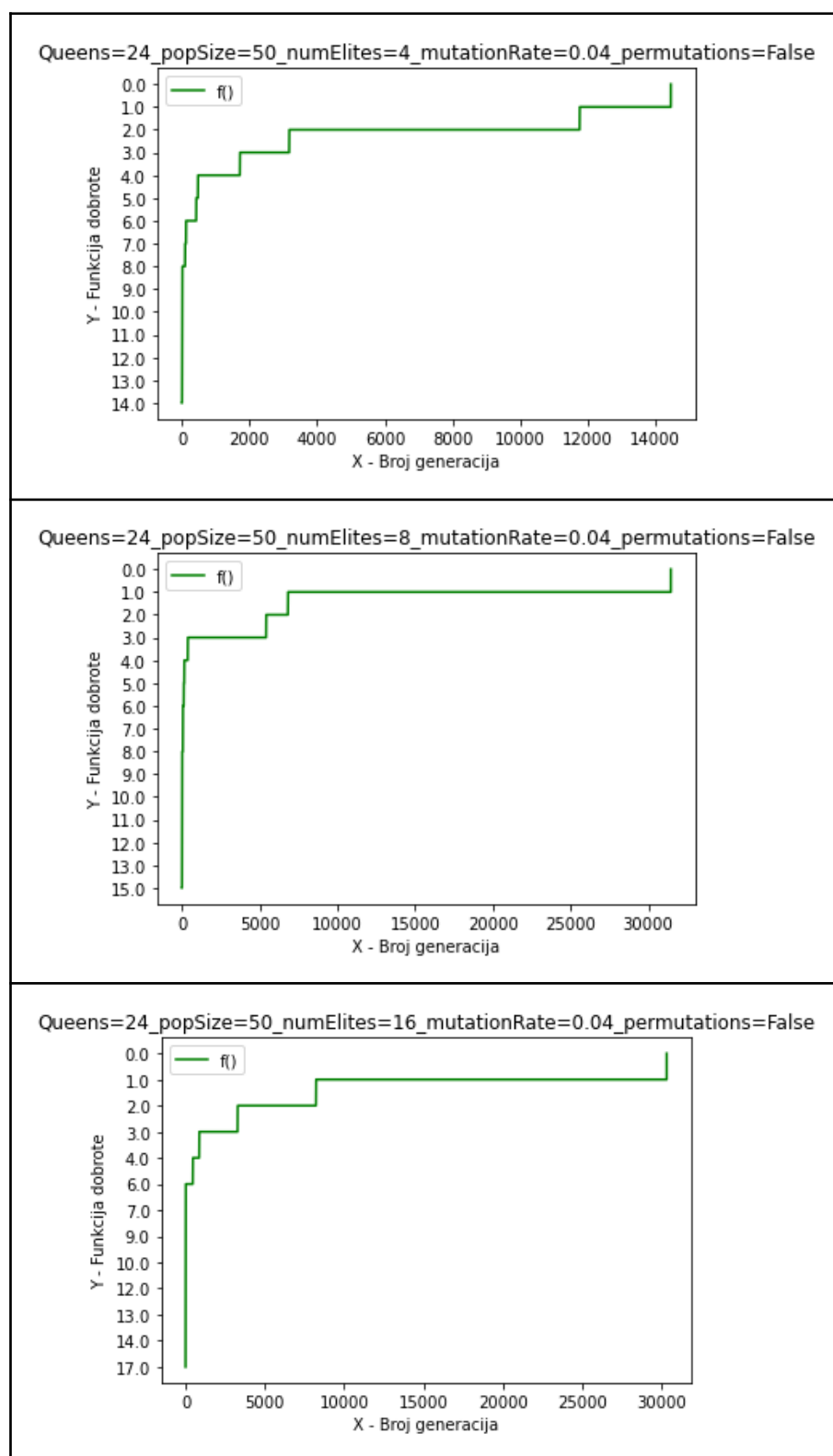




Tablica 3.2.2. Fitness funkcija u ovisnosti o broju elitnih članova za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	50000		
Mutacija	4%		
Broj elitnih članova	4	8	16
Broj generacija rješenja	6953,11742,14457,16606,24763	22049,22843,31458,33068,38155	12243,18263,30312,38642,45036
Prosječan broj generacija rješenja	14904.2	29514.6	28899.2
Broj pokušaja	16	20	11

Tablica 3.2.3. Utjecaj postotka broja elitnih članova za N=24

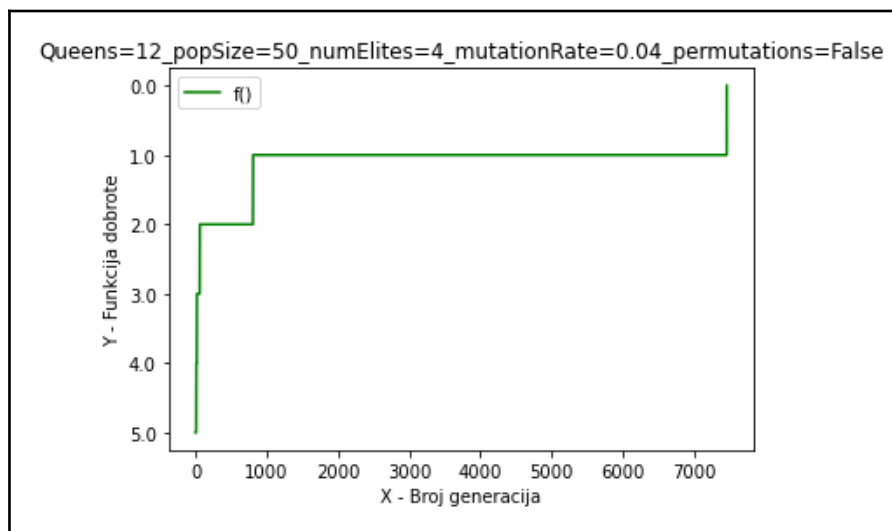


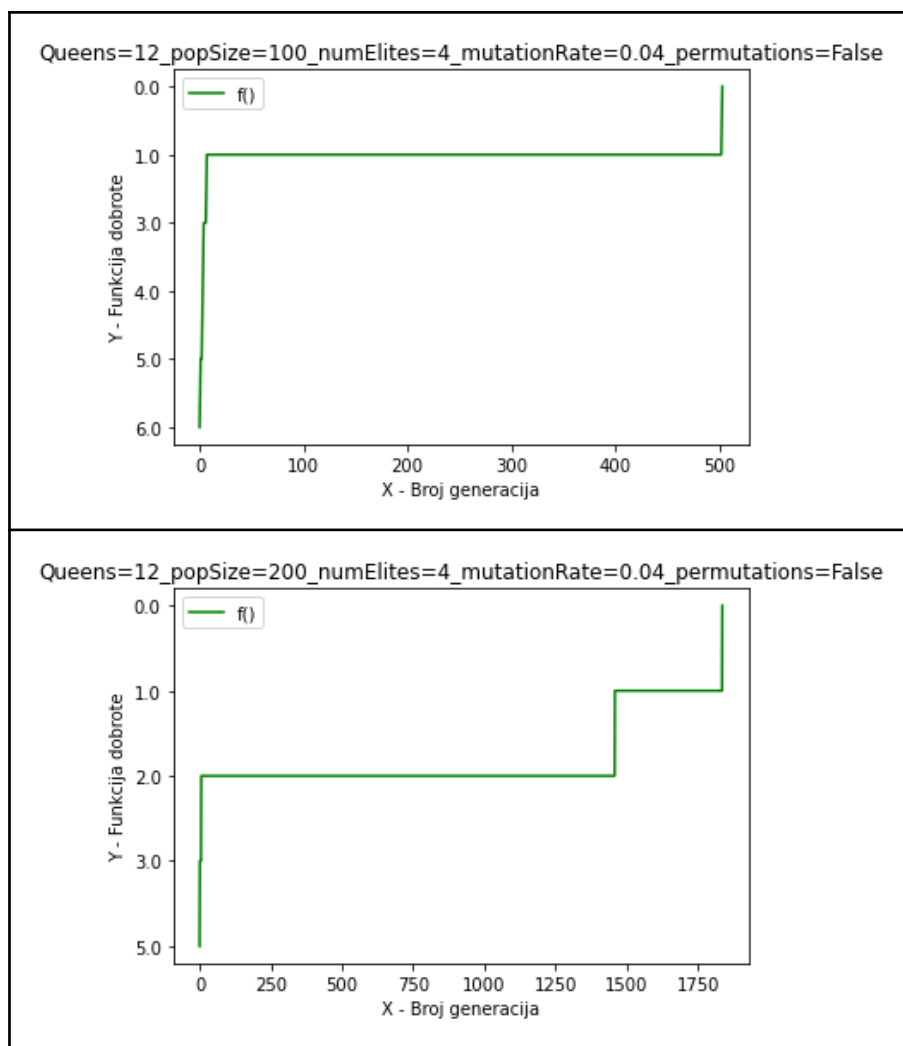
Tablica 3.2.4. Fitness funkcija u ovisnosti o broju elitnih članova za N=24

### 3.3. Ovisnost o broju populacija (bez permutacija)

Veličina ploče N	12		
Broj elitnih članova	4		
Broj generacija	50000		
Mutacija	4%		
Populacija	50	100	200
Broj generacija rješenja	1230,5846,7464, 8963,17604	107,182,504,1072 ,42372	12,1026,1836, 1979,8228
Prosječan broj generacija rješenja	8221.4	8847.4	2616.2
Broj pokušaja	12	9	7

Tablica 3.3.1. Utjecaj broja populacija za N=12

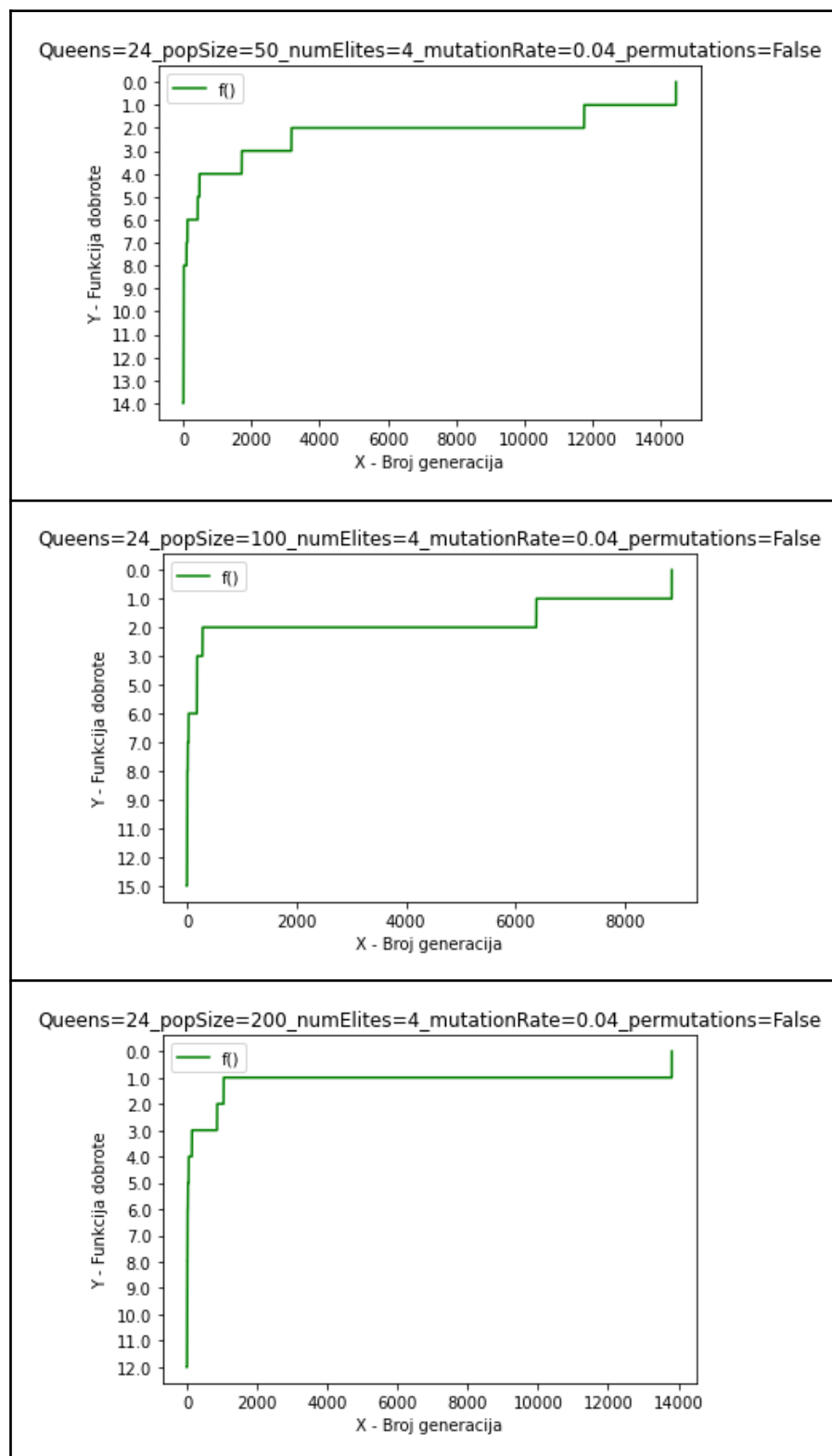




Tablica 3.3.2. Fitness funkcija u ovisnosti o broju populacija za N=12

Veličina ploče N	24		
Broj elitnih članova	4		
Broj generacija	50000		
Mutacija	4%		
Populacija	50	100	200
Broj generacija rješenja	6953,11742,14457,16606,24763	6478,8559,8849,24243,45445	2385,6140,13816,16038,18774
Prosječan broj generacija rješenja	14904.2	18714.8	11430.6
Broj pokušaja	16	15	12

Tablica 3.3.3. Utjecaj broja populacija za N=24

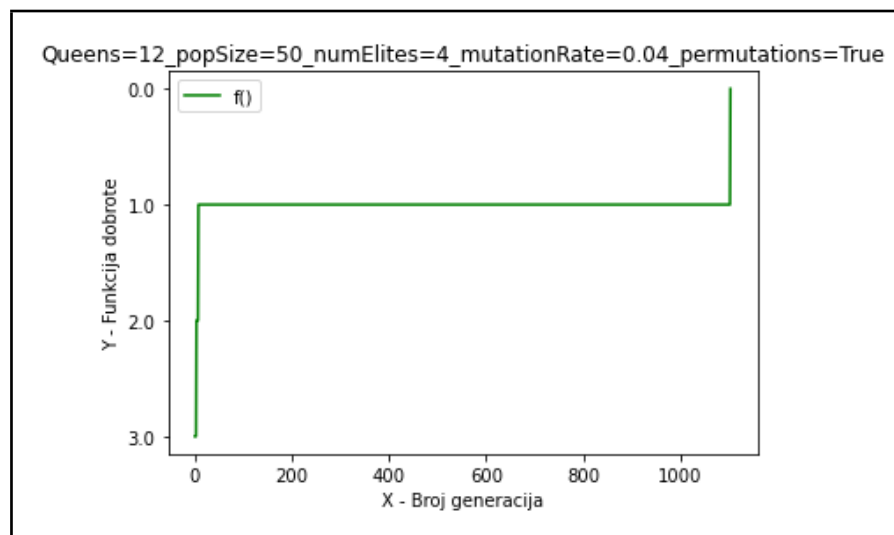


Tablica 3.3.4. Fitness funkcija u ovisnosti o broju populacija za  $N=24$

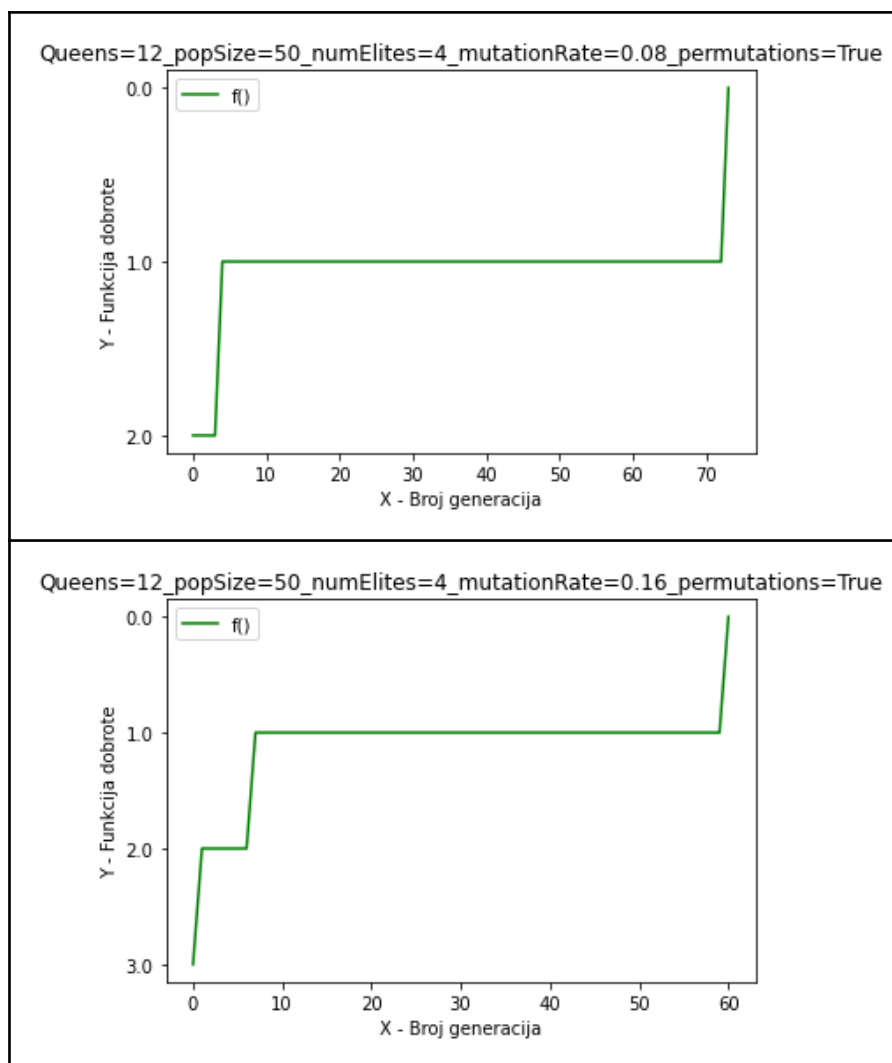
### 3.4. Ovisnost o postotku mutacije (s permutacijama)

Veličina ploče N	12		
Populacija	50		
Broj generacija	50000		
Broj elitnih članova	4		
Mutacija	4%	8%	16%
Broj generacija rješenja	44,438,1105,1252,1620	6,11,74,415,982	6,9,61,351,514
Prosječan broj generacija rješenja	891.8	297.6	188.2
Broj pokušaja	5	5	5

Tablica 3.4.1. Utjecaj postotka mutacije za N=12



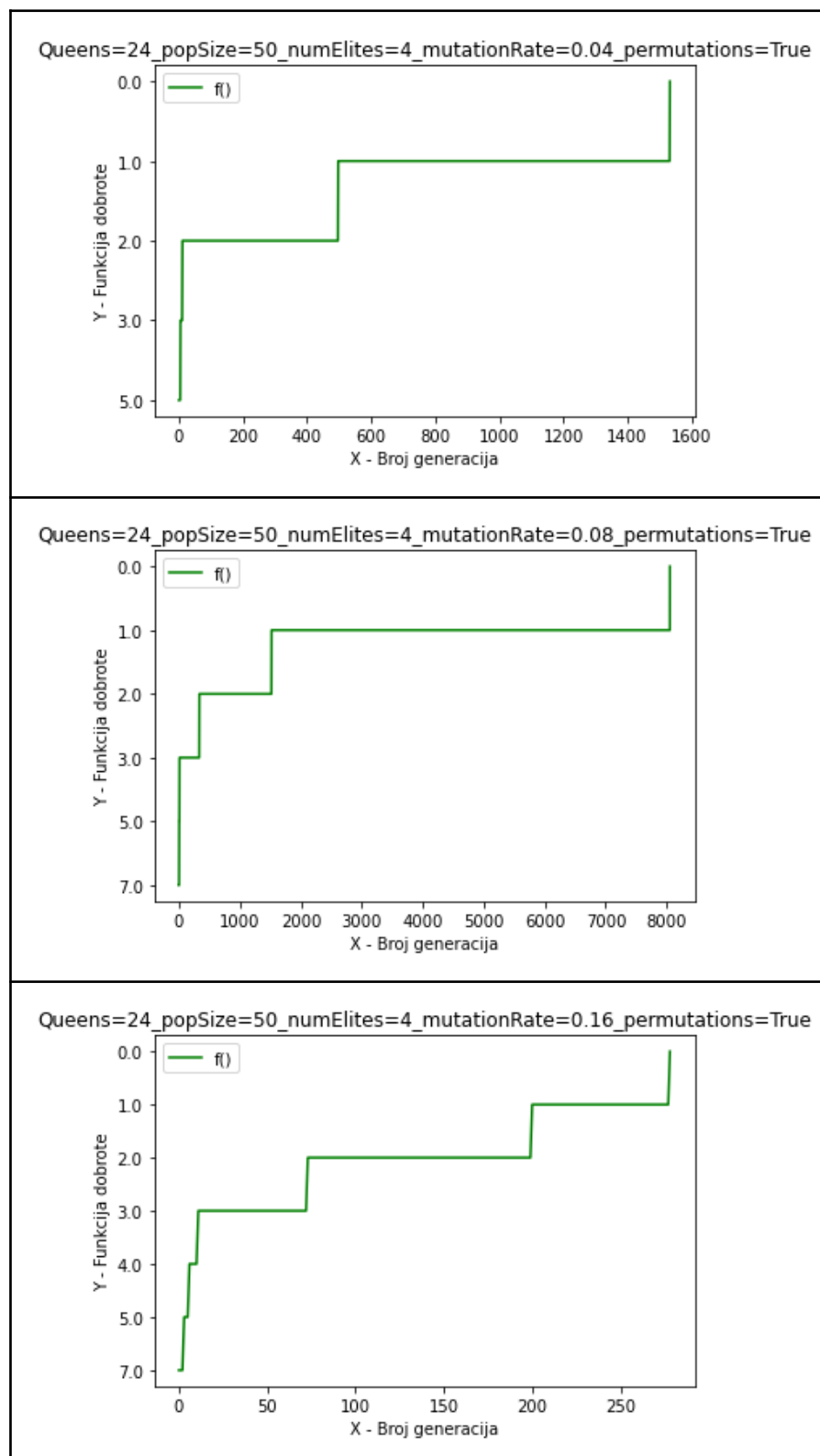




Tablica 3.4.2. Fitness funkcija u ovisnosti o postotku mutacije za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	50000		
Broj elitnih članova	4		
Mutacija	4%	8%	16%
Broj generacija rješenja	17,51,1533,19704,32931	1299,7437,8055,10023,11813	125,179,279,1323,2309
Prosječan broj generacija rješenja	10847.2	7725.4	843.0
Broj pokušaja	6	5	5

Tablica 3.4.3. Utjecaj postotka mutacije za N=24

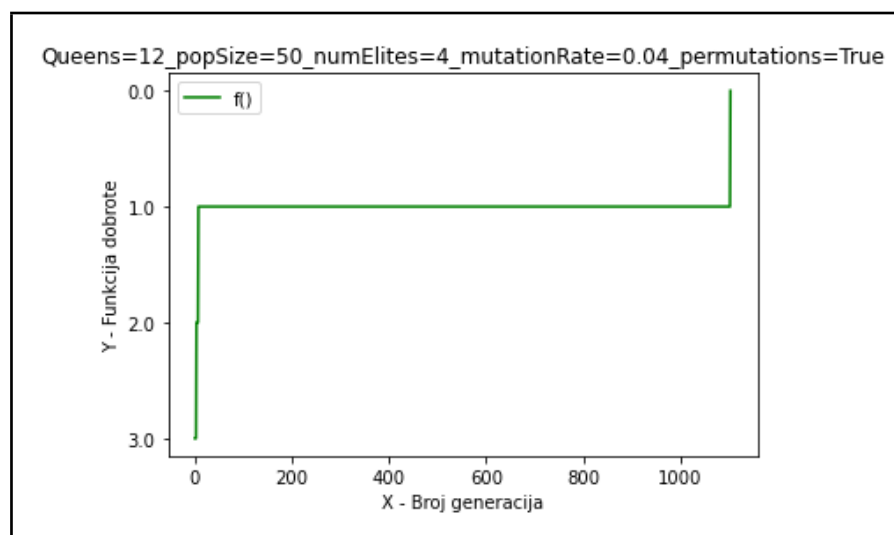


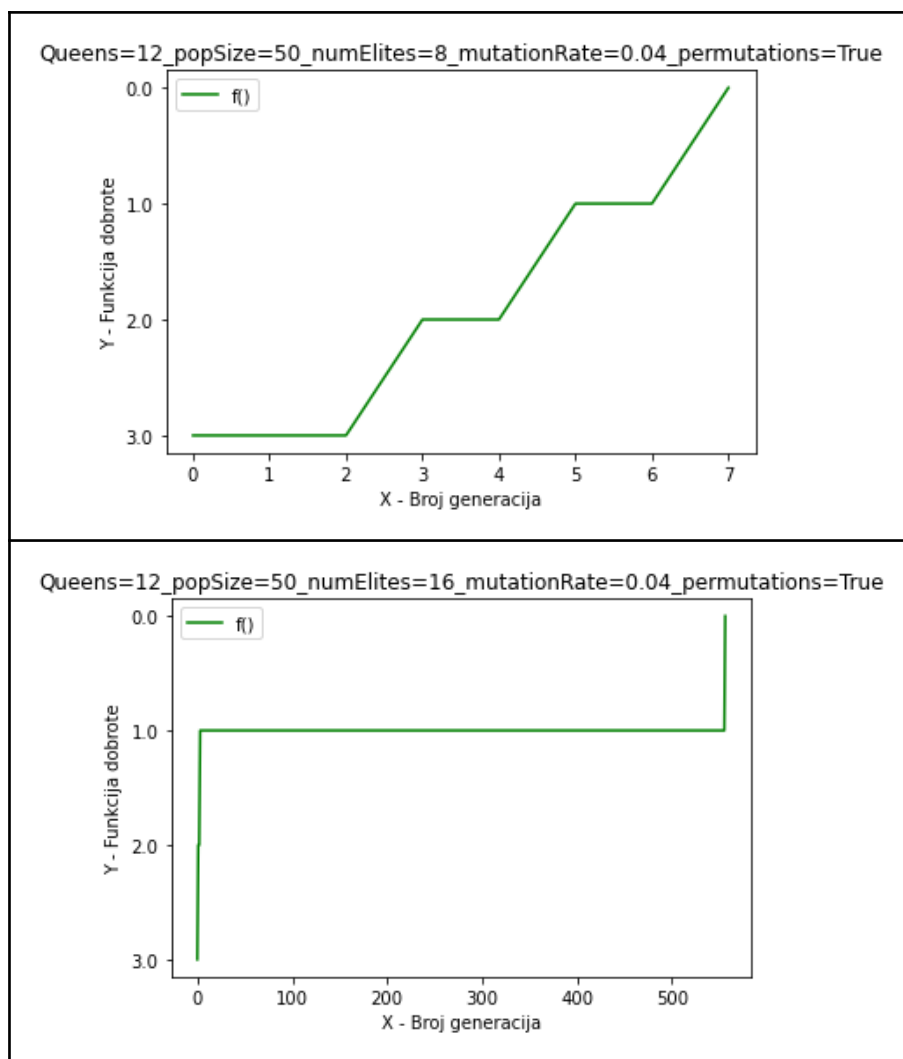
Tablica 3.4.4. Fitness funkcija u ovisnosti o postotku mutacije za N=24

### 3.5. Ovisnost o broju elitnih članova (s permutacijama)

Veličina ploče N	12		
Populacija	50		
Broj generacija	50000		
Mutacija	4%		
Broj elitnih članova	4	8	16
Broj generacija rješenja	44,438,1105,1252,1620	6,7,8,9,1514	228,429,557,800,6497
Prosječan broj generacija rješenja	891.8	308.8	1702.2
Broj pokušaja	5	5	5

Tablica 3.5.1. Utjecaj postotka broja elitnih članova za N=12

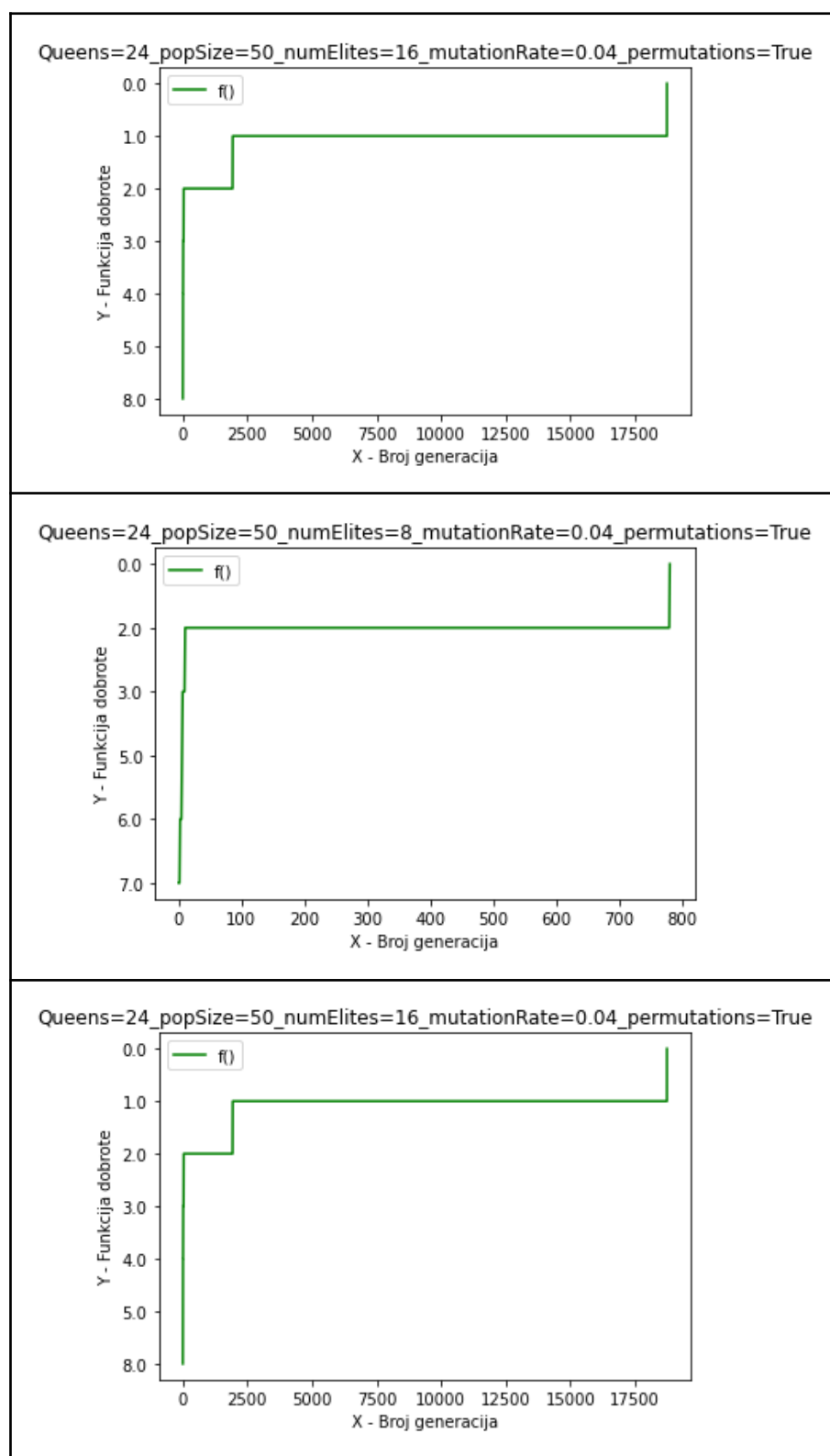




Tablica 3.5.2. Fitness funkcija u ovisnosti o broju elitnih članova za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	50000		
Mutacija	4%		
Broj elitnih članova	4	8	16
Broj generacija rješenja	17,51,1533,19704,32931	273,334,782,6161,16095	2612,3621,18734,21180,48196
Prosječan broj generacija rješenja	10847.2	4729.0	18868.6
Broj pokušaja	6	5	5

Tablica 3.5.3. Utjecaj postotka broja elitnih članova za N=24

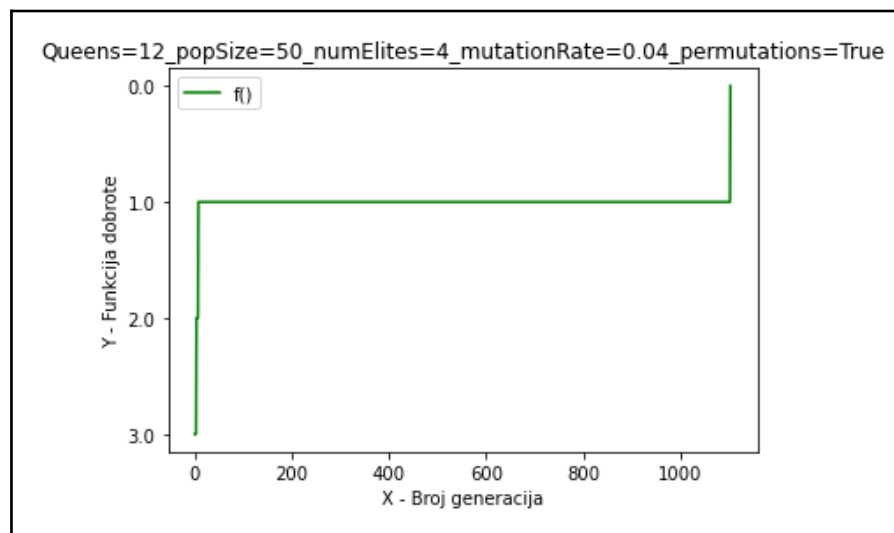


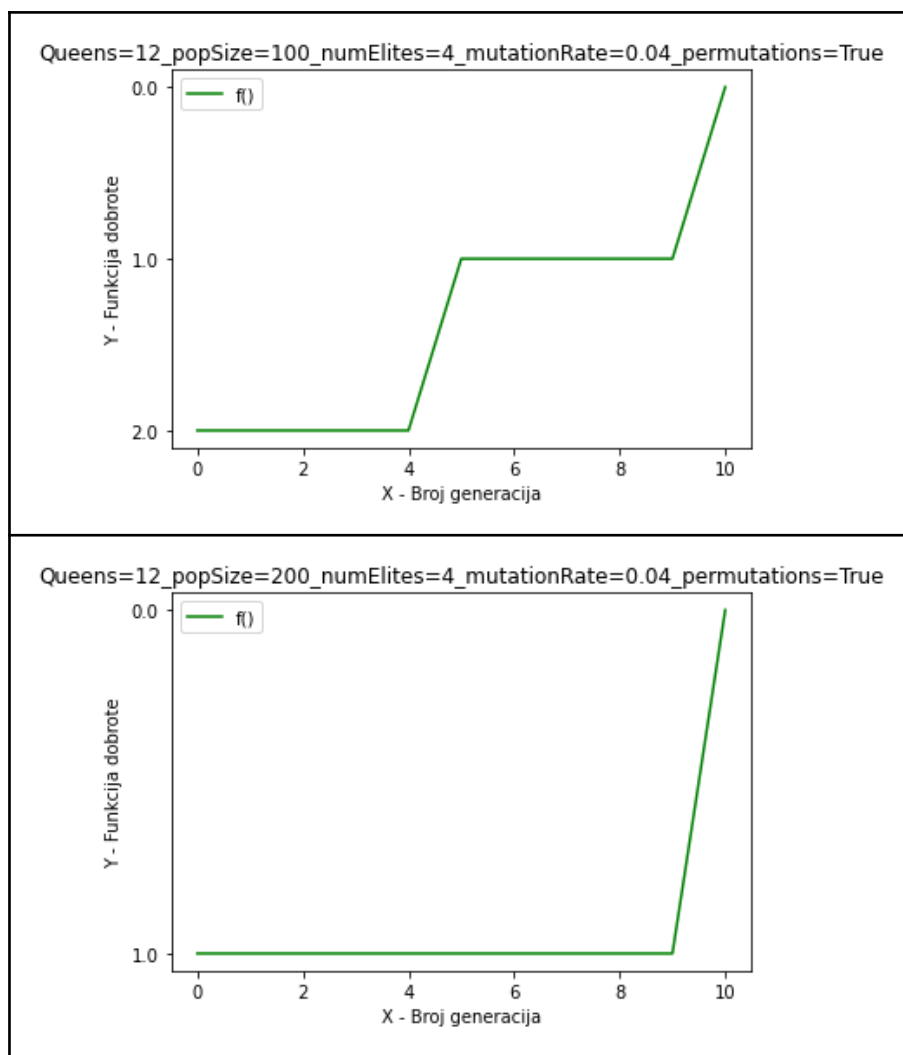
Tablica 3.5.4. Fitness funkcija u ovisnosti o broju elitnih članova za N=24

### 3.6. Ovisnost o broju populacija (s permutacijama)

Veličina ploče N	12		
Broj elitnih članova	4		
Broj generacija	50000		
Mutacija	4%		
Populacija	50	100	200
Broj generacija rješenja	44,438,1105,1252,1620	9,9,11,201,826	7,8,11,18,173
Prosječan broj generacija rješenja	891.8	211.2	43.4
Broj pokušaja	5	5	5

Tablica 3.6.1. Utjecaj broja populacija za N=12

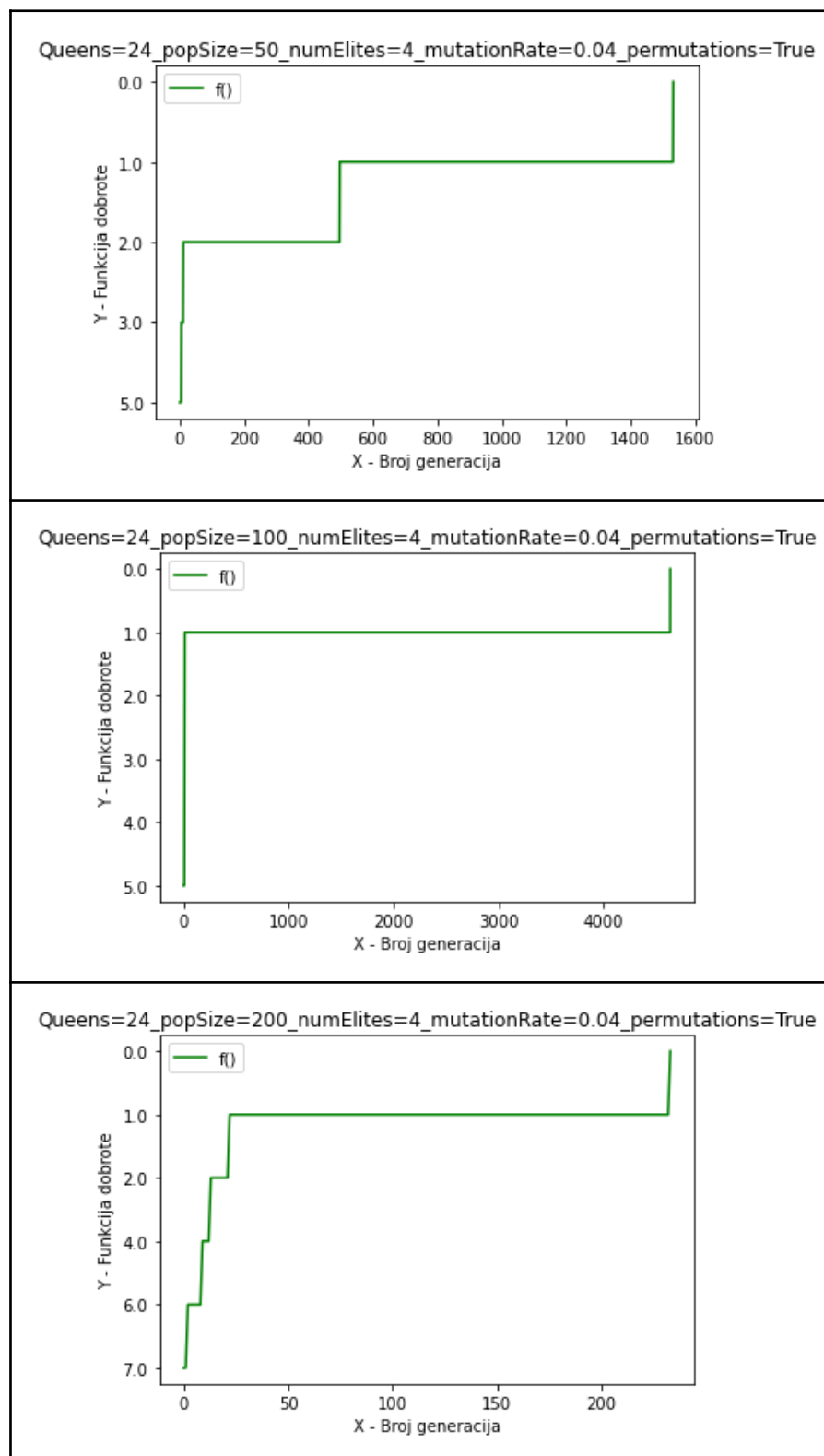




Tablica 3.6.2. Fitness funkcija u ovisnosti o broju populacija za N=12

Veličina ploče N	24		
Broj elitnih članova	4		
Broj generacija	50000		
Mutacija	4%		
Populacija	50	100	200
Broj generacija rješenja	17,51,1533,19704,32931	25,734,4641,8164,35160	25,29,234,768,1654
Prosječan broj generacija rješenja	10847.2	9744.8	542.0
Broj pokušaja	6	5	5

Tablica 3.6.3. Utjecaj broja populacija za N=24



Tablica 3.6.4. Fitness funkcija u ovisnosti o broju populacija za  $N=24$



### 3.7. ZAKLJUČAK

Nakon izvršenih kombinacija parametara za navedeni algoritam računanja problema N dama, možemo zaključiti da promjenom bilo kojeg parametra možemo ubrzati ili usporiti pronalaženje točnog rješenja. Ta ovisnost najbolje se može vidjeti na grafovima.

Može se vidjeti da povećanje mutacija utječe tako što se dobar materijal gubi, a neki novi mutirani geni zauzimaju njihovo mjesto. Možemo vidjeti da s povećanjem broja populacije ubrzavamo proces pronalaska rješenja.

Za točnije rješenje bilo bi potrebno naći više od 5 točnih rješenja. Problem N-dama bilo je brže riješiti korištenjem permutacijskog cjelobrojnog kodiranja nego cjelobrojnog kodiranja. Cjelobrojno kodiranje nije čak moglo pronaći neka rješenja (izvrtiti algoritam do kraja za veličinu  $N=48$ , zaustavljeno nakon 35 pokušaja).

### 3.8. Popis tablica

#### BEZ PERMUTACIJA:

Tablica 3.1.1. Utjecaj postotka mutacije za  $N=12$

Tablica 3.1.2. Fitness funkcija u ovisnosti o postotku mutacije za  $N=12$

Tablica 3.1.3. Utjecaj postotka mutacije za  $N=24$

Tablica 3.1.4. Fitness funkcija u ovisnosti o postotku mutacije za  $N=24$

Tablica 3.2.1. Utjecaj postotka broja elitnih članova za  $N=12$

Tablica 3.2.2. Fitness funkcija u ovisnosti o broju elitnih članova za  $N=12$

Tablica 3.2.3. Utjecaj postotka broja elitnih članova za  $N=24$

Tablica 3.2.4. Fitness funkcija u ovisnosti o broju elitnih članova za  $N=24$

Tablica 3.3.1. Utjecaj broja populacija za  $N=12$

Tablica 3.3.2. Fitness funkcija u ovisnosti o broju populacija za  $N=12$

Tablica 3.3.3. Utjecaj broja populacija za  $N=24$

Tablica 3.3.4. Fitness funkcija u ovisnosti o broju populacija za  $N=24$

#### S PERMUTACIJAMA:

Tablica 3.4.1. Utjecaj postotka mutacije za  $N=12$

Tablica 3.4.2. Fitness funkcija u ovisnosti o postotku mutacije za  $N=12$

Tablica 3.4.3. Utjecaj postotka mutacije za  $N=24$

Tablica 3.4.4. Fitness funkcija u ovisnosti o postotku mutacije za  $N=24$

Tablica 3.5.1. Utjecaj postotka broja elitnih članova za  $N=12$

Tablica 3.5.2. Fitness funkcija u ovisnosti o broju elitnih članova za  $N=12$

Tablica 3.5.3. Utjecaj postotka broja elitnih članova za  $N=24$

Tablica 3.5.4. Fitness funkcija u ovisnosti o broju elitnih članova za  $N=24$

Tablica 3.6.1. Utjecaj broja populacija za  $N=12$

Tablica 3.6.2. Fitness funkcija u ovisnosti o broju populacija za  $N=12$

Tablica 3.6.3. Utjecaj broja populacija za  $N=24$

Tablica 3.6.4. Fitness funkcija u ovisnosti o broju populacija za  $N=24$