

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

Rješavanje problema N-dama koristeći genetski algoritam

Meko računarstvo

Laboratorijska vježba 1

Aleksandra Kuridža

Diplomski studij računarstva, DRB

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 C# programskom jeziku, gdje je moguće unositi veličinu N (broj dama) od 4 do 100. 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

Prvo ćemo opisati osnove genetskog algoritma. Genetski algoritam je metaheuristička metoda optimiranja koja se temelji na ideji biološke evolucije vrsta, tj. preživljavanje najsposobnijih vrsta. Evolucija je prirodni proces traženja najbolje i najprilagodljivije jedinke u okolini i uvjetima u prirodi. Jedinica 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. Ta se pojava događa vrlo rijetko. 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 bolja svojstva.

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:

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 prikažu 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, 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.

3. RJEŠENJE PROBLEMA 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 ova 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 pamtiti koliko je puta bilo potrebno pokrenuti algoritam s nekom konfiguracijom kako bi se došlo do 5 točnih rješenja.

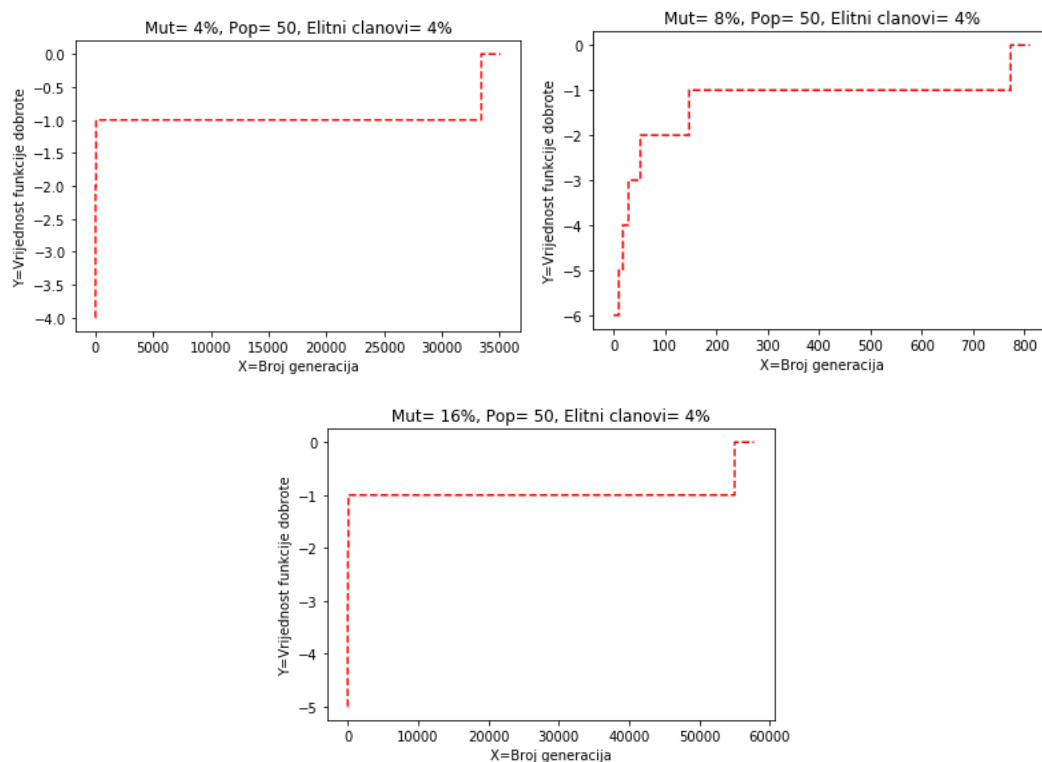
Parametre genetskog algoritma bilo je potrebno mijenjati na slijedeće vrijednosti:

- 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	60000		
Broj elitnih članova	4		
Mutacija	4	8	16
Broj generacija rješenja	206, 265, 30276, 33305, 33420	171, 258, 410, 728, 24565	52, 136, 331, 596, 39677
Prosječan broj generacija rješenja	19495	5227	8159
Broj neuspjelih	6	1	1

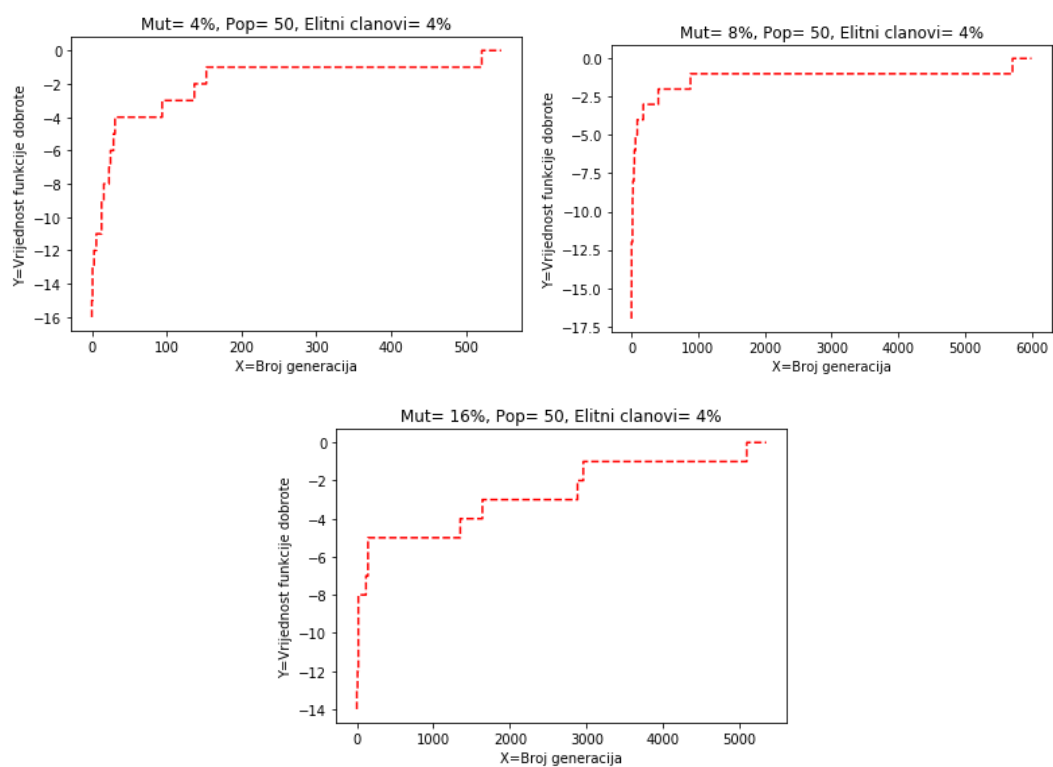
Tablica 3.1.1. Utjecaj postotka mutacije za $N=12$



Slika 3.1.1. Fitness funkcija u ovisnosti o postotku mutacije za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	60000		
Broj elitnih članova	4		
Mutacija	4	8	16
Broj generacija rješenja	523, 2381, 4840, 10096, 29079	1360, 5553, 17512, 36337, 48869	1420, 2312, 5491, 12738, 27242
Prosječan broj generacija rješenja	9386	21927	9841
Broj neuspjelih	8	6	7

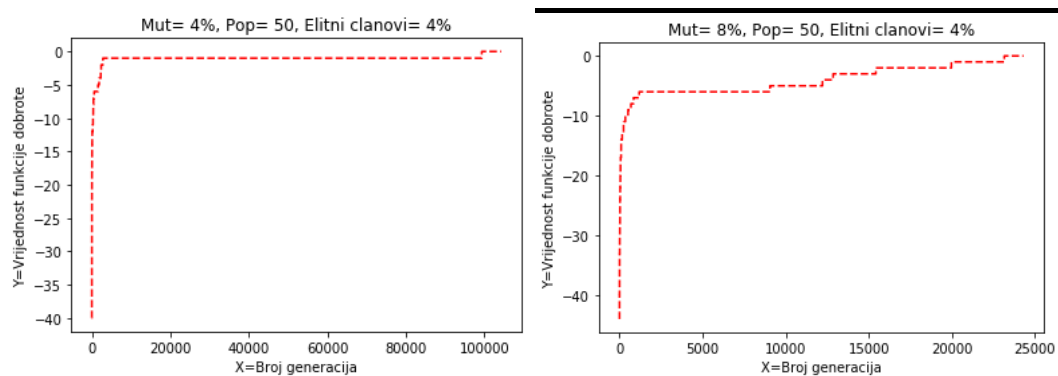
Tablica 3.1.2. Utjecaj postotka mutacije za N=24



Slika 3.1.2. Fitness funkcija u ovisnosti o postotku mutacije za N=24

Veličina ploče N	48		
Populacija	50		
Broj generacija	80000		
Broj elitnih članova	4		
Mutacija	4	8	16
Broj generacija rješenja	18693, 99341, 19230, 2584, 1520	23222	Program nije uspio pronaci rjesenje
Prosječan broj generacija rješenja	28219	23222	
Broj neuspjelih	93	-	-

Tablica 3.1.3. Utjecaj postotka mutacije za N=48

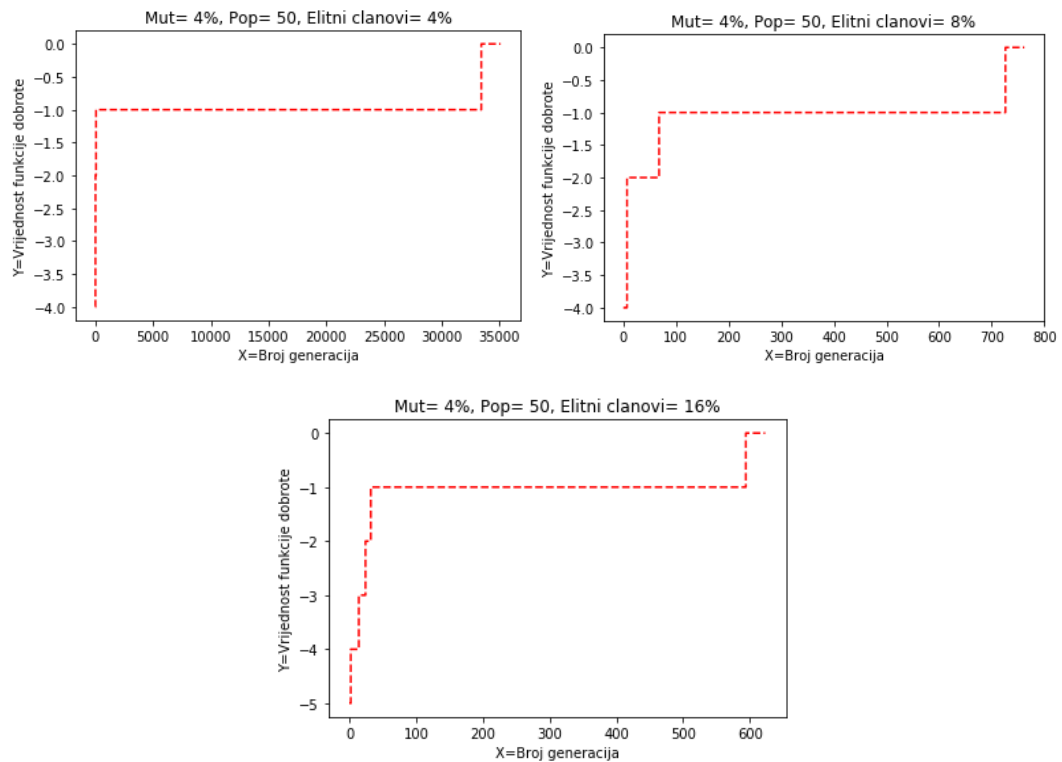


Slika 3.1.3. Fitness funkcija u ovisnosti o postotku mutacije za N=48

3.2. OVISNOST O BROJU ELITNIH ČLANOVA (BEZ PERMUTACIJA)

Veličina ploče N	12		
Populacija	50		
Broj generacija	60000		
Mutacija	4		
Broj elitnih članova	4	8	16
Broj generacija rješenja	206, 265, 30276, 33305, 33420	264, 492, 775, 1055, 1658	246, 1432, 1711, 16013, 55025
Prosječan broj generacija rješenja	19495	846	14886
Broj neuspjelih	6	1	1

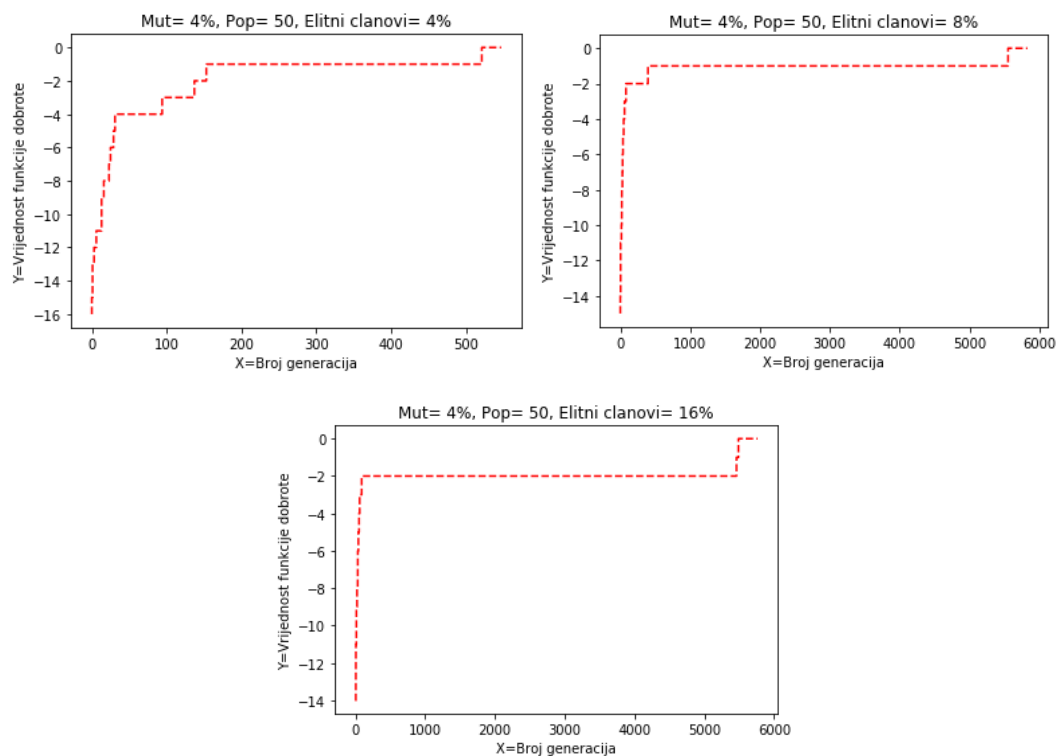
Tablica 3.2.1. Utjecaj postotka broja elitnih članova za N=12



Slika 3.2.1. Fitness funkcija u ovisnosti o broju elitnih članova za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	60000		
Mutacija	4		
Broj elitnih članova	4	8	16
Broj generacija rješenja	523, 2381, 4840, 10096, 29079	5014, 5717, 18632, 54748, 56166	4133, 5095, 12217, 35278, 48909
Prosječan broj generacija rješenja	9386	28056	21127
Broj neuspjelih	8	6	7

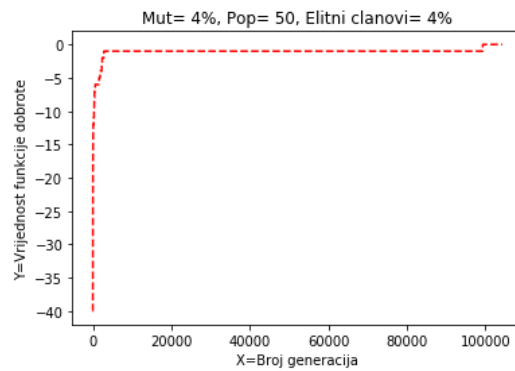
Tablica 3.2.2. Utjecaj postotka broja elitnih članova za N=24



Slika 3.2.2. Fitness funkcija u ovisnosti o broju elitnih članova za N=24

Veličina ploče N	48		
Populacija	50		
Broj generacija	80000		
Mutacija	4		
Broj elitnih članova	4	8	16
Broj generacija rješenja	18693, 99341, 19230, 2584, 1520	Program nije uspio pronaci rjesenje	Program nije uspio pronaci rjesenje
Prosječan broj generacija rješenja	28219		
Broj neuspjelih	93	-	-

Tablica 3.2.3. Utjecaj postotka broja elitnih članova za N=48

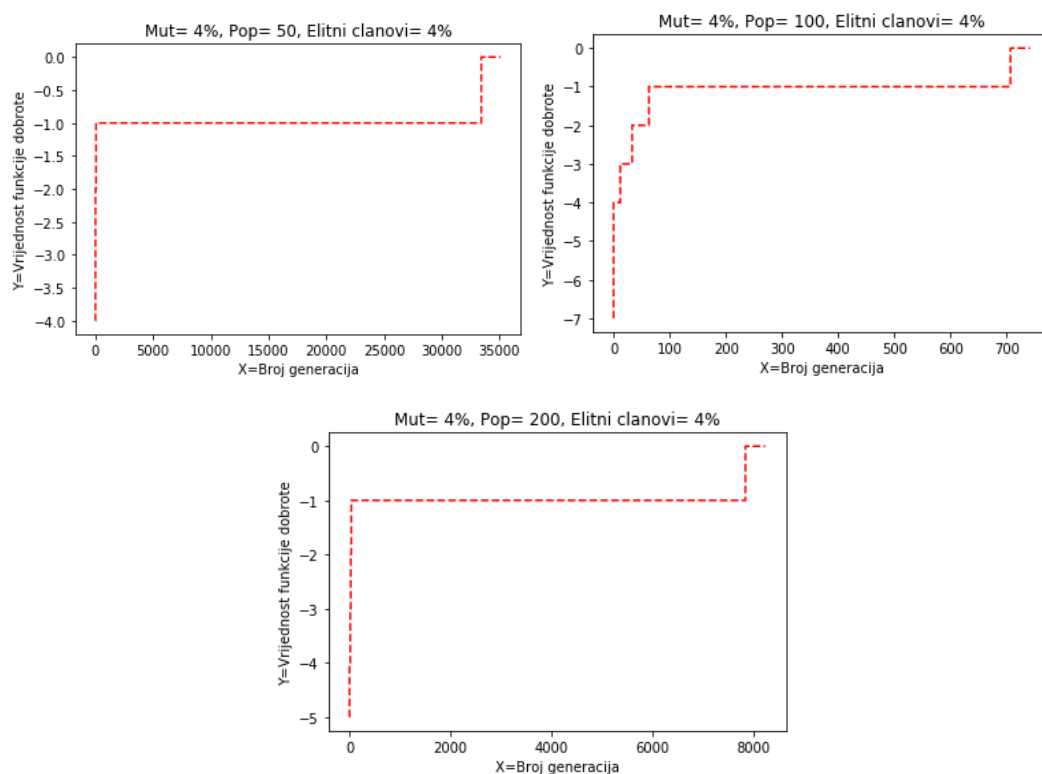


Slika 3.2.3. Fitness funkcija u ovisnosti o broju elitnih članova za N=48

3.3. OVISNOST O BROJU POPULACIJA (BEZ PERMUTACIJA)

Veličina ploče N	12		
Broj elitnih članova	4		
Broj generacija	60000		
Mutacija	4		
Populacija	50	100	200
Broj generacija rješenja	206, 265, 30276, 33305, 33420	154, 222, 709, 2043, 5841	70, 72, 281, 3499, 7852
Prosječan broj generacija rješenja	19495	1794	2355
Broj neuspjelih	6	2	0

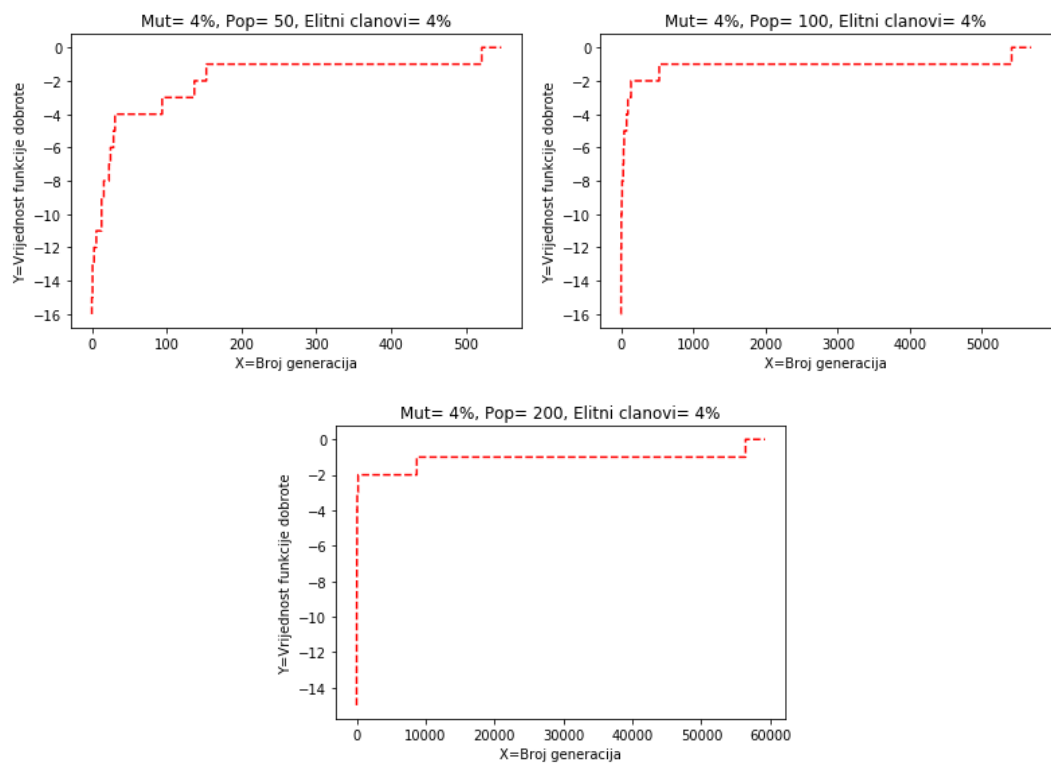
Tablica 3.3.1. Utjecaj broja populacija za N=12



Slika 3.3.1. Fitness funkcija u ovisnosti o broju populacija za N=12

Veličina ploče N	24		
Broj elitnih članova	4		
Broj generacija	60000		
Mutacija	4		
Populacija	50	100	200
Broj generacija rješenja	523, 2381, 4840, 10096, 29079	290, 2476, 3537, 5412, 14500	1009, 2243, 30991, 33505, 56427
Prosječan broj generacija rješenja	9386	5243	24835
Broj neuspjelih	8	9	7

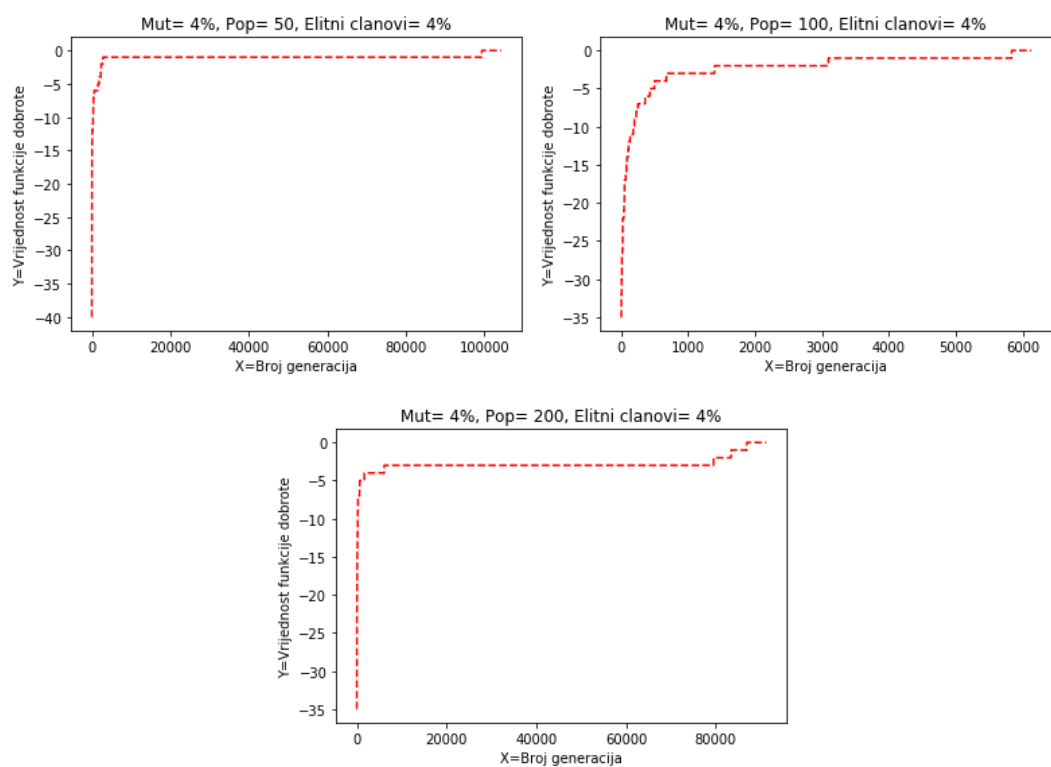
Tablica 3.3.2. Utjecaj broja populacija za N=24



Slika 3.3.2. Fitness funkcija u ovisnosti o broju populacija za N=24

Veličina ploče N	48		
Broj elitnih članova	4		
Broj generacija	60000 – 120000		
Mutacija	4		
Populacija	50	100	200
Broj generacija rješenja	18693, 99341, 19230, 2584, 1520	5832	10316, 22861, 65111, 84208, 86870
Prosječan broj generacija rješenja	28219	5832	53874
Broj neuspjelih	93	22	88

Tablica 3.3.3. Utjecaj broja populacija za N=48

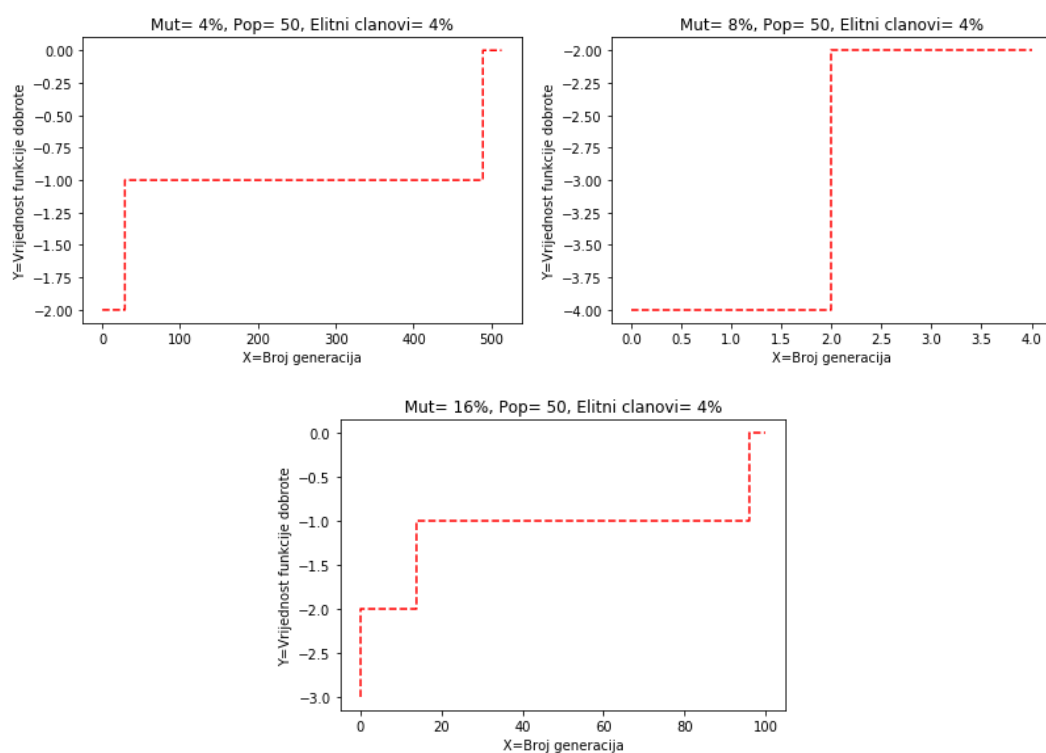


Slika 3.3.3. Fitness funkcija u ovisnosti o broju populacija za N=48

3.4. OVISNOST O POSTOTKU MUTACIJE (S PERMUTACIJAMA)

Veličina ploče N	12		
Populacija	50		
Broj generacija	60000		
Broj elitnih članova	4		
Mutacija	4	8	16
Broj generacija rješenja	138, 203, 234, 239, 491	7, 32, 58, 110, 1637	18, 40, 89, 101, 299
Prosječan broj generacija rješenja	261	369	109
Broj neuspjelih	0	0	0

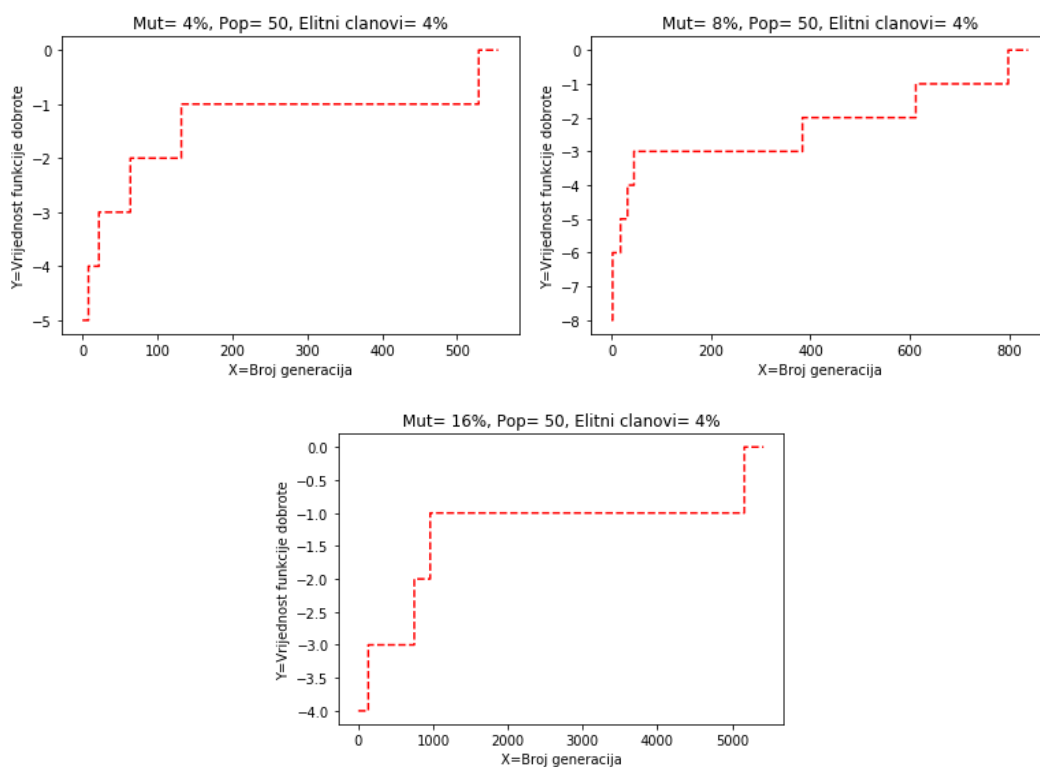
Tablica 3.4.1. Utjecaj postotka mutacije za N=12



Slika 3.4.1. Fitness funkcija u ovisnosti o postotku mutacije za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	60000		
Broj elitnih članova	4		
Mutacija	4	8	16
Broj generacija rješenja	132, 159, 531, 15887, 20050	67, 339, 968, 3258, 3491	96, 363, 402, 421, 500
Prosječan broj generacija rješenja	7352	1625	357
Broj neuspjelih	0	0	0

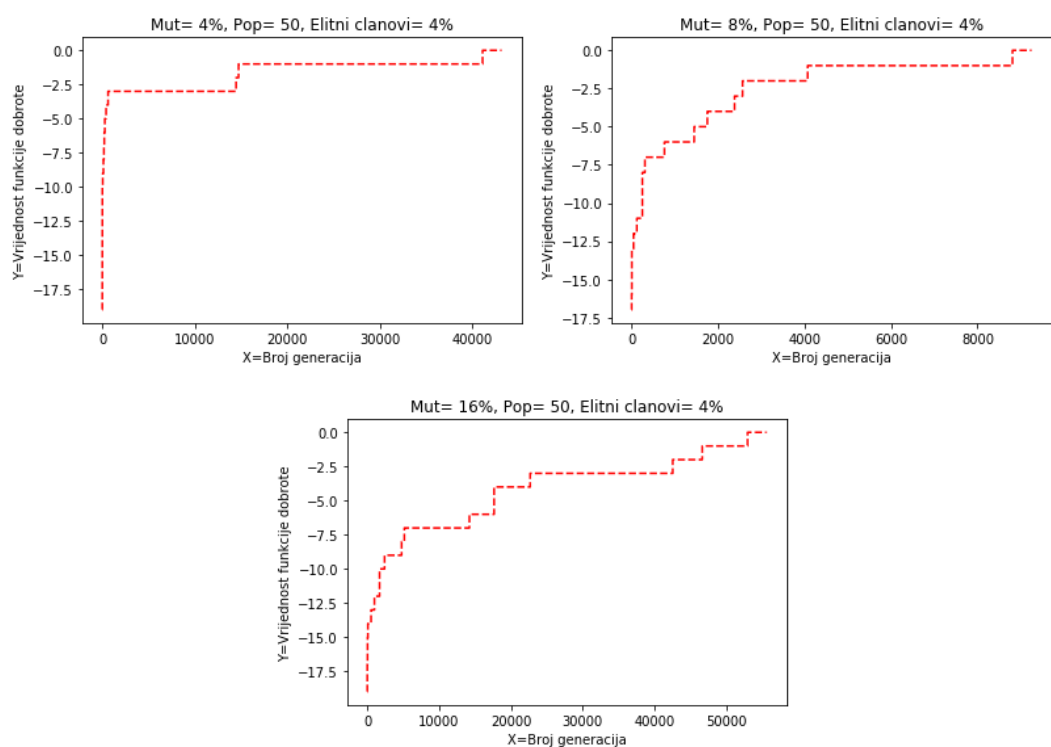
Tablica 3.4.2. Utjecaj postotka mutacije za N=24



Slika 3.4.2. Fitness funkcija u ovisnosti o postotku mutacije za N=24

Veličina ploče N	48		
Populacija	50		
Broj generacija	60000		
Broj elitnih članova	4		
Mutacija	4	8	16
Broj generacija rješenja	2489, 2818, 20185, 35781, 41133	555, 629, 1672, 4901, 5664	786, 987, 1125, 1586, 1678
Prosječan broj generacija rješenja	20482	2685	1233
Broj neuspjelih	1	7	293

Tablica 3.4.3. Utjecaj postotka mutacije za N=48

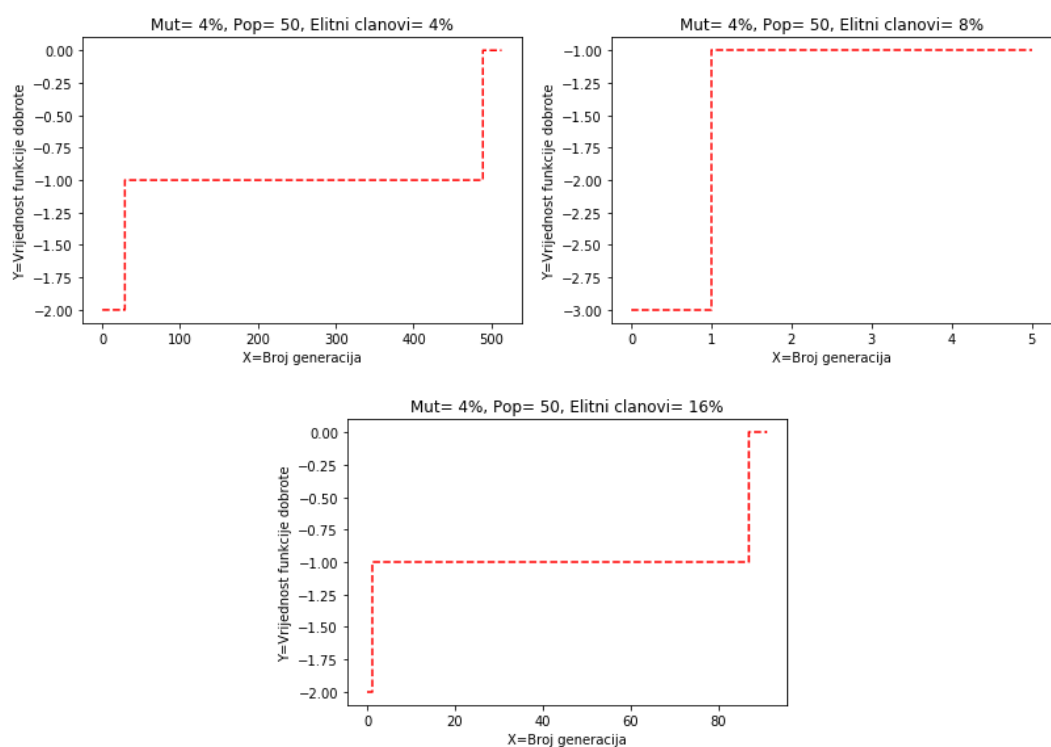


Slika 3.4.3. Fitness funkcija u ovisnosti o postotku mutacije za N=48

3.5. OVISNOST O BROJU ELITNIH ČLANOVA (S PERMUTACIJAMA)

Veličina ploče N	12		
Populacija	50		
Broj generacija	60000		
Mutacija	4		
Broj elitnih članova	4	8	16
Broj generacija rješenja	138, 203, 234, 239, 491	6, 12, 649	21, 82, 98, 131, 594
Prosječan broj generacija rješenja	261	223	186
Broj neuspjelih	0	0	0

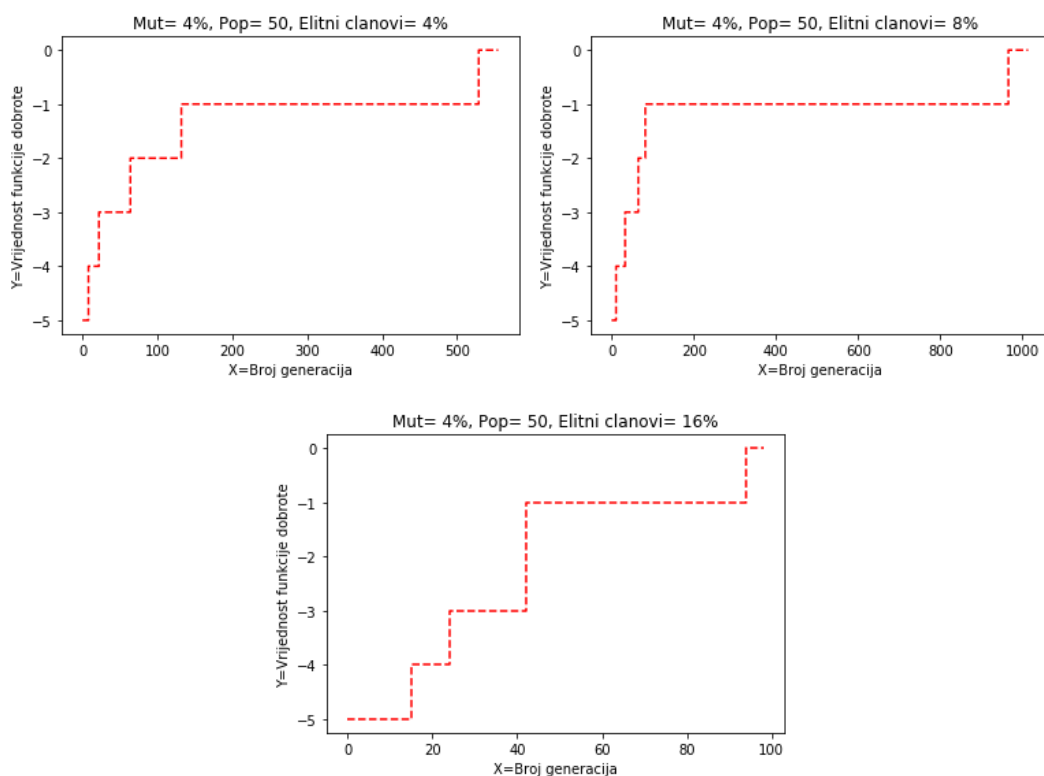
Tablica 3.5.1. Utjecaj postotka broja elitnih članova za N=12



Slika 3.5.1. Fitness funkcija u ovisnosti o broju elitnih članova za N=12

Veličina ploče N	24		
Populacija	50		
Broj generacija	60000		
Mutacija	4		
Broj elitnih članova	4	8	16
Broj generacija rješenja	132, 159, 531, 15887, 20050	176, 367, 800, 26988, 37807	3606, 4995, 5165, 16147, 24912
Prosječan broj generacija rješenja	7352	13228	10965
Broj neuspjelih	0	0	0

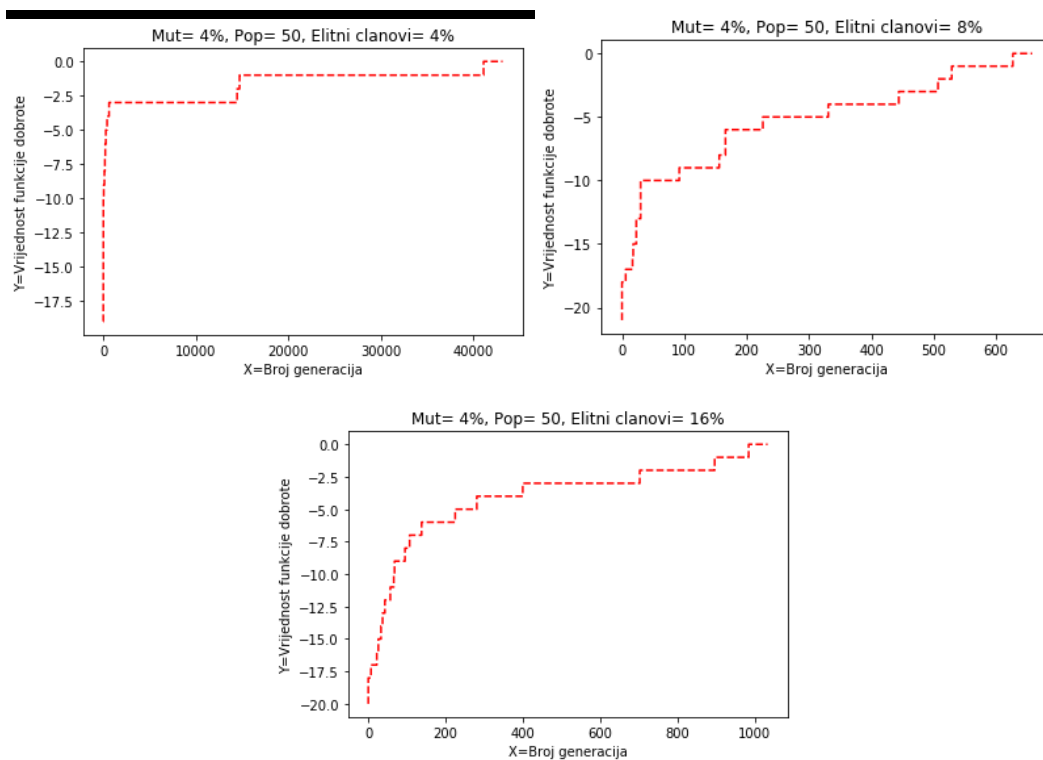
Tablica 3.5.2. Utjecaj postotka broja elitnih članova za N=24



Slika 3.5.2. Fitness funkcija u ovisnosti o broju elitnih članova za N=24

Veličina ploče N	48		
Populacija	50		
Broj generacija	60000		
Mutacija	4		
Broj elitnih članova	4	8	16
Broj generacija rješenja	2489, 2818, 20185, 35781, 41133	3031, 5529, 8522, 8811, 11015	27383, 45083, 47171, 48470, 52964
Prosječan broj generacija rješenja	20482	7242	44215
Broj neuspjelih	1	0	1

Tablica 3.5.3. Utjecaj postotka broja elitnih članova za N=48

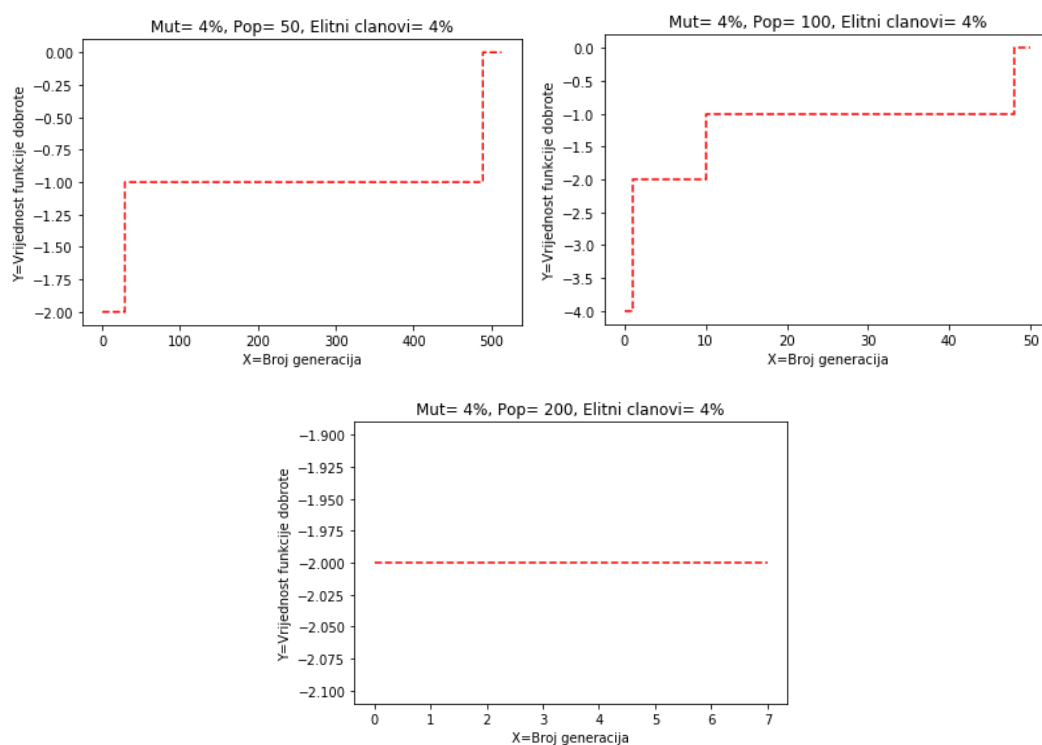


Slika 3.5.3. Fitness funkcija u ovisnosti o broju elitnih članova za N=48

3.6. OVISNOST O BROJU POPULACIJA (S PERMUTACIJAMA)

Veličina ploče N	12		
Broj elitnih članova	4		
Broj generacija	60000		
Mutacija	4		
Populacija	50	100	200
Broj generacija rješenja	138, 203, 234, 239, 491	18, 25, 50, 228, 283	7, 9, 25, 46, 126
Prosječan broj generacija rješenja	261	121	213
Broj neuspjelih	0	0	0

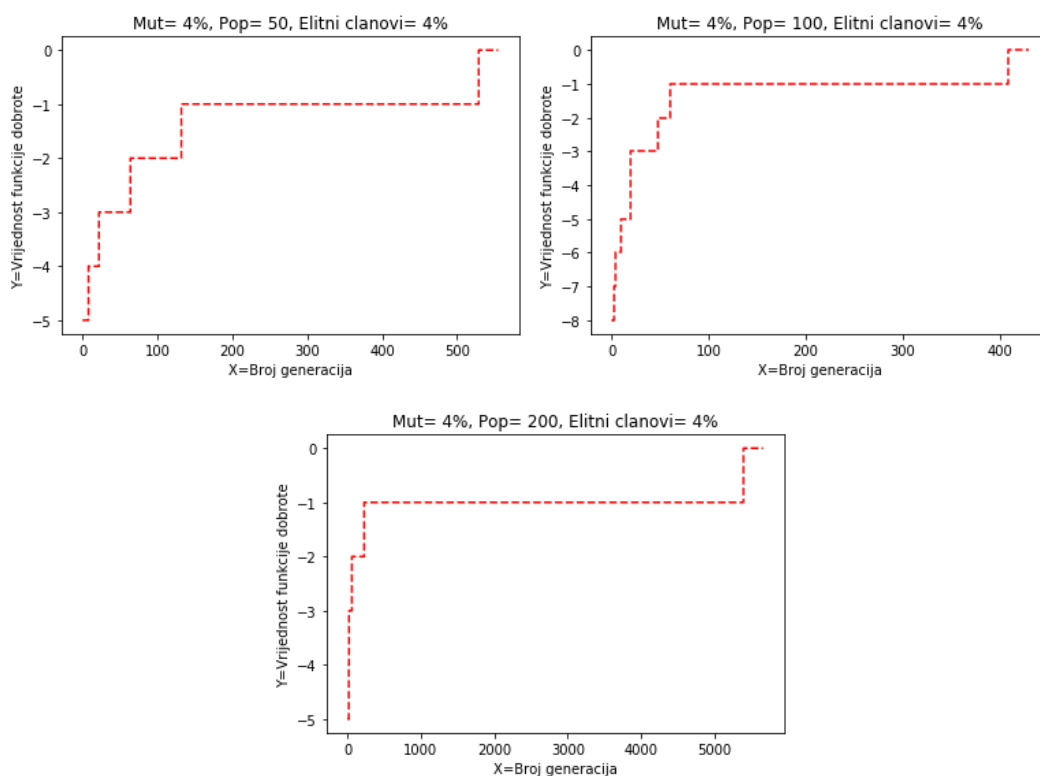
Tablica 3.6.1. Utjecaj broja populacija za N=12



Slika 3.6.1. Fitness funkcija u ovisnosti o broju populacija za N=12

Veličina ploče N	24		
Broj elitnih članova	4		
Broj generacija	60000		
Mutacija	4		
Populacija	50	100	200
Broj generacija rješenja	132, 159, 531, 15887, 20050	249, 411, 1685, 11768, 24914	103, 1530, 2555, 4867, 5397
Prosječan broj generacija rješenja	7352	7806	2891
Broj neuspjelih	0	1	0

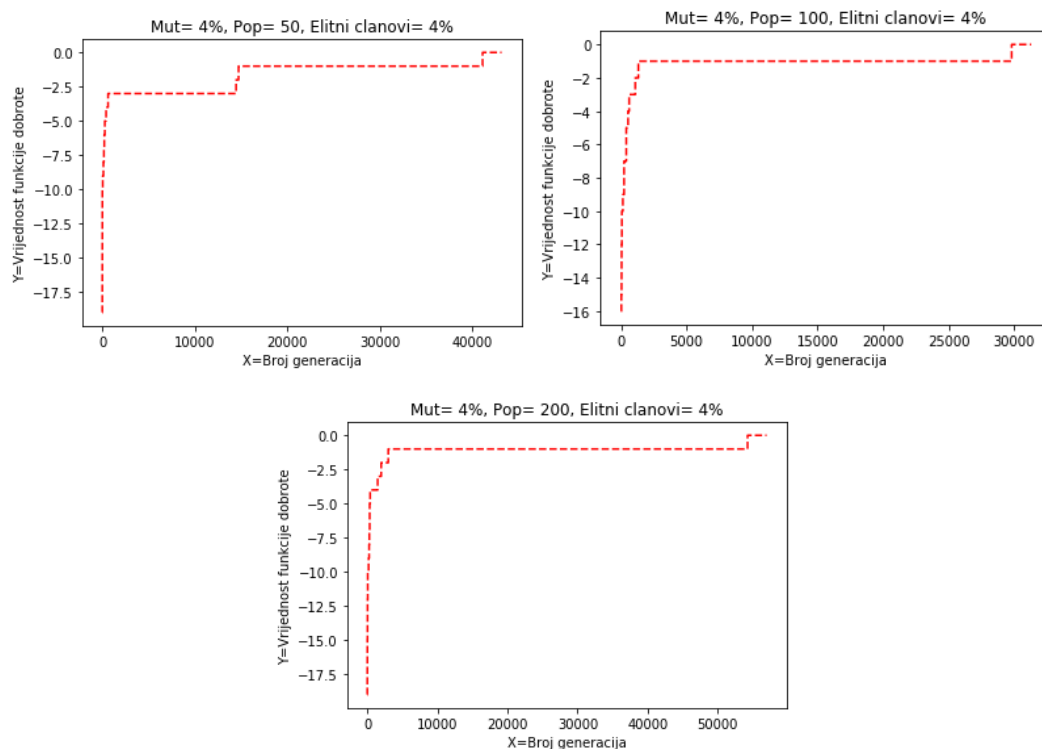
Tablica 3.6.2. Utjecaj broja populacija za N=24



Slika 3.6.2. Fitness funkcija u ovisnosti o broju populacija za N=24

Veličina ploče N	48		
Broj elitnih članova	4		
Broj generacija	60000		
Mutacija	4		
Populacija	50	100	200
Broj generacija rješenja	2489, 2818, 20185, 35781, 41133	1510, 1831, 1877, 2331, 29816	3874, 25823, 30725, 40803, 54264
Prosječan broj generacija rješenja	20482	7473	31098
Broj neuspjelih	1	1	4

Tablica 3.6.3. Utjecaj broja populacija za N=48



Slika 3.6.3. Fitness funkcija u ovisnosti o broju populacija za N=48

4. ZAKLJUČAK

Nakon svih 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 se vidjeti da povećanjem broja populacije ubrzavamo proces pronalaska rješenja, ali je bilo potrebno povećati i broj generacija. 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).