

Optimizacija parametara napada na GPS sustav uz pomoć evolucijskih algoritama

Močinić, Paula

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:577519>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-28**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1528

**OPTIMIZACIJA PARAMETARA NAPADA NA GPS SUSTAV UZ
POMOĆ EVOLUCIJSKIH ALGORITAMA**

Paula Močinić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1528

**OPTIMIZACIJA PARAMETARA NAPADA NA GPS SUSTAV UZ
POMOĆ EVOLUCIJSKIH ALGORITAMA**

Paula Močinić

Zagreb, lipanj 2024.

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1528

Pristupnica: **Paula Močinić (0036543385)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Domagoj Jakobović

Zadatak: **Optimizacija parametara napada na GPS sustav uz pomoć evolucijskih algoritama**

Opis zadatka:

Opisati mehanizam rada GPS sustava pozicioniranja i prikazati ga u obliku optimizacijskog problema. Navesti uobičajene postupke rješavanja problema GPS pozicioniranja. Proučiti mogućnosti napada na sustav GPS te uvjete pod kojima su napadi mogući s obzirom na broj prijemnika i mogućnosti napadača. Ostvariti programski sustav za optimizaciju parametara napada temeljen na evolucijskim algoritmima. Ispitati učinkovitost programskog rješenja na različitim scenarijima napada s obzirom na vremensku složenost i mogućnosti napadača. Radu priložiti izvorne tekstove programa, dobivene rezultate uz potrebna objašnjenja i korištenu literaturu.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod	3
2. Globalni pozicijski sustav	4
2.1. Način rada	4
2.2. Sigurnosni izazovi	5
2.3. Opis problema i teorijsko rješenje	5
2.3.1. Teorijsko rješenje	6
3. Evolucijski algoritmi	8
3.1. Selekcija	9
3.2. Križanje	10
3.3. Mutacija	10
4. Programska implementacija	11
4.1. Računanje pozicije prijemnika	11
4.2. Lažiranje jedne lokacije	11
4.3. Lažiranje dvije lokacije	12
4.4. Korištene kombinacije operatora	13
4.4.1. Selekcija	14
4.4.2. Križanje	14
4.4.3. Mutacija	15
4.4.4. Lokalna pretraga	15
5. Rezultati	16
5.1. Postavljanje parametara	16
5.1.1. Jedna lokacija	16

5.1.2. Dvije lokacije	18
5.2. Lažiranje jedne lokacije	20
5.3. Lažiranje dvije lokacije	21
6. Zaključak	25
Literatura	26
Sažetak	27
Abstract	28

1. Uvod

Globalni pozicijski sustavi (GPS) postali su neizostavan dio suvremenog društva, omogućujući precizno određivanje položaja na Zemlji koristeći satelitske signale. Međutim, s porastom ovisnosti o GPS sustavima, povećava se i svjesnost o njihovoj ranjivosti na različite oblike napada.

GPS sustav pozicioniranja temelji se na slanju signala iz svemira prema Zemlji putem mreže satelita u orbiti. Prijemnici na Zemlji koriste ove signale za određivanje svoje geografske pozicije. Kako bi se osigurala točnost pozicioniranja, GPS sustav koristi složen sustav satelita i matematičkih algoritama za obradu signala.

U ovom radu, pristupamo GPS pozicioniranju kao optimizacijskom problemu u kojem je cilj odrediti geografske koordinate prijemnika na temelju primljenih satelitskih signala. Također, istražujemo mogućnosti napada na GPS sustav, uključujući scenarije u kojima se napadi provode s različitim brojem prijemnika i različitim kapacitetima napadača.

Dodatno, razvijamo programski sustav za optimizaciju parametara napada temeljen na evolucijskim algoritmima. Koristimo ovaj sustav kako bismo ispitivali učinkovitost različitih scenarija napada s obzirom na vremensku složenost i mogućnosti napadača.

U nastavku ovog rada, detaljno ćemo analizirati mehanizam rada GPS sustava, opisati postupke rješavanja problema GPS pozicioniranja te istražiti mogućnosti napada na sustav. Također, pružit ćemo pregled razvijenog programskog rješenja i analizirati rezultate eksperimenata provedenih na različitim scenarijima napada.

2. Globalni pozicijski sustav

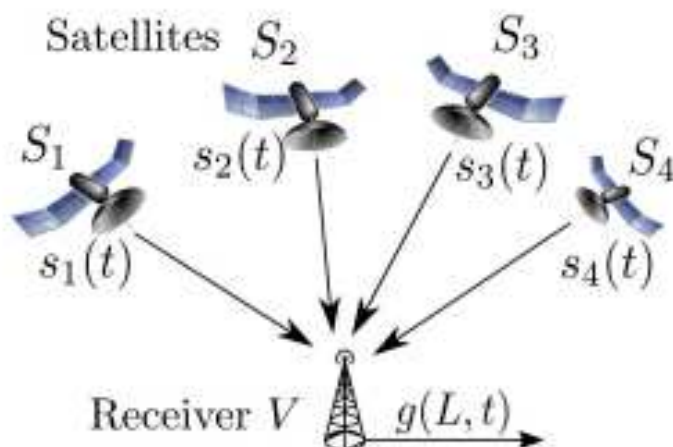
Globalni pozicijski sustav (GPS) predstavlja ključnu tehnologiju u današnjem digitalnom dobu. Originalno je razvijen za potrebe američke vojske, no postao je neizostavan dio brojnih civilnih aplikacija jer omogućuje precizno određivanje lokacije na Zemlji pomoću satelitskih signala.

2.1. Način rada

GPS se sastoji od 3 segmenta koji zajedno pružaju informacije o lokaciji [1]:

1. Sateliti - Oko Zemlje orbitira mreža satelita pri čemu svaki satelit emitira signale koji sadrže informacije o vremenu i položaju satelita.
2. Nadzor na tlu - kontrolni segment koji prati i upravlja satelitima u svemiru
3. Korisnička oprema (GPS prijemnici i odašiljači) - primaju signale od satelita i na temelju vremena dolaska signala od različitih satelita te lokacijama satelita, izračunavaju vlastitu geografsku lokaciju.

Kako bi prijemnik ispravno izračunao svoju lokaciju mora uzeti u obzir 3 dimenzije: dužinu, širinu i visinu. Tri satelita mu omogućuju određivanje položaja na dvodimenzionalnoj površini, dok četvrti satelit pruža informacije o visini. Veći broj satelita također smanjuje mogućnost pogreške i povećava otpornost GPS-a na ometanje signala



Slika 2.1. Prijemnik prima signale s četiri satelita te na temelju njih izračunava svoju poziciju. Slika preuzeta iz [2]

2.2. Sigurnosni izazovi

Postoji nekoliko načina na koje se GPS signali mogu ometati. Prirodne pojave poput solarnih oluja mogu privremeno prekinuti ili degradirati signale, dok je GPS oprema, kao i svaka tehnologija, podložna kvarovima. Takvi uzroci su očekivani te predstavljaju maleni rizik za sigurnost. S druge strane ometanje GPS-a je puno ozbiljnija pojava pri kojoj se GPS prijemnik zasićuje nepoznatim signalima kako bi se onemogućila njegova upotreba. Lažiranje GPS-a je još jedan način namjernog ometanja signala pri čemu napadač šalje lažne signale koje će GPS prijemnik dekodirati i smjestiti se na pogrešnu poziciju [3].

2.3. Opis problema i teorijsko rješenje

Problem koji se razmatra u ovom radu odnosi se na GPS spoofing. Ovaj problem predstavlja veliki sigurnosni izazov, posebno u kontekstu kritičnih sustava poput zrakoplovstva, pomorstva i vojnih aplikacija. GPS spoofing može dovesti do narušavanja funkcionalnosti navigacijskih sustava, što može rezultirati potencijalno katastrofalnim incidentima.

Detaljnije, fokusiramo se na optimizaciju parametara za GPS spoofing korištenjem evolucijskih algoritama, opisujemo prirodu GPS spoofing napada i pružamo teorijski okvir za njihovo otkrivanje i ublažavanje. Cilj je identificirati optimalne parametre koji će maksimizirati učinkovitost napada, istovremeno minimizirajući detekciju od strane

obrambenih mehanizama.

2.3.1. Teorijsko rješenje

GPS spoofing napadi mogu se općenito klasificirati u dvije kategorije: jednostavni spoofing i sofisticirani spoofing. Jednostavni spoofing uključuje prijenos lažnog GPS signala kako bi se prevario prijemnik. Nasuprot tome, sofisticirani spoofing zahtijeva od napadača da sinkronizira lažni signal s legitimnim signalom, što otežava otkrivanje i ublažavanje napada.

Napad uključuje napadača koji prenosi lažni GPS signal koji oponaša legitimni signal. Primarni cilj je da GPS prijemnik žrtve izračuna pogrešnu poziciju ili vrijeme. Napad se može izvesti pomoću jedne prijenosne antene ili više antena kako bi se povećala učinkovitost i složenost napada.

Svaki od satelita nalazi se na poznatoj lokaciji $L_i^s \in \mathbb{R}^3$ te ima savršeno sinkronizirani sat t^s . Prilikom slanja signala $s_i(t)$ satelit također šalje i vremensku oznaku trenutka slanja signala. Prijemnik V_j koji se nalazi na lokaciji $L \in \mathbb{R}^3$ prima signale satelita te na temelju njih može izračunati udaljenost njega i satelita:

$$d_i = |L_i^s - L| \quad (2.1)$$

Zbog toga što prijemnik nema visoko stabilni sat, njegov sat će imati vremenski odmak δ u odnosu na sat satelita :

$$t = t^s + \delta \quad (2.2)$$

Zbog tog odmaka prijemnik više ne računa pravu udaljenost već pseudoudaljenost R_i :

$$R_i = d_i + c \cdot \delta \quad (2.3)$$

Ako se u obzir uzmu signali sa četiri satelita onda prijemnik može izračunati svoju poziciju i odmak ispravno iako nema visoko precizni sat.

Imajući na umu da je $L_i^s = (x_i^s, y_i^s, z_i^s)$, $L = (x, y, z)$ i $\Delta = c \cdot \delta$, formula 2.3 može se

pretvoriti u :

$$(x - x_i^s)^2 + (y - y_i^s)^2 + (z - z_i^s)^2 = (R_i - \Delta)^2 \quad \forall S_i \quad (2.4)$$

kojom za svaki od satelita S_i dobijemo kuglu u čijem je središtu satelit na poziciji L_i .

Pod pretpostavkom da nema stvarnih satelitskih signala te da je napadač sveznajući možemo izvesti formulu za izračun pseudoudaljenosti koje prijemnik računa :

$$R_{ij}^A = |P_j - P_i^A| + c \cdot \delta_i^A \quad (2.5)$$

gdje je P_i fizička lokacija napadača, δ vremenski odmak slanja signala i P_j fizička lokacija prijemnika.

Kako bi prijemnik izračunao stvarnu lokaciju na temelju pseudoudaljenosti on mora riješiti sustav jednažbi:

$$|L_j' - L_i^A| = R_{ij}' - \Delta_j' \quad (2.6)$$

gdje L_i^A predstavlja koordinate svakog od satelita S_i koje je prijemnik dobio iz signala, R_{ij}' je pseudoudaljenost do satelita S_i dok je $\Delta_j' = \delta_j' \cdot c$

Kako bi napadač naveo prijemnika da misli da se nalazi na određenoj lokaciji on mora poslati svaki od svojih signala s_i tako da prijemnik rješavanjem jednažbe iznad dobije željenu lokaciju L_i' i odmak δ_j' . Zato mora vrijediti da je $R_{ij}^A = R_{ij}'$ odnosno :

$$|P_j - P_i^A| + \Delta_i^A = |L_j' - L_i^A| + \Delta_j' \quad (2.7)$$

3. Evolucijski algoritmi

Evolucijski algoritmi su metoda za rješavanje složenih problema koja koristi principe prirodne evolucije. Ovi algoritmi inspirirani su procesima koji se odvijaju u prirodi, poput selekcije, križanja i mutacije, kako bi iterativno poboljšali skup potencijalnih rješenja i postigli optimalna ili blizu optimalna rješenja [4].

Dobrota se u evolucijskim algoritmima koristi za ocjenjivanje kvalitete rješenja. Funkcija dobrote mjeri koliko je određeno rješenje dobro u odnosu na postavljeni cilj ili zadatak odnosno kvantificira uspješnost pojedine jedinke u populaciji, omogućavajući algoritmu da razlikuje bolje jedinke od lošijih.

Algorithm 1: Genetski algoritam

Ulaz: Početna populacija p (veličina = vel_pop)

Izlaz: Najbolje rješenje S^*

```
1  $P \leftarrow stvori\_pocetnu\_populaciju(vel\_pop)$  ;
2  $evaluiraj(P)$  ;
3 while uvjet zaustavljanja nije zadovoljen do
4    $nova\_populacija\ P' = \emptyset$  ;
5   while  $P' < vel\_pop$  do
6     odaberi  $r1, r2$  iz  $P$  ;
7      $\{d1, d2\} = krizaj(r1, r2)$  ;
8     mutiraj  $d1$  ;
9     mutiraj  $d2$  ;
10    dodaj  $d1$  i  $d2$  u  $P'$  ;
11  end
12   $P = P'$  ;
13   $evaluiraj(P)$  ;
14 end
15  $S^* \leftarrow$  Najbolje rješenje iz  $P$  ;
16 return  $S^*$ 
```

3.1. Selekcija

Na početku se populacija obično inicijalizira nasumično, te tijekom svake iteracije (generacije) algoritam primijenjuje selekciju kako bi odabrao najbolja rješenja koja će koristiti za stvaranje nove generacije. Selekcija osigurava da se jedinke s boljim karakteristikama češće odabiru za križanje i reprodukciju, čime se povećava vjerojatnost da će se pozitivne osobine prenijeti na potomstvo. Postoji nekoliko metoda selekcije koje se često koriste:

- **Turnirska selekcija:** Skupina jedinki se nasumično odabire iz populacije, a najbolja jedinka iz tog skupa se odabire za reprodukciju.
- **Ruletska selekcija:** Svaka jedinka ima vjerojatnost odabira proporcionalnu svom fitnessu. Jedinke s boljim fitnessom imaju veću šansu biti odabrane, ali sve jedinke imaju neku šansu, što omogućava održavanje genetske raznolikosti.

- **Rang-selekcija:** Jedinke se rangiraju prema svom fitnessu, a vjerojatnosti odabira se dodjeljuju prema tom rangu čime se smanjuje rizik dominacije nekoliko jedinki u populaciji.

3.2. Križanje

Križanje, ili rekombinacija, kombinira dijelove genetskog materijala roditelja kako bi stvorilo nove jedinke u populaciji. Postoji nekoliko uobičajenih metoda križanja:

- **Jednostruko križanje (Single-point crossover):** Nasumično se odabire točka križanja na genetskom kodu roditelja. Genetski materijal se razmjenjuje između roditelja na toj točki, stvarajući dva nova potomka.
- **Dvostruko križanje (Two-point crossover):** Dvije točke križanja se nasumično odabiru na genetskom kodu roditelja što omogućava veću raznolikost potomaka jer se razmjenjuje više segmenata genetskog materijala.
- **Uniformno križanje (Uniform crossover):** Svaki gen u potomku se nasumično bira od jednog od roditelja, čime se omogućava veća genetska raznolikost.

3.3. Mutacija

Mutacija uvodi slučajne promjene u genetski kod jedinki kako bi se održala raznolikost u populaciji te kako bi se spriječilo konvergiranje na lokalne optimume. Postoji nekoliko metoda mutacije:

- **Bitna mutacija (Bit-flip mutation):** Nasumično odabrani bit u genetskom kodu se mijenja iz 0 u 1 ili obrnuto. Ova metoda je česta kod binarnih genetskih kodova.
- **Gaussova mutacija (Gaussian mutation):** Dodaje se nasumična vrijednost iz Gaussove distribucije svakom genu u kontinuiranom genetskom kodu, omogućavajući finije prilagodbe.
- **Inverzija (Inversion mutation):** Dio genetskog materijala se nasumično odabire i obrće. Ova metoda može drastično mijenjati jedinku.

4. Programska implementacija

4.1. Računanje pozicije prijemnika

izračun pozicije se temelji na idućem sustavu jednažbi :

$$\begin{aligned}(x - A_1)^2 + (y - B_1)^2 + (z - C_1)^2 - (c(t_1 - d))^2 &= 0 \\(x - A_2)^2 + (y - B_2)^2 + (z - C_2)^2 - (c(t_2 - d))^2 &= 0 \\(x - A_3)^2 + (y - B_3)^2 + (z - C_3)^2 - (c(t_3 - d))^2 &= 0 \\(x - A_4)^2 + (y - B_4)^2 + (z - C_4)^2 - (c(t_4 - d))^2 &= 0\end{aligned}\tag{4.1}$$

gdje su x, y i z koordinate prijemnika, A_i, B_i, C_i koordinate satelita S_i , d je razlika sata satelita i prijemnika te je t_i vrijeme potrebno da signal dođe od satelita S_i do prijemnika. Za rješavanje ovih jednažbi korištena je višedimenzionalna Newtonova metoda [5] koja iterira kroz uzastopne procjene dok ne konvergira do ispravnog rješenja, što obično zah-
tjeva manje od 20 iteracija.

4.2. Lažiranje jedne lokacije

Scenarij uključuje napadača s jednom antenom koji može slati signal do bilo koje loka-
cije te je pretpostavljeno da on zna gdje se nalazi prijemnik te može precizno upravljati
vremenom slanja signala.

Potrebni ulazni parametri su pozicije 4 satelita $S_A(x, y, z), S_B(x, y, z), S_C(x, y, z), S_D(x, y, z)$
te željena pozicija prijemnika $SP_i(x, y, z)$. Prvo se odredi udaljenost od najbližeg satelita
do željene pozicije prijemnika koja se računa kao euklidska udaljenost

$$\text{minSatDist} = \min(\text{distance}(S_A, SP), \text{distance}(S_B, SP), \text{distance}(S_C, SP), \text{distance}(S_D, SP))$$

te se na temelju toga izračuna vrijeme propagacije signala od najbližeg satelita do prijemnika

$$\text{minTime} = \frac{\text{minSatDist}}{c}$$

Genetski algoritam optimizira vremena propagacije ostalih signala (t_1, t_2, t_3) kako bi se minimizirala pogreška između stvarne i lažirane pozicije prijemnika. Algoritam koristi genotip s brojevima s pomičnom točkom velicine tri, gdje svako vrijeme propagacije predstavlja jedan gen.

Za svaki prijemnik izračunava se:

$$t_A = \text{minTime}$$

$$t_B = \text{minTime} + t1$$

$$t_C = \text{minTime} + t2$$

$$t_D = \text{minTime} + t3$$

Fitness funkcija izračunava pogrešku koristeći euklidsku udaljenost između željene i izračunate pozicije prijemnika na temelju simuliranih vremena propagacije:

$$\text{error} = \sqrt{(x_r - x_f)^2 + (y_r - y_f)^2 + (z_r - z_f)^2} \quad (4.2)$$

gdje su (x_r, y_r, z_r) koordinate pozicije koju je prijemnik izračunao koristeći simulirana vremena propagacije, a (x_f, y_f, z_f) lažne koordinate prijemnika. Zbog toga unutar implementacije ovog rada najbolja jedinka ima najmanji fitness.

4.3. Lažiranje dvije lokacije

U ovom scenariju napadač pokušava lažirati lokaciju dvaju prijemnika koji u svakom trenutku mogu provjeriti međusobnu udaljenost te na temelju toga znati radi li se o napadu ili ne [6]. Ovo postavlja dodatne zahtjeve na lažiranje kako se napad ne bi otkrio.

Potrebni ulazni parametri su pretpostavljene pozicije 4 satelita $S_A(x, y, z)$, $S_B(x, y, z)$, $S_C(x, y, z)$, $S_D(x, y, z)$ na kojima se nalaze antene napadača, stvarne pozicije 2 prijem-

nika $P_i(x, y, z)$, te željene pozicije 2 prijemnika $SP_i(x, y, z)$. Genetski algoritam koristi se za optimizaciju razmaka vremena slanja signala za svaki od četiri satelita (t_1, t_2, t_3, t_4). Udaljenost satelita do stvarnih pozicija prijemnika $satDist_i[A, B, C, D]$ računa se kao euklidska udaljenost.

Zatim se za svaki prijemnik i izračuna:

$$\begin{aligned} t_A &= \frac{satDist_i[A]}{c} + t1 \\ t_B &= \frac{satDist_i[B]}{c} + t2 \\ t_C &= \frac{satDist_i[C]}{c} + t3 \\ t_D &= \frac{satDist_i[D]}{c} + t4 \end{aligned}$$

Ponovo se koristi euklidska udaljenost za izračun pogreške, no u ovom slučaju fitness funkcija zbraja pogrešku za oba prijemnika. Dodatno, provjerava se i razlika između međusobnih udaljenosti prijemnika na lažnim pozicijama i njihovih stvarnih pozicija. Ako je ta razlika različita od nule, dodaje se odgovarajuća kazna fitness vrijednosti jedinke.

4.4. Korištene kombinacije operatora

Korištene su različite vrste operatora kako bi se analiziralo njihovo ponašanje, usporedilo djelovanje i ustanovilo kako svaki od njih djeluje na optimizaciju. Na taj način se može otkriti koje kombinacije daju najbolje rezultate za problem ovog rada.

Oznaka	Selekcija	Križanje	Mutacija
1	Turnirska	S jednom točkom	Gausova
2	Turnirska	S jednom točkom	Uniformna
3	Turnirska	Uniformno	Gausova
4	Turnirska	Uniformno	Uniformna
5	Proporcionalna	S jednom točkom	Gausova
6	Proporcionalna	S jednom točkom	Uniformna
7	Proporcionalna	Uniformno	Gausova
8	Proporcionalna	Uniformno	Uniformna

Tablica 4.1. Kombinacije operatora korištene u implementaciji genetskog algoritma

U tablici (4.1.) su prikazane kombinacije operatora korištene u genetskom algoritmu za optimizaciju. Svaka kombinacija je označena brojem za jednostavnije referenciranje.

4.4.1. Selekcija

Kao metode selekcije odabrane su turnirska selekcija i modificirana proporcionalna selekcija.

U turnirskoj selekciji nasumično je odabran skup jedinki veličine tournamentSize koji je u ovoj implementaciji postavljen na vrijednost tri, te se zatim kao roditelj odabire ona s najmanjim fitnessom. Postupak se ponavlja sve dok se ne odaberu svi roditelji potrebni za novu generaciju.

Modificirana proporcionalna selekcija odabire roditelje proporcionalno njihovom inverznom fitnessu. Dakle, jedinke s manjim fitnessom imaju veću vjerojatnost da budu odabrane kao roditelji

4.4.2. Križanje

Korištene metode križanja su križanje s jednom točkom prekida i uniformno križanje.

Križanje s jednom točkom prekida nasumično odabire jednu točku te dijeli roditelje na toj točki. Zatim se dijelovi između 2 roditelja mijenjaju kako bi se stvorila djeca.

Uniformno križanje stvara djecu tako da nasumično odabere gene iz oba roditelja. Odabir gena se vrši neovisno gdje svaki roditelj ima jednaku vjerojatnost.

4.4.3. Mutacija

Implementirane vrste mutacije su gaussova i uniformna.

Gaussova mutacija mijenja vrijednost slučajno odabranog gena dodavanjem nasumične vrijednosti prema Gaussovoj distribuciji, što omogućuje fine promjene vrijednosti gena.

Uniformna mutacija mijenja vrijednost slučajno odabranog gena na nasumičnu vrijednost iz uniformne distribucije unutar određenog raspona, što doprinosi širem istraživanju prostora rješenja.

4.4.4. Lokalna pretraga

Korištena je i lokalna pretraga kako bi se pokušali poboljšati dobiveni rezultati. Ona iterativno mijenja jedinku malim nasumičnim izmjenama, tražeći bolje fitness vrijednosti

Algorithm 2: Lokalna pretraga

Ulaz: Početna jedinka S , funkcija fitnessa f , veličina izmjene ϵ , maksimalni broj

iteracija max_iter

Izlaz: Poboljšana jedinka S^*

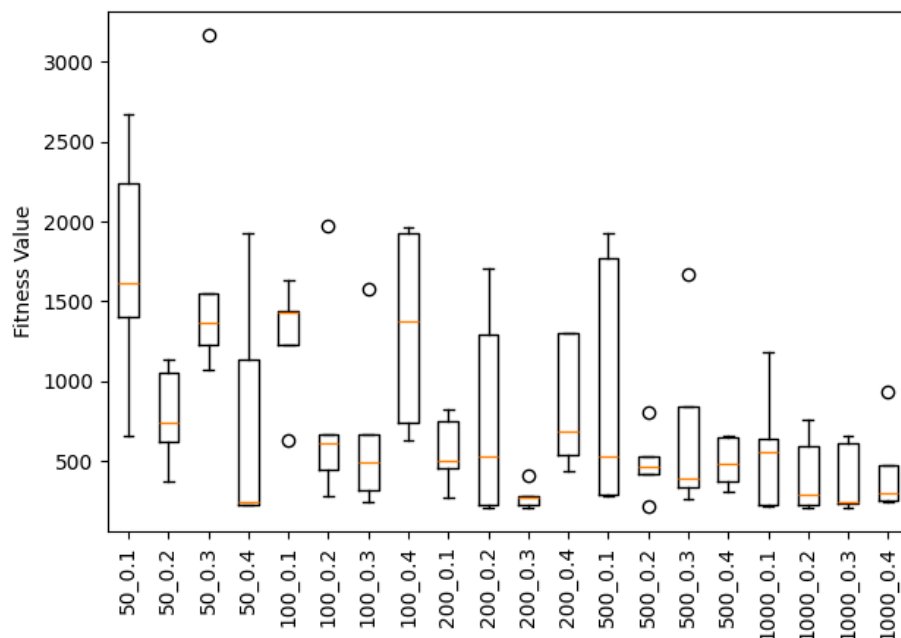
```
1  $S^* = S$  ;
2  $najbolji\_fitnes = f(S^*)$  ;
3 for  $i = 1$  to  $max\_iter$  do
4    $nova\_jedinka = S^* + \mathcal{N}(0, \epsilon)$  ;
5    $nova\_jedinka = osiguraj\_granice(nova\_jedinka, 0, 1)$  ;
6    $novi\_fitnes = f(nova\_jedinka)$  ;
7   if  $novi\_fitnes < najbolji\_fitnes$  then
8      $S^* = nova\_jedinka$  ;
9      $najbolji\_fitnes = novi\_fitnes$  ;
10  end
11 end
12 return  $S^*$ 
```

5. Rezultati

5.1. Postavljanje parametara

5.1.1. Jedna lokacija

Za pronalazak optimalnih vrijednosti za algoritam, testirane su različite kombinacije parametara na lažiranju jedne lokacije. Kao osnova za postavljanje parametara korištena je turnirska selekcija i križanje na jednoj točki. Istraživane su vrijednosti stope mutacije 0.1, 0.2, 0.3 i 0.4 te veličine populacije od 25, 50, 100, 200, 500 i 1000 kako bi se utvrdilo koje vrijednosti parametara daju najbolje rezultate. Svaka kombinacija parametara je ispitana 10 puta radi bolje analize rezultata.

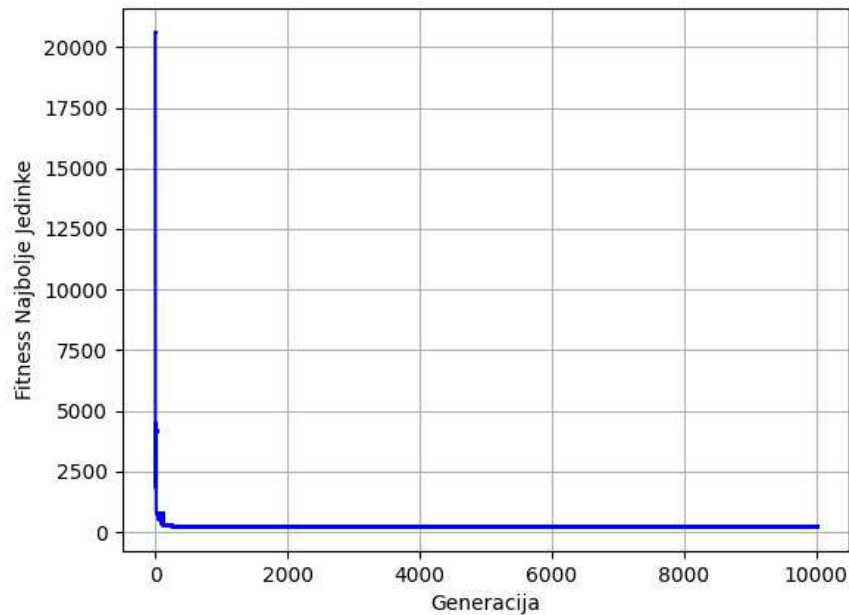


Slika 5.1. Rezultati algoritma s turnirskom selekcijom i križanjem u jednoj točki pri vrijednosti mutacije 0.1, 0.2, 0.3 i 0.4 te veličine populacije od 25, 50, 100, 200, 500 i 1000.

Kao što se može vidjeti na slici (5.1.), rješenje je najbolje za kombinaciju stope muta-

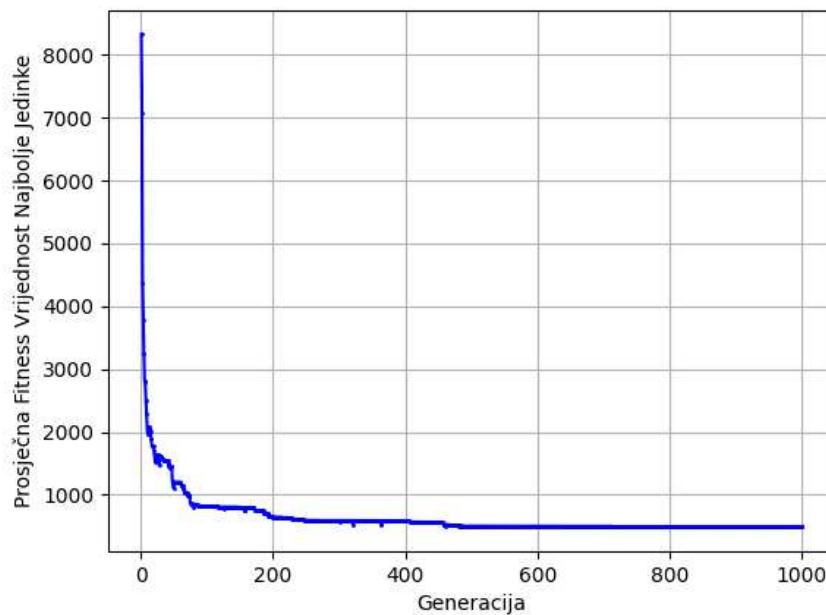
cije 0.3 i veličine populacije od 200 jedinki. U nastavku istraživanja korištene su upravo te vrijednosti mutacije i veličine populacije.

Zatim se nastavilo s pronalaskom broja generacija u kojem će algoritam naći rješenje s najmanjom mogućom pogreškom, idealno blizu nule. Za broj generacija, testiranja su provedena sve do 10,000 generacija kako bi se osiguralo da algoritam ima dovoljno vremena za konvergenciju prema optimalnom rješenju.



Slika 5.2. Ovisnost fitnessa o broju generacija koristeći turnirsku selekciju i križanje na jednoj točki

Na slici (5.2.) vidimo da fitness vrijednost značajno opada već u ranim generacijama te se stabilizira već nakon 1000 iteracija.

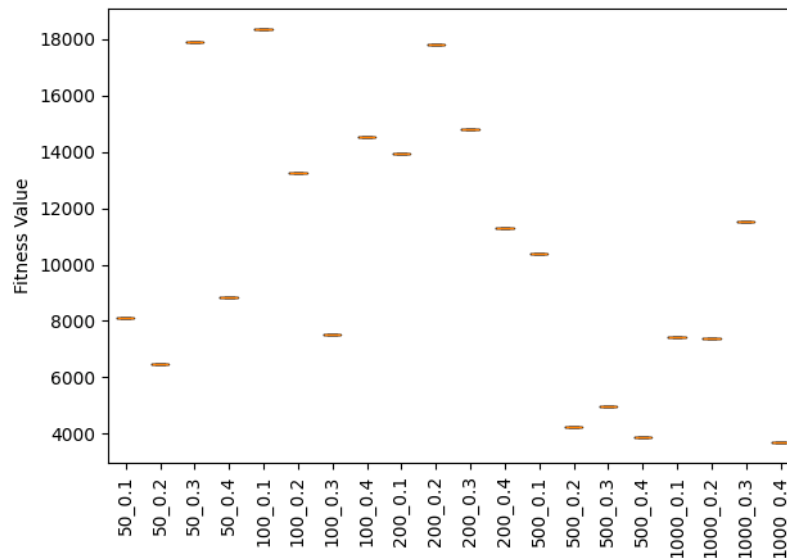


Slika 5.3. Ovisnost fitnessa o broju generacija koristeći turnirsku selekciju i križanje na jednoj točki

Slika (5.3.) prikazuje isti algoritam kao i slika (5.2.) s ograničenjem broja generacija na 1000 umjesto na 10,000 kako bi se mogao utvrditi minimalni broj generacija potreban za postizanje zadovoljavajućeg rješenja. Jasno se može primijetiti da fitness vrijednost počinje stagnirati već nakon 500 generacija. Stoga će se 500 generacija koristiti kao optimalan broj generacija za postizanje zadovoljavajućeg rješenja.

5.1.2. Dvije lokacije

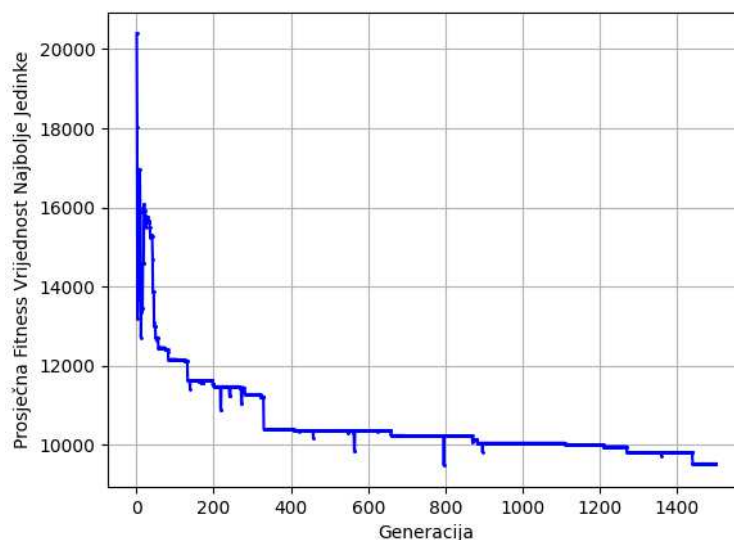
Također su traženi optimalni parametri i za lažiranje dvije lokacije. Kao i kod lažiranja jedne lokacije, korištena je turnirska selekcija i križanje u jednoj točki. Isprobane su stope mutacije 0.1, 0.2, 0.3 i 0.4 te veličine populacije 50, 100, 200, 500 i 1000.



Slika 5.4. Rezultati algoritma s turnirskom selekcijom i križanjem u jednoj točki pri vrijednosti mutacije 0.1, 0.2, 0.3 i 0.4 te veličine populacije od 50, 100, 200, 500 i 1000.

Na slici (5.4.) vidljivo je da algoritam postiže najmanju vrijednost fitnessa korištenjem stope mutacije 0.4 i veličine populacije 1000.

Zatim je istraživao broj generacija potrebnih da se postigne zadovoljavajuće rješenje. Korištena je stopa mutacije 0.4 i veličina populacije 1000. Ograničenje na broj generacija postavljeno je na 1500 zbog dužine izvođenja.



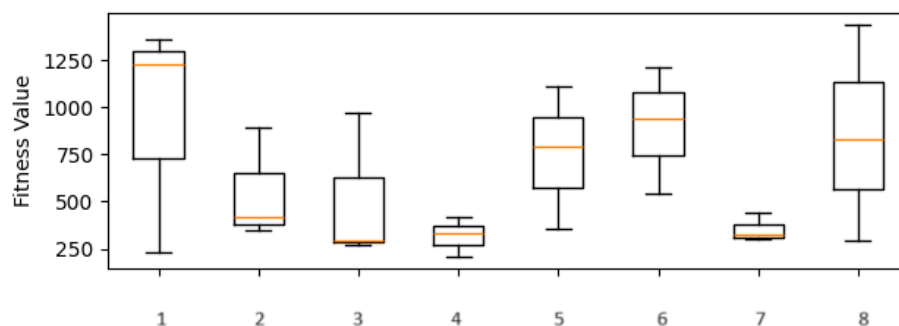
Slika 5.5. Rezultati algoritma s turnirskom selekcijom i uniformnim križanjem pri vrijednosti mutacije 0.4 te veličine populacije 1000.

Na slici (5.5.) je vidljivo kako fitness vrijednost opada te se nazire stagnacija pri kraju

grafa no ne može se reći sa sigurnošću da je vrijednost prestala opadati nakon 1500 generacija.

5.2. Lažiranje jedne lokacije

Na slici (5.6.) prikazana je vrijednost fitnessa za različite kombinacije operatora selekcije, križanja i mutacije prilikom lažiranja jedne lokacije. Svaka kombinacija operatora testirana je koristeći optimalne parametre stope mutacije od 0.3, veličine populacije od 200 jedinki i broja generacija od 500 te je za svaku kombinaciju test ponovljen 10 puta.

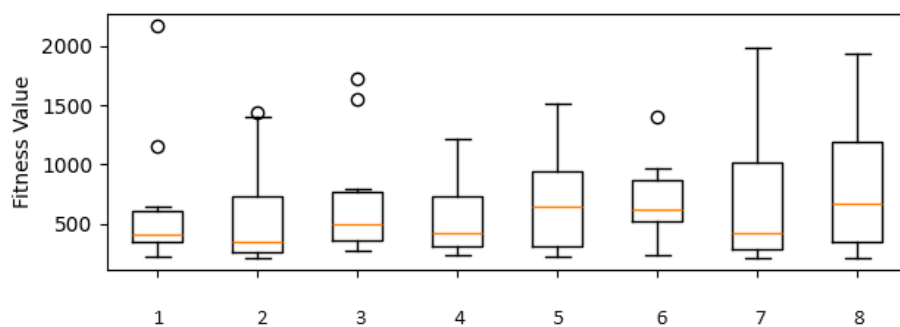


Slika 5.6. Rezultati lažiranja jedna lokacije različitim kombinacijama operatora

Uočavamo da sve kombinacije operatora postižu slična rješenja (slika 5.6.), no ipak kombinacije s turnirskom selekcijom pokazuju bolja rješenja od proporcionalne selekcije.

Kao najbolje kombinacije ističu se ona s turnirskom selekcijom, uniformnim križanjem i uniformnom mutacijom (br.4) te ona s proporcionalnom selekcijom, uniformnim križanjem te gausovom mutacijom (br.7). Primjećujemo da ove kombinacije pokazuju malu oscilaciju u rezultatima za razliku od ostalih kombinacija operatora.

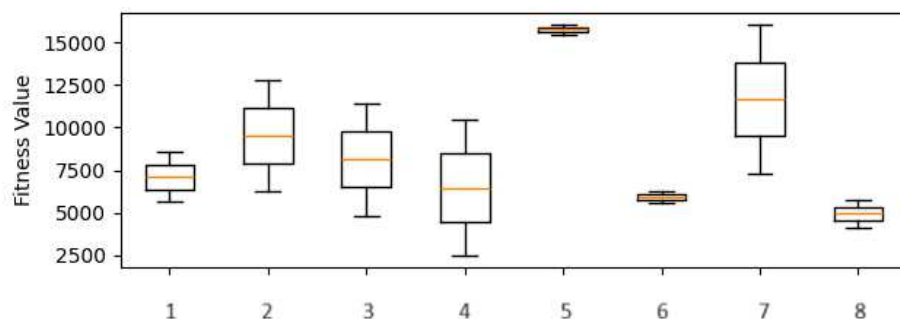
Rezultati korištenja lokalne pretrage prilikom lažiranja jedne lokacije (slika 5.7.) pokazuju poboljšanje svih kombinacija operatora.



Slika 5.7. Rezultati lažiranja jedna lokacije različitim kombinacijama operatora i lokalnom pretragom

5.3. Lažiranje dvije lokacije

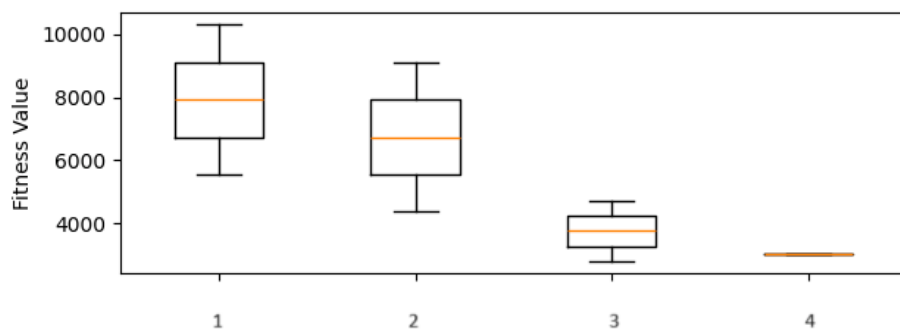
Na slici (5.8.) prikazana je vrijednost fitnessa za različite kombinacije operatora selekcije, križanja i mutacije prilikom lažiranja dvije lokacije. Svaka kombinacija operatora testirana je koristeći optimalne parametre stope mutacije od 0.4, veličine populacije od 1000 jedinki i broja generacija od 1500 te je za svaku kombinaciju test ponovljen 4 puta.



Slika 5.8. Rezultati lažiranja dvije lokacije različitim kombinacijama operatora

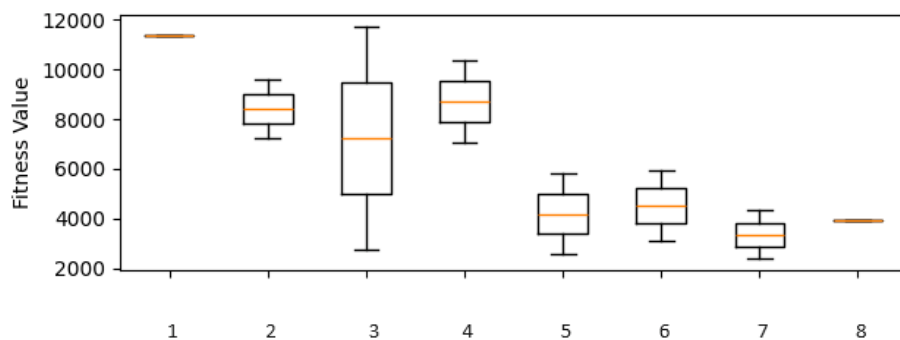
Jedina kombinacija operatora koja postiže zadovoljavajuće rješenje je ona s turnirskom selekcijom, uniformnim križanjem te uniformnom mutacijom. Sve ostale kombinacije su bile manje uspješne te bi njihovim korištenjem napad vjerojatno bio otkriven. Ispitivanje je provedeno ponovo te su ispitane samo kombinacije s turnirskom selekcijom pošto su se one imale bolja rješenja.

Na slici (5.9.) uočljivo je da kombinacije s uniformnim križanjem daju bolje rezultate od onih u kojima se koristi križanje s jednom točkom.



Slika 5.9. Rezultati lažiranja dvije lokacije turnirskom selekcijom

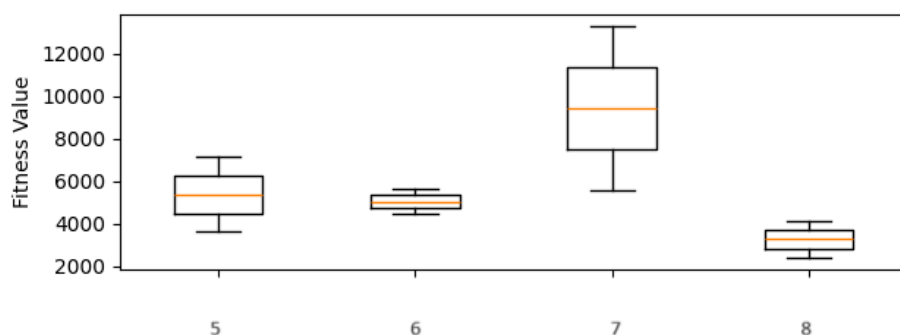
Isti parametri su ponovo testirani koristeći i lokalnu pretragu.



Slika 5.10. Rezultati lažiranja dvije lokacije različitim kombinacijama operatora i lokalnom pretragom

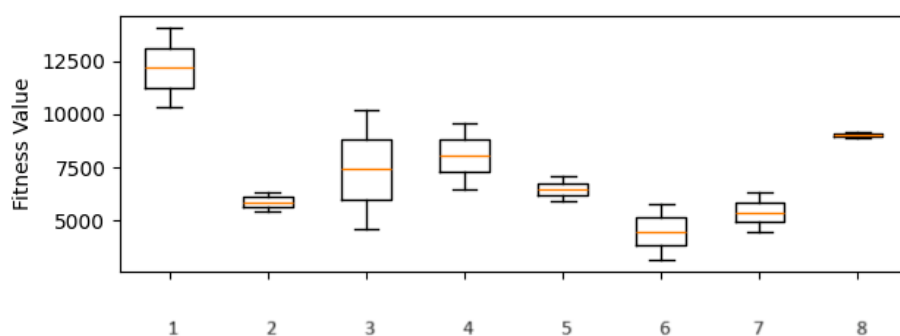
Možemo uočiti kako su sada kombinacije koje uključuju proporcionalnu selekciju pokazale bolje rezultate od onih s turnirskom selekcijom, za razliku od testiranja gdje nije bila korištena lokalna pretraga. Također vidimo da većina kombinacija postiže bolje rezultate (imaju nižu fitness vrijednost) nego na slici (5.8.)

Detaljniji prikaz lažiranja dviju lokacije koristeći lokalnu pretragu i proporcionalnu selekciju prikazan je na slici (5.11.).



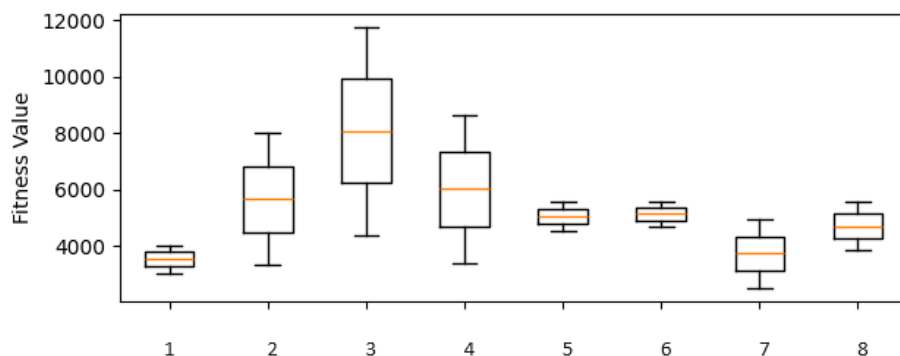
Slika 5.11. Rezultati lažiranja dvije lokacije proporcionalnom selekcijom i lokalnom pretragom

Operatori su isprobani i na veličini populacije od 1500 jedinki, mutaciji 0.4 te broju generacija 1500. Pretpostavljeno je da će veća populacija dati bolje rezultate i omogućiti daljnu pretragu prostora rješenja.



Slika 5.12. Rezultati lažiranja dvije lokacije različitim kombinacijama parametara

Na slici (5.12.) vidimo da najbolje rješenje daje kombinacija s proporcionalnom selekcijom, križanjem s jednom točkom te uniformnom mutacijom.



Slika 5.13. Rezultati lažiranja dvije lokacije različitim kombinacijama parametara i lokalnom pretragom

Koristeći lokalnu pretragu, na slici (5.13.) uočavamo da su se rezultati većine kombinacija operatora poboljšali u odnosu na sliku (5.12.). Također, drukčija kombinacija istaknula se kao najbolja: proporcionalna selekcija, uniformno križanje i gausova mutacija.

Svi rezultati lažiranja dvije lokacije podložni su oscilacijama jer je pojedina kombinacija operatora za specifični slučaj pokrenuta po četiri puta zbog duljine izvođenja programa. Zbog toga se rezultati za istu kombinaciju operatora na slici (5.10.) i (5.11.) međusobno razlikuju. Bilo bi potrebno ispitati slučajeve puno više puta kako bi sa sigurnošću mogli reći koje kombinacije daju bolje rezultate od drugih, no to bi zahtjevalo više vremena. Nasuprot tome, lažiranje jedne lokacije bilo je moguće ispitati deset puta za svaku kombinaciju jer su optimalni broj jedinki u generaciji i broj generacija bili manji.

6. Zaključak

U ovom radu istraženi su različiti pristupi lažiranju GPS lokacija koristeći genetske algoritme. Fokus je bio na analizi kako kombinacije selekcije, križanja i mutacije utječu na efikasnost algoritma.

Proučavano je lažiranje jedne i dviju lokacija. Prilikom lažiranja jedne lokacije algoritam mora poslati signal s vremenskim odmacima koje simuliraju stvarnu lokaciju satelita. Za lažiranje dviju lokacija, algoritam mora koristiti složenije metode kako bi prevario žrtvu.

Opisan je evolucijski algoritam čiji su ključni dijelovi selekcija, križanje i mutacija. Koristile su se turnirska i proporcionalna selekcija, križanje s jednom točkom i uniformno križanje te gaussova i uniformna mutacija.

Implementirani su scenariji lažiranja jedne lokacije te lažiranja dvije lokacije s napadačima postavljenim na pozicijama satelita. Ispitivane su sve kombinacije navedenih operatora s ciljem pronalaska najbolje. Pri lažiranju jedne lokacije sve kombinacije su postigle slična rješenja dok su kod lažiranja dviju lokacija kombinacije s uniformnim križanjem pokazivale bolja rješenja od križanja s jednom točkom. Za razliku od lažiranja jedne lokacije, lažiranje dviju lokacija rezultiralo je puno većom greškom. Dodatno, pokazalo se da povećanje broja jedinki u generaciji doprinosi većoj preciznosti u lažiranju dviju lokacija.

Implementirana je i lokalna pretraga s ciljem poboljšanja dobivenih rezultata. Došlo je do poboljšanja rezultata gotovo svih kombinacija operatora u oba scenarija. Pri lažiranju dvije lokacije uz upotrebu lokalne pretrage najbolje rezultate dale su kombinacije s proporcionalnom selekcijom. Povećavanje broja jedinki u generaciji s 1000 na 1500 rezultiralo je ujednačavanjem dobivenih rezultata među kombinacijama operatora.

Literatura

- [1] Geotab, “What is gps?”. [Mrežno]. Adresa: <https://www.geotab.com/blog/what-is-gps/>
- [2] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, i S. Čapkun, “On the requirements for successful gps spoofing attacks”. [Mrežno]. Adresa: <https://www.cs.ox.ac.uk/files/6489/gps.pdf>
- [3] Flightradar24, “Types of gps jamming”. [Mrežno]. Adresa: <https://www.flightradar24.com/blog/types-of-gps-jamming/>
- [4] M. Gupta, “What are evolutionary algorithms?”. [Mrežno]. Adresa: <https://medium.com/data-science-in-your-pocket/what-are-evolutionary-algorithms-fb642f72db34>
- [5] H. Arneja, A. Bender, S. Jugus, i T. Reid, “Solving the gps equations”. [Mrežno]. Adresa: <https://mason.gmu.edu/~treid5/Math447/GPSEquations/>
- [6] E. Zurich, “Gps spoofing”. [Mrežno]. Adresa: <https://www.zemris.fer.hr/~yeti/studenti/radovi/gps-spoofing.pdf>

Sažetak

GPS spoofing parameters optimization based on evolutionary algorithms

U ovom radu istražuje se optimizacija parametara napada na GPS sustav korištenjem evolucijskih algoritama. Fokus je na simulaciji GPS spoofinga, posebno na lažiranju jedne i dvije lokacije prijemnika. Korištena su dva scenarija: u prvom napadač lažira lokaciju jednog prijemnika, dok u drugom lažira lokaciju dva prijemnika, uz pretpostavku da se oni međusobno provjeravaju. Za svaki scenario, razvijen je genetski algoritam koji optimizira vrijeme slanja signala s četiri satelita kako bi se maksimizirala učinkovitost napada, minimizirajući detekciju. Testirane su različite kombinacije parametara kako bi se odredile optimalne vrijednosti za algoritam. Uočeno je da kod lažiranja dviju lokacija uniformno križanje pokazuje bolje rezultate od križanja s jednom točkom. Također povećavanje veličine generacija doprinosi preciznijem lažiranju dviju lokacija, dok se greška znatno povećava u usporedbi s lažiranjem jedne lokacije. Lokalna pretraga poboljšala je rezultata gotovo svih kombinacija operatora te je pri lažiranju dviju lokacija proporcionalna selekcija pružala bolje rezultate od turnirske.

Ključne riječi: Optimizacija; GPS; Spoofing; Genetski; Algoritam

Abstract

GPS spoofing parameters optimization based on evolutionary algorithms

This paper investigates the optimization of parameters for attacking the GPS system using evolutionary algorithms. The focus is on simulating GPS spoofing, particularly on falsifying the location of one or two receiver locations. Two scenarios are considered: in the first, the attacker falsifies the location of one receiver, while in the second, the attacker falsifies the location of two receivers, assuming they cross-validate each other. For each scenario, a genetic algorithm has been developed to optimize the timing of signal transmission from four satellites to maximize the attack efficiency while minimizing detection. Various combinations of parameters have been tested to determine the optimal values for the algorithm. In the case of spoofing two locations, uniform crossover yields better results compared to single-point crossover. Additionally, increasing the generation size contributes to more precise spoofing of two locations, although the error significantly increases compared to spoofing a single location. Local search improved the results of nearly all operator combinations, and in spoofing two locations, proportional selection outperformed tournament.

Keywords: Optimization, GPS, spoofing, genetic, algorithms