

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6393

**Optimizacija parametara GPS napada
temeljena na evolucijskim
algoritmima**

Ante Žužul

Zagreb, lipanj 2019.

Zagreb, 14. ožujka 2019.

ZAVRŠNI ZADATAK br. 6393

Pristupnik: **Ante Žužul (0036498016)**
Studij: Računarstvo
Modul: Računarska znanost

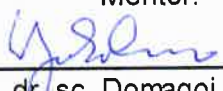
Zadatak: **Optimizacija parametara GPS napada temeljena na evolucijskim algoritmima**

Opis zadatka:

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

Zadatak uručen pristupniku: 15. ožujka 2019.
Rok za predaju rada: 14. lipnja 2019.

Mentor:



Prof. dr. sc. Domagoj Jakobović

Djelovođa:



Izv. prof. dr. sc. Tomislav Hrkać

Predsjednik odbora za
završni rad modula:



Doc. dr. sc. Marko Čupić

SADRŽAJ

1. Uvod	1
2. Globalni položajni sustav	2
2.1. Mehanizmi GPS-a	2
2.2. Opis problema	3
2.2.1. Oblikovanje pseudoudaljenosti	3
2.2.2. Lažiranje jedne lokacije	4
2.2.3. Lažiranje više lokacija	4
2.2.4. Moguća rješenja	5
3. Optimizacijski algoritmi	6
3.1. Evolucijski algoritmi	6
3.1.1. Steady State Tournament	6
3.1.2. Evolution Strategy	7
3.2. Imunološki algoritmi	7
3.2.1. Clonal Selection Algorithm	7
3.2.2. Immune Optimization Algorithm	8
4. Programsko ostvarenje	9
4.1. Računanje lokacije i vremenskog odmaka	9
4.2. Implementirani scenariji	9
4.2.1. Lažiranje jedne lokacije	9
4.2.2. Lažiranje više lokacija - pozicije napadača	10
4.2.3. Lažiranje više lokacija - pozicije satelita	11
5. Rezultati i diskusija	13
5.1. Optimizacija parametara algoritama	13
5.2. Ispitivanje algoritama	16
6. Zaključak	26

1. Uvod

Globalni položajni sustav (eng. Global Positioning System, GPS), osmišljen od strane američke vojske, postao je neizostavna komponenta mnogih korisničkih aplikacija. Za razliku od vojnih GPS signala, korisnički GPS signali nisu šifrirani. Unatoč tome, GPS lokacije koriste se u sigurnosno kritičnim aplikacijama kao što su navigacija vozila, praćenje vrijednih predmeta te raznim ostalim uslugama baziranim na lokacijama. Usluge GPS-a također uključuju i sinkronizaciju vremena, npr. vremenske oznake u sigurnosnim videozapisima, financijskim, telekomunikacijskim i računalnim mrežama.

Korisnici uvelike ovise o preciznosti i točnosti GPS lokacija i vremena. Upravo zbog toga, pokrenute su brojne istrage o sigurnosti samog sustava. Istraživanja su pokazala kako je civilni GPS osjetljiv na napade ometanjem i lažiranjem signala. Uspješno provedeni napadi lažiranjem signala pokazuju kako komercijalni prijemnici ne mogu otkriti takve napade. GPS prijemnici, koji se nalaze u dometu napadača, određuju lokaciju različitu od njihove stvarne fizičke lokacije.

Ideja rješavanja ovoga problema temelji se na korištenju više GPS prijemnika. Oni se mogu rasporediti u statičnoj, poznatoj formaciji. Protumjera je sljedeća. Ako GPS prijemnici mogu izmjenjivati svoje individualne GPS lokacije, tada mogu i provjeriti je li njihova formacija očuvana. U slučaju da izračunata GPS lokacija ne odgovara poznatoj formaciji, treba posumnjati na napad. Svaki dodatni prijemnik uvelike otežava lažiranje signala, pogotovo ako je moguće barem jedan prijemnik sakriti od napadača.

U ovom radu, ispituju se mogućnosti lažiranja GPS signala optimizacijom parametara GPS napada evolucijskim algoritmima.

2. Globalni položajni sustav

2.1. Mehanizmi GPS-a

Globalni položajni sustav koristi brojne satelitske odašiljače koji se nalaze na poznatim lokacijama $L_i^S \in \mathbb{R}^3$. Svaki odašiljač opremljen je savršeno sinkroniziranim satom t^S i odašilje pažljivo odabran navigacijski signal s_i koji propagira brzinom c .

Prijemnik se nalazi na lokaciji $L \in \mathbb{R}^3$ i koristeći jednu višesmjernu antenu prima signale svih satelita u doseg. Na temelju vremenskog kašnjenja primljenog signala s_i , prijemnik može zaključiti svoju udaljenost do pojedinog satelita

$$d_i = |L_i^S - L|. \quad (1)$$

S tri poznate udaljenosti d_i do poznatih lokacija L_i^S , prijemnik može iz tri izraza (1) izračunati svoju lokaciju. Međutim, visoko stabilni satovi su preskupi pa GPS prijemnici ne mogu sudjelovati u dvosmjernoj sinkronizaciji satova. Stoga će prijemnik imati vremenski odmak δ od stvarnog vremena: $t = t^S + \delta$. Prijemnik više ne zaključuje stvarne udaljenosti, nego pseudoudaljenosti

$$R_i = d_i + c \cdot \delta \quad (2)$$

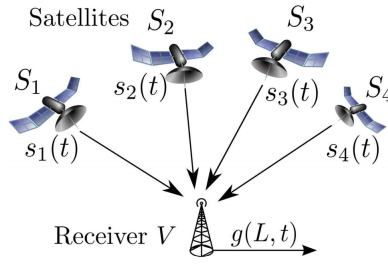
koje računa kao $(\text{trenutak_primanja_signala} - \text{trenutak_slanja_signala}) \cdot c$.

Vremenski odmak δ predstavlja četvrtu nepoznanicu. Pseudoudaljenosti do barem četiri odašiljača S_i (Slika 2.1) omogućavaju rješavanje sustava s četiri izraza (2) i dobivanje točne lokacije L i vremena δ bez potrebe za preciznim lokalnim satom.

Neka je $L_i^S = (x_i^S, y_i^S, z_i^S)$, $L = (x, y, z)$ i $\Delta = c \cdot \delta$, tada izraz (2) možemo zapisati kao

$$(x - x_i^S)^2 + (y - y_i^S)^2 + (z - z_i^S)^2 = (R_i - \Delta)^2 \quad \forall S_i. \quad (3)$$

Izraz za neki S_i geometrijski predstavlja kuglu u čijem se središtu nalazi satelit i na čijem se oplošju nalazi moguća lokacija prijemnika. Izraz (3) je preodređen za više od četiri satelita i u većini slučajeva nema jedinstveno rješenje zbog šuma podataka.



Slika 2.1: Prijemnik V prima signale $s_i(t)$ koje mu šalju sateliti. Iz signala, rješavajući četiri jednadžbe s četiri nepoznanice, prijemnik može odrediti svoju poziciju i vremenski odmak. Slika preuzeta iz [3].

2.2. Opis problema

U ovom radu, najviše pozornosti posvetit će se napadaču i njegovim mogućnostima da, korištenjem ispravnih parametara, uspješno provede napad lažiranjem signala nad jednom ili više žrtvi.

Implementacijski problemi su zanemareni i pretpostavka je da stvarni satelitski signali nisu prisutni.

2.2.1. Oblikovanje pseudoudaljenosti

Na izračun lokacije žrtve L'_j utječu fizička lokacija napadača P_i^A , vremenski odmak slanja signala δ_i^A i pretpostavljeni položaj satelita L_i^A . Napadač može utjecati na svoju fizičku lokaciju P_i^A i vremenski odmak δ_i^A . Postavljanjem tih parametara napadač može utjecati na pseudoudaljenost koju računa žrtva. Očekivana pseudoudaljenost koju će žrtva na fizičkoj lokaciji P_j izračunati je

$$R_{ij}^A = |P_j - P_i^A| + \delta_i^A \cdot c. \quad (4)$$

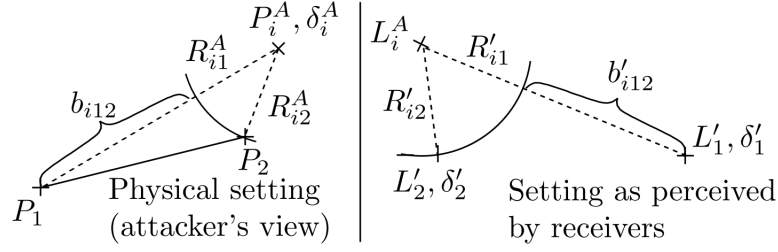
Kako bi odredila svoju lokaciju L'_j , svaka žrtva rješava sustav

$$|L'_j - L_i^A| = R'_{ij} - \Delta'_j \quad (5)$$

gdje je L_i^A pretpostavljeni položaj satelita, R'_{ij} pseudoudaljenost koju je izračunala žrtva i $\Delta'_j = \delta'_j \cdot c$ vremenski odmak koji je izračunala žrtva, pomnožen brzinom propagacije signala.

Za svaki satelit koji želi imitirati, napadač mora poslati signal s_i^A takav da žrtva, rješavajući izraz (5), odredi željenu lokaciju L'_j i vremenski odmak δ'_j . Kako bi napadač uspio u tome, mora zadovoljiti uvjet $R_{ij}^A = R'_{ij}$ (Slika 2.2), odnosno mora vrijediti

$$|P_j - P_i^A| + \Delta_i^A = |L'_j - L_i^A| + \Delta'_j. \quad (6)$$



Slika 2.2: Scenarij lažiranja GPS signala s dvije žrtve i jednim napadačem. Prikaz pogleda napadača (lijevo) i pogleda žrtve (desno). Slika preuzeta iz [3].

2.2.2. Lažiranje jedne lokacije

Jedan ili više prijemnika mogu biti vrlo lako prevareni tako da izračunaju bilo koju željenu, ali jednaku lokaciju. Napadaču je za takav pothvat potrebna samo jedna antena. Jedina razlika u izračunu pojaviti će se kod vremenskog odmaka svake žrtve.

Napadač koristi samo jednu antenu pa dolazi do vremenskih razlika primitka signala između prijemnika. Vrlo jednostavan dokaz detaljno je objašnjen u [3].

2.2.3. Lažiranje više lokacija

Uzimajući u obzir da napadač može prevariti neograničeno mnogo prijemnika da odrede jednaku željenu poziciju uz različite vremenske odmake, nailazimo na problem prilikom lažiranja više lokacija. Ako prijemnici imaju mogućnost provjere međusobnih pozicija ili vremenskih odmaka, tada mogu i otkriti takve napade. Prijemnici mogu provjeriti kako je nemoguće da svi imaju iste pozicije te da vremenski odmaci ne odgovaraju.

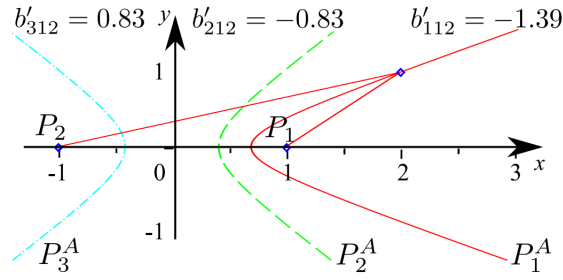
Iz tog razloga napadač ne smije koristiti samo jednu antenu kako bi pomaknuo žrtve na različite lokacije.

Dva prijemnika koja se nalaze na fizičkim lokacijama $P_1 \neq P_2$ moguće je prevariti da izračunaju lokacije $L'_1 \neq L'_2$ i vremenske odmake δ'_1, δ'_2 ako napadač smije odabrati bilo koju svoju lokaciju P_i^A i bilo koji pretpostavljeni satelit L_i^A . Za svaki signal s_i^A moguće lokacije napadača P_i^A leže na jednoj polovici dvoplošnog hiperboloida definiranog s $L'_1, L'_2, \delta'_1, \delta'_2, L_i^A$ te P_1 i P_2 . Slika 2.3 prikazuje takav slučaj u 2D.

Kako bi napadač uspješno prevario dvije žrtve, mora poslati takav signal s_i da pristigne s ispravnim vremenskim odmakom na lokacije žrtvi. Stoga mora vrijediti

$$|P_j - P_i^A| - |P_k - P_i^A| = |L'_j - L_i^A| - |L'_k - L_i^A| + \Delta'_j - \Delta'_k. \quad (7)$$

Iz izraza (7) jasno je vidljivo kako napadač uvijek može pronaći kombinacije P_i^A i L_i^A da bi dobio ispravne pseudoudaljenosti i zatim iz izraza (6) pronašao odgovarajuće vremenske odmake slanja signala δ_i^A .



Slika 2.3: Hiperbole mogućih lokacija antena napadača P_i^A u slučaju dvije žrtve (u 2D). Slika preuzeta iz [3].

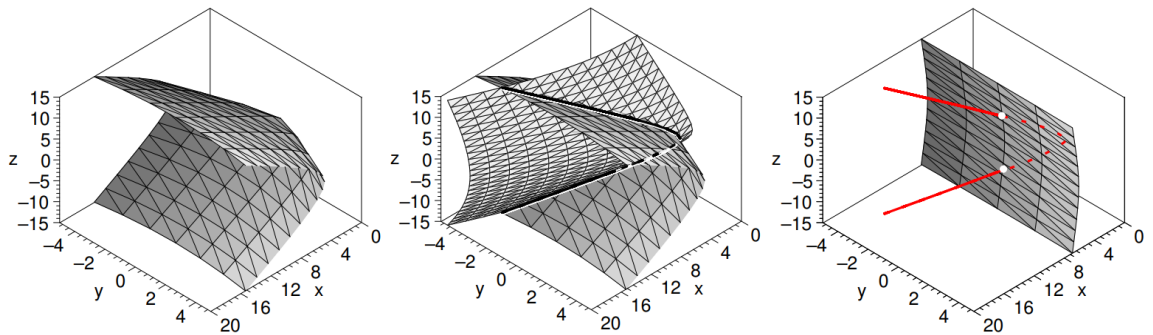
n	Lažiranje jedne lokacije	Lažiranje više lokacija (očuvana formacija)
1	$P_i^A \in \mathbb{R}^3$	-
2	$P_i^A \in \mathbb{R}^3$	skup hiperboloida
3	$P_i^A \in \mathbb{R}^3$	skup sjecišta između dva hiperboloida
4	$P_i^A \in \mathbb{R}^3$	skup dviju točaka
≥ 5	$P_i^A \in \mathbb{R}^3$	skup točaka

Tablica 2.1: Pregled mogućih pozicija napadača P_i^A u ovisnosti o broju žrtava n .

2.2.4. Moguća rješenja

Tablica 2.1 daje pregled mogućih pozicija antena napadača P_i^A u ovisnosti o broju žrtava n i broju lokacija koje se žele lažirati. Vidljivo je da za bilo koji broj žrtava, ne postoje ograničenja prilikom lažiranja jedne lokacije ($P_i^A \in \mathbb{R}^3$), dok se prilikom lažiranja više lokacija, broj mogućih rješenja drastično smanjuje porastom broja žrtava.

Slika 2.4 omogućava lakšu vizualizaciju mogućih pozicija antena napadača. Jasno je da broj mogućih rješenja značajno opada porastom broja prijemnika čija se lokacija želi lažirati.



Slika 2.4: Vizualizacija mogućih pozicija napadača. Za dvije žrtve (lijevo) sve točke hiperboloida su prihvatljiva rješenja, za 3 žrtve (sredina) prihvatljiva rješenja leže na presjecištu hiperboloida i za 4 žrtve (desno) samo dvije točke su prihvatljiva rješenja. Slika preuzeta iz [3].

3. Optimizacijski algoritmi

3.1. Evolucijski algoritmi

Evolucijske algoritme karakterizira svojstvo adaptacije kroz iterativan proces pokušaja i pogreške. Inspirirani su procesima i mehanizmima biološke evolucije. Potencijalna rješenja zapravo su jedinke virtualne populacije koje nastoje preživjeti u okruženju definiranom specifičnom ciljnom funkcijom. Glavni mehanizmi procesa evolucije, koji omogućavaju unapređenje prilagodbe potencijalnih rješenja na okruženje, su genetska rekombinacija i mutacija. Od evolucijskih algoritama u ovom radu opisat će se Steady State Tournament te Evolution Strategy.

3.1.1. Steady State Tournament

Jedinke populacije doprinose ukupnom genetskom materijalu (genotipu) ovisno o tome koliko je njihov izraženi genom (fenotip) prikladan okruženju u kojem se ispituju. Sljedeća generacija odbacit će određen broj najlošijih jedinki, a od ostalih će se, u procesu rekombinacije dva individualna genoma i mutacije, stvoriti nove jedinke. Ovaj iterativan proces može rezultirati poboljšanjem prilagodbe populacije na zadano okruženje. Detaljnije na [2].

Algoritam 1: Steady State Tournament

```
1 begin jedna generacija
2   for  $i = 1$  to veličina populacije do
3     nasumično dodaj  $n$  jedinki u turnir;
4     izaberi najgoru jedinku na turniru;
5     nasumično odaberi 2 roditelja između preostalih jedinki na turniru;
6     najgoru jedinku zamijeni jedinkom djeteta nastalom križanjem roditelja;
7     izvrši mutaciju djeteta;
8   end
9 end
```

3.1.2. Evolution Strategy

Za razliku od ostalih algoritama, ovaj algoritam inspiriran je procesom evolucije na razini vrste (fenotip, nasljeđivanje, varijacija) i ne bavi se genetskim mehanizmima evolucije (genom, kromosomi, geni, aleli). Sažeto se može opisati terminologijom μ i λ , gdje je μ broj potencijalnih rješenja u roditeljskoj generaciji, a λ broj potencijalnih rješenja nastalih iz roditeljske generacije. Razlikujemo dva tipa konfiguracija. Zarez selekcija $(\mu, \lambda) - ES$ stvara potomke samo iz trenutne generacije μ . Plus selekcija $(\mu + \lambda) - ES$ stvara potomke iz trenutne generacije i roditeljske generacije. Za više detalja pogledati [2].

Algoritam 2: Evolution Strategy

```
1 begin jedna generacija
2   foreach subpopulaciju do
3     nasumično dodaj  $\mu$  jedinki roditeljskoj generaciji;
4     stvori  $\lambda$  potomaka koristeći rho roditelja za svaku jedinku;
5     if zarez selekcija then
6       stvori novu roditeljsku generaciju s  $\mu$  najboljih potomaka;
7     end
8     else
9       stvori novu roditeljsku generaciju s  $\mu$  najboljih potomaka i jedinki iz
        roditeljske generacije;
10    end
11  end
12 end
```

3.2. Imunološki algoritmi

Inspirirani su procesima i mehanizmima biološkog imunološkog sustava. Pojednostavljen opis imunološkog sustava je sustav organa koji štite organizam domaćina od raznih prijetnji kao što su patogeni i toksične tvari. Od imunoloških algoritama u ovom radu opisat će se Clonal Selection Algorithm te Immune Optimization Algorithm.

3.2.1. Clonal Selection Algorithm

Inspiriran ponašanjem i sposobnošću antitijela u stečenom imunološkom sustavu. Potencijalna rješenja u ovom modelu predstavljaju antitijela čija se selekcija obavlja na temelju

uspješnosti protiv uzorka antigena. Odabrana antitijela podvrgnuta su kloniranju proporcionalno afinitetu, a hipermutacija klonova obrnuto proporcionalno afinitetu klonova. Dobi-
veni klonalni skup natječe se s postojećom populacijom antitijela za članstvo u sljedećoj
generaciji. Dodatno, članovi populacije s niskim afinitetom zamijenjeni su nasumično stvo-
renim antitijelima. Više informacija na [4].

Algoritam 3: Clonal Selection Algorithm

Data: populacija veličine N

```

1 while uvjet zaustavljanja nije zadovoljen do
2   foreach jedinku iz populacije do
3     |   izračunaj afinitet jedinke;
4   end
5   provedi selekciju iz populacije;
6   foreach selektiranu jedinku do
7     |   kloniraj jedinku;
8   end
9   foreach kloniranu jedinku do
10    |   hipermutiraj jedinku;
11    |   izračunaj afinitet jedinke;
12  end
13  stvori novu populaciju iz selektiranih i kloniranih jedinki;
14  najlošije jedinke populacije zamijeni nasumično stvorenima;
15 end

```

3.2.2. Immune Optimization Algorithm

Kao i Clonal Selection Algorithm, Immune Optimization Algorithm koristi dvije populacije, antigene i antitijela. Međutim, razlika je u tome što Immune Optimization Algorithm, uz operatore kloniranja i hipermutacije, koristi i operator starenja. Taj operator izbacuje stara antitijela iz populacije kako bi se izbjegla preuranjena konvergencija. Prilikom kloniranja, klonirana jedinka preuzima starost svojih roditelja, a nakon faze hipermutacije, klonirana jedinka koja uspješno mutira imat će starost 0. Tako svaka nova jedinka ima jednaku mogućnost učinkovitog istraživanja okoline. Detaljnije na [4].

4. Programsko ostvarenje

4.1. Računanje lokacije i vremenskog odmaka

U svim implementacijama korišten je programski kod koji lokaciju i vremenski odmak prijemnika računa prema izrazu

$$\begin{aligned}\sqrt{(x - x_A)^2 + (y - y_A)^2 + (z - z_A)^2} &= c \cdot (t_A - d) \\ \sqrt{(x - x_B)^2 + (y - y_B)^2 + (z - z_B)^2} &= c \cdot (t_B - d) \\ \sqrt{(x - x_C)^2 + (y - y_C)^2 + (z - z_C)^2} &= c \cdot (t_C - d) \\ \sqrt{(x - x_D)^2 + (y - y_D)^2 + (z - z_D)^2} &= c \cdot (t_D - d)\end{aligned}\tag{8}$$

gdje su x , y i z koordinate prijemnika, $x_{A,B,C,D}$, $y_{A,B,C,D}$ i $z_{A,B,C,D}$ koordinate satelita, d vremenska razlika sata prijemnika i satova satelita i $t_{A,B,C,D}$ razlike trenutka primitka i vremenske oznake slanja pojedinog signala.

Problem se tada svodi na rješavanje sustava četiri jednačbe s četiri nepoznanice. Rješavanje sustava provodi se programskom implementacijom koja koristi Newton-Raphsonovu metodu s deset iteracija. Za dokumentaciju i više detalja pogledati [1].

4.2. Implementirani scenariji

4.2.1. Lažiranje jedne lokacije

Kao što je objašnjeno u poglavlju 2.2.2, lažiranje jedne lokacije je problem koji se može vrlo jednostavno riješiti analitički. Unatoč tome, ovaj scenarij poslužio je za postavljanje modela te izgradnju temelja za suočavanje s ozbiljnijim slučajevima. Stoga će biti ukratko opisan.

U ovom scenariju, jedan napadač (koristeći jednu antenu) nastoji zavarati jednu žrtvu (prijemnik). Pretpostavka je da napadač ima neograničene mogućnosti, odnosno da može u svakom trenutku poslati signal do bilo koje željene lokacije.

Potrebni ulazni parametri su: pretpostavljene koordinate četiri satelita te željena pozicija na koju se želi navesti prijemnik.

Rješavanje problema započinje računanjem vremena propagacije signala od najbližeg satelita do prijemnika. Napadač će u prvom poslanom signalu navesti vremensku oznaku slanja

koja odgovara upravo tom vremenu propagacije. Vrlo jasno je vidljivo kako se pretpostavlja da je napadač sveznajući, on zna stvarnu poziciju prijemnika i zna kad treba poslati signal. Ovisno o vremenu propagacije prvog signala, optimizacijskim algoritmima želimo saznati potrebna propagacijska vremena ostala tri signala (po jedan signal za svaki satelit).

Za dobivanje preostalih vremena propagacije signala koristi se floating-point genotip veličine tri. Implementirana su dva različita načina. Prvi način uključuje apsolutna propagacijska vremena jer ovise samo o navedenom početnom vremenu propagacije, dok se u drugom načinu koristi relativno vrijeme propagacije jer svako sljedeće vrijeme ovisi o svim prethodnim vremenima. I jedan i drugi način uglavnom daju jednake rezultate te zbog toga nisu pojedinačno implementirani u scenarijima koji slijede.

Koristeći generirana propagacijska vremena, postupkom opisanim u poglavlju 4.1 možemo odrediti trenutnu poziciju. Nakon određivanja trenutne pozicije, pogreška se utvrđuje računanjem euklidske udaljenosti željene pozicije i trenutne pozicije

$$error = d(P_w, P_c) = \sqrt{(x_w - x_c)^2 + (y_w - y_c)^2 + (z_w - z_c)^2} \quad (9)$$

gdje su P_w željena pozicija prijemnika, P_c trenutna pozicija prijemnika, x_w, y_w, z_w koordinate željene pozicije prijemnika i x_c, y_c, z_c koordinate trenutne pozicije prijemnika.

Iako se u ovom primjeru računa s propagacijskim vremenima, u stvarnoj implementaciji napadač mora znati koju vremensku oznaku će poslati prilikom slanja signala kako bi prijemnik dobio dojam kao da je taj signal stvarno toliko vremenski propagirao.

4.2.2. Lažiranje više lokacija - pozicije napadača

Lažiranje više lokacija s nepoznatim pozicijama napadača temeljna je implementacija ovog rada i bit će detaljno opisana.

Početni scenarij uključuje jednog napadača s četiri antene i dvije žrtve (dva prijemnika), a kasnije će biti proširen na neograničeni broj prijemnika. Problem je pojednostavljen time što se uvijek koriste četiri antene od kojih svaka imitira jedan od pretpostavljenih satelita. Kada ne bi bilo tog pojednostavljenja, dodatno bi se trebao ispitivati odnos više satelita i antena odjednom, a time se situacija znatno otežava.

Ulazni parametri početnog scenarija s dva prijemnika su: pretpostavljene koordinate četiri satelita, stvarne pozicije dvaju prijemnika te željene pozicije dvaju prijemnika. Pritom treba pripaziti da stvarne pozicije prijemnika i željene pozicije prijemnika budu jednako udaljene. Prema izrazu (7), u realnoj implementaciji prijemnici mogu provjeriti svoju poziciju i vremenski odmak u odnosu na pozicije i vremenske odmake drugih prijemnika i tako shvatiti radi li se o napadu ili ne. U ovoj implementaciji te provjere nema, ali taj uvjet nije zanemaren baš zbog toga što razmak između stvarnih i željenih pozicija ostaje isti.

Potrebno je pronaći pozicije na koje treba postaviti antene napadača, ali i vremenske odmake s kojima napadač treba poslati signale prema prijemnicima. Stoga se koristi folating-point genotip veličine šesnaest. Na koordinate četiri antene otpada dvanaest vrijednosti, a preostale četiri vrijednosti predstavljaju vremenske odmake slanja signala s pojedine antene.

Evaluacija jedinice započinje određivanjem pozicija antena napadača iz genotipa. Zatim, za svaku antenu treba izračunati vrijeme propagacije signala od antene do prijemnika. Postupkom iz poglavlja 4.1 možemo dobiti trenutnu poziciju prijemnika, a kao ukupno vrijeme propagacije signala od antene do prijemnika uvrštava se zbroj generiranog vremenskog odmaka iz genotipa i izračunatog vremena propagacije za određenu antenu i određeni prijemnik. Pogreška se računa prema izrazu (9) koristeći trenutnu i željenu poziciju prijemnika. Postupak se ponavlja za oba prijemnika zasebno, a pogreške se zbrajaju.

Prethodni scenarij može se generalizirati za jednog napadača s četiri antene i neograničeno mnogo žrtava (prijemnika). Evaluacija će se sastojati od računanja pogreške između trenutne i željene pozicije za svaki prijemnik, a rezultat evaluacije bit će ukupni zbroj svih pogrešaka. U generaliziranom scenariju potrebni parametri su: pozicije pretpostavljenih satelita, broj prijemnika, stvarne pozicije svakog prijemnika i željene pozicije svakog prijemnika.

Iako je omogućeno dodavanje neograničenog broja prijemnika, u poglavlju 2.2.4 je vrlo jasno prikazano da porastom broja prijemnika, u slučaju lažiranja više lokacija, dolazi do drastičnog pada broja mogućih rješenja. Stoga implementacija nije ispitana za veći broj prijemnika i ponašanje modela u takvom slučaju nije poznato.

Postoji i poseban slučaj kada su pozicije antena napadača postavljene na pretpostavljene pozicije satelita, odnosno pojedina antena doslovno imitira zadani satelit. Ovakav scenarij dodan je kako bi se provjerila implementirana konfiguracija. Provjera se pokazala interesantnom i što je još važnije, uspješnom.

4.2.3. Lažiranje više lokacija - pozicije satelita

Za razliku od prethodnog poglavlja, gdje je opisan način traženja optimalnih pozicija antena napadača, u ovom poglavlju situacija je obrnuta. Uz vremenske odmake slanja signala, koji su uvijek u središtu zanimanja, potrebno je pronaći i pretpostavljene pozicije satelita. Slučaj je detaljno opisan u nastavku.

Scenarij uključuje jednog napadača s četiri antene, četiri satelita i neograničen broj prijemnika. Pozicije antena unaprijed su određene, kao i stvarne te željene pozicije prijemnika. Cilj je pronaći pretpostavljene koordinate svih satelita koje bi odgovarale trenutnoj konfiguraciji napadača i prijemnika, odnosno koje bi omogućile napadaču da uspješno izvede napad.

Ulazni parametri su: pozicije četiri antene napadača, broj prijemnika, stvarne pozicije prijemnika i željene pozicije prijemnika.

Potrebno je pronaći pozicije satelita koje se želi imitirati sa zadanih pozicija antena, kao i vremenske odmake slanja lažiranih signala. Ponovno se koristi floating-point genotip veličine šesnaest. Dvanaest vrijednosti otpada na pozicije satelita, a četiri na vremenske odmake slanja signala.

Evaluacija jedinice započinje određivanjem pozicija satelita iz genotipa. Potom slijedi izračun trenutne pozicije prijemnika postupkom opisanom u poglavlju 4.1. Kao ukupno vrijeme propagacije signala uzima se zbroj iznosa propagacijskog vremena signala od određenog napadača do određenog prijamnika i vremenskog odmaka dobivenog iz genotipa. Pogreška se ponovno računa na isti način, koristeći izraz (9). Rezultat evaluacije je ukupan zbroj pogrešaka za svaki prijemnik.

Vidljivo je da se postupak evaluacije ne razlikuje značajno od postupka iz prošlog poglavlja. Razlika je u tome što se u ovom slučaju pozicije satelita optimiraju i takve se koriste u izračunu, a u prošlom poglavlju koristile su se zadane vrijednosti pozicija satelita. Jednako tako se u ovom slučaju koriste zadane pozicije antena napadača, dok su se u prošlom poglavlju iste optimirale.

Treba napomenuti da u opisanom scenariju ne postoji slučaj postavljanja antena napadača direktno na pozicije satelita, kakav se javlja u prošlom poglavlju. Jednostavno takva situacija nema smisla jer se pozicije satelita tek trebaju optimirati.

5. Rezultati i diskusija

Za potrebe testiranja koristi se ECF - Evolutionary Computation Framework čija se detaljna dokumentacija može pronaći na [2].

5.1. Optimizacija parametara algoritama

Optimizacija parametara provodit će se na scenariju iz poglavlja 4.2.2. Prije nego što se krene s usporedbom algoritama, trebamo provjeriti koji parametri algoritama su optimalni. Početna konfiguracija je sljedeća:

$$\text{broj evaluacija} = 200\,000$$

$$\text{veličina floating-point genotipa} = 16$$

$$\text{vrijednosti genotipa} = [-30\,000, 30\,000]$$

$$\text{veličina populacije} = ?$$

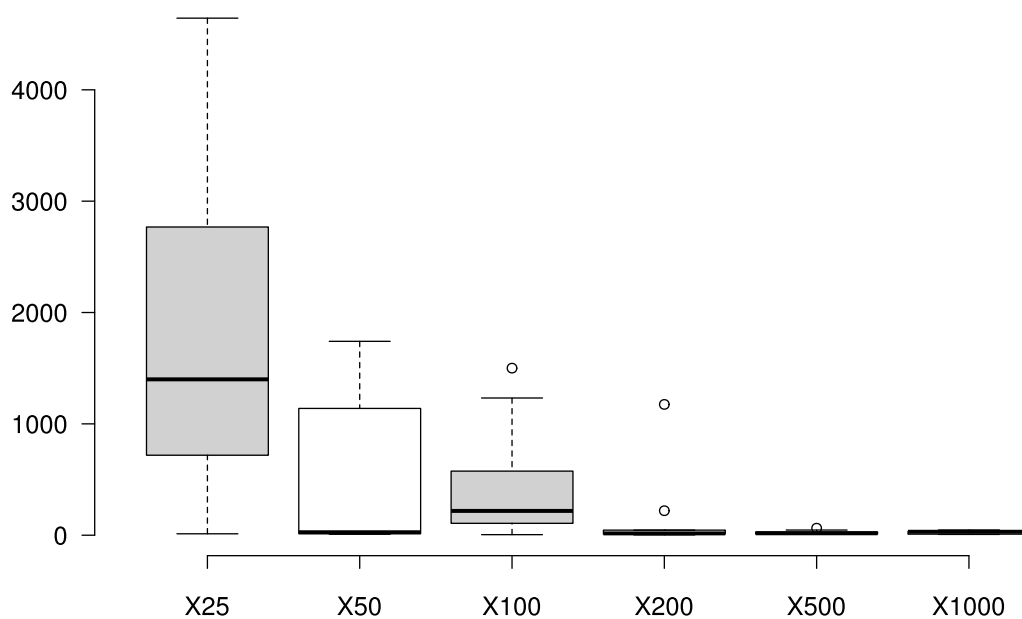
$$\text{vjerojatnost mutacije} = ?$$

Cilj je optimizirati parametre *veličina populacije* i *vjerojatnost mutacije*. Parametri algoritama optimizirat će se na algoritmima Steady State Tournament (3.1.1) i Clonal Selection Algorithm (3.2.1). Optimizacija započinje postavljanjem jednog od ta dva parametra. Parametar *vjerojatnost mutacije* postavlja se na vrijednost 0.3 te se ispituje uspješnost algoritama za vrijednosti 25, 50, 100, 200, 500 i 1000 parametra *veličina populacije*. Testiranje se pokreće 10 puta za sve kombinacije parametara.

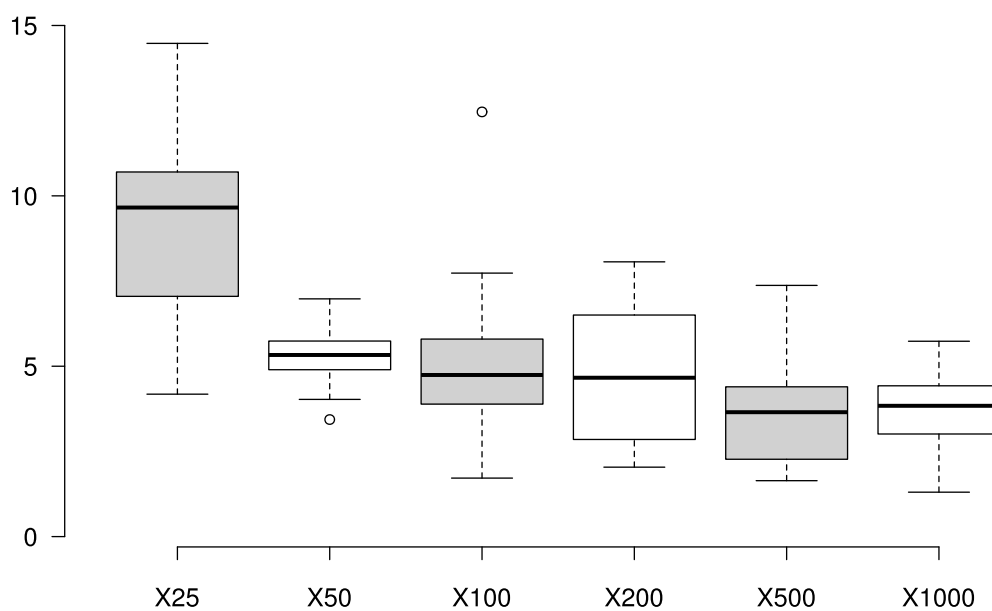
Na slici 5.1 prikazani su rezultati algoritma Steady State Tournament. Iako za vrijednosti 100, 500 i 1000 parametra *veličina populacije* algoritam daje poprilično dobre rezultate, velike pogreške javljaju se za vrijednosti 50 i 100, a pogotovo 25.

Slika 5.2 prikazuje rezultate algoritma Clonal Selection Algorithm. Vidljivo je da skoro sve vrijednosti parametra *veličina populacije* daju solidne rezultate. Međutim, valja primijetiti da su najlošiji rezultati dobiveni za vrijednost 25, a najbolji za vrijednost 500.

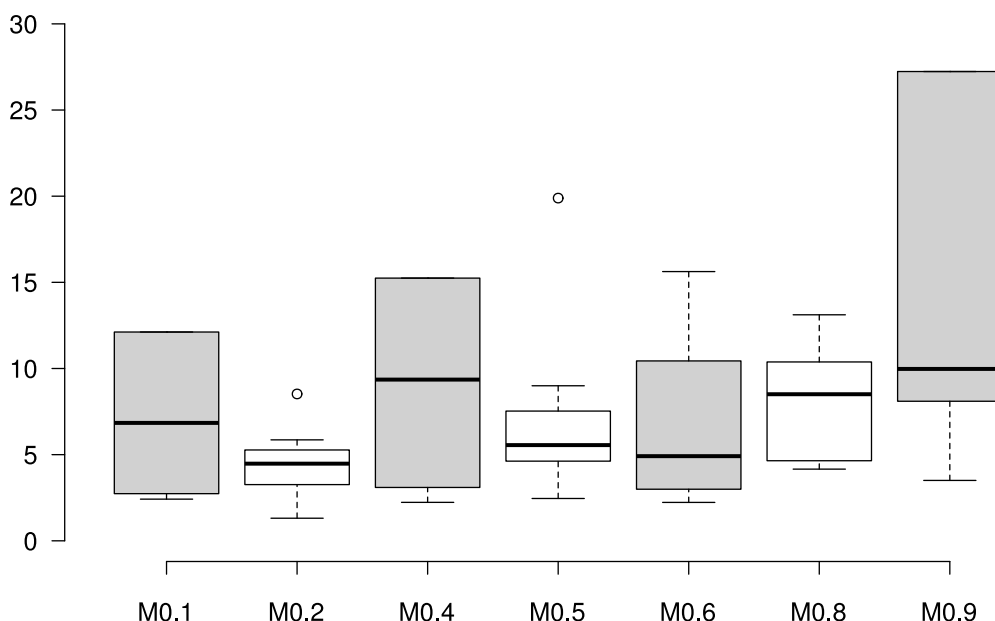
Iz slika 5.1 i 5.2 može se zaključiti da algoritmi najbolje rezultate ostvaruju za vrijednost 500 parametra *veličina populacije*. Stoga se testovi nastavljaju koristeći upravo tu vrijednost.



Slika 5.1: Rezultati algoritma Steady State Tournament s postavljenom vrijednosti 0.3 parametra *vjerojatnost mutacije* i vrijednostima 25, 50, 100, 200, 500 i 1000 parametra *veličina populacije*.



Slika 5.2: Rezultati algoritma Clonal Selection Algorithm s postavljenom vrijednosti 0.3 parametra *vjerojatnost mutacije* i vrijednostima 25, 50, 100, 200, 500 i 1000 parametra *veličina populacije*.



Slika 5.3: Rezultati algoritma Steady State Tournament s postavljenom vrijednosti 500 parametra *veličina populacije* i vrijednostima 0.1, 0.2, 0.4, 0.5, 0.6, 0.8 i 0.9 parametra *vjerojatnost mutacije*.

Parametar *veličina populacije* postavlja se na vrijednost 500, a ispituje se uspješnost algoritama za vrijednosti 0.1, 0.2, 0.4, 0.5, 0.6, 0.8 i 0.9 parametra *vjerojatnost mutacije*. Testiranje se pokreće 10 puta za sve kombinacije parametara.

Na slici 5.3 mogu se vidjeti rezultati algoritma Steady State Tournament. Rezultati za vrijednost 0.2 parametra *vjerojatnost mutacije* definitivno odskakuju od ostalih jer su konstantno najbolji. Rezultati postaju lošiji porastom vrijednosti parametra *vjerojatnost mutacije*, a uvjerljivo najlošiji rezultat dobiva se za vrijednost 0.9.

Slika 5.4 prikazuje rezultate algoritma Clonal Selection Algorithm. Vidljivo je da su rezultati zadovoljavajući i podjednaki za sve vrijednosti parametra *vjerojatnost mutacije*.

Uzimajući u obzir rezultate prikazane na slikama 5.3 i 5.4 može se tvrditi da algoritmi najbolje rezultate ostvaruju za vrijednost 0.2 parametra *vjerojatnost mutacije*. Iako to nije odmah jasno iz rezultata sa slike 5.4, algoritmi konstantno polučuju dobre rezultate za vrijednost 0.2 parametra *vjerojatnost mutacije*, bez značajnijih odstupanja. Završna konfiguracija je sljedeća:

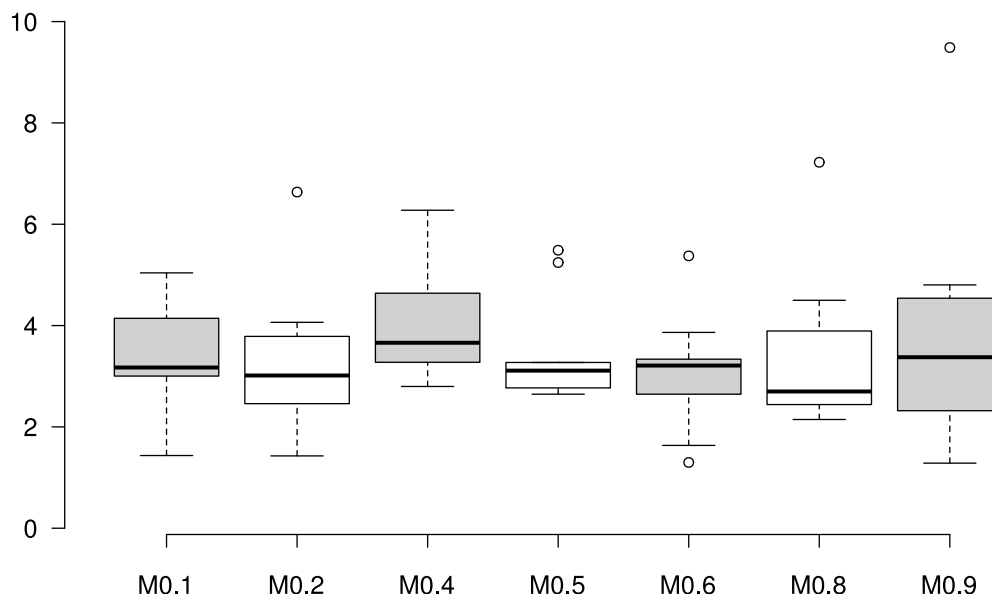
broj evaluacija = 200 000

veličina floating-point genotipa = 16

vrijednosti genotipa = [−30 000, 30 000]

veličina populacije = **500**

vjerojatnost mutacije = **0.2**



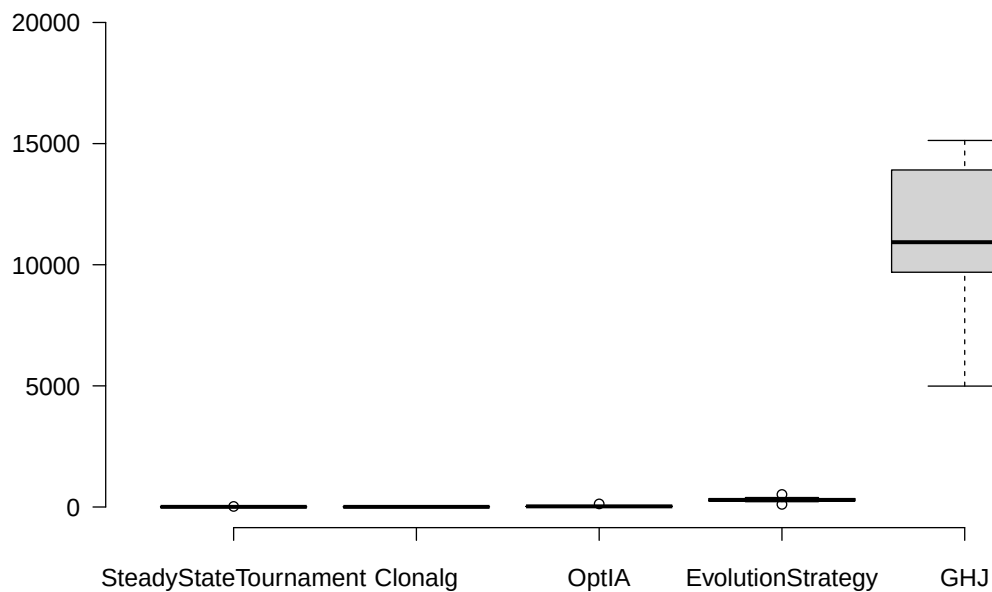
Slika 5.4: Rezultati algoritma Clonal Selection Algorithm s postavljenom vrijednosti 500 parametra *veličina populacije* i vrijednostima 0.1, 0.2, 0.4, 0.5, 0.6, 0.8 i 0.9 parametra *vjerojatnost mutacije*.

Nakon što smo dobili optimalne parametre, vrijeme je za testiranje uspješnosti pojedinih algoritama upravo s tim parametrima. Uz već poznate algoritme Steady State Tournament i Clonal Selection Algorithm, testiranju će biti podvrgnuti i Evolution Strategy (3.1.2), Immune Optimization Algorithm (3.2.2) te GenHookeJeeves algoritam koji neće biti zasebno opisan, ali čija se dokumentacija može pronaći na [2]. Po preporuci autora algoritma, GenHookeJeeves bit će pokretan s vrijednost 50 za parametar *veličina populacije*.

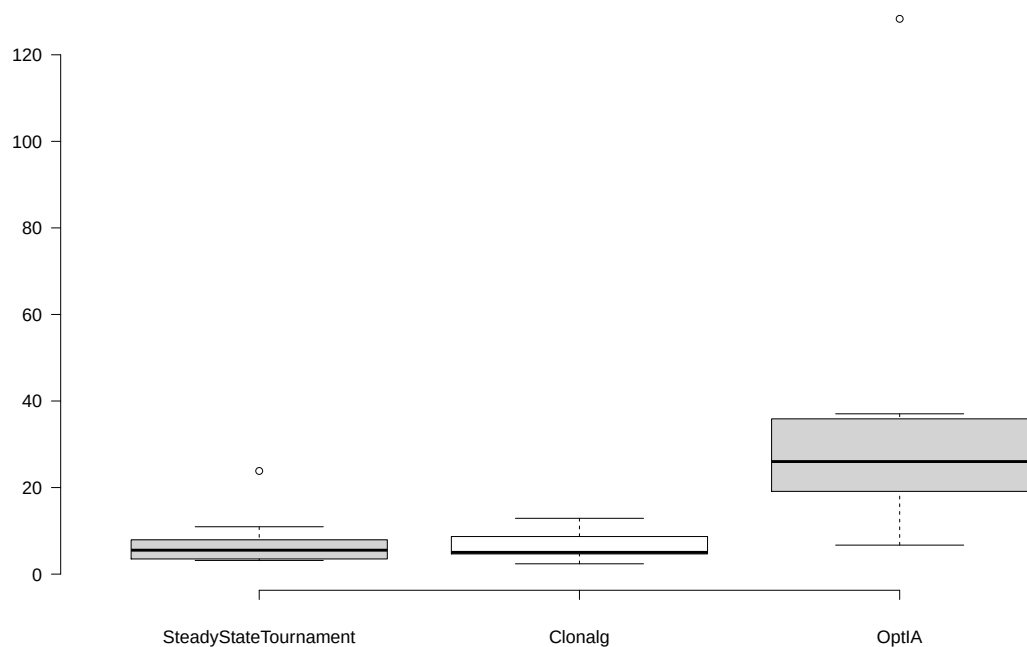
Završni rezultati prikazani su na slici 5.5. Nedvojbeno najbolje rezultate ostvarili su algoritmi Steady State Tournament, Clonal Selection Algorithm i Immune Optimization Algorithm. Na slikama 5.6 i 5.7 detaljnije su prikazani rezultati tri, odnosno dva najbolja algoritma po rezultatima. Immune Optimization Algorithm ne može se mjeriti s preostala dva i ne konkuriра za najbolji algoritam. Teško je odabrati koji je algoritam stvarno najbolji za ovaj problem jer Clonal Selection Algorithm ima manji medijan rezultata, a Steady State Tournament može pružiti dosta bolje pojedinačne rezultate.

5.2. Ispitivanje algoritama

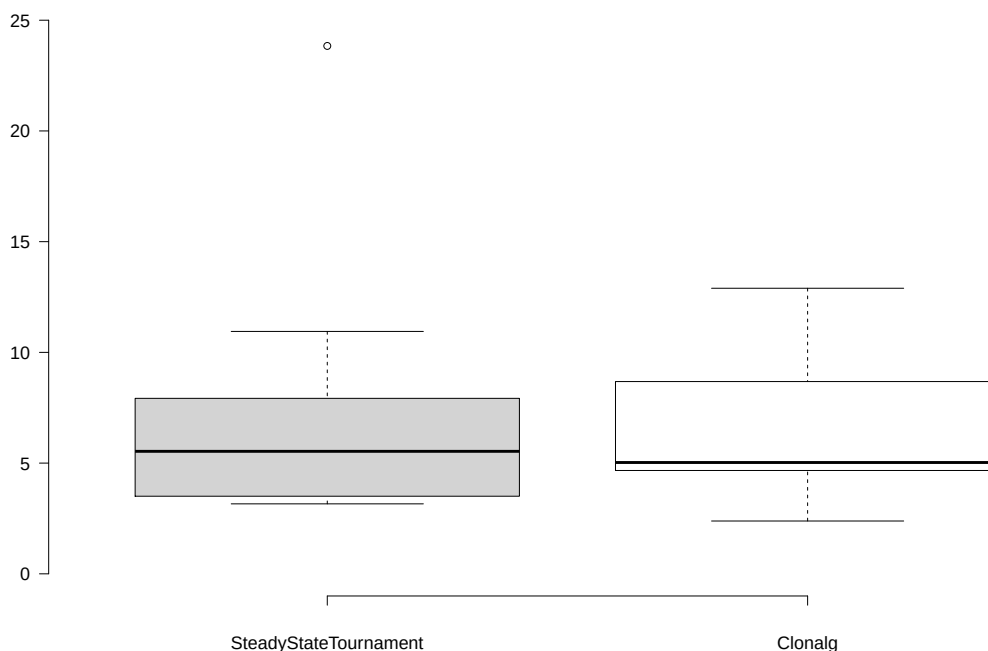
U ovom poglavlju ispitivat će se učinkovitost algoritama na scenarijima iz poglavlja 4.2. Algoritmi će se testirati na više testnih slučajeva s parametrima dobivenima u poglavlju 5.1. Svaki algoritam pokreće se deset puta za jedan ulazni skup podataka. Cilj je minimizirati pogrešku procjene željene lokacije. Rezultati prikazuju iznos pogreške algoritama.



Slika 5.5: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm, Evolution Strategy, Immune Optimization Algorithm te GenHookeJeeves za *veličina populacije: 500* i *vjerojatnost mutacije: 0.2*.



Slika 5.6: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm i Immune Optimization Algorithm za *veličina populacije: 500* i *vjerojatnost mutacije: 0.2*.



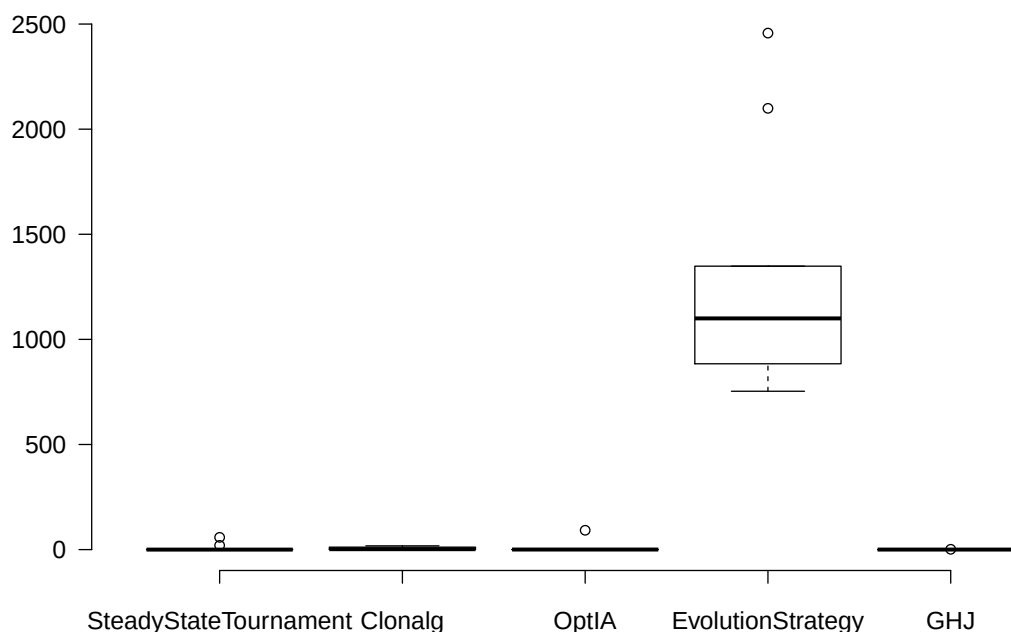
Slika 5.7: Rezultati algoritama Steady State Tournament i Clonal Selection Algorithm, Evolution Strategy za veličina populacije: 500 i vjerojatnost mutacije: 0.2.

Ispitivanje započinje lažiranjem jedne lokacije (poglavlje 4.2.1). Ulazni skup podataka prikazan je u tablici 5.1. Dobiveni rezultati za pojedini algoritam prikazani su na slici 5.8. Jasno je da je problem jednostavan jer su ga gotovo svi algoritmi uspjeli riješiti poprilično dobro. Algoritam Evolution Strategy nije proizveo zadovoljavajuće rezultate za ovaj scenarij. Najboljima su se pokazali algoritmi GenHookeJeeves te Steady State Tournament.

pozicija / koordinate	X	Y	Z
satelit 1	17330	10611	15289
satelit 2	10035	13957	12025
satelit 3	17257	11378	16590
satelit 4	16200	12164	15772
željena pozicija	6072	6524	6829

Tablica 5.1: Ulazni skup podataka za scenarij lažiranja jedne lokacije.

Sljedeći scenarij koji se ispituje je lažiranje dvije lokacije (poglavlje 4.2.2), ali jednostavniji slučaj. Četiri napadača postavljena su na pozicije pretpostavljenih satelita. Ulazni skup podataka za ovaj slučaj prikazan je u tablici 5.2, a rezultati su prikazani na slici 5.9. Ponovno se algoritam Evolution Strategy ne može mjeriti s ostalima. Stoga slika 5.10 daje detaljniji uvid u rezultate za ovaj slučaj. Iako je algoritam GenHookeJeeves mnogo uspješniji od Evolution Strategy algoritma, jasno je da su preostala tri algoritma proizvela bolje



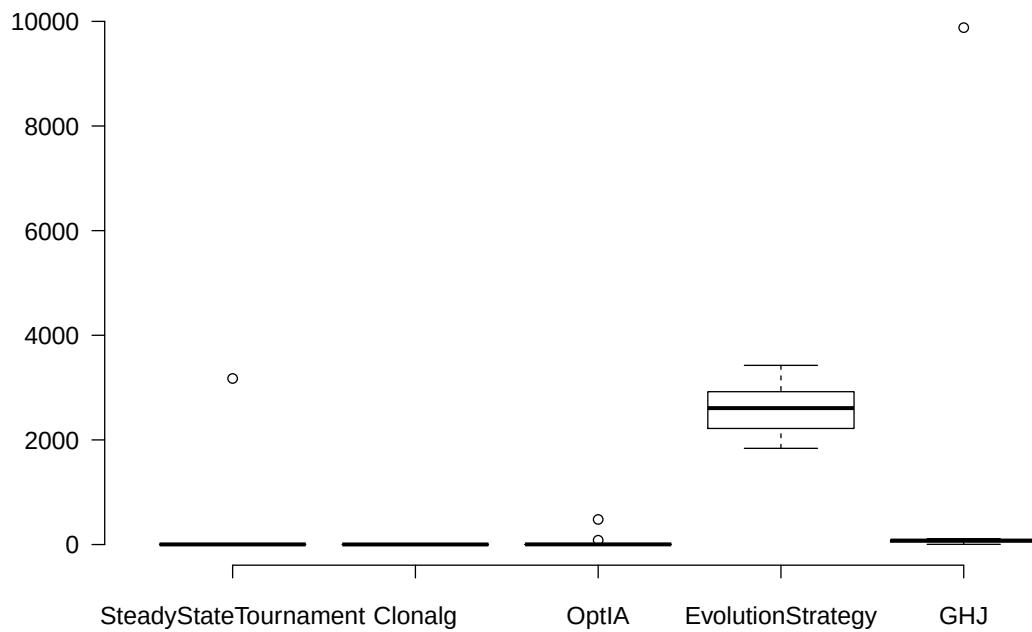
Slika 5.8: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm, Evolution Strategy, Immune Optimization Algorithm te GenHookeJeeves za scenarij lažiranja jedne lokacije.

rezultate. Među njima se ističe Clonal Selection Algorithm koji je konstantno najbolji.

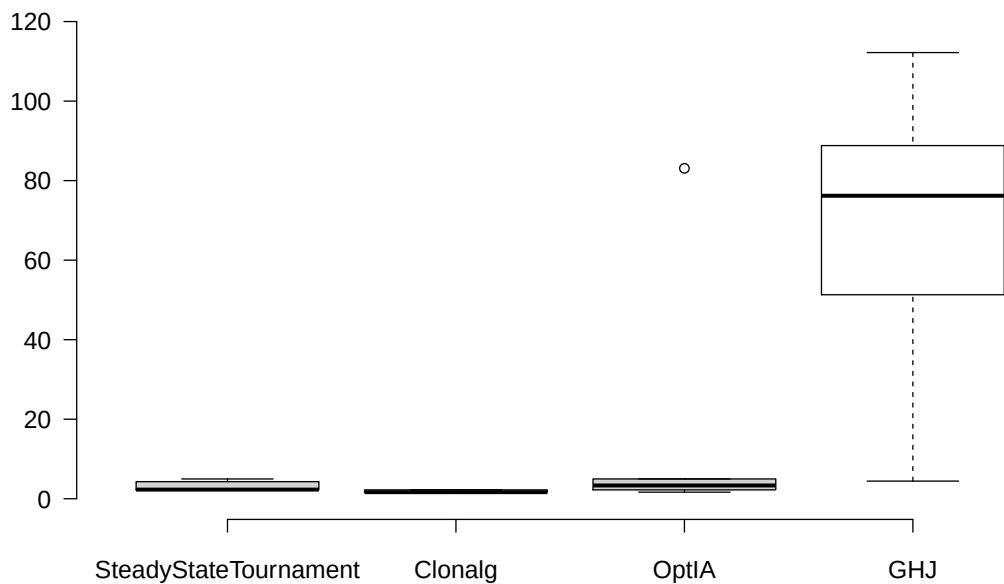
pozicija / koordinate	X	Y	Z
satelit 1 (napadač 1)	17398	11219	19534
satelit 2 (napadač 2)	17495	13229	19245
satelit 3 (napadač 3)	13857	16966	13816
satelit 4 (napadač 4)	18070	12316	16666
pozicija 1	6903	6400	6159
pozicija 2	6903	6400	6162
željena pozicija 1	6937	6457	6161
željena pozicija 2	6937	6457	6164

Tablica 5.2: Ulazni skup podataka za scenarij lažiranja dvije lokacije kad su napadači postavljeni na pretpostavljene pozicija satelita.

Scenarij lažiranja dvije lokacije s četiri napadača nepoznatih pozicija središte je ovog rada. Sukladno tome će algoritmi biti ispitani na dva skupa ulaznih podataka. Jedina promjena u parametrima algoritama bit će ta da se vrijednost parametra *broj evaluacija* povećava s 200 000 na 500 000 kako bi se omogućila veća preciznost. U tablici 5.3 prikazan je prvi skup ulaznih podataka, a dobiveni rezultati algoritama vidljivi su na slici 5.11. Može se uočiti kako je GenHookeJeeves algoritam, iako najbolji u prošlom scenariju, u ovom slučaju



Slika 5.9: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm, Evolution Strategy, Immune Optimization Algorithm te GenHookeJeeves za scenarij lažiranja dvije lokacije kad su napadači postavljeni na pretpostavljene pozicija satelita.



Slika 5.10: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm, Immune Optimization Algorithm te GenHookeJeeves za scenarij lažiranja dvije lokacije kad su napadači postavljeni na pretpostavljene pozicija satelita.

dao najlošije rezultate. Slika 5.12 prikazuje rezultate bez tog algoritma. Nema sumnje da Clonal Selection Algorithm proizvodi daleko najbolje rezultate za ovaj scenarij.

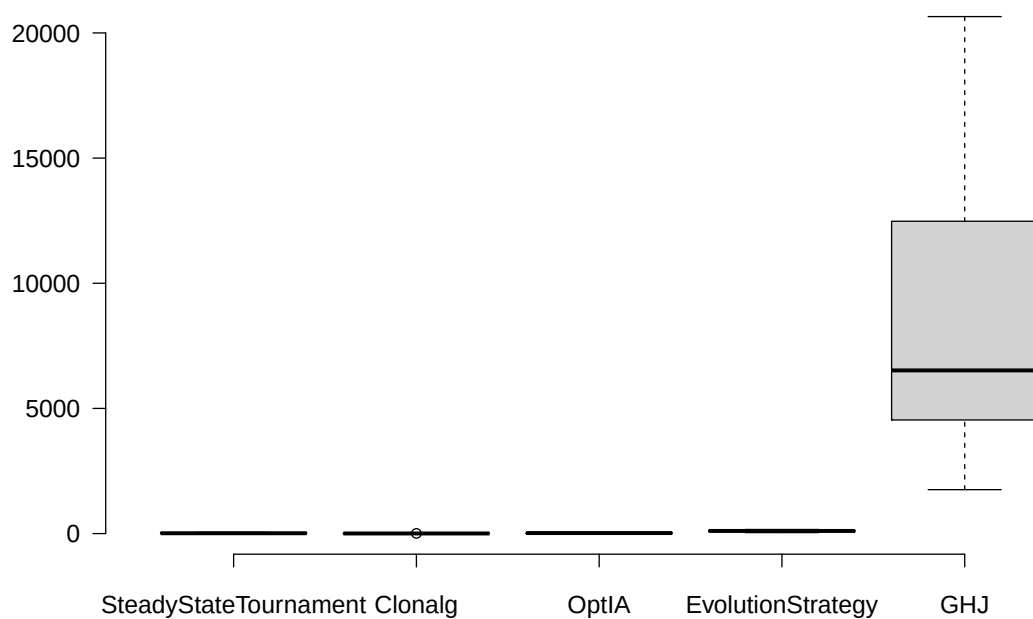
pozicija / koordinate	X	Y	Z
satelit 1	13031	14999	12847
satelit 2	15449	13554	15540
satelit 3	14808	16155	10037
satelit 4	10530	10167	16092
pozicija 1	6955	6519	6254
pozicija 2	6957	6522	6257
željena pozicija 1	6955	6136	6603
željena pozicija 2	6957	6139	6606

Tablica 5.3: Ulazni skup podataka I za scenarij lažiranja dvije lokacije s četiri napadača nepoznatih lokacija.

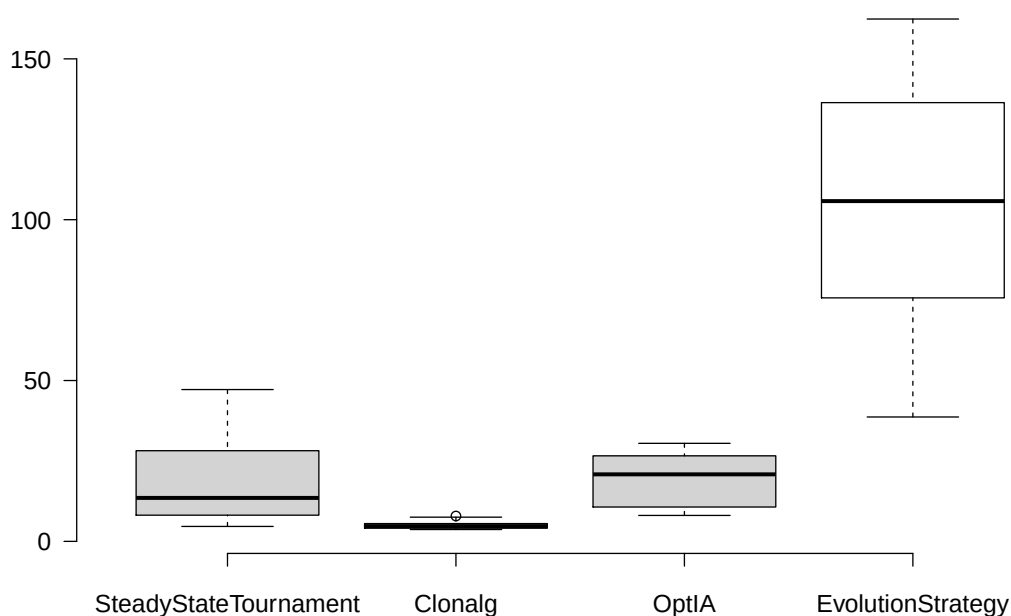
Rezultati na drugom skupu ulaznih podataka vrlo su slični onima dobivenima na prvom. GenHookeJeeves algoritam ponovno odskaka pa ga se neće ni prikazati. Ulazni skup podataka vidljiv je u tablici 5.4, a rezultati na slici 5.13. Rezultati općenito na drugom skupu više variraju, ali Clonal Selection Algorithm i dalje pokazuje najbolja svojstva, dok mu Steady State Tournament i Immune Optimization Algorithm neuspješno pokušavaju parirati.

Zadnji scenarij uključuje lažiranje dvije lokacije s četiri napadača poznatih pozicija, ali su pozicije pretpostavljenih satelita nepoznate (poglavlje 4.2.3). Stoga ulazni skup podataka uključuje koordinate napadača i može se vidjeti u tablici 5.5. Poprilično dobri rezultati za sve algoritme vidljivi su na slici 5.14. Kako bi se dobio bolji uvid u najbolje rezultate, slika 5.15 ne prikazuje rezultate algoritma Evolution Strategy. Iako nijedan od algoritama nije dao rezultat bez pogreške, svi algoritmi ostvarili su konstantno zadovoljavajuće rezultate bez mnogo oscilacija. Ponovno je najbolje rezultate proizveo Clonal Selection Algorithm.

Svi scenariji uspješno su implementirani i na njima je ispitana učinkovitost algoritama. Iz provedenih ispitivanja može se zaključiti da je definitivno najbolje rezultate dao Clonal Selection Algorithm, dok su najlošiji rezultati dobiveni za algoritam Evolution Strategy. Za scenarij lažiranja jedne lokacije uglavnom nije bilo većih odstupanja, odnosno ako je postojala pogreška, bila je vrlo mala. S druge strane, prilikom ispitivanja scenarija lažiranja dvije lokacije postavljanjem napadača na pozicije satelita te lažiranja dvije lokacije s nepoznatim i poznatim pozicijama napadača, vrlo mali broj rezultata bio je bez pogreške. I za one najbolje rezultate ne može se reći da bi u potpunosti zavarali prijemnike, ali bi bili vrlo blizu tome. Najbolji rezultati, odnosno najmanje pogreške prikazane su u tablici 5.6.



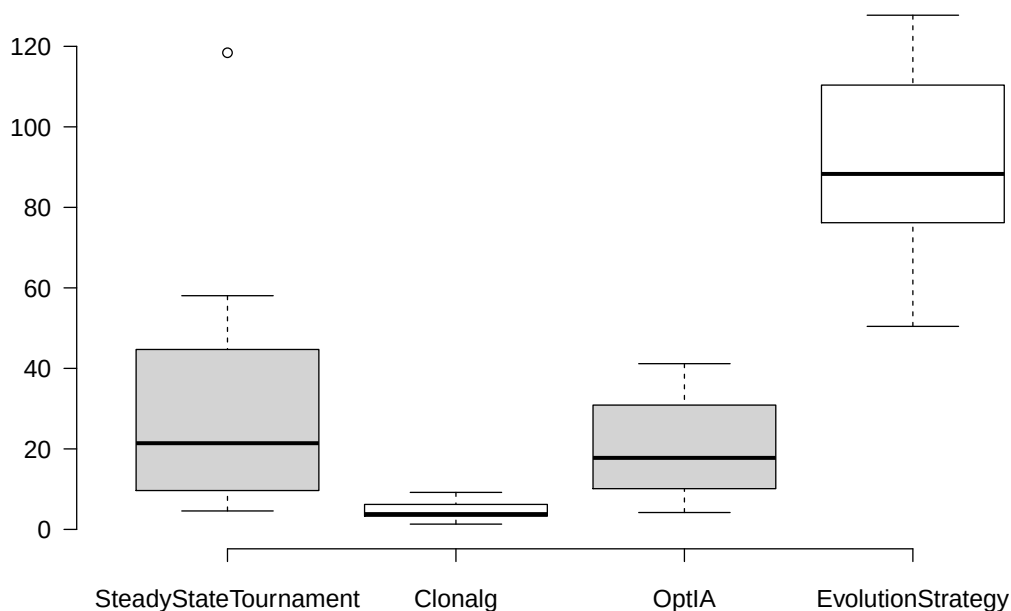
Slika 5.11: Rezultati I algoritama Steady State Tournament, Clonal Selection Algorithm, Immune Optimization Algorithm, Evolution Strategy te GenHookeJeeves za scenarij lažiranja dvije lokacije s četiri napadača nepoznatih lokacija.



Slika 5.12: Rezultati I algoritama Steady State Tournament, Clonal Selection Algorithm, Immune Optimization Algorithm te Evolution Strategy za scenarij lažiranja dvije lokacije s četiri napadača nepoznatih lokacija.

pozicija / koordinate	X	Y	Z
satelit 1	18351	19264	19845
satelit 2	19819	16447	19626
satelit 3	14543	16075	10943
satelit 4	12293	11532	18044
pozicija 1	6953	6459	6087
pozicija 2	6956	6461	6089
željena pozicija 1	6511	6675	6962
željena pozicija 2	6514	6677	6964

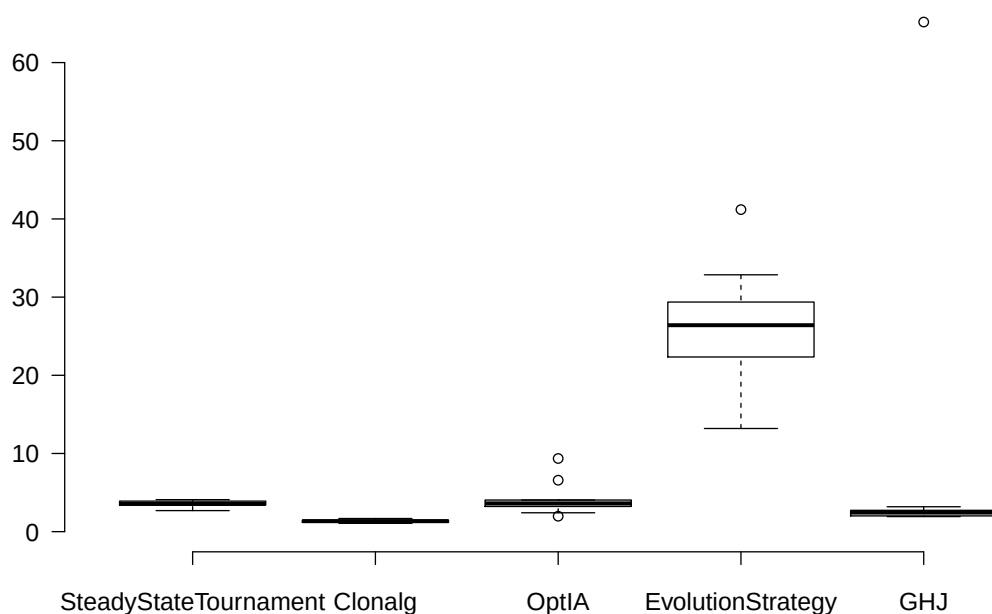
Tablica 5.4: Ulazni skup podataka II za scenarij lažiranja dvije lokacije s četiri napadača nepoznatih lokacija.



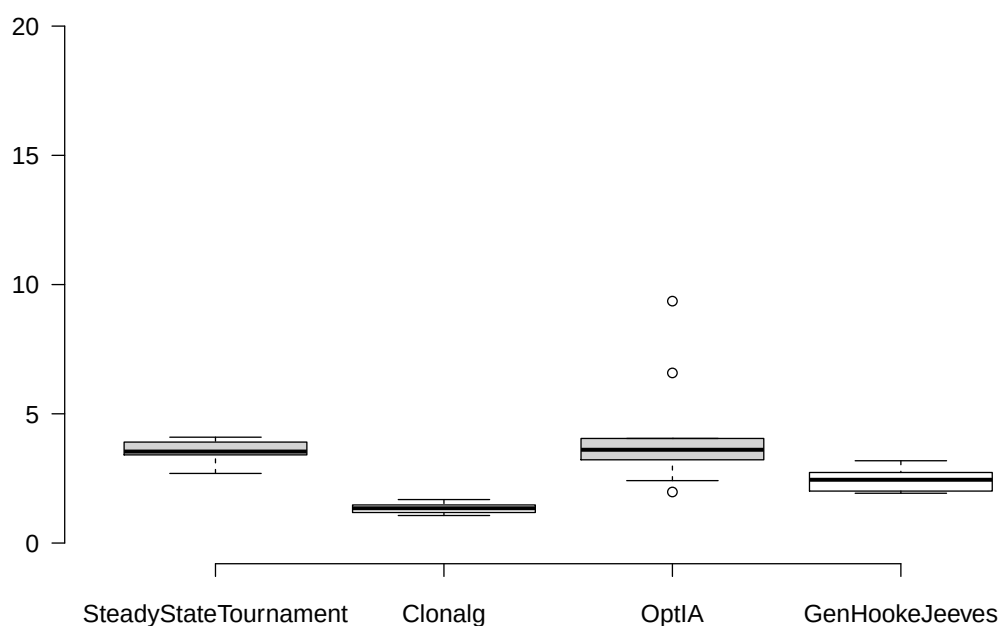
Slika 5.13: Rezultati II algoritama Steady State Tournament, Clonal Selection Algorithm, Immune Optimization Algorithm te Evolution Strategy za scenarij lažiranja dvije lokacije s četiri napadača nepoznatih lokacija.

pozicija / koordinate	X	Y	Z
napadač 1	12000	13200	12100
napadač 2	11000	14400	9300
napadač 3	15000	12000	8400
napadač 4	10700	16000	14100
pozicija 1	6055	6548	6943
pozicija 2	6056	6548	6945
željena pozicija 1	6002	6482	6976
željena pozicija 2	6003	6482	6978

Tablica 5.5: Ulazni skup podataka za scenarij lažiranja dvije lokacije s četiri napadača poznatih lokacija.



Slika 5.14: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm, Immune Optimization Algorithm, Evolution Strategy te GenHookeJeeves za scenarij lažiranja dvije lokacije s četiri napadača poznatih lokacija.



Slika 5.15: Rezultati algoritama Steady State Tournament, Clonal Selection Algorithm, Immune Optimization Algorithm te GenHookeJeeves za scenarij lažiranja dvije lokacije s četiri napadača poznatih lokacija.

scenarij / algoritam	SST	CLONALG	OptIA	ES	GHJ
lažiranje jedne lokacije	7.45811e-05	0.407172	0.160289	753.249	6.52226e-07
lažiranje dvije lokacije (pozicije satelita)	2.22983	1.65257	1.65565	1838.91	4.43678
lažiranje dvije lokacije (nepoznate pozicije)	4.58761	1.31632	4.19626	38.6677	1754.34
lažiranje dvije lokacije (poznate pozicije)	2.69282	1.0653	1.97603	13.1936	1.95609

Tablica 5.6: Najbolji rezultati za svaki implementirani scenarij po algoritmima. SST - Steady State Tournament, CLONALG - Clonal Selection Algorithm, OptIA - Immune Optimization Algorithm, ES - Evolution Strategy te GHJ - GenHookeJeeves.

6. Zaključak

U ovom radu, opisani su mehanizmi globalnog položajnog sustava. Posebna pažnja se posvetila načinu na koji prijemnik može, iz signala primljenih od satelita, izračunati svoju poziciju. Upoznavanjem tog načina računanja pozicije, mogu se uvidjeti mogućnosti zavaravanja istog. Temeljni cilj napadača je uvjeriti žrtvu da se nalazi tamo gdje on želi. Kako bi uspio u tome, svakim lažiranim signalom treba poslati takvu vremensku oznaku slanja tog signala, da žrtva dobije dojam kao da je taj signal propagirao drugačije nego stvarno jest.

Obrađena su dva glavna slučaja lažiranja lokacija, lažiranje jedne i više lokacija. Teorijski je utvrđeno da je lažiranje jedne lokacije poprilično jednostavan analitički problem, dok je lažiranje više lokacija mnogo kompliciranije. Za lažiranje više lokacija napadač mora koristiti više antena, a povećanjem broja prijemnika broj rješenja drastično opada.

Ukratko su opisani optimizacijski algoritmi. Oni se mogu podijeliti u dvije skupine, evolucijske i imunološke algoritme. Evolucijski algoritmi oponašaju biološku evoluciju, a imunološki funkcioniranje imunološkog sustava. Od evolucijskih algoritama koriste se Steady State Algorithm i Evolution Strategy, a od imunoloških Clonal Selection Algorithm te Immune Optimization Algorithm.

Uspješno su implementirani scenariji lažiranja jedne lokacije, lažiranja dvije lokacije s četiri napadača postavljanjem napadača na pozicije pretpostavljenih satelita, lažiranja dvije lokacije s četiri napadača nepoznatih i poznatih pozicija. Na svim scenarijima provedena su ispitivanja učinkovitosti algoritama. Kao najbolji algoritam istaknuo se Clonal Selection Algorithm.

Rezultati su zadovoljavajući. Zaključeno je da je moguće lažirati jednu lokaciju, kao i dvije lokacije, ali uz poprilično veću pogrešku i mnogo manje rješenja. Valja napomenuti kako bi stvarna implementacija uključivala mnogo strože uvjete koje treba zadovoljiti, kao i sklopovsku podršku bez koje lažiranje ne bi bilo moguće.

LITERATURA

- [1] HARNAM ARNEJA, ANDREW BENDER, S. J. T. R. Solving the gps equations. <http://mason.gmu.edu/~treid5/Math447/GPSEquations/>.
- [2] JAKOBOVIĆ, D. ECF - Evolutionary Computation Framework. <http://ecf.zemris.fer.hr/>.
- [3] NILS OLE TIPPENHAUER, CHRISTINA PÖPPER, K. B. R. S. C. On the requirements for successful gps spoofing attacks.
- [4] VINCENZO CUTELLO, GIUSEPPE NARZISI, G. N., AND PAVONE, M. Clonal selection algorithms: A comparative case study using effective mutation potentials.

Optimizacija parametara GPS napada temeljena na evolucijskim algoritmima

Sažetak

Civilni GPS osjetljiv je na napade lažiranjem signala. Lažiranje jedne lokacije vrlo je jednostavno ostvarivo jer se problem svodi na rješavanje sustava četiri jednačbe s četiri nepoznanice. Lažiranje više lokacija također je izvedivo, ali je dosta kompliciranije i može se dodatno otežati dodavanjem prijemnika koje treba zavarati. Evolucijski i imunološki algoritmi, optimizacijski su mehanizmi kojima se mogu ispitati mogućnosti lažiranja GPS signala. Lažiranje jedne lokacije je uspješno provedeno, dok je lažiranje dvije lokacije rezultiralo malim pogreškama. Najboljim se pokazao Clonal Selection Algorithm koji pripada imunološkim algoritmima. Stvarna implementacija bila bi mnogo zahtjevnija zbog potrebne dodatne sklopovske i programske podrške.

Ključne riječi: Optimizacija, GPS, napad, lažiranje, genetski, evolucijski, imunološki, algoritmi.

GPS spoofing parameters optimization based on evolutionary algorithms

Abstract

Civilian GPS is susceptible to spoofing attacks. Spoofing to one location is easily accomplished as the problem consists of solving a system of four equations and four unknowns. Spoofing to multiple locations can also be done, but it is quite complicated and it can be made even more difficult by installing additional receivers that need to be deceived. Evolutionary and immune algorithms are optimization mechanisms that can be used for examining the possibilities of GPS spoofing. The most successful was Clonal Selection Algorithm which belongs to immune algorithms. Real life implementation would be much more demanding due to necessary additional hardware and software support.

Keywords: Optimization, GPS, attack, spoofing, genetic, evolutionary, immune, algorithms.