SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Upravljanje svemirskim brodom pomoću genetskog programiranja

Hrvoje Ban

Voditelj: *Domagoj Jakobović*

Zagreb, svibanj 2011.

Sadržaj

Uvod

Postoji veliki skup problema za koje danas nisu poznati algoritmi koji ih mogu riješiti u razumnom vremenu ili koji opće mogu pronaći najbolje rješenje. Najpoznatiji primjeri za takve probleme su skupovi NP-potpunih (engl. *NP-complete*) i NP-teških (engl. *NP-hard*) problema za koje se smatra da ih nikada neće biti moguće efikasno rješavati.

Od posebnog su interesa tzv. optimizacijski problemi. Rješavanje tih problema zahtjeva pronalazak najboljeg rješenja iz skupa mogućih rješenja. Primjeri optimizacijskih problema su: raspoređivanje poslova (engl. *job scheduling*), krojenje (engl. *cutting stock*) i izrada rasporeda predavanja. Osim najboljeg rješenja, za optimizacijske probleme moguće je pronaći i druga rješenja koja su ispravna, ali lošija od najboljeg.

Ako se odustane od zahtjeva da je jedino prihvatljivo rješenje ono najbolje, mogu se razviti algoritmi koji nastoje pronaći dobra rješenja ili aproksimacije najboljeg rješenja. Takva rješenja su lošija od najboljeg mogućeg, ali mogu biti prihvatljiva za korištenje. Algoritmi koji traže takva rješenja nazivaju se heuristikama.

Jedna vrsta heuristike koja se može koristiti za optimizacijske probleme su genetski algoritmi čiji se rad temelji na oponašanju procesa evolucije u prirodi (prirodna selekcija, reprodukcija i mutacija). Korištenje genetskih algoritama za traženje računalnih programa naziva se genetskim programiranjem.

Ovaj rad podijeljen je u četiri poglavlja. U prvom poglavlju opisan je problem upravljanja svemirskim brodom. U drugom poglavlju opisan je način korištenja genetskog programiranja za traženje prihvatljivog rješenja. U trećem poglavlju prikazani su i obrazloženi rezultati. Na kraju je dan zaključak.

1 Opis problema

Rješenje problema upravljanja svemirskim brodom je računalni program koji može usmjeriti svemirski brod kroz skup kontrolnih točaka u unaprijed zadanom redoslijedu bez sudara s okolnim planetima. Brod cijelo vrijeme ubrzava u smjeru prema kojemu je rotiran dok ga planeti privlače gravitacijskim silama. S obzirom na to da nema trenja jedini način usporavanja broda jest ubrzavanje u smjeru suprotnom od smjera gibanja.

Tri vrste entiteta ovog problema su: planeti, kontrolne točke i svemirski brod koji su smješteni u 2D prostor. Koordinate svih entiteta nalaze se unutar intervala (0, 1000), s tim da svemirski brod može izaći iz tog područja tijekom svog gibanja. Za svaki planet poznata je njegova lokacija i polumjer. Za svaku kontrolnu točku poznata je njezina lokacija. Za svemirski brod poznata je njegova lokacija, brzina i kut za koji je rotiran.

Simulacija se odvija u diskretnim koracima, a 25 koraka čini jednu sekundu simulacije. Jedan korak simulacije odvija se na sljedeći način:

- Za svaki planet računa se vektor gravitacijske sile kojom planet djeluje na svemirski broj po Newtonovom izrazu za gravitaciju: $F = (G\ m_p\ m_b) / r^2$, gdje je r vektor udaljenosti između planeta i svemirskog broda, m_p masa planeta, a m_b masa broda. Radi jednostavnosti, brod i planeti imaju jediničnu masu. Za gravitacijsku konstantu G koristi se vrijednost 0,015;

- Poziva se algoritam za upravljanje brodom koji određuje novi kut rotacije broda. Iznos koji algoritam vraća shvaća se kao razlika trenutnog i željenog kuta. Najveća moguća promjena kuta u jednom koraku određena je tako da brodu treba barem jedna sekunda da se okrene za puni krug. Stoga najveća promjena kuta iznosi 360 / 25 = 14,4°;

- Računa se rezultanta svih sila koje djeluju na brod kao suma gravitacijskih sila planeta i sila kojom djeluje pogon broda. Sila pogona je vektor iznosa 0,24 zarotiran za isti kut kao i brod (brod ubrzava prema smjeru u kojem je okrenut);

- Kako brod ima jediničnu masu, vektor ubrzanja jednak je vektoru rezultante svih sila koje djeluju na brod;

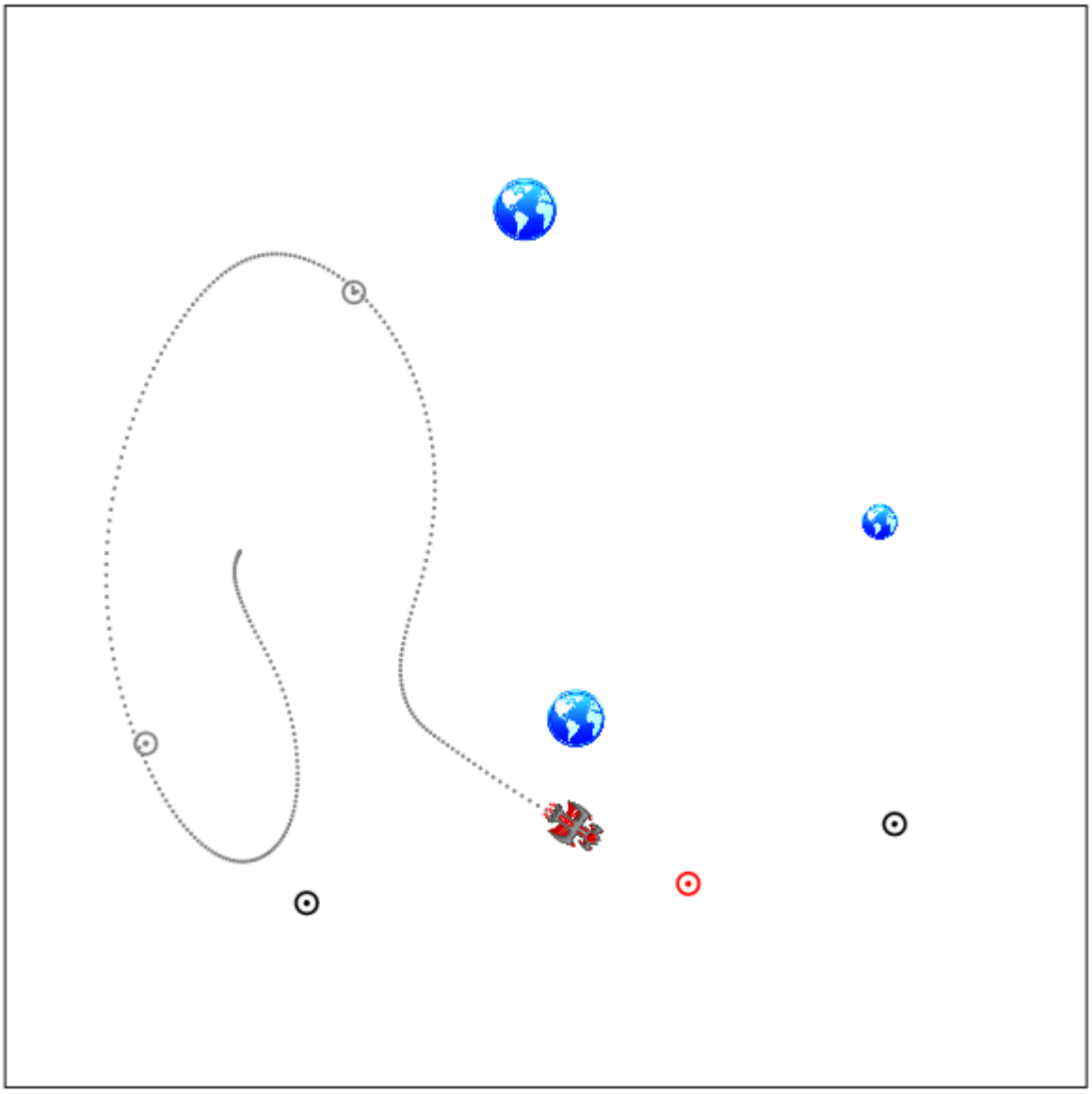
- Novi iznosi položaja i brzine računaju se Eulerovom metodom integracije. Novi vektor brzine dobiva se dodavanjem vektora ubrzanja na trenutni vektor brzine. Novi vektor položaja dobiva se dodavanjem novog vektora brzine na trenutni vektor položaja;

- Ako je udaljenost između broda i planeta manja ili jednaka polumjeru tog planeta došlo je sudara te se simulacije zaustavlja;

- Ako je udaljenost između broda i sljedeće kontrolne točke manja od 20 smatra se da je brod prošao kroz točku. Simulacija se zaustavlja kada brod prođe kroz posljednju kontrolnu točku.

Slika 1.1 prikazuje jedan ispitni slučaj koji se sastoji od 3 planeta i 5 kontrolnih točaka. Na početku simulacije brod miruje (iznos brzine je nula) i okrenut je prema prvog kontrolnoj točki. Točke kroz koje je brod već prošao označene su sivom bojom. Sljedeća kontrolna točka označena je crvenom bojom dok su sve ostale kontrolne točke označene crnom bojom. Plavi objekti različitih polumjera su planeti.

Putanja broda vizualizira se crtanjem sivih točkica koje predstavljaju položaj broda u pojedinim koracima simulacije. Točkice su gušće tamo gdje se brod gibao sporije (kao npr. kod skretanja), a rjeđe gdje se gibao brže.



Slika 1.1: Primjer gibanja broda kroz kontrolne točke

2 Parametri genetskog programiranja

Za rad genetskog algoritma korišteni su sljedeći parametri:

- Dopuštene dubine stabla početne generacije: 1 – 5;
- Dopuštene dubine stabla ostalih generacija: 1 – 10;
- Funkcijski (engl. *function*) čvorovi: `+`, `-`, `*`, `/`, `iflt`;
- Završni (engl. *terminal*) čvorovi: `td`, `ta`, `vm`, `va`, `npd`, `npa`, `ERC`;
- Operator selekcije: turnirska selekcija uz veličinu turnira 3;
- Operator križanja: jednostavno križanje uz vjerojatnost od 90%;
- Operatori mutacije: mutacija podstabla (25%), mutacija smanjivanjem (25%), mutacija konstanti dodavanje Gaussovog šuma (50%);
- Veličina populacije: 1500 jedinki;
- Broj generacija: 100.

Funkcijski čvorovi `+`, `-` i `*` su standardni operatori zbrajanja, oduzimanja i množenja. Čvor `/` je operator zaštićenog dijeljenja. Čvor `iflt` je operator grananja koji prima četiri argumenta: ako je vrijednosti prvog argumenta manja od vrijednosti drugog rezultat je treći argument, inače je četvrti.

Završni čvorovi, osim čvora `ERC`, sadrže informacije o svemirskom brodu i njegovoj okolini:

- `td` (*target distance*): udaljenost do sljedeće kontrolne točke;
- `ta` (*target angle*): kut prema sljedećoj kontrolnoj točki tj. kut za koji se brod treba zarotirati da bi bio okrenut prema toj točki;
- `vm` (*velocity magnitude*): iznos brzine kojom se brod trenutno giba;
- `va` (*velocity angle*): kut prema smjeru gibanja;
- `npd` (*nearest planet distance*): udaljenost do najbližeg planeta;
- `npa` (*nearest planet angle*): kut prema najbližem planetu.

Uz navede, kao završni čvorovi koriste se i `ERC` konstante (*ephemeral random constant*) koje se biraju iz intervala (-10, 10).

Aritmetičke operacije su izabrane jer su jednostavne i po intuiciji potrebne za rješavanje ovog problema. Operator grananja dodan je kako bi se povećala ekspresivnost programa jer određene funkcionalnosti nije moguće ostvariti samo aritmetičkim operacijama.

Pri izboru završnih čvorova nastojalo se izabrati one informacije koje će biti najkorisnije za upravljanje brodom. Podaci o sljedećoj kontrolnoj točki važni su kako bi program mogao usmjeriti brod prema njoj. Podaci o najbližem planetu važni su kako bi ga brod mogao zaobići i time izbjeći sudar. Kako program za upravljanje određuje novi kut broda te samim time novi iznos i smjer brzine, podaci o trenutnoj brzini su također bitni. Konstante su uključene kako bi program mogao formirati razne faktore koji su mu potrebni za rad.

2.1 Računanje dobrote programa

Dobrota programa računa se na temelju njegovih performansi na pet ispitnih slučajeva. Za svaki ispitni slučaj simulacija se izvršava za najviše 2500 koraka (100 sekundi) iako može završiti ranije ako dođe do sudara s planetom ili ako brod prođe kroz sve kontrolne točke. Dobrota programa boduje se na temelju dva kriterija:

- Prvih 1000 bodova program može dobiti na temelju broja točaka kroz koje je uspješno usmjerio brod. Broj bodova računa se kao omjer između broja točaka kroz koje je brod prošao i ukupnog broja točaka, pomnoženo s 1000.
- Program koji je uspješno usmjerio brod kroz sve točke može dobiti do 1000 dodatnih bodova na temelju vremena koje mu je za to trebalo, tj. ako je broj koraka simulacije manji od maksimalnog i ako nije došlo do sudara. Broj dodatnih bodova računa se kao omjer broja preostalih koraka i maksimalnog broja koraka, pomnoženo s 1000.

Na primjer, program koji brod usmjeri kroz pola točaka dobiti će 500 bodova. Program koji brod usmjeri kroz sve točke u maksimalnom broju koraka dobiti će 1000 bodova, a program koji to uspije za pola od maksimalnog broja koraka dobiti će 1500 bodova.

Kako se koristi više ispitnih slučajeva, konačna dobrota je aritmetička sredina dobrota dobivenih za pojedine slučajeve, uz jedno ograničenje. Dobrota za jedan slučaj ne može biti za više od 250 bodova veća od dobrote dobivene za najlošiji slučaj. Ovo ograničenje je uvedeno kako bi se spriječila specijalizacija za pojedine slučajeve nauštrb drugih.

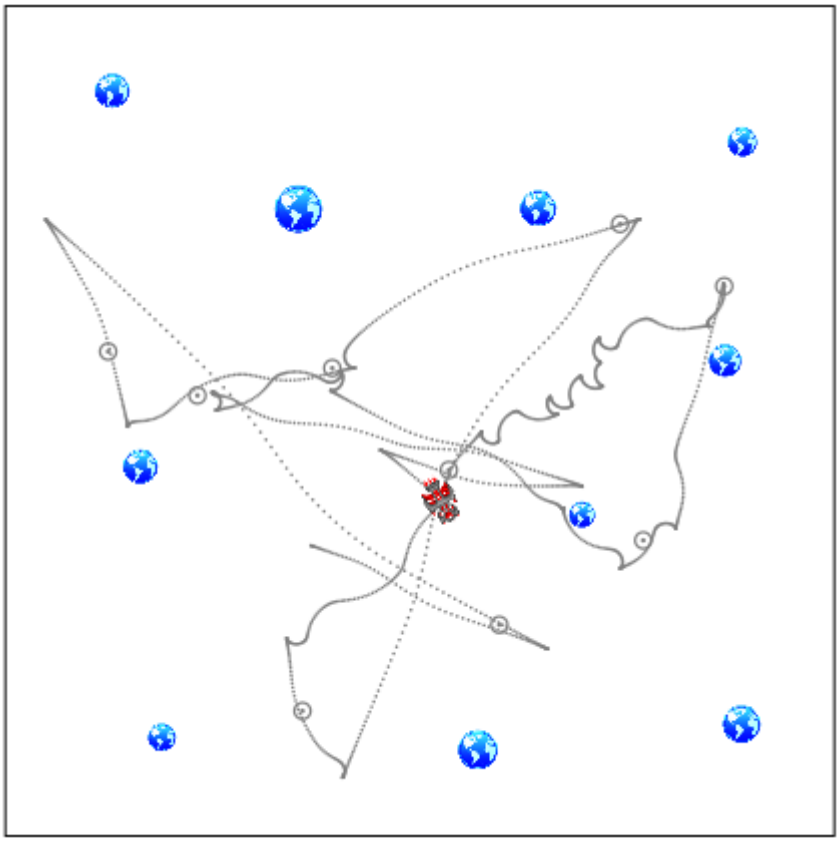
3 Rezultati

Svaki od pet ispitnih slučajeva sastoji se od 9 ili 10 planeta i 10 kontrolnih točaka, slučajno smještenih u prostoru. Nakon pokretanja genetskog programiranja, u početnoj generaciji pojavljuje se jedinka koja ima iznos dobrote 500, što znači da može uspješno proći kroz 5 kontrolnih točaka. Srednja vrijednosti dobrote cijele populacije iznosi vrlo niskih 18,5.

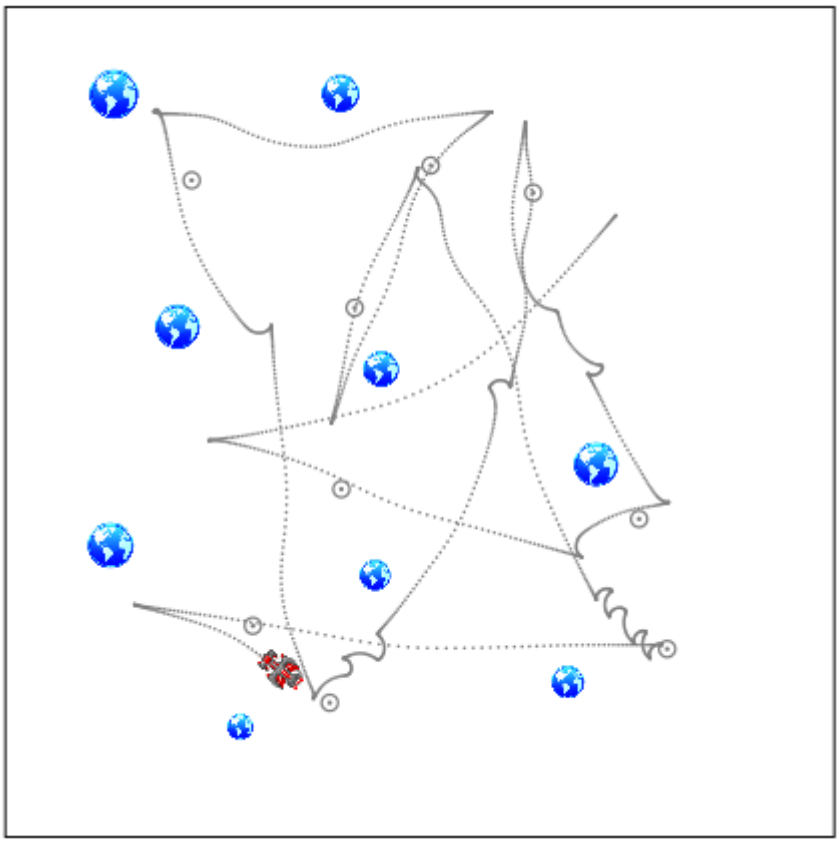
U 18. generaciji pojavljuje se jedinka koja ima iznos dobrote malo veći od 1000 (1037) što znači da može uspješno proći kroz sve kontrolne točke, ali joj za to treba skoro cijelo dozvoljeno vrijeme simulacije. Srednja vrijednosti dobrote cijele populacije iznosi 687.

Nakon toga iznos dobrote najbolje jedinke povećava se dosta sporijim tempom. Tek se u 72. generaciji pojavljuje najbolja jedinka s iznosom dobrote 1458 kojoj u prosjeku trebaju 54 sekunde za prolazak kroz sve kontrolne točke. Ova jedinka ostaje najbolja i u narednih 28 generacija nakon čega je zaustavljeno izvođenje genetskog programiranja.

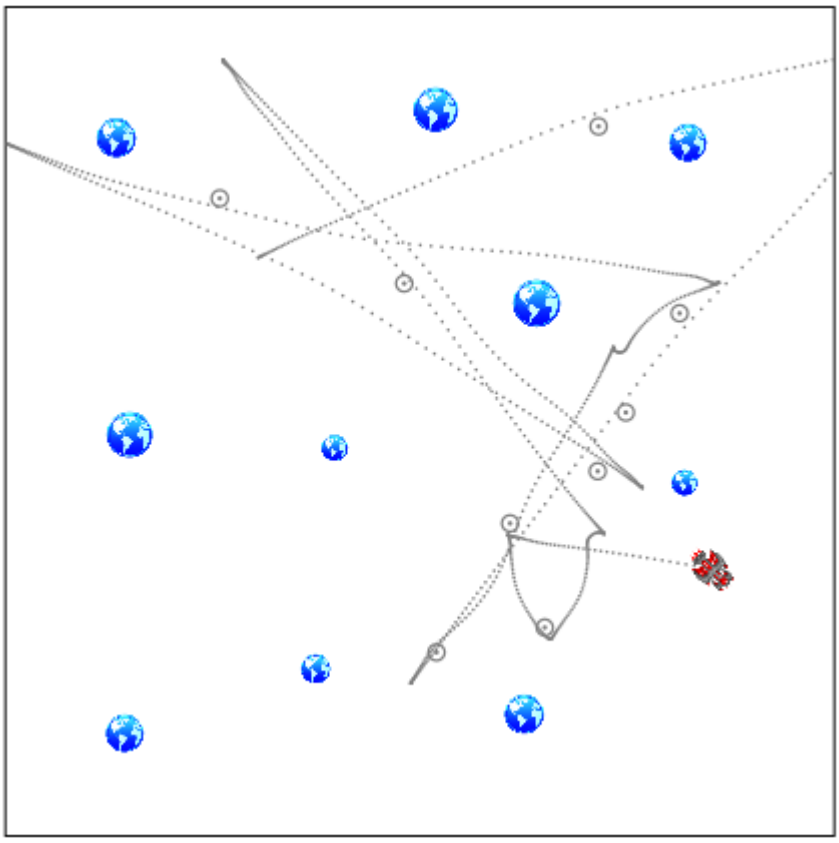
Slika 3.1 prikazuje gibanje kroz prvi ispitni slučaj koje je trajalo 62 sekunde. Slika 3.2 prikazuje gibanje kroz drugi ispitni slučaj koje je trajalo 63,5 sekundi. Slika 3.3 prikazuje gibanje kroz treći ispitni slučaj koje je trajalo 41,2 sekunde. Slika 3.4 prikazuje gibanje kroz četvrti ispitni slučaj koje je trajalo 47,6 sekundi. Slika 3.5 prikazuje gibanje kroz peti ispitni slučaj koje je trajalo 50,4 sekundi.



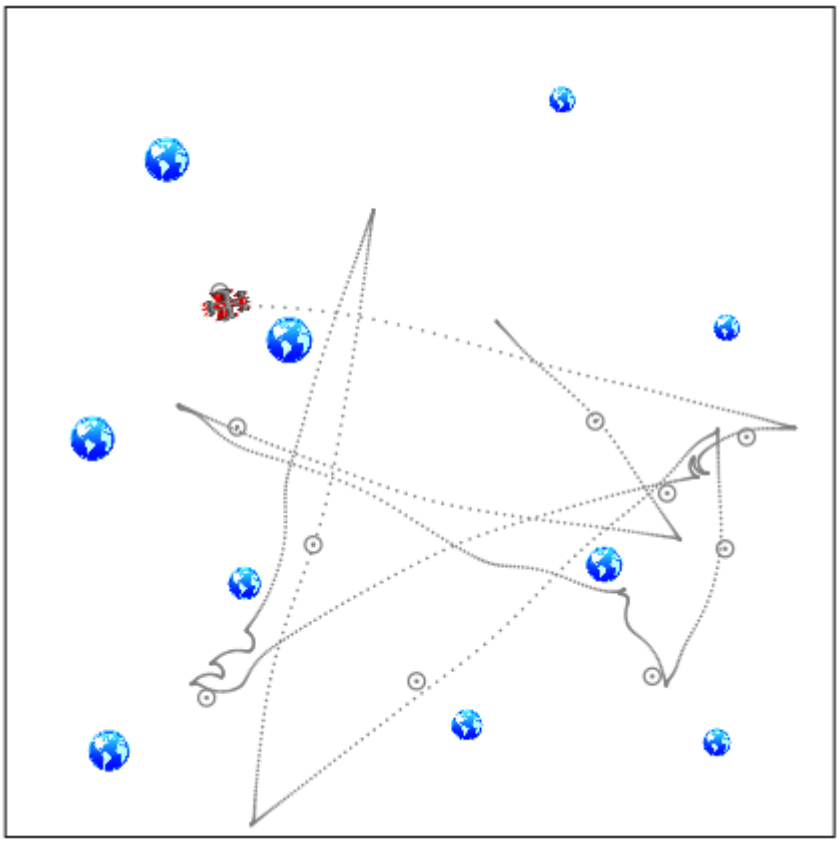
Slika 3.1: Prvi ispitni slučaj



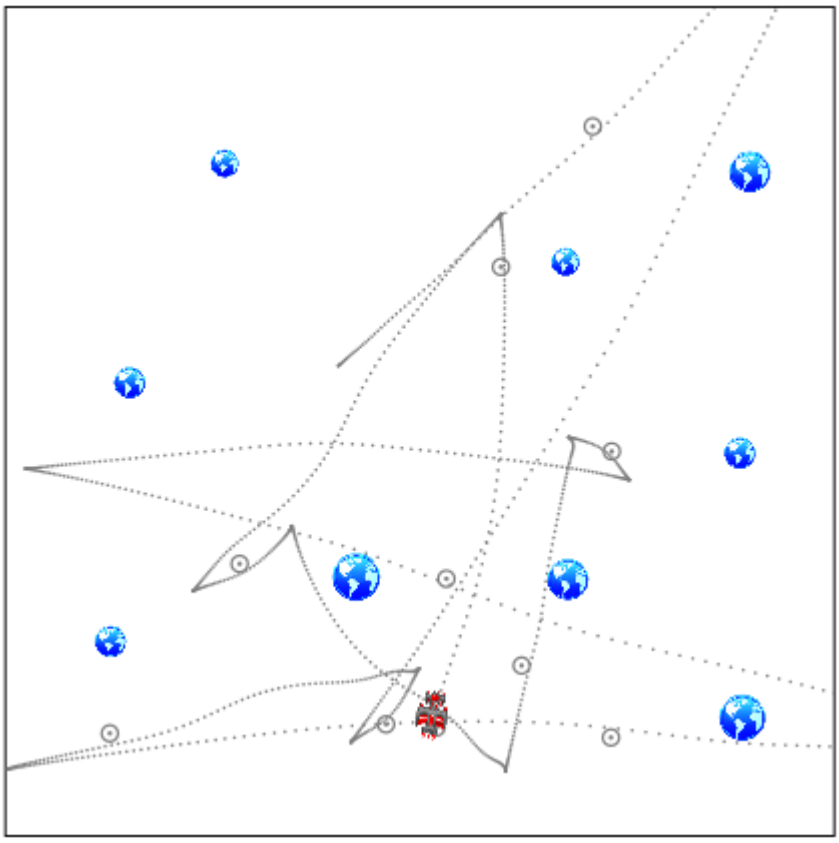
Slika 3.2: Drugi ispitni slučaj



Slika 3.3: Treći ispitni slučaj



Slika 3.4: Četvrti ispitni slučaj



Slika 3.5: Peti ispitni slučaj

Kod programa dobivenog genetskim programiranjem postoje tri uočljiva obrasca ponašanja:

- U većini slučajeva program usmjerava brod izravno prema sljedećoj kontrolnoj točki kompenzirajući pomake koji bi nastali zbog gravitacijskog privlačenja planeta;
- Kada se nađe relativno blizu sljedeće točke, program okreće brod prema suprotnom smjeru čime usporava njegovo gibanje. To usporavanje često počinje prekasno da bi se brod mogao zaustaviti točno nakon kontakta s točkom zbog čega će brod preletjeti preko točke. Kada brod, osim što preleti također i promaši točku, program će izgubiti određeno vrijeme da brod vrati nazad. Veliki preleti preko kontrolnih točaka najprimjetniji su u trećem ispitnom slučaju.
- U nekim situacijama program će, bez očitog razloga, drastično usporiti brod iako u blizini nema planeta ili sljedeće kontrolne točke. Nakon par (uzaludno potrošenih) sekundi, program će nastaviti s normalnim gibanjem. Ovaj obrazac ponašanja najviše je primjetan u prvom ispitnom slučaju.

Iz čudnovatog gibanja u pojedinim situacijama te iz činjenice da program pokazuje dosta bolje ponašanje u pojedinim ispitnim slučajevima može se ustanoviti da dolazi do određene razine specijalizacije programa. Bolje upravljanje brodom u nekim situacijama kao protutežu ima lošije ponašanje u nekim drugim situacijama.

Konačno, primjećuje se relativno loša generalizacija programa. Iako sve ispitne slučajeve nad kojima je treniran može proći bez većih problema, kod drugih slučajeva, koji nisu korišteni u postupku učenja, često ili dođe do sudara s planetom ili programu treba dosta dugo da prođe kroz sve kontrolne točke. Slika 3.6 prikazuje gibanje broda za novi ispitni slučaj. Program prvo praktički zaustavlja brod u blizini jedne točke te mu treba dosta dugo da prođe kroz nju (što se vidi po čudnom obliku putanje u blizini te točke). Pri kretanju prema predzadnjoj točki program ne uspijeva izbjeći planet koji se nalazi između broda i točke te dolazi do sudara.

3.1 Alternativni parametri

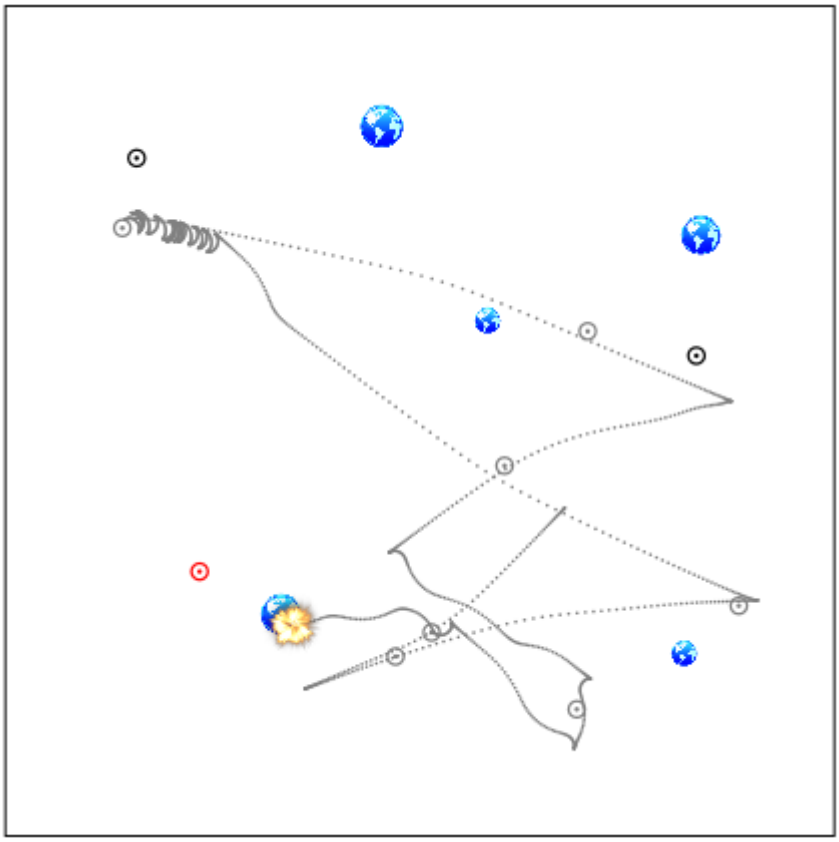
Parametri genetskog programiranja mijenjani su na nekoliko načina u nadi da će uspjeti proizvesti bolji program za upravljanje svemirskim brodom:

- Zamjena `ERC` konstanti s malim skupom unaprijed zadanih konstanti. Genetskom programiranju na raspolaganje su stavljane konstante 2, 5, 100 i π. Nove konstante nisu polučile nikakve razlike u ponašanju genetskog programiranja.
- Dodavanje trigonometrijskim funkcija u skup funkcijskih čvorova. Kako se među završnim čvorovima pojavljuju razni kutovi, trigonometrijske funkcije izgledale su kao koristan dodatak. Programi koji koriste ove funkcije pokazali su osjetno lošije ponašanje. To može biti pokazatelj da te funkcije nisu potrebne niti korisne za rješavanje ovog problema.

- Dodavanje automatski definiranih funkcija. Umjesto jednog razvijaju se dva programa od kojih drugi služi kao potprogram prvome. Oba programa imaju pristup istim podacima (završnim čvorovima). Prvi program na raspolaganju ima dodatni funkcijski čvor koji poziva potprogram. Ideja je bila da se određena logika koja se često koristi može razviti kao poseban potprogram. Ni korištenjem ove tehnike nisu dobiveni bolji programi. Dapače, najbolja jedinka je bila ona koja opće nije koristila potprogram!

- Povećavanjem najveće dopuštene dubine stabla na 20. Dobiveni programi su dosta veći, ali ne pokazuju bolje ponašanje. Rezultat je dosta sporije izvršavanje genetskog programiranja i sporija konvergencija.

- Smanjivanjem veličine populacije. S obzirom da pokušaji dobivanja boljeg rješenja nisu uspjeli, pokušalo se ubrzati konvergenciju korištenjem manje populacije. Iako je izvođenje genetskog programiranje ubrzano, ono često konvergira prema rješenju s vrlo malom dobrotom.



Slika 3.6: Ispitni slučaj nad kojim program nije treniran

4 Zaključak

U ovom radu prikazan je način korištenja genetskog programiranja za rješavanje problema upravljanja svemirskim brodom. Demonstrirano je da je moguće pronaći program koji može uspješno upravljati brodom, tj. proći kroz sve kontrolne točke bez sudara s planetima.

Za skup korištenih ispitnih slučajeva pronađen je program koji može brod usmjeriti kroz sve kontrolne točke u razumnom vremenu. Korištenjem programa za ispitne slučajeve nad kojima nije treniran ustanovljeno je da ih program ne može dobro riješiti tj. da ima loše svojstvo generalizacije.

Pokušaji dobivanja boljeg programa korištenjem drugih parametara genetskog programiranja nisu uspjeli. Najbolje rješenje dobiva se korištenjem malog skupa jednostavnih funkcijskih čvorova, uz malu najveću dopuštenu dubinu stabla i relativno veliku populaciju.

5 Literatura

[1] KOZA, J.R., RICE, J.P. *Genetic Programming*, MIT Press: Cambridge, 1992.

[2] POLI, R., LANGDON, W.B., MCPHEE, N.F. *A Field Guide to Genetic Programming*, Lulu Enterprises: Ujedinjeno Kraljevstvo, 2008.