
Vuk Vuković

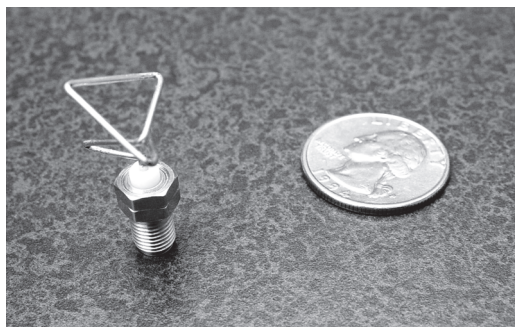
Razvoj točka od slučajno generisanih mnogouglova primenom genetskih algoritama

Ispitivano je da li se primenom genetskih algoritama od slučajno generisanih mnogouglova može razviti pravilan mnogougao (točak). Genetski algoritmi su tehnika u programiranju koja se oslanja na biološke koncepte selekcije, ukrštanja i mutacije. Inicijalno, vrši se slučajno generisanje generacije mnogouglova. Ovim mnogouglovima se predaje početni impuls i računa se njihov pređeni put na zadatom terenu. Nakon sortiranja po kriterijumu pređenog puta, vrši se primena genetskih algoritama: selekcija (zadržavanje najboljih mnogouglova), mutacija (slučajna promena mnogouglova), ukrštanje (stvaranje dva nova mnogougla kombinacijom dva trenutna) i generisanje novih jedinki. Ovaj proces se ponavlja iznova i iznova. Očekivani ishod je konvergencija ka pravilnom mnogouglu (točku), što je u radu i pokazano.

Uvod

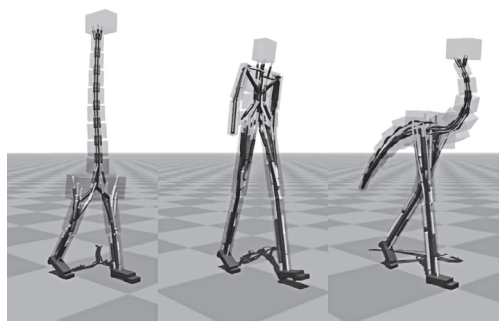
Genetski algoritmi su tehnika koja je inicijalno počela da se razvija još od vremena Alana Tjuringa (1950-ih godina), dok se danas koristi za pronalaženje rešenja problema iz najrazličitijih oblasti nauke. Jedna od najpoznatijih primena ove tehnike jeste generisanje antene za svemirsku letelicu koju je NASA kreirala 2006. godine (slika 1).

U jednom radu iz ove tematike koji se bavi simulacijom kretanja dvonožnih bića pomoću 3D pokreta i mišića (Geijtenbeek *et al.* 2013), genetski algoritmi su korišćeni kako bi jedinka doslovno naučila da se kreće (slika 2). U zavi-



Slika 1. Antena za svemirsku letelicu NASA ST5

Figure 1. NASA ST5 spaceship antenna



Slika 2. Prikaz simulacije iz referentnog rada (Geijtenbeek *et al.* 2013)

Figure 2. Simulation done in the referenced paper (Geijtenbeek *et al.* 2013)

snosti od bića, brzine kretanja i nagiba terena, dobijeni su rezultati koji su prikazani u tabeli 1. Najbitnija informacija jeste broj generacija nakon kojih je jedinka naučila da se pravilno kreće i ta informacija može se iskoristiti za poređenje sa ovim radom.

Vuk Vuković (1998), Beograd, Trajka Stamenkovića 1, učenik 3. razreda Matematičke gimnazije u Beogradu

MENTOR: Dragan Toroman, Istraživačka stanica Petnica

Tabela 1. Zavisnost brzine, nagiba terena i broja generacije od modela (Geijtenbeek *et al.* 2013)

Model	Brzina	Nagib terena	Broj generacija
Čovek	2.0 m/s	$\pm 5^\circ$	197
Čovek	4.0 m/s	$\pm 3^\circ$	760
Noj	1.2 m/s	$\pm 5^\circ$	698
Dug vrat	1.5 m/s	$\pm 5^\circ$	541
Duge noge	2.0 m/s	$\pm 3^\circ$	403

Tema ovog rada jeste još jedna od primena genetskih algoritama: da li bi se korišćenjem principa „prirodne” evolucije mogao hipotetički razviti točak.

Genetski algoritmi su tehnika koja se koristi za rešavanje problema optimizacije. Inspirisani su biološkim konceptima evolucije. Upravo zbog zasnovanosti na biološkim konceptima se i u programiranju koriste termini kao što su jedinka (hromozom) i generacija.

Inicijalno, vrši se slučajno generisanje generacije koja se sastoji od određenog broja kandidata za rešenje (jedinki, hromozoma). Ove jedinke se zatim evaluiraju zadatom funkcijom. Naredne generacije potiču od incijale primenom postupaka selekcije, ukrštanja i mutacije. Na ovaj način, u svakoj sledećoj generaciji rešenje konvergira ka boljem.

Selekcija predstavlja zadržavanje određenog dela prethodne generacije koja ima najbolji rezultat. Ukrštanje je proces u kom dolazi do razmene karakteristika dve jedinke i nastajanja dve nove. Svako dete preuzima deo karakteristika od jednog roditelja, a ostatak od drugog. Mutacijom se vrši slučajna promena određenih karakteristika roditelja čime nastaje dete koje postaje deo nove generacije. Zajedno sa stvaranjem slučajno generisanih jedinki u svakoj generaciji, mutacije se koriste kako bi se izbegli problemi lokalnog maksimuma.

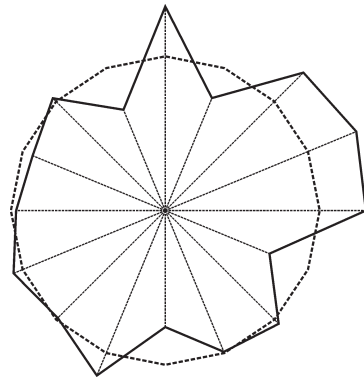
U ovom problemu, za početak se slučajno generišu mnogouglovi koji se uz predaju početnog momenta impulsa kreću po zadatom terenu. Sortiranjem mnogouglova (u odnosu na pređeni put) i primenom genetskih algoritama (procesa selekcije, ukrštanja i mutacije) dolazi do promena ovih mnogouglova. Pretpostavljeni ishod nakon ponavljanja ovog procesa određeni broj puta jeste konvergencija ka pravilnom mnogo-

uglu (koji je ujedno najbliži točku, a samim tim i najpogodniji za kretanje). Za potrebe izračunavanja kretanja i simulacije korišćena je biblioteka Box2D.

Metod

Problem je rešavan u dvodimenzionalnom prostoru. Svakom mnogouglu je zadavana početna sila i moment impulsa kako bi započeo kretanje. Između mnogougla i podloge je postojalo trenje.

Mnogougao. Mnogougao je definisan kao udeo odstupanja od pravilnog mnogougla (slika 3), tj. koliko dužina koja spaja centar sa temenom mnogougla (poluprečnik opisanog kruga) svakog temena odstupa od iste dužine u pravilnom.



Slika 3. Odstupanje mnogougla od pravilnog

Figure 3. Generated olygon's deviation from a regular one

Teren. U istraživanju su korišćene dve vrste terena – teren sa uzvišenjima i udubljenjima i ravan teren (slika 4).

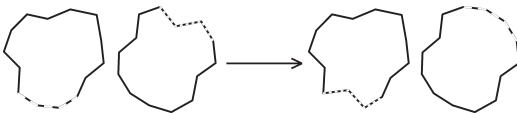


Slika 4. Dve vrste terena

Figure 4. Two types of surface

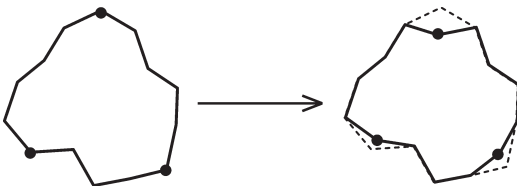
Selekcija. Selekcija predstavlja zadržavanje dela prethodne generacije koja ima najbolji rezultat, u ovom slučaju to je pređeni put. U okviru ovog istraživanja, pretpostavka koju takođe proveravamo jeste da li duži pređeni put direktno zavisi od toga koliko je mnogougao blizak pravilnom.

Ukrštanje. Princip ukrštanja koji se pokazao kao najpogodniji za ovaj problem jeste ukrštanje u dve tačke sa pomeranjem. Svako dete preuzima deo karakteristika od jednog roditelja, a ostatak od drugog. Slučajno se određuju tri parametra: početak segmenta za razmenu na prvoj jedinci, početak segmenta za razmenu na drugoj jedinci i dužina segmenata (broj temena) koji se razmenjuju. Razmenom ovih karkteristika nastaju dve nove jedinke (slika 5).



Slika 5. Ukrštanje

Figure 5. Crossover



Slika 6. Mutacija

Figure 6. Mutation

Mutacija. Mutacija je implementirana tako da za svako teme postoji verovatnoća promene.

Promena se vrši slučajnim generisanjem nove vrednosti odstupanja za to teme (slika 6).

Pored ovih procesa, u svakoj sledećoj generaciji se javlja i određeni udeo novih jedinki.

Za mnogougao i teren po kom se on kreće korišćene su klase `BodyDef`, `Body`, `ChainShape` i `FixtureDef` iz biblioteke `Box2D`.

Implementacija

Struktura projekta

Za implementaciju projekta korišćen je programski jezik Java u Eclipse razvojnom okruženju. Takođe je korišćena i biblioteka `Box2D` u okviru `libGDX` frameworka.

Projekat se sastoji iz sledećih klasa:

- `Settings.java` – klasa koja sadrži sve javne konstante kojima se može pristupiti iz bilo koje od ostalih klasa (broj temena mnogougla, broj generacija, verovatnoća mutacije itd);
- `PublicRandom.java` – klasa koja omogućava generisanje slučajnih celih i decimalnih brojeva u zadatim granicama iz svih ostalih klasa;
- `TerrainGenerator.java` – klasa koja generiše niz sa koordinatama terena i u njoj se zadaje da li će teren biti ravan ili brdovit;
- `Polygon.java` – klasa u kojoj se čuva niz prethodno pomenutih odstupanja; Pored ovog niza kao atributa, postoje i metode koje vrše mutaciju, ukrštanje, slučajno generišu novi mnogougao ili računaju koeficijent odstupanja;
- `Chromosome.java`, `PairChromosome.java`, `Generation.java`, `PairPolygon.java`, `Result.java`, `PolygonPoint.java` – organizacione klase bez funkcionalnog značaja;
- `Calculate.java` – pomoću biblioteke `Box2D` za prosleđeni mnogougao kao povratnu informaciju daje pređeni put tog mnogougla;
- `Simulation.java` – pomoću biblioteke `Box2D` vrši iscertavanje i simulaciju kretanja zadatog mnogougla (u svrhu praćenja napretka, tj. razvoja mnogouglova kroz generacije).

Uprošćeni kodovi najznačajnijih delova

Primena genetskog algoritma

```
1. for (int i=0; i < broj_generacija;i++)
2. {
3.     for (int j=0; j < veličina_generacije; j++)
4.         jedinke[j].izračunajPreдениPut();
5.
6.     sortiranje(); //po pređenom putu
7.     iscrtajNajbolju();
8.
9.     ukrštanje();
10.    mutacija();
11.    generišiNoveJedinke();
12. }
```

Generisanje slučajnog mnogougla

```
1. for (int i=0; i < broj temena; i++)
2. {
3.     teme[i] = randomDouble(-1,1)*max_odstupanje;
4. }
```

Mutacija

```
1. for (int i=0;ii++)
2. {
3.     if (randomDouble(0,1) verovatnoća_mutacije_temena)
4.         teme[i] = randomDouble(-1,1)*max_odstupanje;
5. }
```

Ukrštanje

```
1. int start = randomInt(0, broj_temena-1);
2. int dužina = randomInt(min_dužina, max_dužina);
3. int pomeranje = randomInt(min_pomeranje, max_pomeranje);
4.
5. for (int i=start;i ina;i++)
6. {
7.     int t1 = i%N;
8.     int t2 = (i+pomeranje)%N;
9.     novi_mnogougao1.teme[t2] = mnogougao2.teme[t1];
10.    novi_mnogougao2.teme[t1] = mnogougao1.teme[t2];
11. }
```

Rezultati i diskusija

Rezultati su dobijeni pri sledećoj konfiguraciji promenljivih:

- Broj temena mnogougla: 16;
- Trenje: 0.6;
- Gravitacija: 13;
- Gustina mnogougla: 10;
- Početni ugaoni impuls: -700 ;
- Početna sila: 1300;
- Veličina generacije: 100 jedinki;
- Broj generacija: 100;
- Maksimalno odstupanje poluprečnika opisanog kruga svakog temena: 40%;
- Verovatnoća promene temena tokom mutacije: 0.64;
- Selekcija: 40%;
- Ukrštanje: 25%;
- Mutacija: 25%;
- Nove jedinke: 10%.

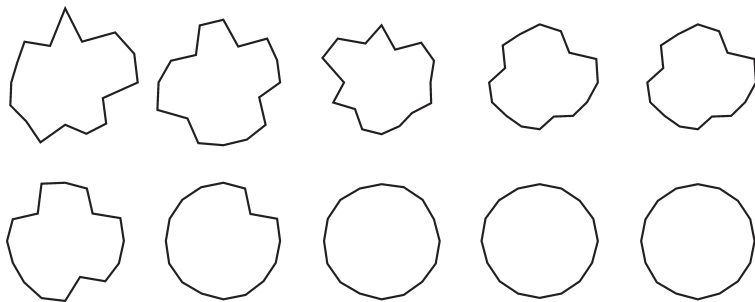
Na slikama 7 i 8 prikazan je razvoj najboljih jedinki u svakoj desetoj generaciji na dve vrste terena (ravnom i brdovitom).

Sa grafika na slici 9 zaključuje se da se razvoj ka pravilnom mnogouglu odvijao sličnom brzinom na oba tipa terena. Takođe, što je mnogougao više težio ka pravilnom, to mu je bilo potrebno više vremena da pređe u sledeći bolji stadijum što pokazuju horizontalne linije na grafiku. Ovo se najbolje uočava između generacija 60 i 90, gde su se javili veći regioni sa konstantnim pređenim putem. Sa druge strane, promene u početku dešavaju se mnogo brže.

Rezultati pokazuju da je konačan ishod na oba terena bio približno jednak na uzorku od 100 generacija.

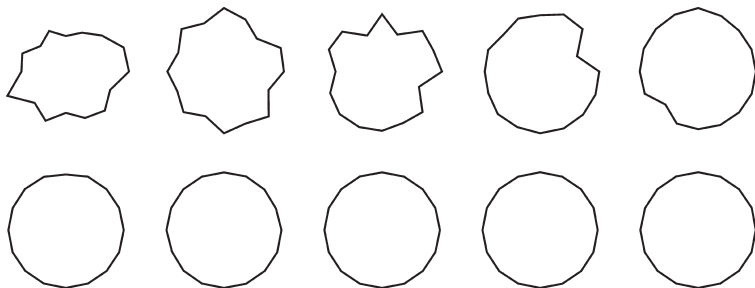
U poređenju sa referentnim radom (Geijtenbeek *et al.* 2013), potreban je manji broj generacija kako bi se dostiglo optimalno rešenje. Ovo je bilo i za očekivati s obzirom na manju kompleksnost ovog problema u odnosu na kretanje dvožnih modela.

Radi jednostavnijeg i objektivnijeg poređenja mnogouglova, uvodi se koeficijent odstupanja k poluprečnika opisanog kruga r_s od pravilnog mnogougla, koji je definisan kao standardna devijacija ovih poluprečnika:



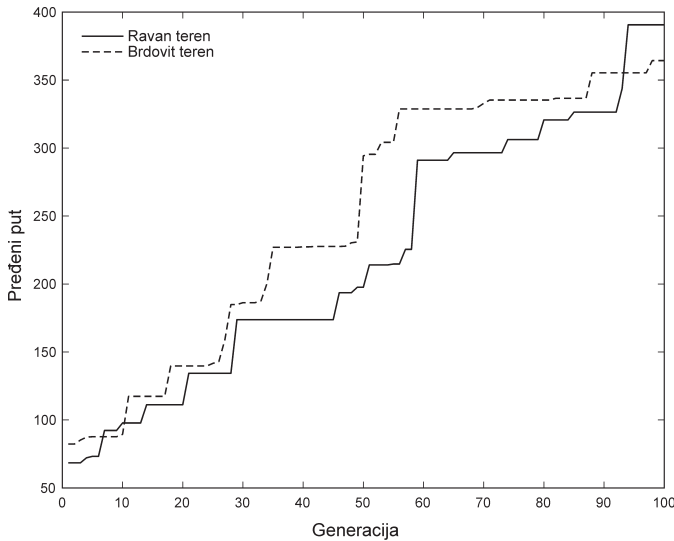
Slika 7. Napredak najboljih jedinki kroz generacije na ravnom terenu

Figure 7. Development of the best polygon through generations on a flat surface



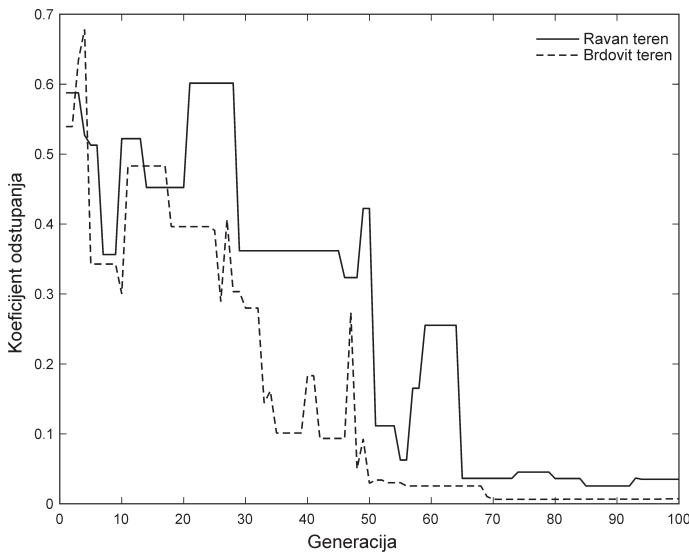
Slika 8. Napredak najboljih jedinki kroz generacije na brdovitom terenu

Figure 8. Development of the best polygon through generations on a hilly surface



Slika 9. Zavisnost pređenog puta najbolje jedinke od generacije

Figure 9. Number of generation versus the distance traveled by the best polygon (full line – flat surface, dotted line – hilly surface)



Slika 10. Zavisnost koeficijenta odstupanja najbolje jedinke od broja generacije

Figure 10. Number of generation versus the deviation coefficient of the best polygon (full line – flat surface, dotted line – hilly surface)

$$r_s = \frac{\sum_{i=1}^N r_i}{N}, \quad k = \sqrt{\frac{\sum_{i=1}^N (r_s - r_i)^2}{N}}$$

Iako je bilo za očekivati da koeficijent odstupanja konstantno opada, tokom generacija došlo je do nekoliko skokova ove promenljive, što pokazuje grafikon na slici 10. To se objašnjava činjenicom da određeni mnogouglovi koji su približniji pravilnom prelaze manji put od onih koji su manje približni. Ovakve jedinice (mnogouglovi) predstavljaju lokalne maksimume koji se uz pomoć mutacije i ukrštanja prevazilaze

u narednim generacijama. Ovo pokazuje da postoje slučajevi kada pređeni put ne zavisi direktno od odstupanja od pravilnog mnogougla, međutim ti slučajevi samo usporavaju potragu ka konačnom globalnom maksimumu.

Zaključak

Na osnovu dobijenih rezultata prema kojima je mnogougao, prilikom kretanja na obe vrste terena, ravnom i brdovitom, već posle 60 generacija dostigao približno pravilan oblik, može se

zaključiti da se primenom genetskih algoritama može razviti točak od slučajno generisanih mnogouglova. Iz generacije u generaciju, mnogouglovima je trebalo sve više vremena da se poboljšaju i pređu duži put. Takođe, bitno je napomenuti da pređeni put nije u direktnoj zavisnosti sa oblikom mnogougla, jer postoje slučajevi kada mnogouglovi približniji pravilnom prelaze kraći put od onih koji su manje približni. Ovakvi mnogouglovi predstavljaju lokalne maksimume.

Neka unapređenja ovog modela mogla bi se realizovati zadavanjem više stepeni slobode mnogouglu (odstupanja uglova na primer), predstavljanjem mnogouglova u trodimenzionom prostoru i promenom ostalih konstanti koje se tiču genetskih algoritama. Ishod bi, najverovatnije, bio istovetan, ali bi bilo potrebno da protekne više generacija.

Literatura

Geijtenbeek T., van de Panne M., van der Stappen A. F. 2013. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics (TOG)*, 32 (6): 206.

Vuk Vuković

Development of a Wheel from Randomly Generated Polygons Using Genetic Algorithms

The purpose of this project is to check whether it is possible to develop a regular polygon (wheel) from randomly generated polygons through the application of genetic algorithms. Genetic algorithm is a technique used in programming which is based on the biological concepts of selection, cross-over, and mutation. Firstly, the generation of polygons is randomly generated. An initial momentum is given to each polygon whose distance traveled is then calculated. After sorting these polygons by using the traveled distance criteria, the application of genetic algorithms takes place: selection (preserving the best polygons), mutation (random changes in a polygon), cross-over (creation of two new polygons by combining two current ones) and generating new polygons. This process is repeated over and over again. The expected outcome was the convergence to a regular polygon (wheel), which was eventually proven. 