

Rešavanje problema sletanja u igrici Lunar Lander korišćenjem genetskog algoritma

Seminarski rad u okviru kursa

Naučno izračunavanje

Matematički fakultet

Aleksandra Kovačević, Filip Novović

12. septembar 2019.

Sažetak

Već neko vreme je u svetu računarstva jedna od zanimljivih tema učenje računara da igra igrice i pokušava da ih pređe. U ovom radu igrice u pitanju će biti Lunar Lander, a nalaženju rešenja je pristupljeno primenom genetskog algoritma. Na kraju se nalaze i rezultati pokretanja algoritma sa različitim podešavanjima i komentar o rezultatima.

Sadržaj

1	Uvod	2
2	Implementacija igrice	2
2.1	Letilica	2
2.2	Mapa	3
3	Genetski algoritam	3
4	Osnovni oblik	3
4.1	Struktura jedinke	4
4.2	Funkcija prilagođenosti	4
4.3	Inicijalna populacija	4
4.4	Selekcija	5
4.5	Reprodukcija	5
4.6	Mutacija	5
4.7	Politika zamene generacije	6
4.8	Kriterijum zaustavljanja	6
5	Merenja i neki rezultati	6
5.1	Komentar	9
6	Zaključak	9
	Literatura	10

1 Uvod

Lunar lander je igrice iz 1979. godine koja je pravljen za Atari. Cilj igrice je letilicom kojom upravlja igrač, sleteti na platformu za to predviđenu. Akcije koje korisnik može da primeni jesu: rotacija letilice na levo, rotacija na desno i paljenje motora što omogućava letilici da se kreće u smeru koji je određen uglom izrotiranosti letilice. Kada se letilica približi platformi, da bi uspešno sletela, njena brzina i horizontalna i vertikalna mora biti jako mala, a ugao mora biti normalan u odnosu na tlo i to sa kljunom koji usmerava ka gore, jer u suprotnom eksplodira.



Slika 1: Slika iz igrice Lunar Lander

Genetski algoritam je pretraživačka heuristika koja oponaša prirodnu selekciju. Oblik rešenja problema se prilagodi strukturi koja omogućava primenu genetskog algoritma (obično niz nula i jedinica). Pojedinačni delovi te strukture predstavljaju analogiju gena iz hromozoma.

Ciljevi rada su:

1. Implementacija igrice Lunar Lander
2. Prilagođavanje struktura igrice primeni genetskog algoritma
3. Definisanje i implementacija okruženja za menjanje parametara genetskog algoritma i njegovo pokretanje
4. Testiranje sa različitim parametrima i beleženje rezultata rada algoritma

2 Implementacija igrice

2.1 Letilica

Prikazana je sa tri kruga, prva dva označavaju samu letilicu (sivi i plavi krug), treći predstavlja rad motora i narandžaste je boje.

Kretanje je razloženo na horizontalnu i vertikalnu brzinu, znakovi ispred brojeva koji ih opisuju ukazuju na smer. Na njega utiče sila gravitacije

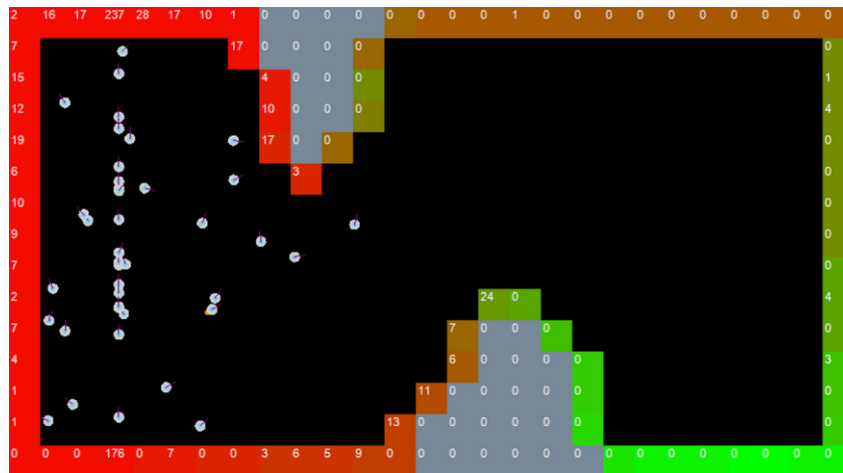
i sila prouzrokovana radom motora. Uticaj sile rada motora na trenutni vektor kretanja određen je uglom izrotiranosti letilice i da bi postojao neophodno je da je aktivan motor.

2.2 Mapa

Opisana je mrežom blokova koji predstavljaju zemlju ili vazduh (letilica prolazi kroz vazduh, a sa zemljom se proverava kolizija). Mape se čuvaju u fajlovima koji u sebi sadrže matricu brojeva gde svaki broj predstavlja blok. Ukoliko je broj ≤ 0 on predstavlja vazduh, a u suprotnom zemlju. Broj u matrici koji je pridružen bloku označava koliko je on daleko od cilja koji je označen brojem 100. Ovaj broj ćemo kasnije koristiti kao jedan od faktora za izračunavanje funkcije cilja jedinke. Mesto na kom se stvara letilica na početku igre je označena brojem -1. Na slici 3 je veličina koja predstavlja udaljenost bloka od ciljnog prikazana spektrom boja (zeleno je cilj).

5	5	40	50	60	60
5	-1	0	0	0	70
5	0	0	0	0	70
5	0	60	70	0	80
5	30	5	90	100	90

Slika 2: Primer opisa mape u fajlu



Slika 3: Prikaz implementacije igrice

3 Genetski algoritam

4 Osnovni oblik

Tok rada genetskog algoritma prikazan je na slici 4.

Algoritam: Opšti genetski algoritam**Ulaz:** podaci koji određuju funkciju cilja i podešavanja algoritma**Izlaz:** najkvalitetnija jedinka u tekućoj populaciji

- 1: generiši početnu populaciju jedinki;
- 2: izračunaj prilagođenost svake jedinke u populaciji;
- 3: **ponavljaj**
- 4: izaberi iz populacije skup jedinki za reprodukciju;
- 5: primenom operatora ukrštanja i mutacije kreiraj nove jedinke (i računaj njihovu prilagođenost);
- 6: na osnovu starih i novih jedinki, kreiraj novu generaciju;
- 7: **dok nije ispunjen** uslov zaustavljanja
- 8: vrati najkvalitetniju jedinku u poslednjoj populaciji

Slika 4: Tok genetskog algoritma

4.1 Struktura jedinke

Rešenje je opisano nizom parova (akcija, trajanje akcije)[2] jer ono je jedinstveno opisuje jedan ceo let letilice, a odgovara prirodi genetskog algoritma. Akcija može biti:

1. Rotacija levo (RL)
2. Rotacija desno (RR)
3. Uključivanje motora (TH)
4. Mirovanje (NT)

A1	T1	A2	T2	A3	T3	...	AN	TN
-----------	-----------	-----------	-----------	-----------	-----------	------------	-----------	-----------

Slika 5: Struktura jedinke

4.2 Funkcija prilagođenosti

Funkcija prilagođenosti daje ocenu kvaliteta jedinke. Što je vrednost funkcije prilagođenosti za neku jedinku veća, to će biti veća verovatnoća da se ta jedinka koristi za generisanje sledeće generacije. Formula funkcije cilja u našoj implementaciji:

$$F_c = c_1 \cdot BlockValue + c_2 \cdot \left(\frac{1}{VerticalSpeed} \right) + c_3 \cdot \left(\frac{1}{HorizontalSpeed} \right)$$

gde c_1 , c_2 i c_3 predstavljaju konstante koje smo prilagodili veličinama faktora koji su definisani u igrici (npr. horizontalna i vertikalna brzina označavaju broj piksela koji se pređe u određenom broju milisekundi).

4.3 Inicijalna populacija

Generiše se na slučajan način. Za svaku jedinku se u petlji generiše njen run vektor koji označava niz aktivnosti i njihovo trajanje. Bira se

slučajan broj od nula do 4 i u zavisnosti od toga odgovarajuća akcija. Ukoliko je odabran broj 0, akcija je mirovanje u trajanju od 100 do 200 milisekundi. Ukoliko su odabrani 1 ili 2, akcije su redom rotacija (u levo ili desno) i gas u trajanju od 10 do 25 milisekundi. Parametri koji određuju trajanje akcija izabrani su na osnovu posmatranja ponašanja letilice.

4.4 Selekcija

Obezbeđuje čuvanje i prenošenje dobrih osobina populacije na sledeću generaciju. Parametar "veličina reprodukcije", unosi se preko forme i označava broj jedinki koje će biti odabrane da učestvuju u reprodukciji. Implementirane su 2 vrste selekcije:

1. **Ruletska selekcija** Veće šanse za učestvovanje u procesu selekcije imaju prilagođene jedinke. Verovatnoća da jedinka i bude izabrana da učestvuje u reprodukciji, jednaka je:

$$p_i = \frac{f(i)}{\sum_j^N f(j)} \quad [1]$$

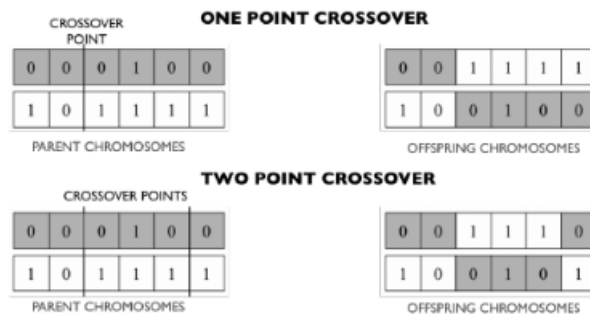
gde je $f(i)$ vrednost funkcije prilagođenosti za jedinku i , a N broj jedinki u populaciji.

2. **Turnirska selekcija** Jedinke odigravaju turnire u kojima veće šanse za pobedu imaju one sa boljom prilagođenošću. Parametar K označava veličinu turnira, odnosno k slučajnih jedinki koje učestvuju u turniru, a pobednik je jedinka sa najvećom funkcijom prilagođenosti.

Ni u jednoj selekciji nije dozvoljeno ponavljanje jedinki.

4.5 Reprodukcija

U procesu reprodukcije učestvuju jedinke koje su izabrane u procesu selekcije. Implementirano je jednopoziciono i dvopoziciono ukrštanje u kome od dve jedinke (roditelja) nastaju dva deteta. Grafički prikaz jednopozicionog i dvopozicionog ukrštanja prikazan je na slici 11.



Slika 6: Jednopoziciono i dvopoziciono ukrštanje

4.6 Mutacija

Uloga mutacije u genetskom algoritmu sprečava da jedinke u populaciji postanu suviše slične. Parametar mutacije se unosi na formi i označava

verovatnoću sa kojom će jedan hromozom, odnosno jedna akcija jedinke, i njeno trajanje, mutirati.

4.7 Politika zamene generacije

Elitizam je strategija pomoću koje se nekoliko najboljih jedinki u generaciji štite od eliminisanja ili bilo kakvih izmena i takve se prenose u sledeću generaciju. Parametar koji označava broj elitnih jedinki koje se prenose u narednu generaciju unosi se na formi.

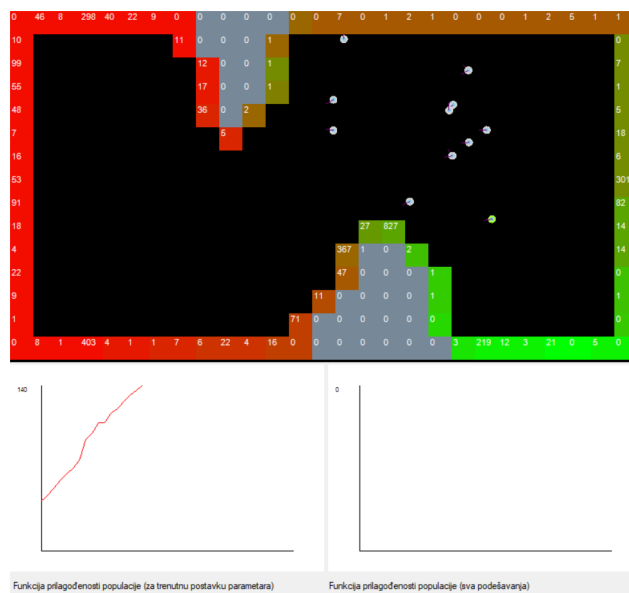
4.8 Kriterijum zaustavljanja

Algoritam prestaje sa radom kada se dostigne maksimalan broj iteracija koji se zadaje na formi.

5 Merenja i neki rezultati

Podešavanje 1

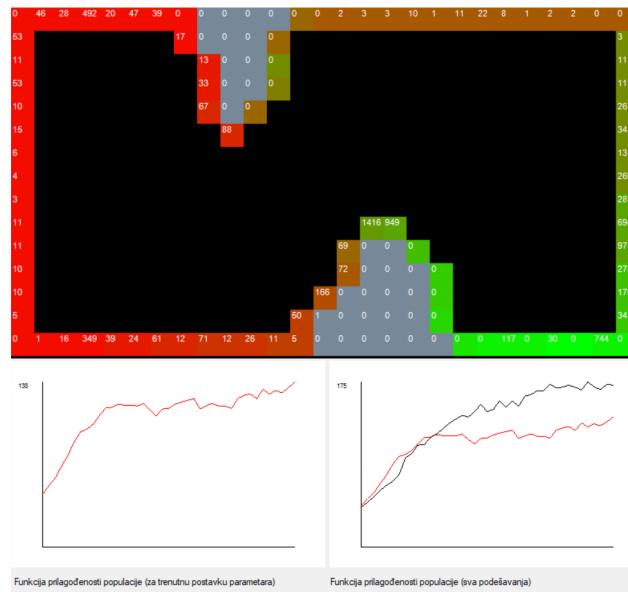
1. Veličina populacije: 200
2. Tip selekcije: ruletska
3. Tip ukrštanja: jednopoziciono
4. Veličina grupe za reprodukciju: 50
5. Procenat mutacije: 0.3
6. Broj elitnih jedinki: 20
7. Broj iteracija: 40



Slika 7: Podešavanje 1

Podešavanje 2

1. Veličina populacije: 200
2. Tip selekcije: ruletska
3. Tip ukrštanja: jednopoziciono
4. Veličina grupe za reprodukciju: 70
5. Procenat mutacije: 0.4
6. Broj elitnih jedinki: 20
7. Broj iteracija: 40



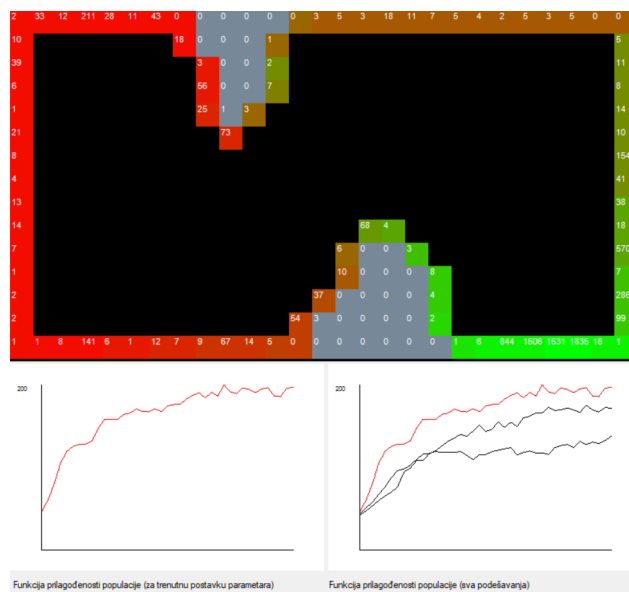
Slika 8: Podešavanje 2

Podešavanje 3

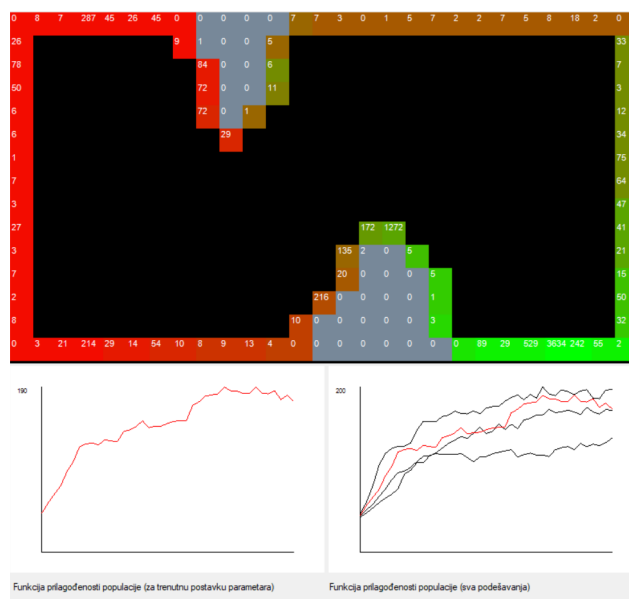
1. Veličina populacije: 200
2. Tip selekcije: turnirska (veličina turnira 20)
3. Tip ukrštanja: jednopoziciono
4. Veličina grupe za reprodukciju: 70
5. Procenat mutacije: 0.4
6. Broj elitnih jedinki: 10
7. Broj iteracija: 40

Podešavanje 4

1. Veličina populacije: 200
2. Tip selekcije: turnirska (veličina turnira 20)
3. Tip ukrštanja: jednopoziciono
4. Veličina grupe za reprodukciju: 70
5. Procenat mutacije: 1
6. Broj elitnih jedinki: 1



Slika 9: Podešavanje 3



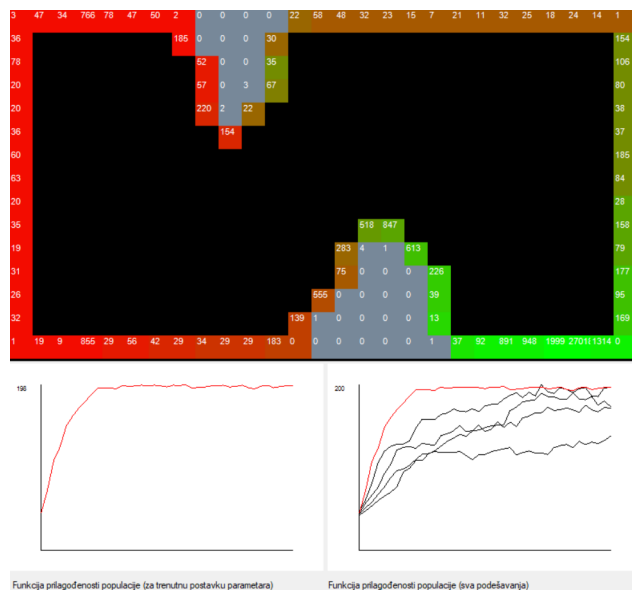
Slika 10: Podešavanje 4

7. Broj iteracija: 40

Podešavanje 5

1. Veličina populacije: 1000
2. Tip selekcije: turnirska (veličina turnira 20)

3. Tip ukrštanja: dvopoziciono
4. Veličina grupe za reprodukciju: 80
5. Procenat mutacije: 0.4
6. Broj elitnih jedinki: 30
7. Broj iteracija: 40



Slika 11: Podešavanje 5

5.1 Komentar

Na graficima ispod mape se iscrtavaju funkcije koje opisuju ponašanje proseka funkcije prilagođenosti populacije za svaku iteraciju. Y osi odgovara prosek funkcije prilagođenosti populacije, a X osi iteracija. Na slici koja opisuje pokretanje pri podešavanju 1 smo prikazali da veliki broj jedinki već oko 15. iteracije uspeva da pređe na desnu stranu mape, gde nam se nalazi cilj. Najbolji rezultat je ostvarila populacija pri pokretanju sa podešavanjem 3. Tada je prosek funkcija prilagođenosti u jednom trenutku bio 200. Interesantno ponašanje napretka populacije možemo videti pri podešavanju 5 gde populacija broji čak 1000 jedinki. Pri tom pokretanju vidimo da je rast proseka funkcije prilagođenosti ubedljivo najbrže rastao u poređenju sa ostalim podešavanjima, ali je onda oko 198 počeo da stagnira i nije imao nagle promene koje možemo primetiti kod ostalih pokretanja.

6 Zaključak

Primećujemo da su uspeh i napredovanje populacije za svako od podešavanja bili uglavnom rastući. Za menjanje vrednosti nekih parametara smo videli veće razlike u ponašanju proseka funkcije prilagođenosti, a za neke

manje. Svakako vidimo da genetski algoritam uspeva da za određena podešavanja nađe dosta dobro rešenje u malom broju iteracija, što može biti korisno (npr. podešavanje broj 5). Proširivanje projekta bi podrazumevalo eventualno dalje menjanje koeficijenata u funkciji cilja i pravljenje nekog jednostavnog algoritma koji bi prošao mapom i sam dodelio procenu udaljenosti od cilja prohodnim putem.

Literatura

- [1] Predrag Janičić and Mladen Nikolić. Veštačka inteligencija. *Matematički fakultet u Beogradu*, 2010.
- [2] Zhangbo Liu. A guided genetic algorithm for the planning in lunar lander games, 2006.