

Simulacija kretanja loptice po poluzi korišćenjem genetskog algoritma

Maja Gavrilović 489/2017
Januar 2020

Sažetak:

Ovaj rad se bavi simuliranjem kretanja loptice po poluzi primenom genetskog algoritma.

Za implementaciju i razvijanje funkcionalnosti korišćen programski jezik JavaScript

Sadržaj

1. Genetski algoritmi	
1.1 Šta su genetski algoritmi i čemu služe	3
2. Opis problema koji se rešava.	3
3. Opis rešenja zadatog problema.	4
4. Poglavlje sa eksperimentalnim rezultatima	5
5. Tehnički detalji	6
6. Zaključak.	7
Literatura.	8

1. Genetski algoritmi

1.1 Šta su genetski algoritmi i čemu služe

Genetski algoritmi (Genetic Algorithms, GA) su porodica algoritama koja je inspirisana Darwinovom teorijom evolucije.

Tvorcem ove oblasti se smatra John Holland. Iako ne nalaze uvek optimalno rešenje, njihova prednost je što mogu da u razumnom vremenu nađu rešenje koje je svega 2-3% lošije od optimalnog.

Genetski algoritam je jedan od najstarijih metaheurističkih algoritama kojem su mnogo pažnje posvetili istraživači širom sveta.

Algoritam se koristi za rešavanje kombinatorno optimizacionih problema zasnovanih na principima prirodne selekcije i genetike. GA kreće od grupe inicijalnih rešenja koja se zovu inicijalna populacija.

Zatim, koristi se fitnes funkcija koja predstavlja ocenu kvaliteta jedinke.

Svaki put se izaberu dva rešenja iz populacije, koja se zovu roditelji, prema verovatnoći selekcije koja je proporcionalna njihovoj fitnes vrednosti. Roditelji se ukrštaju i dobijaju se dva nova rešenja u sledećoj generaciji. Nova rešenja će da zamene stara, ako imaju bolju fitnes funkciju. Operacija mutacije se primenjuje na nova rešenja koja su zasnovana na verovatnoći mutacije. Selekcija, ukrštanje i mutacija se ponavljaju da proizvedu nova rešenja dok veličina nove populacije ne bude jednaka staroj. Iteracija zatim kreće od nove populacije. Pošto nova rešenja imaju veću verovatnoću da budu izabrana za ukrštanje i prenose osobine roditelja, očekuje se da će nova generacija biti bolja od stare.

Proces se nastavlja dok broj generacija ne stigne do n ili se kvalitet rešenja ne može lako poboljšati.

2. Opis problema koji se rešava

Problem se zasniva na simulaciji loptice koja se kreće po poluzi.

Početna pozicija je sredina poluge. S obzirom na gravitaciju očekivanje je da će loptica da sklizne nakon nekog vremena. Cilj je da uz pomoć

genetskog algoritma kroz nekoliko generacija pronadjem način da se loptica sto duže zadrži na poluzi. Ukoliko uspe da je zadrži određeni broj sekundi izvršavanje je uspešno. Cela ideja je ispraćena kroz grafički prikaz i logove na kojima ce biti ispisivana najbolja “jedinka” i njene vrednosti. Kao koren svoje ideje bih navela članak na koji sam naisla i kog mozete pronaci kao pdf u okviru linka https://research.google/pubs/pub42902/?fbclid=IwAR279o_5uzabmq1bpCnZBpsuRwmpLF9Zx3GmnZDM-H4VypS6mpNKbRjPu7U. Sami rezultati i ciljevi projekta nisu uporedivi ali određeni koraci u implementaciji se baziraju na toj ideji.

3. Opis rešenja zadatog problema

Bot pokušava da balansira lopticu primenom genetskog algoritma. Rad algoritma isti kao i svaki drugi genetski algoritam.

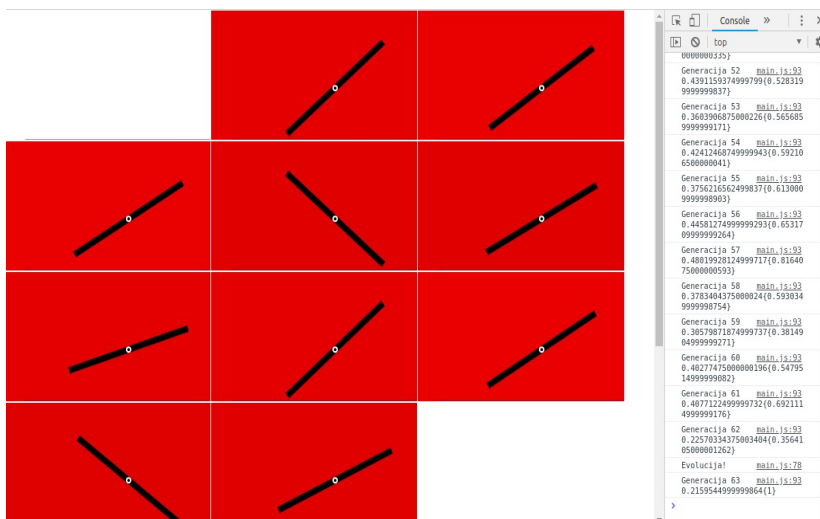
Funkcija cilja je predstavljena vremenom koje je loptica provela na dasci. Sto je veca vrednost rezultat je bolji.

Za prirodnu selekciju koristim Monte Karlo metod. Tacka kidanja DNK je direktno proporcionalna sa maksimalnim fitnessom oba roditelja sto ubrzava evoluciju, zato sto su početni pokreti bitniji od završnih.

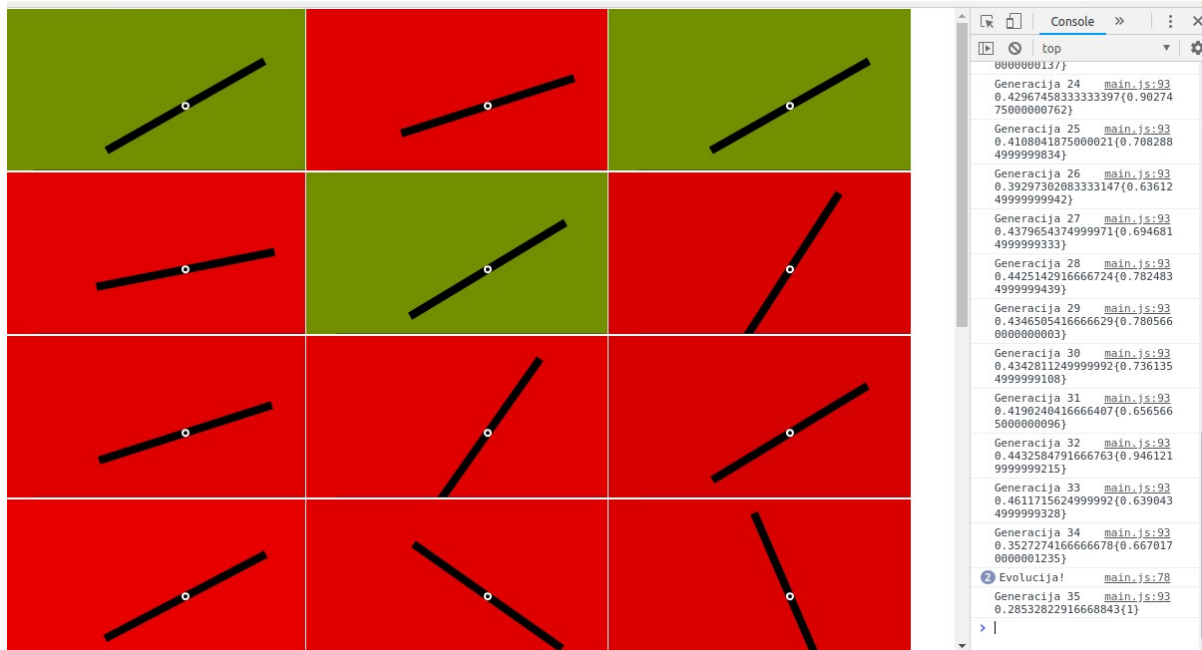
Stepen mutacije smanjujemo vremenom kako se ne bi premašilo dobro rešenje.

Za fiziku koristim Matter.js biblioteku.

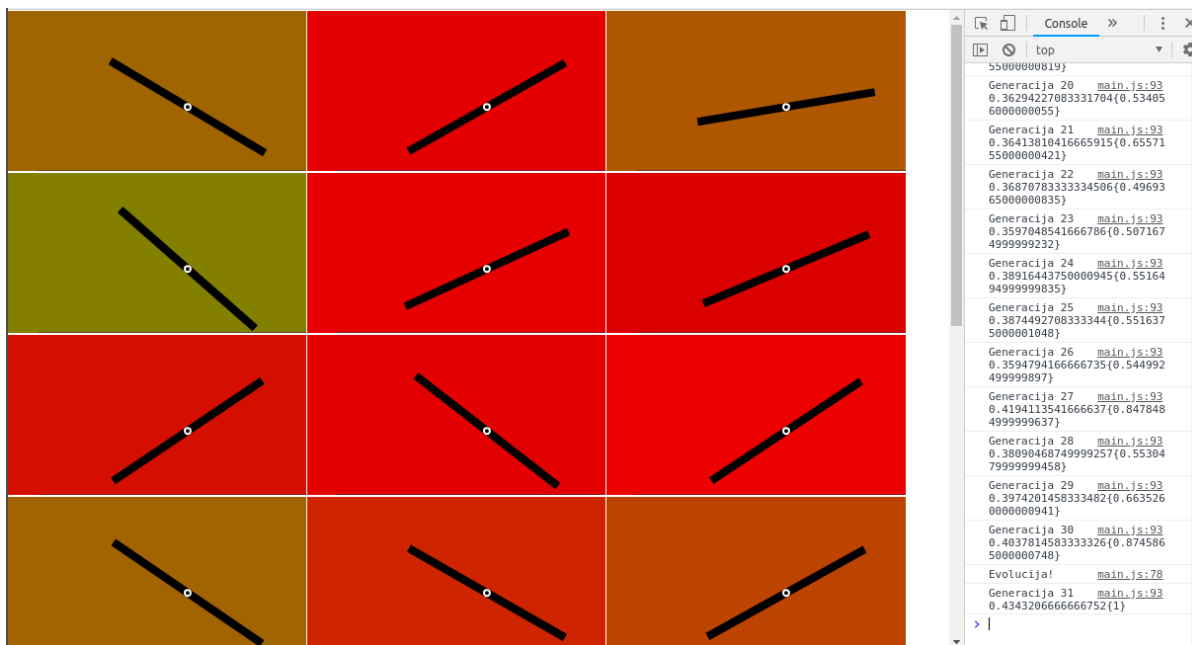
4. Poglavlje sa eksperimentalnim rezultatima



U ovom delu ću prikazati odnos rezultata simulacije na promenu parametra. Na prvoj slici možemo videti kako povećanjem broja populacije (sa početnog 16 uvećan na 28) dolazi ranije do zadate funkcije cilja, odnosno u manjem broju iteracija uspevamo da zadržimo lopticu na poluzi desetak sekundi (unapred definisan parametar).



Takodje pogodnim uvecanjem broja iteracija povecavamo verovatnocu da uopšte dodje do evolucije predstavljene simulacijom. Sličan ishod se desava i ako uvećamo stopu mutacije brže dolazi do evolucije sto se poklapa i sa logičkim aspektom razmišljanja.



5. Tehnički detalji

Tehnologije koje su korišćene za razvijanje funkcionalnosti HTML5, CSS3, Java Script. Za pisanje programskog koda korišćen tekst editor Visual Studio Code. Implementacija na virtuelnoj mašini sa operativnim sistemom Ubuntu. Pregledači koji će biti korišćeni pri demonstraciji projekta Chrome, Mozilla Firefox.

6. Zaključak

Genetski algoritmi primenljivi su na veoma širok skup problema. Ipak, za uspešno rešavanje konkretnih problema potrebno je napraviti mnogo dobrih izbora (na primer, za funkciju prilagođenosti i za parametre). Pogodan izbor operatora ukrštanja i parametara genetskog algoritma (velicina populacije, verovatnoća ukrštanja, verovatnoća mutacije, itd) veoma je važan za njegove performanse. S druge strane, upravo velika sloboda u izboru parametara istovremeno je i pretnja da mogu da budu korišćeni parametri koji daju loše performanse.

Optimizovanje parametara genetskog algoritma je kompleksan problem koji se najčešće rešava izvođenjem eksperimenata – probnih rešavanja. Za izbor pogodnih parametara često se koriste sami genetski algoritmi.

Parametri genetskog algoritma ne moraju biti fiksirani, vec mogu da se menjaju i prilagođavaju tokom rada. Na primer, ukoliko su tekuće jedinke raznolike, onda se može povećati verovatnoća ukrštanja, a smanjiti mutacija, a ukoliko su slične, onda se može uraditi obratno, kako bi se povećale šanse za bekstvo jedinki iz lokalnog optimuma.

Popularnost genetskih algoritama potiče iz njihove uspesnosti, ali i jednostavnosti. Naime, ideje na kojima su genetski algoritmi zasnovani su jednostavne za razumevanje i implementiranje, a daju opšti sistem pretrage primenljiv na veliki broj problema. Pored toga, i u situacijama kada ne nalaze globalne ekstremume, rešenja koja daju su često dovoljno dobra. S druge strane, još uvek nema mnogo teorijskih rezultata koji govore o svojstvima genetskih algoritama, o kvalitetu rešenja koja daju, pa čak ni o tome zašto su genetski algoritmi uspešni.

Literatura:

- [1] Predrag Janicic, Vestacka inteligencija, Matematicki fakultet Beograd
- [2] Brodman, G. and Voldstad, R. (2012). QWOP Learning. Department of Computer Science CS 229 class project, Stanford University, 2012. Retrieved 4/13/2014 from <http://cs229.stanford.edu/projects2012.html>
- [3] Foddy, B. QWOP [Flash Game] (2008). Retrieved 11-6-13 from <http://www.foddy.net/Athletics.html>
- [4] Gordon, V. and Whitley, D. (1993). Serial and Parallel Genetic Algorithms as Function Optimizers. Proceedings of the 5th Int. Conf. on Genetic Algorithms (ICGA-93).
- [5] Mahnig, T. and Muhlenbein, H. (2002). A Comparison of Stochastic Local Search and Population Based Search. Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02).
- [6] Parker, G., Braun, D., and Cyliax, I. (1997). Evolving Hexapod Gaits Using a Cyclic Genetic Algorithm. Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing (ASC '97).
- [7] Ramachandra, R. (2013). Fastest 100m run, QWOP (flash game). Guinness World Records. Retrieved 11-6-13 from <http://challengers.guinnessworldrecords.com/challenges/160-fastest-100m-run-qwop-flash-game>
- [8] Vaucher, L. (2011). Genetically Engineered QWOP (Part 1). Slow Frog Blog [blog]. Retrieved 11-6-13 from <http://slowfrog.blogspot.com/2011/03/genetically-engineered-qwop-part-1.html>