

AI pristup za igru Triba

Amna Omerović, TKN, Odsjek za matematiku, PMF Sarajevo

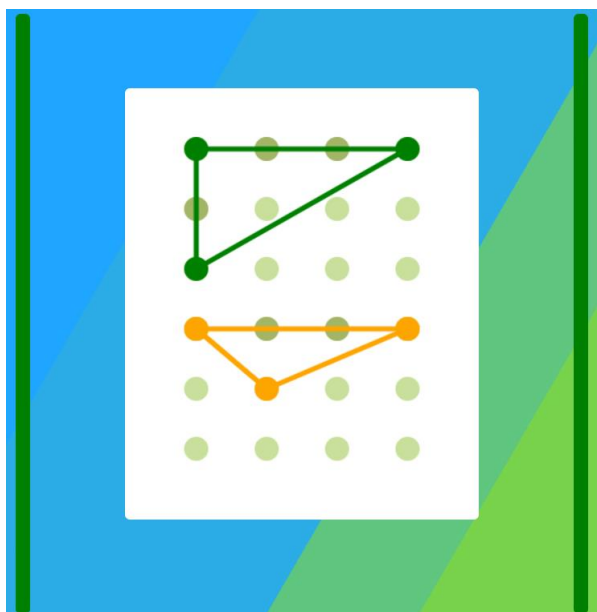
Abstract—Projekat se sastoji od implementacije AI igrača za igru pod nazivom Triba, te implementacije spomenute igre. U ovom radu bit će objašnjen implementirani AI pristup. Za AI pristup se koristi Generic Algorithm.

I. UVOD

OVAJ rad će se prvo kratko osvrnuti na strategiju igre Triba. Igra Triba (slika 1.) se sastoji od ploče na kojoj se nalaze tačke. Svaki igrač u svakom svom potezu, klikom miša, bira tri tačke koje čine trougao. Trouglovi se ne smiju sjeći niti imati zajedničke vrhove. Gubitnik u igri je igrač koji nema slobodnih tačaka koje čine trougao. Linije sa strane ploče predstavljaju koji je igrač na potezu.

U igri je moguće odabrati da li igrate sa AI ili sa prijateljem. Prvi igrač koji je na potezu se bira slučajno. Na kraju igre ispituje se ko je pobjednik. Osim toga ponuđene su ploče sa dimenzijama 10x8, 8x6, 6x4. AI igrač je u stanju igrati i na 10x8 ploči.

Za implementaciju AI pristupa koristi se Genetički algoritam [2 3 4]. U nastavku rada prvo će biti pojašnjen genetički algoritam, a zatim njegova primjena na ovu igru.

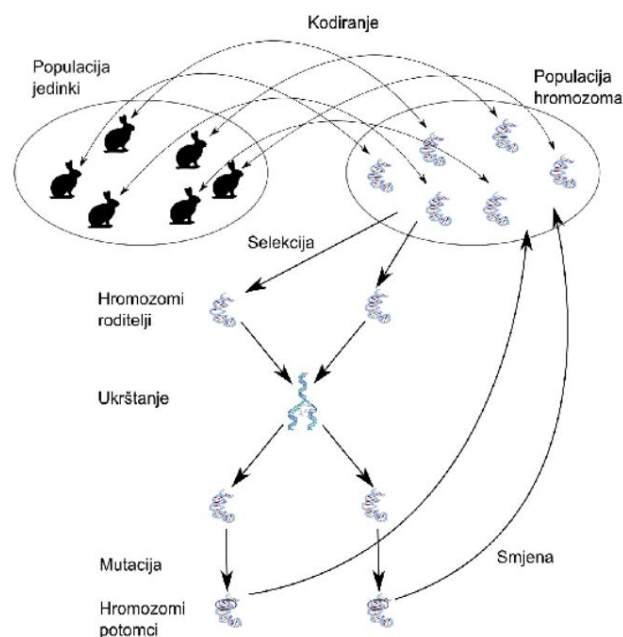


Slika 1. Igra Triba

II. AI PRISTUP

Genetički algoritam oponaša evoluciju tako što na populaciju vještačkih jedinki, od kojih svaka predstavlja potencijalno rješenje problema, primjenjuje operatore koji pokušavaju stvoriti nove jedinke sa poboljšanom performansom. Operatori koji će se koristiti su operatori ukrštanja i mutacije [1].

Na slici 2. možemo vidjeti osnovni put genetičkog algoritma. Svaki od dijelova bit će objašnjen kroz implementaciju.



Slika 2. Osnovna ideja Genetičkog algoritma

A. Populacija

Početna populacija se generiše slučajno unutar funkcije `runGeneticAlgorithm()`, koja je ujedno i glavna funkcija, a vraća najbolje rješenje. Najbolje rješenje je trougao tj. jedinka (hromozom) koja se prikaže na ploči. Za generisanje jednog trougla koristi se funkcija `generateRandomSolution()` koja generiše tri validne tačke (validna je ona tačka koja nije vrh drugog trougla, niti leži na stranama trougla) i nakon toga se provjeri da li je trougao validan (validan trougao je onaj koji

se ne presjeca i ne dodiruje sa drugim trouglovima na ploči). Ako je trougao validan funkcija ga vrati, a inače generiše novi trougao. Time se garantuje da je početna populacija slučajna, ali i da su svi trouglovi validni.

B. Fitness funkcija

Fitness funkcija zapravo daje vrijednost svakoj jedinki. Naime, pristup u radu se sastoji od toga da svaki trougao (jedinka) ima vrijednost fitnessa onoliko koliki je broj tačaka koji taj trougao prekriva. Broj takvih tačaka se računa pomoću funkcije `isThereMorePoints()`. Koristeći kolinearnost zaključuje se da li tačka leži na stranici trougla. Trouglove koji nisu validni (nastali mutacijom i ukrštanjem) kažnjava tako da im fitness umanjuje za vrijednost -1000.

Pored toga, najbolja jedinka iz trenutne populacije bezuslovno se prenosi u sljedeću populaciju, što se naziva elitizam. Najbolja jedinka je ona sa najvećim fitnessom.

C. Selekcija

Mehanizam selekcije korišten u ovom radu je tzv. selekcija na bazi točka ruleta. Selekcija služi za odabir jedinke iz populacije. Biraju se dva roditelja čijim ukrštanjem nastaje nova jedinka (dijete).

Funkcija koja vrši selekciju je funkcija `selection()`. Unutar nje biramo slučajnu vrijednost u opsegu `[0, totalFitness]` Nakon toga se za svaku jedinku iz populacije računa opseg kojem pripada. Ako se uzme prva jedinka iz populacije njen opseg je `[0, fitness(trenutnaJedinka)]`, druga jedinka imat će opseg `[fitness(prethodnaJedinka), fitness(trenutnaJedinka)]` i tako za svaku jedinku. Bira se ona jedinka unutar čijeg opsega je generisani slučajni broj.

D. Ukrštanje

Ukrštanje koje se ovdje koristi je ukrštanje u jednoj tački. Funkcija koja vrši ovu operaciju je `crossover()` funkcija. Proces se sastoji od uzimanja slučajnog indeksa od 0 do 2 (uključujući) i kreira se dijete tj. novi trougao. Prva tačka novog trougla je tačka na indeksu 0 prvog roditelja, a druga tačka je na indeksu 1 prvog roditelja, dok se zadnja tačka uzima od drugog roditelja, na slučajno generisanom indeksu.

E. Mutacija

Naredna operacija je mutacija. Vršiti se pomoću funkcije `mutation()` i mutira jedinke. Mutiranje se ne izvršava na svakoj jedinki, nego samo kada je slučajno generisani broj manji od `mutationRate` varijable koju zadajemo. Sastoji se od generisanja slučajnog indeksa tačke koju mutiramo. Nakon čega odabranu tačku mijenjamo sa novom slučajno odabranom

tačkom.

F. Smjena

Na kraju još ostaje funkcija `evolve()` unutar koje se vrši smjena populacija. Nova populacija se popunjava sa jedinkama nastalim ukrštanjem jedinki iz prethodne populacije. Prethodna populacija se ažurira sa novom populacijom, ostavljajući samo svoju najbolju jedinku još u populaciji. Svaka ima isti broj jedinki (veličina populacije). Evaluacija se vrši onoliko puta koliki je broj generacija [1].

III. ZAKLJUČAK

Rad objašnjava AI pristup za igru Triba uz pomoć Genetičkog algoritma. Zaključujemo da genetički algoritam radi efikasno i brzo, davajući dobrog protivnika. Zahvaljujući operacijama **ukrštanja** i **mutacije** dobivaju se novi trouglovi bez da pretraži čitav skup mogućih rješenja. **Fitness** funkcija ne dozvoljava odabir trouglova koji nisu validni i povećava mogućnost odabira trouglova koji smanjuju broj validnih tačaka protivniku. **Selekcija** je ta koja će se zbog slučajnosti malo udaljiti od trouglova koji imaju dobar fitness (iako trouglovi sa većim fitnessom imaju i veći opseg, pa samim tim i veću šansu da budu odabrani), te uzeti neke druge trouglove.

LITERATURA

- [1] S. Konjicija, “Genetički algoritam” in *Etf-Optimizacija resursa* Sarajevo. Dostupno: <https://c2.etf.unsa.ba/mod/resource/view.php?id=5507>
- [2] Paperspace, A. Fawzy Gad. (2020). *Building a Game-Playing Agent for CoinTex Using the Generic Algorithm* [Online]. Available: <https://blog.paperspace.com/building-agent-for-cointex-using-genetic-algorithm>
- [3] W3schools, Javascript, w3.css. Dostupno <https://www.w3schools.com/>
- [4] FullPage.js, W. Davies, (2021, Nov. 15.). 33 *Animated BackgroundsExamples*, [Online]. Dostupno: <https://alvarotrigo.com/blog/animated-backgrounds-css/>