

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1120

Evolucijski algoritmi za rješavanje igre Dots and Boxes

Hana Ujčić

Zagreb, srpanj 2023.

SADRŽAJ

1. Uvod	1
2. Igra Dots and Boxes	2
2.1. Pravila	2
2.2. Matematička razmatranja	3
3. Evolucijski algoritmi	4
3.1. Implementacija bioloških pojmova	4
3.2. Neuronske mreže	5
4. Programska implementacija	7
5. Rezultati	10
5.1. Optimizacija parametara	10
6. Zaključak	13
Literatura	14

1. Uvod

Brz rast u istraživanju umjetne inteligencije je omogućio rješavanje problema u velikom rasponu područja. Sam pojam umjetna inteligencija implicira da se do znanja dolazi tako da se imitiraju ljudi, ali i ostale prirodne pojave. Evolucijski algoritmi su posebna grana koja imitira proces evolucije, u kojem se organizmi poboljšavaju i prilagođavaju svom okolišu tijekom velikog broja uzastopnih generacija.

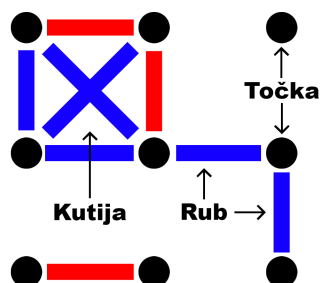
Ovaj rad će se fokusirati na primjenu evolucijskih algoritama na problem pronalaženja strategije u igrama za više igrača. Kroz primjenu na igri Dots and Boxes razmotrit će se učinkovitost ovih algoritama.

2. Igra Dots and Boxes

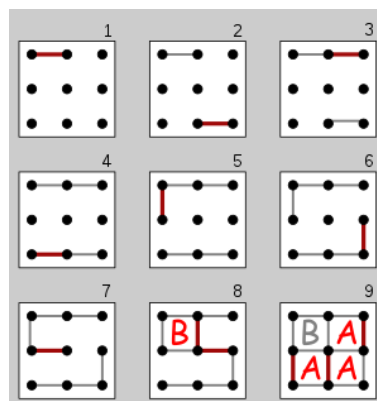
Igra Dots and Boxes je jednostavna igra za dva igrača za koju treba samo olovka i papir. Ploča (slika 2.1) se sastoji od pravokutnog polja točaka (eng. Dots) koje se mogu spajati rubovima (eng. Edges) vertikalno i horizontalno. Spojeni rubovi čine kutije (eng. Boxes) koji mjere uspjeh u igri. Kutija se sastoji od ukupno četiri ruba: dva vertikalna i dva horizontalna. Veličina polja može varirati, te broj redova i stupaca točki može biti različit.

2.1. Pravila

Igra počinje sa praznim poljem točaka gdje igrači izmjenjuju poteze. Igrač u jednom potezu mora spojiti dvije prethodno nespojene točke jednim rubom. Kada igrač spoji četvrti rub neke kutije, on "osvaja" tu kutiju i dobiva još jedan potez koji mora iskoristiti. Kada više ne postoji legalnih poteza u igri, odnosno svi vertikalni i horizontalni rubovi su spojeni, igra završava (slika 2.2). Pobjednik igre je onaj igrač koji ima najviše osvojenih kutija. Ovisno o veličini polja, može biti parni i neparni broj kutija, pa igre s parnim brojem kutija mogu završiti neriješenim rezultatom. [2]



Slika 2.1: Osnovni elementi igre



Slika 2.2: Primjer jedne 2x2 igre

2.2. Matematička razmatranja

Ploču za igru Dots and Boxes možemo opisati tako da ima m redova i n stupaca koje sačinjavaju kutije. U igri onda ima mn kutija i $(m+1)(n+1)$ točaka. Postoji $(m+1)n$ horizontalnih i $m(n+1)$ vertikalnih rubova, što je ukupno $2mn+m+n$ rubova, tj. poteza u igri. [5]

Dots and Boxes ima egzaktno rješenje, no zbog pravila igre veličina prostora pretraživanja stanja je jako velika, što otežava pronalazak poteza u razumnom vremenu. Jedna od posebnosti igre je da zatvaranjem kutije igrač osvaja pravo na još jedan potez, pa igrači ne moraju imati podjednaki broj poteza u igri. U suprotnosti s mnogim drugim igrama, igrač koji postavi zadnji potez na ploču ne mora nužno pobijediti, što dodatno otežava analizu igre.

Najčešći pokušaji za rješavanje igre uključuju smanjivanje prostora pretraživanja stanja koristeći podrezivanje alfa-beta i slične algoritme kako bi se igra mogla lakše analizirati, ili upotrebu podržanog učenja. Najpoznatiji radovi o rješavanju igre podržanim učenjem koriste umjetne neuronske mreže i pretraživanje stabla Monte Carlo tehnikom. [3] U ovom radu će fokus biti na primjeni neuronskih mreža za rješavanje Dots and Boxes.

3. Evolucijski algoritmi

Evolucijski algoritmi su skup stohastičkih metoda pretraživanja koji oponašaju prirodni tijek biološke evolucije. U prirodi svako biće ima svoj kromosom - osnovnu jedinicu nasljeđivanja koja opisuje sve osobine te jedinke, ali i osobine koje će preneti na svoje potomstvo.

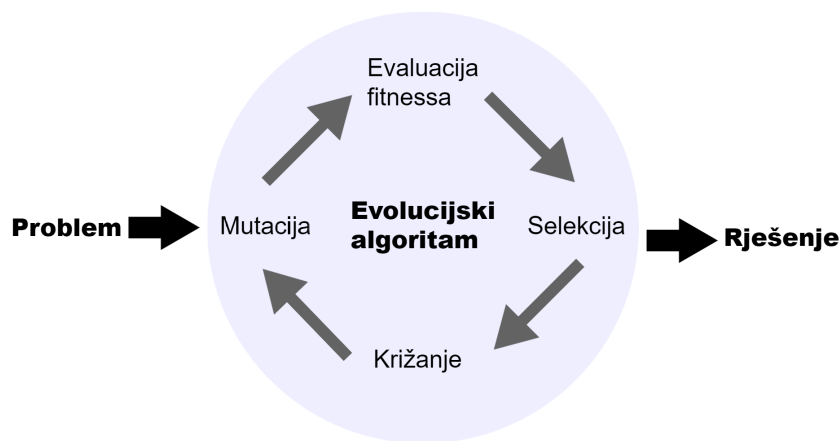
Genetski materijal kromosoma se može križati i mutirati, čime se stvara veća raznolikost u nekoj vrsti. Križanje je biološki proces u kojem dva organizma izmjenjuju svoj genetski materijal kako bi stvorili novog potomka. Mutacija je postupak kojim se genetski materijal stanice nasumično mijenja, čime se mijenjaju karakteristike jedinke i njenih potomaka.

Još jedan bitan pojam je prirodna selekcija. Jedinke koje se bolje prilagode svojoj okolini i uvjetima u prirodi imaju veću vjerojatnost preživljavanja i parenja, a samim time i prenošenja svog genetskog materijala na sljedeće naraštaje. Ovim postupkom se prirodno odabiru najbolji geni, dok geni jedinka koje su lošije prilagođene izumiru. Ove pojmove možemo primijeniti na računarstvo, i pomoću njih napraviti algoritme za pretraživanje.

3.1. Implementacija bioloških pojmova

Evolucijski algoritam počinje s određenim brojem jedinki koje čine populaciju. Populacije prolaze kroz nekoliko generacija u kojima evoluiraju i uče na problemu. Slično kao u prirodi, svaki pojedinac u populaciji sadrži svoj kromosom - skup karakteristika pomoću kojih rješava neki problem. Mjeru uspješnosti i brzine rješavanja problema nazivamo "*fitness*" funkcija ili funkcija dobrote, i pomoću nje rangiramo jedinke.

U postupku selekcije jedinke s najboljom funkcijom dobrote se prenose u sljedeću generaciju, bilo to u izvornom ili djelomično izmijenjenom obliku. Tu se javlja pojam elitizma, gdje se najbolje jedinke zaštićuju od bilo kakve izmjene i prenose se u izvornom obliku u novu generaciju. Time sprječavamo gubitak poželjnih svojstava tih jedinki.



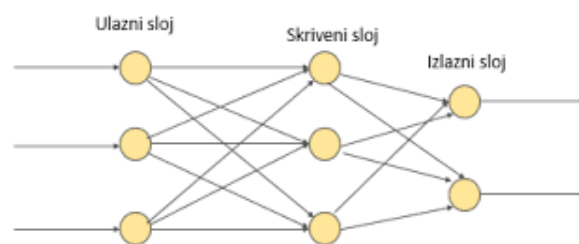
Slika 3.1: Evolucijski algoritam

Iz stare generacije se međusobnim križanjem kromosoma jedinki stvara nova populacija, gdje prednost u križanju imaju najbolje rangirani pojedinci. Geni nove generacije su podložni i mutaciji, čime se neke karakteristike nasumično mijenjaju kako bi se osigurala veća raznolikost. Ovaj proces se ponavlja sve dok nije zadovoljen uvjet zastavljanja evolucije (slika 3.1).

3.2. Neuronske mreže

Svi podaci o jedinki su zapisani u kromosomu, a ima mnogo pristupa programskom ostvarenju. Mi ćemo se fokusirati na neuroevoluciju jer to koristimo u našoj implementaciji. U neuroevoluciji je kromosom ostvaren kao umjetna neuronska mreža, struktura koja imitira ljudski mozak. Posebno nam je zanimljiva aciklička mreža (eng. feed-forward net), koja nema povratnih veza između neurona pa signali koji krenu od ulaznih neurona nakon određenog broja prijelaza dolaze do izlaza mreže. [4]

Umjetna neuronska mreža (slika 3.2) je skup umjetnih neurona koji su međusobno povezani i interaktivni. Svaki neuron na ulazu prima signal koji procesira pa ga šalje svim neuronima s kojima je povezan. Snaga veze između pojedinih neurona se naziva težina. Težine su prilagodljive te ih mreža učenjem namješta kako bi generirala što bolji rezultat. Aciklička mreža se obično sastoji od ulaznog i izlaznog sloja, a može imati i nekoliko skrivenih slojeva. Neuroni ulaznog sloja nemaju ulazne signale nego sadrže podatke o problemu.



Slika 3.2: Primjer acikličke neuronske mreže

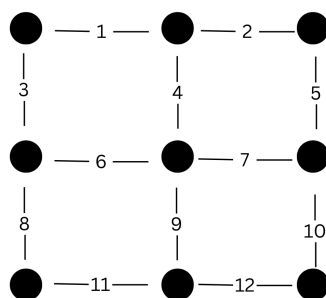
4. Programska implementacija

Učenje igre evolucijskim algoritmom je ostvareno u jeziku C++, te je podijeljeno na implementaciju igre Dots and Boxes, evolucijski algoritam i klasu upravitelja (eng. game manager) koji komunicira između njih dvoje. Upravitelj prenosi podatke o igri svakoj jedinki, ovjerava i procesira poteze te drži zapis o stanju igre.

Ploča u igri je reprezentirana listom veličine $2mn+m+n$, jer je toliko rubova u igri. Za ploču veličine 3×3 , lista koja reprezentira ploču ima 24 elemenata. Svaki element ima vrijednost 0 ili 1, gdje je 1 analogno spojenom rubu, a 0 nespojenom rubu. Jedna reprezentacija stanja igre bi mogla biti $[0, 1, 0, 1, 1, \dots, 1]$. Poredak rubova u listi je prikazan na slici 4.1.

Jedan potez odgovara spajanju jednog ruba, te je predstavljen metodom *one-hot encoding* [1] u listi veličine $2mn+m+n$, gdje odabrani potez ima vrijednost 1, dok svi ostali imaju vrijednost 0. Na primjer, ako bi igrač htio spojiti rub 2, taj potez bi bio enkodiran kao $[0, 0, 1, 0, 0, \dots, 0]$.

Svaka jedinka ima svoju acikličku neuronsku mrežu pomoću koje odlučuje koje će poteze napraviti u igri. Aciklička mreža se sastoji od ulaznog sloja, skrivenih slojeva i izlaznog sloja, a neuroni između slojeva su potpuno povezani. Ulaz u neuronsku mrežu je lista veličine $2 * (2mn+m+n)$, a izlaz je veličine 1. Vrijednosti u ulaznoj listi su dobivene tako da se spoji trenutno stanje ploče i potencijalni potez, predstavljen u



Slika 4.1: Reprezentacija ploče listom

one-hot encoding obliku. Za prijenosnu funkciju se koristi sigmoidalna funkcija. Pojedinač na ulazu dobiva trenutno stanje igre u listi, te za svaki legalan potez stvara ulaz za neuronsku mrežu, tako da spoji trenutno stanje igre i potez enkodiran one-hot metodom. Legalnim potezom se smatra svaki potez koji bi spojio do tad nespojeni rub. Neuronska mreža na ulazu dobiva listu, pa je procesira i vraća vrijednost. Vrijednosti za sve legalne poteze se rangiraju, te se odabire i vraća najbolji potez. Ako više poteza vraća istu najvišu vrijednost, jedan od njih se odabire nasumično. Evaluacija dobrote se radi na principu *round robin tournament*, gdje svaka jedinka igra protiv svih ostalih jedinki u populaciji jedanput. U dvoboju između jedinki se igraju dvije igre, gdje svaki od igrača jednom započinju igru, to jest stavljaju prvi potez na ploču. Rezultat svake igre se boduje i sprema u r , koji se dodaje dobroti.

$$r = \begin{cases} 2 & \text{ako je igrač pobijedio igru} \\ 1 & \text{ako je igra neriješena} \\ 0 & \text{ako je igrač izgubio igru} \end{cases} \quad (4.1)$$

Na kraju evaluacije, svaka jedinka ima vrijednost dobrote između $[0, 4 * (N - 1)]$, a N je veličina populacije. Nakon što se svakoj jedinki dodijeli dobrota, jedinke u populaciji se rangiraju. 10% najboljih jedinki preživljava u sljedeću generaciju, dok se ostalih 90% generira na temelju jednostavnog generacijskog odabira (*roulette-wheel selection*), te se nad njima primjenjuje operator mutacije. Za operator mutacije se koristi jednoliko (uniformno) križanje. Novonastala populacija se ponovo evaluira, te se proces ponavlja.

Data: *populacija*

inicijaliziraj *populacija*;

evaluiraj algoritmom round-robin $f(\textit{populacija})$;

$i \leftarrow 0$;

dok *uvjet zaustavljanja algoritma nije ispunjen* **radi**

$i \leftarrow i + 1$;

 sortiraj *populacija* prema *dobrota*;

Data: *novaPopulacija*

novaPopulacija \leftarrow elitizam : odaberi jedinke s najvećom dobrotom iz
 populacija;

za *novaJedinka* **radi**

 odaberi roditelje p_1 i p_2 iz *populacija* uz korištenje jednostavnog
 generacijskog odabira;

novaJedinka \leftarrow križaj p_1 i p_2 uniformnim križanjem;

novaJedinka \leftarrow mutiraj *novaJedinka* Gaussovom funkcijom s
 vjerojatnošću p ;

 dodaj *novaJedinka* u *novaPopulacija*;

kraj

populacija \leftarrow *novaPopulacija*;

 evaluiraj algoritmom round-robin $f(\textit{populacija})$;

kraj

;

evaluiraj algoritmom round-robin *Main* (*populacija*) :

za *jedinka iz populacija* **radi**

$r \leftarrow$ odigraj igru sa svakom jedinkom iz *populacija*;

dobrota \leftarrow *dobrota* + r

kraj

5. Rezultati

Parametar	Vrijednost
Broj ponavljanja	10
Kriterij zaustavljanja	1000 evaluacija
Veličina populacije	10
Postotak elitizma	0.1
Vjerojatnost mutacije	0.1
Prijenosna funkcija	sigmoid

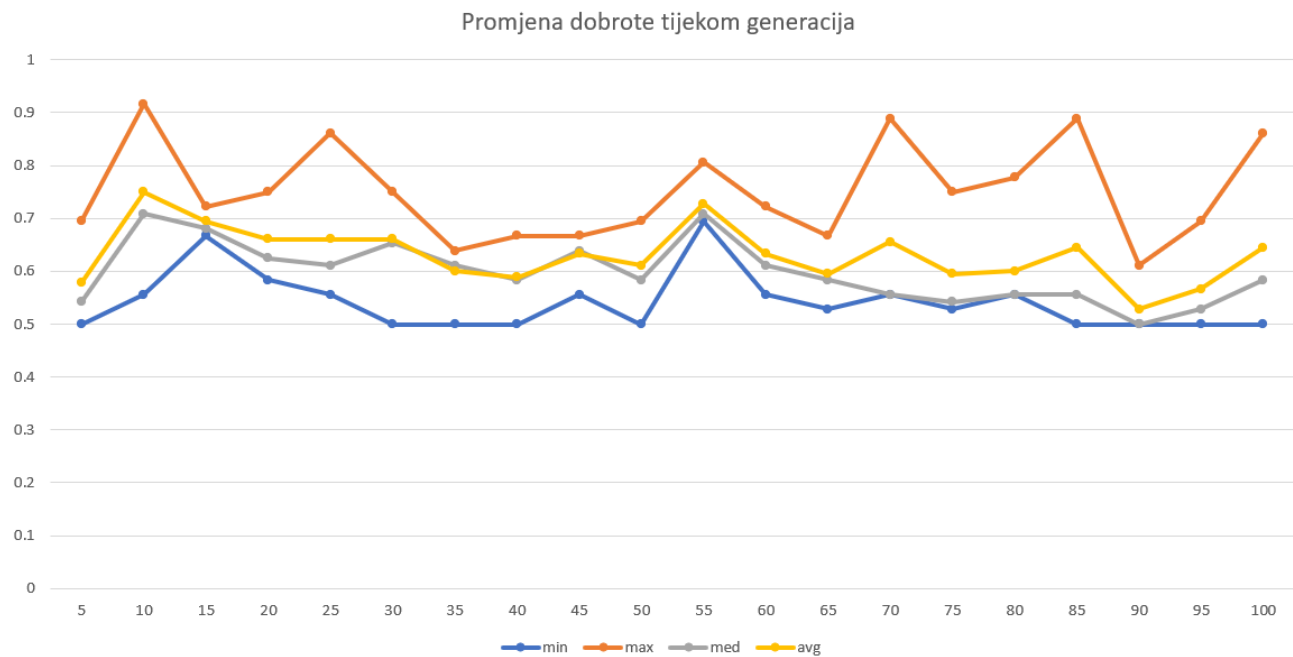
Na slici 5.1. su rezultati učenja igrača tijekom generacija za vrijednosti u tablici. Dobrota je u rasponu $[0, 1]$ koji predstavlja postotak bodova koji je igrač osvojio u odnosu na ukupan broj bodova koji je mogao osvojiti. Dobrota se ne mijenja previše tijekom učenja.

Agente možemo testirati protiv hardkodiranog protivnika koji koristi svoju heuristiku. Strategija hardkodiranog protivnika je da osvaja kutije ako može. Protivnik prvo traži ako postoji kutija s tri spojena ruba, te kao potez vraća četvrti rub, a ako ne postoji takav rub, vraća se nasumičan rub. Testiranje se provodilo svakih nekoliko generacija, te nije utjecalo na učenje. U testiranju je najbolja jedinka odigrala 20 igri s heuristikom, od kojim je polovicu jedinka postavila prvi potez, a drugu polovicu protivnik. Rezultati su prikazani na slici 5.2.

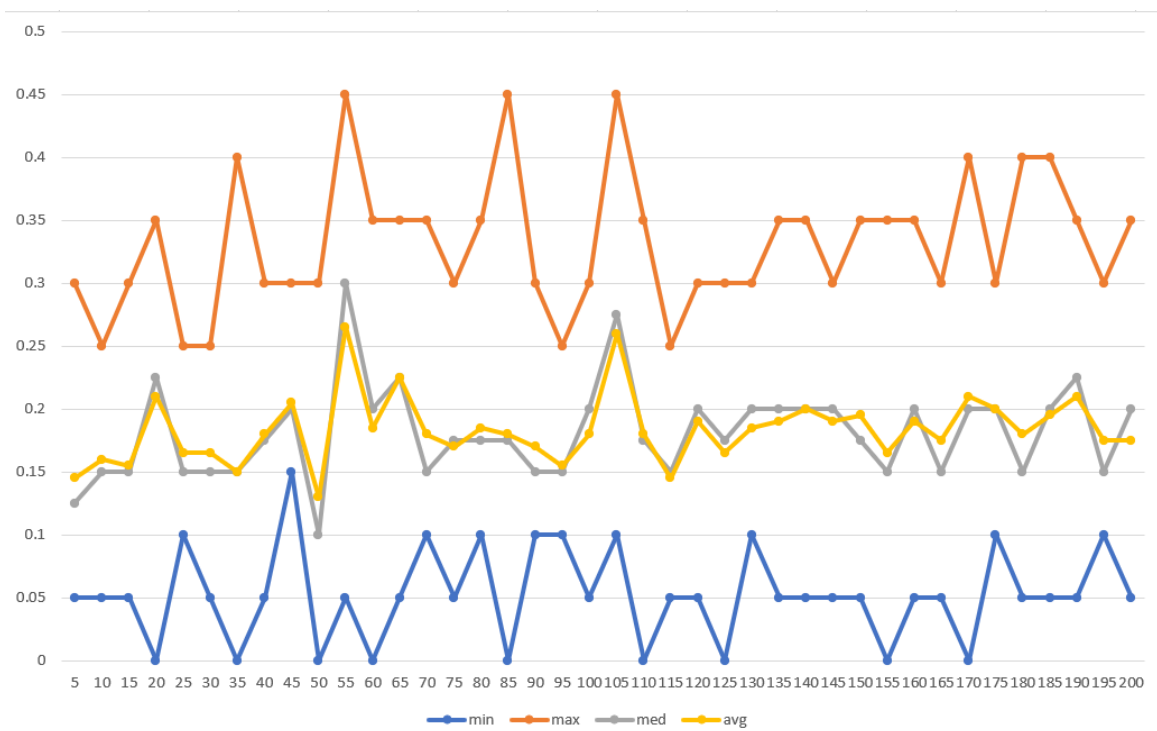
5.1. Optimizacija parametara

Optimizacija parametara se također provela s hardkodiranim protivnikom. Kao uvjet zaustavljanja se koristi broj evaluacija.

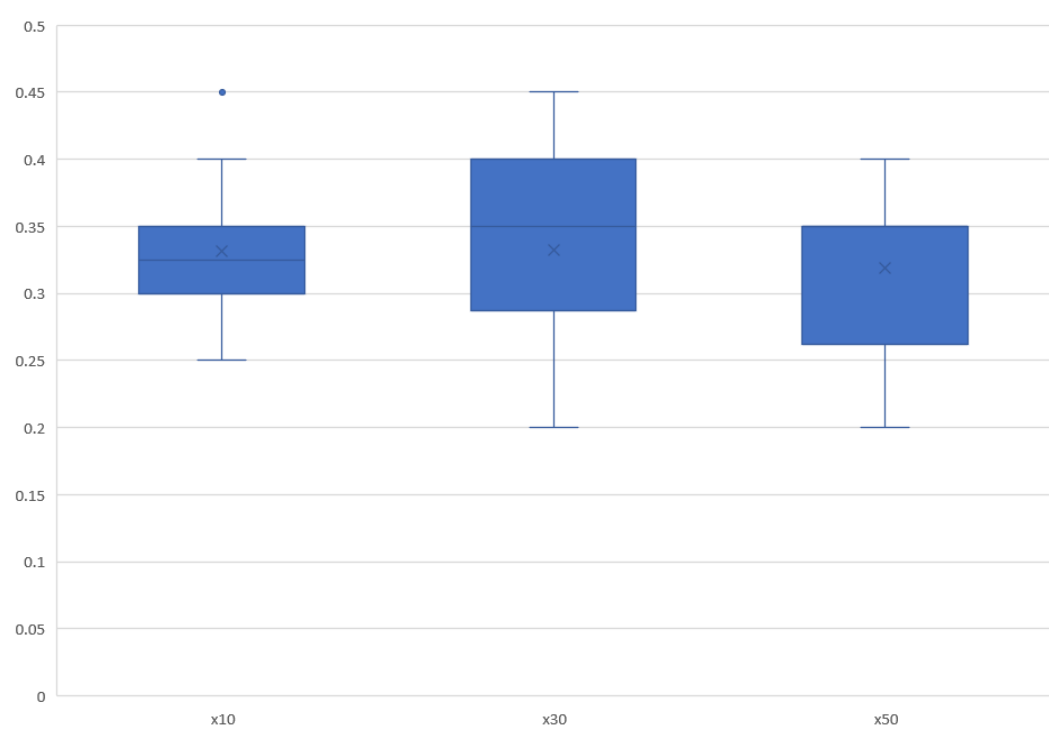
Na slici 5.3. je prikazana optimizacija koja za parametar uzima veličinu populacije, a na slici 5.4. gleda se vjerojatnost mutacije. Promjena nijednog parametra ne utječe znatno na poboljšanje agenta.



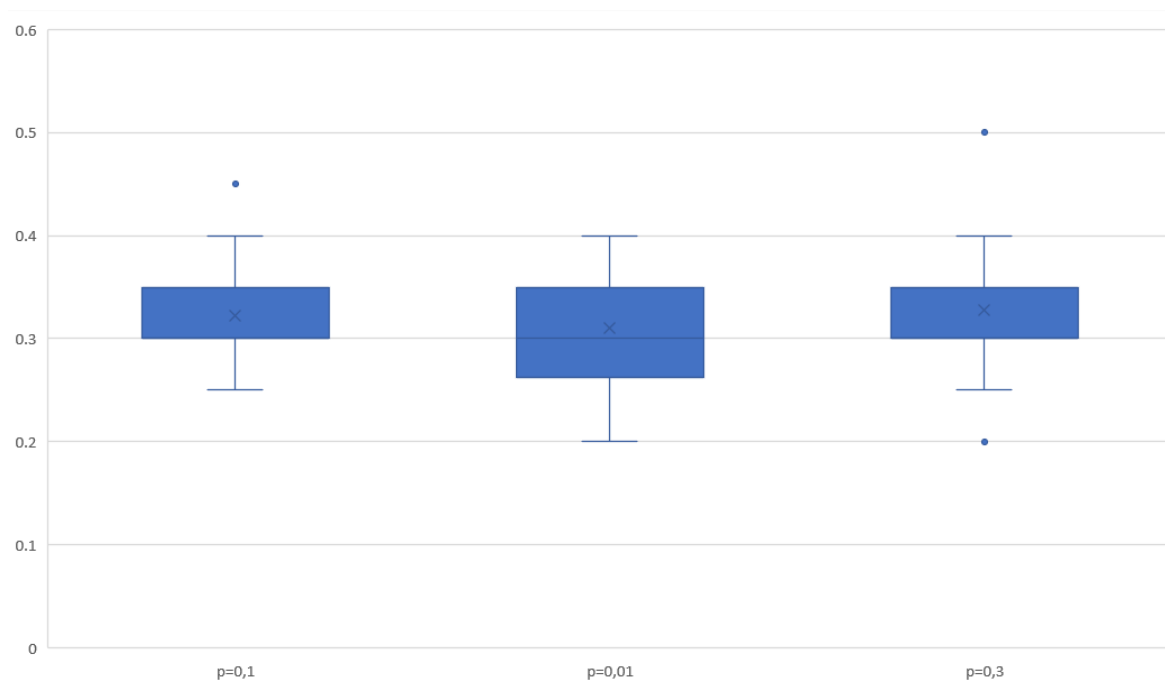
Slika 5.1: Učenje algoritma tijekom generacija



Slika 5.2: Testiranje algoritma protiv heuristike



Slika 5.3: Optimiranje veličine populacije



Slika 5.4: Optimiranje vjerojatnosti mutacije

6. Zaključak

Igra Dots and Boxes je jednostavna igra koja se može prilagoditi za rješavanje podržanim učenjem. Ima jasno definirana pravila, može se lagano skalirati te postoje predefinirane strategije za pobjeđivanje.

U trenutnoj implementaciji agent jako sporo uči te njegova uspješnost varira tijekom učenja. Za poboljšanje agenta bi trebalo uzeti u obzir druge reprezentacije ulaza u mrežu, te dodatno testiranje parametara.

LITERATURA

- [1] Apoorv Pandey: Solving Dots & Boxes Using Reinforcement Learning, 2022.
- [2] Lex Weaver, Terry Bossomaier: Evolution of Neural Networks to Play the Game of Dots-and-Boxes, 1998.
- [3] Shuqin Li, Dongming Li, Xiaohua Yuan: Research and Implementation of Dots-and-Boxes Game System, 2012.
- [4] Bojana Dalbelo Bašić, Marko Čupić, Jan Šnajder: Umjetne neuronske mreže, 2008.
- [5] UCLA: Some Strategy For Dots And Boxes, https://www.math.ucla.edu/~tom/Games/dots&boxes_hin

Evolucijski algoritmi za rješavanje igre Dots and Boxes

Sažetak

Opisati problem pronalaženja optimalnog poteza u igrama s više igrača. Na nekoliko primjera igara oblikovati problem odabira poteza kao problem optimizacije ili strojnog učenja. Posebnu pažnju posvetiti mogućnosti primjene evolucijskih algoritama u pronalasku što kvalitetnijeg poteza. Ostvariti programski sustav za pronalaženje što kvalitetnijeg poteza za igru Dots and Boxes temeljen na evolucijskim algoritmima. Eksperimentalno utvrditi učinkovitost različitih inačica algoritma optimizacije ovisno o parametrima algoritma i postavkama igre. Radu priložiti izvorne tekstove programa, dobivene rezultate uz potrebna objašnjenja i korištenu literaturu. **Ključne riječi:**

strojno učenje, evolucijski algoritmi, neuronska mreža, dots and boxes

Evolutionary algorithms for solving Dots and Boxes

Abstract

Abstract.

Keywords: machine learning, evolutionary algorithms, neural network, dots and boxes