

BRIŠKULA I UMJETNA INTELIGENCIJA

Ivan Laković

ilakovi@student.math.hr
PMF - Matematički odsjek

O briškuli

Briškula (Briscola) je talijanska kartaška igra udomaćena na hrvatskom priobalju i otocima.

- u špilu ima 40 karata.

- postoje 4 tipa po 10 karata: kope, baštone, špade i dinari

- nakon sto svi igrači bace kartu, rundu uzima igrač koji je bacio najjaču briškulu ili ako briškule nema, igrač koji je bacio najjaču kartu iste boje kao i igrač koji je igrao prvi u trenutnoj rundi.

Primjer stabla

```
[M:None W/V:26/2000 U:]potez igra 2
|[M:0 W/V:0/13 U:]potez igra 1
|[M:1 W/V:6/6 U:]potez igra 2
|[M:0 W/V:0/3 U:]potez igra 2
|[M:0 W/V:0/2 U:]potez igra 1
|[M:0 W/V:1/1 U:]potez igra 2
|[M:1 W/V:0/2 U:]potez igra 2
|[M:0 W/V:0/1 U:]potez igra 1
|[M:0 W/V:6/6 U:]potez igra 2
|[M:1 W/V:0/3 U:]potez igra 2
|[M:0 W/V:0/2 U:]potez igra 1
|[M:0 W/V:1/1 U:]potez igra 2
|[M:0 W/V:0/2 U:]potez igra 1
|[M:0 W/V:0/2 U:]potez igra 1
|[M:0 W/V:1/1 U:]potez igra 2
|[M:2 W/V:1974/1974 U:]potez igra 2
|[M:0 W/V:367/987 U:]potez igra 1
|[M:0 W/V:0/493 U:]potez igra 2
|[M:0 W/V:492/492 U:]potez igra 1
|[M:0 W/V:0/491 U:]potez igra 2
|[M:1 W/V:0/493 U:]potez igra 2
|[M:0 W/V:492/492 U:]potez igra 1
|[M:0 W/V:0/491 U:]potez igra 1
|[M:1 W/V:386/986 U:]potez igra 1
|[M:1 W/V:0/493 U:]potez igra 2
|[M:0 W/V:492/492 U:]potez igra 1
|[M:0 W/V:0/491 U:]potez igra 2
|[M:0 W/V:0/492 U:]potez igra 2
|[M:0 W/V:491/491 U:]potez igra 1
|[M:0 W/V:0/490 U:]potez igra 1
|[M:0 W/V:0/490 U:]potez igra 2
|[M:1 W/V:0/13 U:]potez igra 1
|[M:1 W/V:6/6 U:]potez igra 2
|[M:1 W/V:0/3 U:]potez igra 2
|[M:0 W/V:0/2 U:]potez igra 1
|[M:0 W/V:1/1 U:]potez igra 2
|[M:0 W/V:0/2 U:]potez igra 1
|[M:0 W/V:1/1 U:]potez igra 2
|[M:0 W/V:6/6 U:]potez igra 2
|[M:0 W/V:0/3 U:]potez igra 1
|[M:0 W/V:2/2 U:]potez igra 2
|[M:0 W/V:0/1 U:]potez igra 2
|[M:1 W/V:0/2 U:]potez igra 2
|[M:0 W/V:0/1 U:]potez igra 1
```

Kako UI vidi poteze

Stablo pretraživanja UCT-a 3 koraka prije kraja igre.

Vidi se da ako odigra kartu s rednim brojem 2 sigurno pobjeđuje, te je to i razlog zašto bacanje te karte češće analizira u odnosu na druge dvije.

Iz ovakvog stabla vidljivo je koju kartu će UCT odigrati i zašto.



Fig. 1: Podijeljene karte

Kad igra započne svakom igraču se djieli 3/4 karte te se jedna okrene licem prema gore, te ona označava briškulu.

Monte Carlo stablo pretraživanja

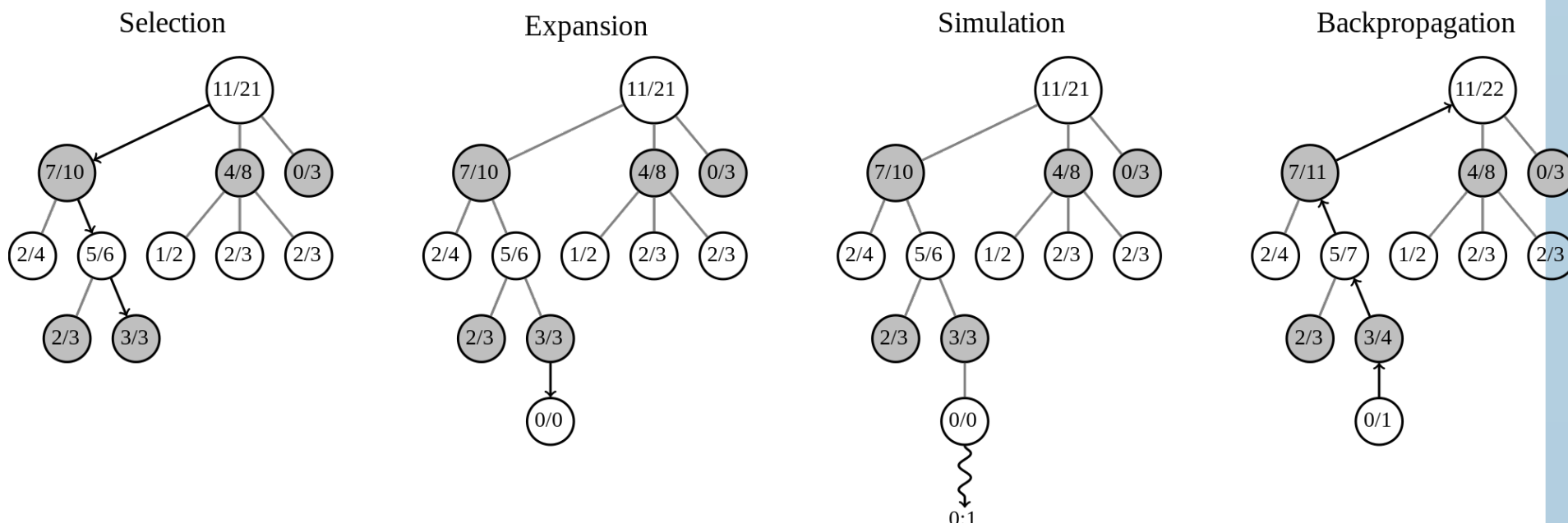
Monte Carlo stablo pretraživanja je heuristički algoritam pretraživanja koji se koristi u procesima donošenja odluka, najčešće u igranju igara. Samom algoritmu prethodio je John von Neumannov teorem.

Teorem. *Minimax teorem. Neka su $X \subset \mathbb{R}^n$ i $Y \subset \mathbb{R}^m$ kompaktni konveksni skupovi. Ako je $f : X \times Y \rightarrow \mathbb{R}$ neprekidna funkcija koja je konveksno-konkavna. Tada slijedi:*

$$\min_{x \in X} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{x \in X} f(x, y) \quad (1)$$

Minimax teorem (1) formirao je bazu za teoriju odlučivanja u računarstvu i AI-u. Algoritam simulira igru zadani broj puta te pronalazi optimalan potez. Svaka simulacija igre se sastoji od iduća 4 koraka:

- Odabir** - odabir najuspješnijega djeteta.
- Proširivanje** - ako nije kraj igre odigra potez.
- Simulacija** - odigra simulaciju do kraja.
- Ažuriranje znanja** - ažurira odigrani potez ovisno o rezultatu.



UCT algoritam

Najteži posao je održati ravnotežu između iskorištavanja kombinacija koje donosi odabir djeteta s visokim stupnjem pobjeda i proširivanja poteza s malim brojem simulacija. Prva uspješna formula koja je to ostvarila je UCB:

$$\frac{w_i}{n_i} + c * \sqrt{\frac{\ln t}{n_i}} \quad (2)$$

gdje je:

- w_i označava broj pobjeda nakon i-tog poteza
- n_i označava broj simulacija nakon i-tog poteza
- c je parametar iskorištavanja, teoretski je uvijek $\sqrt{2}$
- t označava ukupan broj simulacija koji odgovara sumi n_i -ova

Prvi dio formule (2) zadužen je za produbljivanje postojećeg stanja, a drugi za proširivanje stabla radi problema lokalnog ekstrema. Dokazano je da procjena najboljih poteza u MCTS algoritmu konvergira minimax procjeni.

UCT je poseban slučaj MCTS algoritma te se može opisati kao:

$$UCT = MCTS + UCB \quad (3)$$

Sažetak

Cilj projekta je bio dokazati da će se umjetna inteligencija moći nositi s ljudskim igračima u igri briškula za koju se smatra da najvećim dijelom čista sreća, a manjim dijelom znanje i taktika. U slučaju da je umjetna inteligencija bolja od ljudskog igrača želja nam je bila vidjeti razliku u načinu igre, te pokušati i sami naučiti igrati na taj bolji način. Umjetnu inteligenciju odlučili smo implementirati pomoću algoritma **Monte Carlo tree search**. Važan čimbenik u odabiru je bila i nedavna pobjeda Google-ovog AI-a **AlphaGo** u igri Go (posljednja igra na ploči u kojoj je čovjek mogao pobijediti računalno) umjetna inteligencija koristila kombinaciju dubokog učenja i MCTS algoritma.

UCT vs UCT

Borba umjetnih inteligencija s različitim brojem simulacija igre po potezu. U tablici je prosječan

UCT 1	UCT 2	Omjer pobjeda
300	500	25:25
500	750	26:24
300	1000	15:35
500	1000	21:29
800	1000	19:31
900	1000	27:23
1000	1500	25:25
300	2000	16:34
500	2000	22:28
750	2000	22:28
1000	2000	22:28
1500	2000	23:27
1750	2000	22:28
2000	2500	23:27
2000	3000	26:24
2000	4000	27:23
500	5000	22:28
2000	5000	27:23
2000	10000	24:26

Iz tablice je vidljivo da je optimalan broj simulacija oko 2000.

Važni pojmovi

- MCTS - Monte Carlo Tree Search, algoritam za korištenje stabla pretraživanja
- UCT - Upper Confidence Bound 1 applied to trees, implementacija MCTS koja balansira proširivanje i produbljivanje stabla pretraživanja
- Stablo pretraživanja - stablo u kojem svaki čvor sadrži podatke o broju pobjeda u odnosu na broj simulacija za svaki odigrani potez
- Briškula - kartaška igra, popularna u priobalju Jadranskog mora, ujedno i naziv boje koja trenutno vodi igru
- AlphaGo - Googleov program, prvi program koji je pobijedio ljudskog protivnika u igri Go

Testiranje algoritma

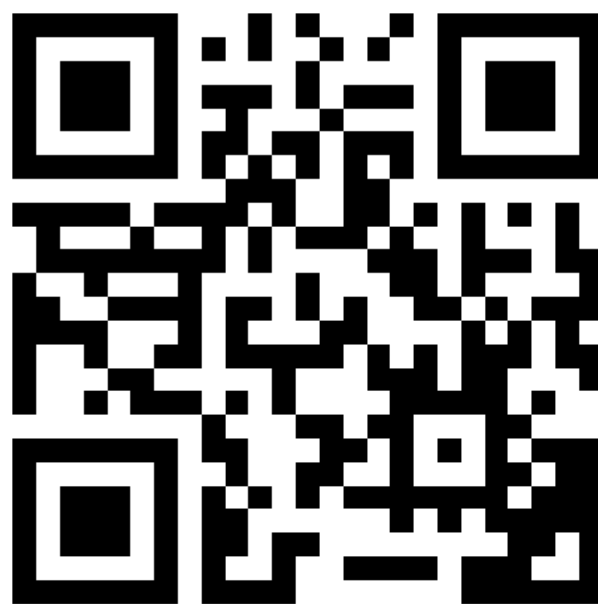
Algoritam smo osim igranjem protiv njega, radi lakšeg i bržeg testiranja odlučili testirati na 3 načina:

- Borbom protiv samog sebe s različitim brojem iteracija.
- Borbom protiv heuristike koja simulira igranje čovjeka
- Borbom protiv samog sebe s različitim koeficijentom c u UCT formuli.

- U borbi protiv samog sebe pokušali smo pronaći optimalan broj iteracija tako da se na odgovor algoritma ne čeka previše, ali opet da bude dobro obaviješten. Kao što se vidi iz tablice, optimalan broj iteracija je oko 2000. Ako mu suprotstavimo UCT s manjim brojem iteracija primjećujemo da UCT s 2000 iteracija uvijek pobijedi više od 50% puta, dok u suprotnom slučaju kad se bori protiv UCT-a s više iteracija rezultat nije predvidljiv.
- U borbi protiv naše heuristike igra završi gotovo uvijek jednako, to jest UCT uvijek prevlada, ali bez obzira na broj iteracija, kako briškula ipak ima udio sreće nikad ne uspije pobijediti u 100% slučajeva.
- Borbom UCT-a s istim brojem iteracija, a različitim koeficijentima pri čemu jedan UCT ima fiksiran koeficijent od $\sqrt{2}$ zaključujemo da veličina izabranoga koeficijenta c ukoliko je on u intervalu između 0.4 i 20 ne mijenja bitno situaciju, te zbog nausumičnosti podjele karata broj pobjeda varira.

Link na igricu

Za pokrenuti igru treba: skinuti i raspakirati mapu s danog linka. Na izboru su 4 opcije igranja: easy, medium, hard te igranje s otvorenim kartama.



<https://goo.gl/a2bMXZ>

Fig. 2: QRcode