

MINIMAX S ALFA - BETA PROŘEZÁVÁNÍM

ZADÁNÍ

Cílem úlohy aplikace algoritmů minimax a minimax s alfa-beta prořezáváním na hru piškvorky.

POPIS ALGORITMU :

Minimax uvažuje dva agenty, kteří se snaží minimalizovat či maximalizovat zisk, respektive. Algoritmus postupně prochází všechny tahy, a vzniknuvší stromová struktura střídá minimalizujícího a maximalizujícího agenta. Každý uzel je ohodnocen:

- Je-li uzel listem, je ohodnocen bodovací funkcí
- V případě uzlu maximalizujícího agenta se vybere největší hodnota z potomků uzlu
- V případě uzlu minimalizujícího agenta se vybere nejmenší hodnota z potomků - uzlu.

Heuristická funkce ohodnocuje stav hry jako součet ohodnocení jednotlivých (vertikálních, horizontálních a diagonálních) čar. Hodnota čáry se spočítá následovně

10^{k-1} , k = počet žetonů hráče na čáře, pokud jsou na čáře žetony pouze jednoho hráče, jinak 0.

Tato hodnota je přičtena/odečtena (záleží, jestli se jedná o min, nebo max hráče), k ohodnocení hry.

Algoritmus je vylepšen o alfa-beta prořezávání, které zrychluje výpočet přeskočením větví (ořezáním), které mají garantovaný horší výsledek pro agenta. Hodnoty alfa a beta jsou nejlepší nalezené hodnoty pro maximum a minimum od aktuálního uzlu ke kořenu stromu. Parametry alfa a beta reprezentují nejlepší možnosti pro agenty od daného uzlu ke kořeni.

Pro uzly, které se snaží hodnoty svých potomků maximalizovat to znamená, že pokud najdou hodnotu větší, než je aktuální beta, tak zbývající potomci mohou být ignorováni, protože minimalizující agent o úroveň výše se touto cestou nevydá. Obdobným způsobem fungují uzly, které hodnotu minimalizují, ovšem ty zjišťují, zda-li není hodnota potomka menší, než alfa.

PROVEDENÉ EXPERIMENTY A JEJICH VÝSLEDKY

Algoritmus byl implementován ve třech fázích, a výsledky obou fází porovnány. Na grafu níže je záznam běhů obou algoritmů. V první fázi byl implementován minimax bez alfa-beta prořezávání, kde první běh, kde bych prozkoumáván prostor velikosti $3 = 19683$ minimaxu běžel zhruba 47 vteřin.

V druhé fázi byl implementován minimax s alfa-beta prořezáváním, který dramaticky zrychlil první běh ze 47 vteřin na 5 vteřin. Následující průchody stavovým prostorem byly vzhledem ke snižování exponentu dramaticky kratší a po třetím průchodu se dají označit za téměř konstantní.

Ve třetí fázi byla přidána omezení doby běhu. Prohledávání herního stromu probíhá jen do určité maximální hloubky stromu, a stejně tak je běh každého vyhodnocení omezen časově. Pro hry do velikosti 5x5 včetně není prohledávání přerušeno překročením doby běhu.

INSTRUKCE KE SPUŠTĚNÍ

Požadavky:

1. Python 3
2. Numpy

Program vyžaduje dva argumenty, a to sice počet lidských hráčů (kteří budou umisťovat své značky zadáváním do konzole) a velikost mřížky – výška, potažmo šířka mřížky. V případě, že počet lidských hráčů je roven nule, hrají proti sobě dvě umělé inteligence, jedna z nichž používá funkci min a druhá max. Příklad spuštění programu:

```
./main.py -players 2 -grid-size 3
```

Program bude spuštěn pro dva lidské hráče na mřížce 3x3.

ZDROJE

jacek, 2020. CodinGame. [Online]

Available at: <https://www.codingame.com/playgrounds/46262/on-generating-opening-book-for-yavalath-mcts-bot>