

# Ročníkový projekt - report za letný semester

Lukáš Hornáček

## Práca počas letného semestra:

Implementoval som algoritmy MTD(f) <sup>1</sup> a MCTS <sup>2</sup>. Implementoval som tiež niekoľko rôznych funkcií na vyhodnocovanie hodnoty konkrétneho stavu šachovnice, ktoré sú opísané nižšie. Porovnal som efektivitu týchto funkcií a algoritmov.

## Výsledky:

Z mojich implementácií algoritmov Alpha-beta pruning <sup>3</sup>, PVS <sup>4</sup> a MTD(f), ktoré vyberajú ťah s rovnakou hodnotou bol nakoniec najrýchlejší jednoduchý Alpha-beta pruning. Tento som ďalej porovnával s dvoma verziami MCTS. Prvá (označovaná ako MCTSR) v simulačnej fáze vyberá ťahy náhodne, zatiaľ čo druhá (označovaná ako MCTSAB) v simulačnej fáze v každom ťahu spustí Alpha-beta pruning s hĺbkou prehľadávania 3.

**Výsledky porovnania algoritmov:** Na testovanie algoritmov bola použitá evaluačná funkcia NumberOfPieces. Množstvo výhier algoritmu v percentách je ďalej označované ako úspešnosť.

Alpha-beta pruning vs MCTSR		
hĺbka prehľadávania	počet iterácií v simulačnej fáze	úspešnosť Alpha-beta pruningu
1	10 000	55
1	50 000	25
1	100 000	20
2	10 000	81
2	50 000	59
2	100 000	56
3	10 000	72
3	50 000	45
3	100 000	28
4	10 000	83
4	50 000	68
4	100 000	74
5	10 000	89
5	50 000	80
5	100 000	72
6	10 000	98
6	50 000	94
6	100 000	91
7	10 000	99
7	50 000	95
7	100 000	89
8	10 000	100
8	50 000	100
8	100 000	99

<sup>1</sup>[https://en.wikipedia.org/wiki/MTD\(f\)](https://en.wikipedia.org/wiki/MTD(f))

<sup>2</sup>[https://en.wikipedia.org/wiki/Monte\\_Carlo\\_tree\\_search](https://en.wikipedia.org/wiki/Monte_Carlo_tree_search)

<sup>3</sup>[https://en.wikipedia.org/wiki/Alpha-beta\\_pruning](https://en.wikipedia.org/wiki/Alpha-beta_pruning)

<sup>4</sup>[https://en.wikipedia.org/wiki/Principal\\_variation\\_search](https://en.wikipedia.org/wiki/Principal_variation_search)

Alpha-beta pruning vs MCTSAB		
hlĺbka prehľadávania	počet iterácii v simulačnej fáze	úspešnosť Alpha-beta pruningu
1	100	40
1	500	64
1	1 000	20
2	100	57
2	500	73
2	1 000	25
3	100	31
3	500	47
3	1 000	11
4	100	64
4	500	73
4	1 000	40
5	100	69
5	500	73
5	1 000	59
6	100	97
6	500	88
6	1 000	88
7	100	97
7	500	97
7	1 000	80
8	100	99
8	500	96
8	1 000	96

MCTSR vs MCTSAB		
počet iterácii v simulačnej fáze MCTSR	počet iterácii v simulačnej fáze MCTSAB	úspešnosť MCTSR
10 000	100	55
10 000	500	52
10 000	1 000	63
50 000	100	60
50 000	500	62
50 000	1 000	50
100 000	100	73
100 000	500	82
100 000	1 000	69

Ďalej som porovnával 4 rôzne funkcie na vyhodnocovanie hodnoty stavu šachovnice, zoradené podľa ich rýchlosti:

1. NumberOfPieces: Hodnota šachovnice je počet vlastných figúrok mínus počet súperových figúrok.
2. Positions: Za každú svoju/súperovú figúrku pridá/odoberie hodnotu podľa toho, na akej pozícii sa figúrka nachádza. Hodnoty pozícii pre biele figúrky sú uvedené v nasledujúcej tabuľke: (pre čierne figúrky je tabuľka prevrátená)

5	15	15	5	5	15	15	5
2	3	3	3	3	3	3	2
4	6	6	6	6	6	6	4
7	10	10	10	10	10	10	7
11	15	15	15	15	15	15	11
16	21	21	21	21	21	21	16
20	28	28	28	28	28	28	20
36	36	36	36	36	36	36	36

3. SafePositions: Pripočítava hodnoty rovnako ako 2. funkcia, ale navyše kontroluje či je daná pozícia bezpečná, teda či daná figúrka nemôže byť v ďalšom ťahu vyhodená. Ak je pozícia bezpečná pridá/odoberie hodnotu pozície \* 1.5, inak pridá/odoberie hodnotu rovnako ako 2. funkcia.
4. AdvanceablePositions: Pripočítava hodnoty rovnako ako 2. funkcia, ale navyše pomocou niekoľkých bitových masiek kontroluje, či sa figúrka z danej pozície už nevie určite do dvoch ťahov dostať na výhernú pozíciu. Ak áno opäť pridá/odoberie hodnotu pozície \* 1.5, inak pridá/odoberie hodnotu rovnako ako 2. funkcia.

**Výsledky porovnania funkcií:** Na testovanie funkcií bol použitý Alpha-beta pruning. Množstvo výhier funkcie v percentách je ďalej označované ako úspešnosť.



