

Varianty Monte Carlo Tree Search

Tomáš Kuča

tomas.kuca@matfyz.cz

Herní algoritmy
MFF UK Praha

2011

O čem bude přednáška?

- Monte Carlo Tree Search
- od her podobných Go (bez Go) k vzdálenějším
- rozdíly a rozšíření MTCS

Postupné budování herního stromu

Algoritmus opakuje čtyři fáze:

- **Selekce**

- vybere list z herního stromu (nejdříve nenavštívené)
- Multi-armed bandit problem
- typický UCT algoritmus: $U_i = v_i + c\sqrt{\frac{\ln N}{n_i}}$

- **Expanze** - přidá nový uzel

- **Simulace** - odhad pravděpodobnosti výhry (bodového zisku) v daném uzlu na základě mnoha simulací

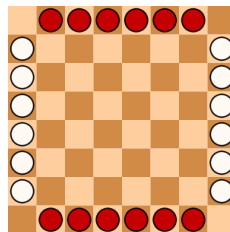
- **Backpropagation** - informace o hodnotě uzlu je propagována stromem na cestě ke kořeni

- nepotřebuje nutně evaluační funkci
- lze přidávat doménově specifické heuristiky

- má problémy s pozicemi, kde je jen jeden "správný" tah, tzv. killer move
- často nedokáže odhalit vyhrané (nebo prohrané pozice), obzvlášť ve hrách s náhlou smrtí (šachy, hex, lines of action)
 - $\alpha\beta$ -prohledávání nebo proof number search dávají v takových situacích lepší výsledky
 - řeší tzv. MCTS solver

Lines of action

- hra dvou hráčů s perfektní informací, nulovým součtem
- hrací deska o velikosti 8×8 (obecně $n \times n$)
- počáteční pozice viz obr.
- hráči se střídají v tazích
- cílem hry je uspořádat kameny tak, aby se všechny kameny hráče dotýkaly (stačí diagonálně)



Tah v Lines of action

- během svého tahu hráč pohne jedním kamenem horizontálně, vertikálně nebo diagonálně
 - kámen se pohne o tolik pozic, kolik je v daném řádku/sloupci/diagonále kamenů
 - nesmí přeskočit soupeřův kámen
 - pokud skončí přesně na pozici soupeřova kamene, zakryje ho (soupeřův kámen je ze hry odstraněn)
- pokud hráč nemůže hrát, passuje
- pokud se třikrát opakuje pozice → remíza
- pokud spojení do jedné komponenty nastane současně → remíza

- při selekci používá "progressive bias"
- $$U_i = v_i + c\sqrt{\frac{\ln N}{n_i}} + \frac{W \times P_c}{n_i + 1}$$
 - W je konstanta (použitá hodnota 100)
 - P_c je pravděpodobnost tahu dané kategorie, získaná na základě předchozích her
 - kategorie jsou určeny dle typu tahu (zajetí, blokování) a dle počáteční a koncové pozice kamene (hrací deska se rozdělí na několik částí - střed, rohy, okraje)
- při simulaci nejsou tahy náhodně, ale též s pravděpodobnostmi dle kategorií
- pokud evaluační funkce přesáhne určitou hranici, považuje se hra za vítěství (podobně prohra)
- při backpropagation se propagují hodnoty 1, 0, -1

- tato varianta má stále problém s náhlou smrtí
- nalezení výherního tahu trvá dlouho

- tato varianta má stále problém s náhlou smrtí
- nalezení výherního tahu trvá dlouho

Backpropagation

- propagují se hodnoty $\{-\infty, 0, \infty\}$
- propagace podobně jako v MINIMAX
- pokud je na tahu hráč a propagují $-\infty$:
 - propaguje $-\infty$, pokud mají všichni sourozenci $-\infty$
 - propaguje -1 jinak
- obdobně pro výherní pozice
- jak to ovlivní selekci?

- hodnota uzlu se může rychle měnit, proto nelze vzít uzel s nejlepší hodnotou
- pokud je nalezena vítězná strategie, hraje se ta, jinak
- $v + \frac{A}{\sqrt{n}}$, kde
 - v je hodnota uzlu
 - A je konstanta (např. 1)
 - n je počet návštěv uzlu

- MTCS-solver porazí MTCS v 65
- MTCS-solver porazí starší verze MIA 2000, 2002 (pravděpodobně nejlepší program, založený na $\alpha\beta$ prohledávání) (získá cca 60% bodů)
- nejnovější verze MIA 2006 výrazně porazí i MTCS-solver (11%)

- hra dvou hráčů s perfektní informací, nulovým součtem
- hrací deska o velikosti 11×11 (obecně $n \times n$)
- hráči v každém tahu přidají jeden kámen své barvy
- hráči se střídají v tazích
- cílem hry spojit protilehlé strany desky (každý hráč má "své" dvě strany)



Mrtvé oblasti

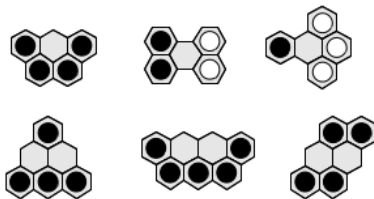


Fig. 2. Local fillin patterns. Top three patterns identify dead cells, and can be filled in with stones of either player. Bottom three patterns identify Black-captured regions, and can be filled in with Black stones.

- mrtvé buňky - nejsou zajímavé ani pro jednoho hráče
- zajaté území - pro jednoho z hráčů nemá smysl tam hrát

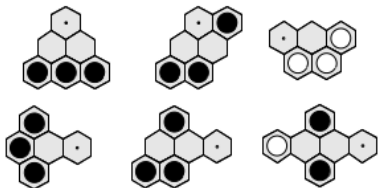


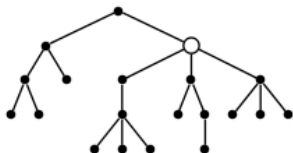
Fig. 3. Local inferior cell patterns. Empty cells can be pruned from consideration by Black based on their reversible/dominated relation to the corresponding dotted cells.

- některé tahy nemusím brát v úvahu, protože jsou dominovány jinými tahy

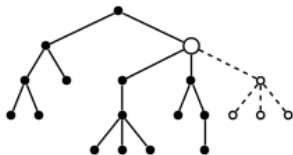
- algoritmus pro nalezení strategie pro spojení dvou buněk
- neúplný, občas přehlíží i pro člověka triviální tahy
- ale pokud existuje řešení, často ho najde

- standartní MTCS
- nejsilnější pokud používá AMAF heuristiku, bez prohledávání (exploration)
- simulace náhodné (permutace prázdných polí)
- pouze "bridge" pattern, ostatní nepřidávají zlepšení
- při určitém počtu navštívení uzlu z něho spustí H-search
- prořezání stromu

Hex - prořezání



1) Node reaches knowledge threshold; perform inferior cell analysis and H-search computations.



2) Prune children that are inferior or outside of mustplay.



- MTCS + bridge pattern vs MTCS → 65% win rate
- MTCS + AMAF vs MTCS → 74% win rate
- MTCS vs Wolve (nejlepší známý program) → 49% win rate

- šachy, kdy neznám pozici soupeřových figur
- jen koncovka King + Rook vs King má prohledávací prostor jako dáma
- rozdíly oproti Phantom Go
 - dynamické
 - získaná informace rychle zastarává
 - hra nemá omezený počet tahů
 - podmínky pro vítězství

 - standartní MTCS selhává
 - místo simulace protivníka simulovat jen kategorie tahů
 - omezit počet tahů při simulaci

- Scrabble, Poker, Game of Scotland Yard, Bridge, generický hráč, evropské hry
- časem možná i RTS