Outline

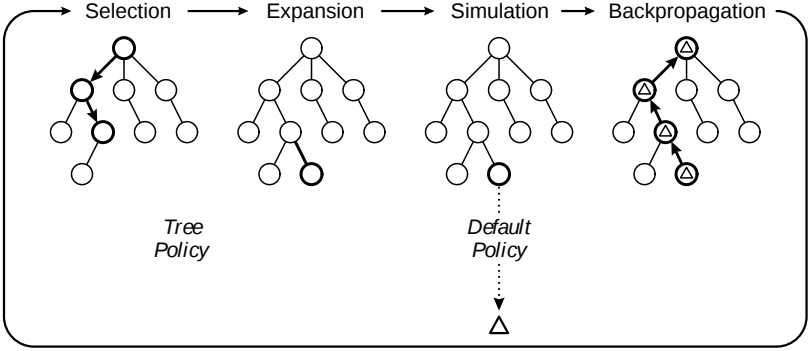
Problem

- drzewo
  - duże. nie mieści się w całości w pamięci
  - stany, akcje
- wartości w liściach
  - stałe
  - losowe, nieznany rozkład
- ścieżka do najlepszego liścia
- liczba graczy
- informacja

Min-max i $\alpha\beta$

MCTS

1. Powtarzaj
   1. Selekcja (tree policy)
   2. Ekspansja
   3. Symulacja (playout, default policy) – rozkład jednostajny lub heurystyka
   4. Propagacja
2. Wybierz najlepszy ruch na podstawie zebranych danych



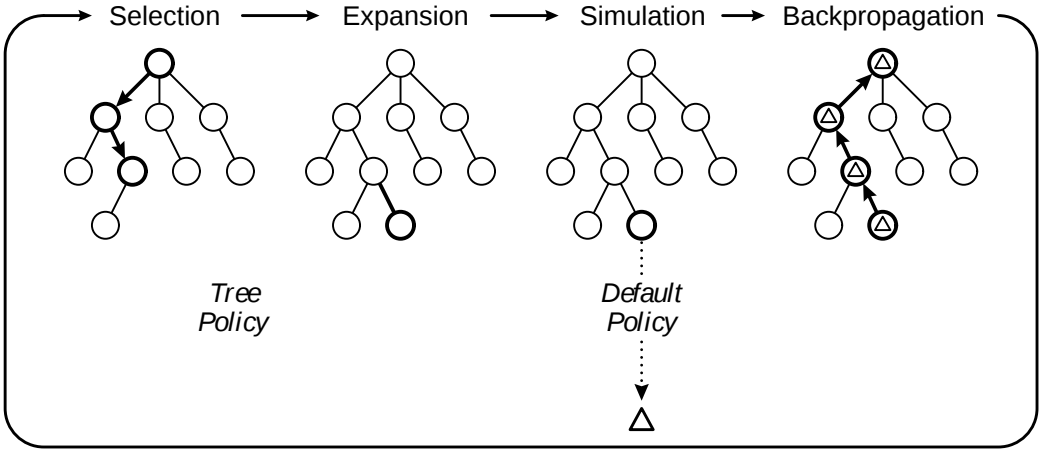
(Źródło: Browne et al. A Survey of MCTS Methods)

Selekcja/Ekspansja

1. Jeśli węzeł ma jakiś nieodwiedzony ruch, wybierz jeden z nich i dodaj do drzewa (ekspansja)
2. W przeciwnym przypadku: wybierz najlepszy ruch według aktualnie zebranych statystyk i wykonaj selekcję na niższym poziomie (selekcja)

Typowe użycie MCTS

1. Wykonaj MCTS dla bieżącego stanu gry
2. Po upływie zadanego czasu wykonaj najlepszy ruch według aktualnej oceny
3. Przenieś zebrane informacje (poddrzewo) do MCTS w nowym stanie



Zalety MCTS

- możliwość przerwania obliczeń w dowolnym momencie
- brak heurystyk związanych z konkretną grą
- asymetryczna budowa drzewa w pamięci

Propagacja z każdego kroku vs propagacja z liścia

- wypłata nie ma wpływu na ruchy poniżej
- dyskontowanie
- możliwość przerwania symulacji przed liściem

Multi-armed bandit-problem

- n automatów, każdy z wypłatami z pewnego (innego) rozkładu z pewną wartością oczekiwaną
- kolejne losowania niezależne
- jaką przyjąć strategię, żeby zmaksymalizować wypłatę, mają zerową wiedzę o tych rozkładach?
- w każdym kroku pozyskujemy nową wiedzę o jednym (wybranym w danym kroku) rozkładzie

Selekcja: UCB1

zbierane dane n_i licznik odwiedzin i-tego ruchu z węzła, Q_i suma wypłat z symulacji dotąd

wybór akcji najlepszy ruch według

$$\frac{Q_i}{n_i} + C\sqrt{\frac{2 \ln N}{n_i}}$$

, $N = \sum n_i$

Eksploracja vs. eksploatacja

Auer, Cesa-Bianchi, Fischer. Finite-time analysis of the multiarmed bandit problem (2002)

UCB1+MCTS=UCT

Dowód: (Theorem 6) Dla UCT zastosowanego do MDP prawdopodobieństwo błędnej oceny ruchu w korzeniu zbiega do 0 wraz ze wzrostem liczby iteracji.
Koscis, Szepesvari. Bandit based Monte-Carlo planning (2006)

Selekcja: Bayesian UCT

- P_i – każdy węzeł zawiera estymację rozkładu prawdopodobieństwa wypłat dla ruchu i.
- $$B_i = \mu_i + \sqrt{2 \ln N / n_i}$$
- $\mu_i = E(P_i)$
- Wersja 2:
$$B_i = \mu_i + \sqrt{2 \ln N} \sigma_i$$
- uzależnienie eksploracji od odchylenia std bieżącej estymacji.

Tesauro, Rajan, Segal. Bayesian Inference in Monte-Carlo Tree Search (2012)

Selekcja: EXP3 (wypłaty z [0,1])

początkowe wartości $w_i(1) = 1$

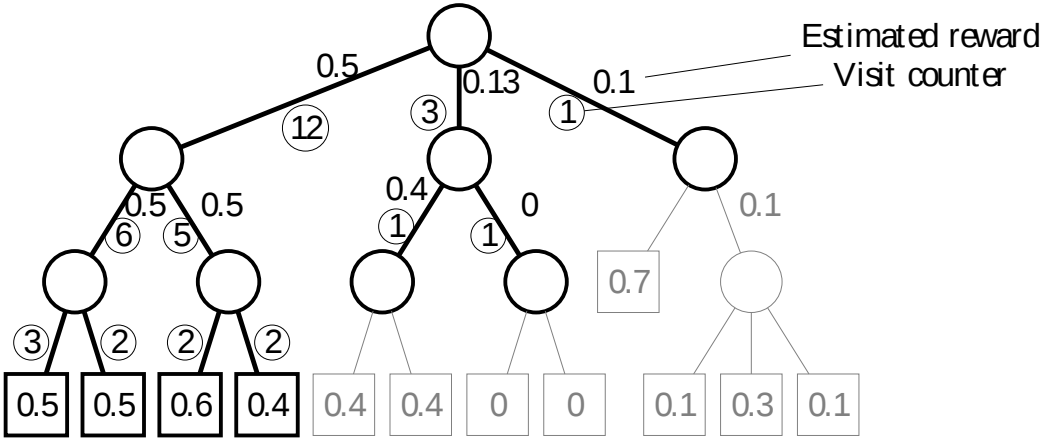
wybór akcji z prawdopodobieństwem

$$p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum w_j} + \frac{\gamma}{K}$$

aktualizacja $\hat{x}_j(t) = x_j(t)/p_j(t)$ dla wybranego ruchu, 0 w p.p.;
$w_j(t+1) = w_j(t)EXP(\gamma\hat{x}_j(t)/K)$

Szybka reakcja na spadek wypłaty, niedeteminizm. Nauka online.
Auer, Cesa-Bianchi, Freund, Schapire. The nonstochastic multiarmed bandit problem. (2002)

Konstrukcja drzewa



Proste zrównoleglenie

Leaf parallelization w kroku symulacji uruchom n niezależnych przebiegów i propaguj uśrednioną wartość.

Root parallelization wykonaj równolegle n procesów MCTS, na koniec sklej drzewa i wybierz najlepszy ruch.

Wady

- brak komunikacji między procesami – wielokrotne wyliczanie tej samej informacji,
- konieczność czekania na wszystkie procesy.

Zrównoleglenie (MPI)

- proces master:
  1. dla każdego z n slave'ów: wykonaj selekcję w aktualnym drzewie, wyślij sekwencję ruchów do slave
  2. czekaj na wyliczenie wyniku gry przez slave
     1. odbierz wynik
     2. aktualizuj drzewo
     3. wykonaj selekcję i wyślij kolejną sekwencję
- procesy slave wykonuj zadaną sekwencję aż do liścia.

Cazenave, Jouandeau. A Parallel Monte-Carlo Tree Search Algorithm. (2008)

Zrównoleglenie (Tree parallelization)

- symetryczne procesy, wspólne drzewo
- mutex w każdym węźle drzewa na dostęp do statystyk
- przebiegi jak w klasycznym MCTS
- virtual loss – dodatkowa kara na wybór ruchu, który został wybrany przez inny wątek, ale wyniki nie zostały przepropagowane.

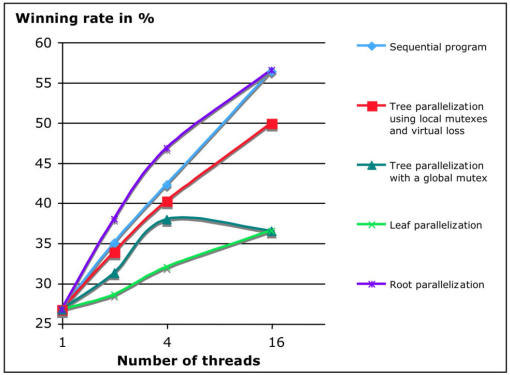


Fig. 5. Performance of the different parallelization methods

Nested MCTS

jeden gracz, statyczne wypłaty

1. Parametr: głębokość przeszukiwania; L
2. Jeśli $L = 1$
   1. Wykonaj losowe symulacje po wykonaniu każdego ruchu, zwróć najlepszy wynik i ruch
3. Jeśli $L > 1$
   1. Wykonaj nested MCTS(L-1) dla każdego ruchu, zwróć najlepszy wynik i ruch.

Procedurę można powtarzać wielokrotnie, zapamiętując najlepsze ruchy miedzy iteracjami.

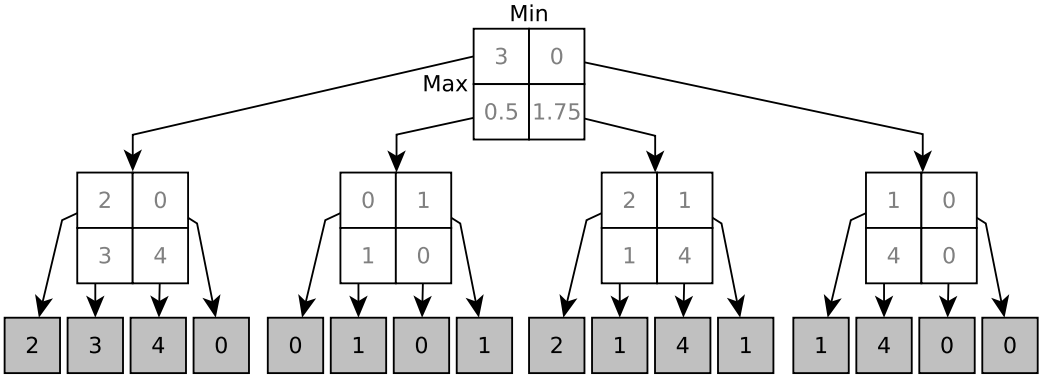
- wybór według maksimum
- łatwe zrównoleglenie

Cazenave. Nested Monte-Carlo Search. (2009)
Cazenave, Jouandeau. Parallel Nested Monte-Carlo search (2009)

Dwóch graczy: Ruchy naprzemienne

- standardowe MCTS, propagowanie całego wektora wypłat (dla każdego gracza). Polityka wyboru względem gracza wykonującego ruch.
- przy odpowiednio dobranej selekcji zbiega do strategii min-max. (Przy założeniu zbudowania pełnego drzewa).

Dwóch graczy: Ruchy jednoczesne



- równowaga Nasha
- strategie mieszane
- SM-MCTS

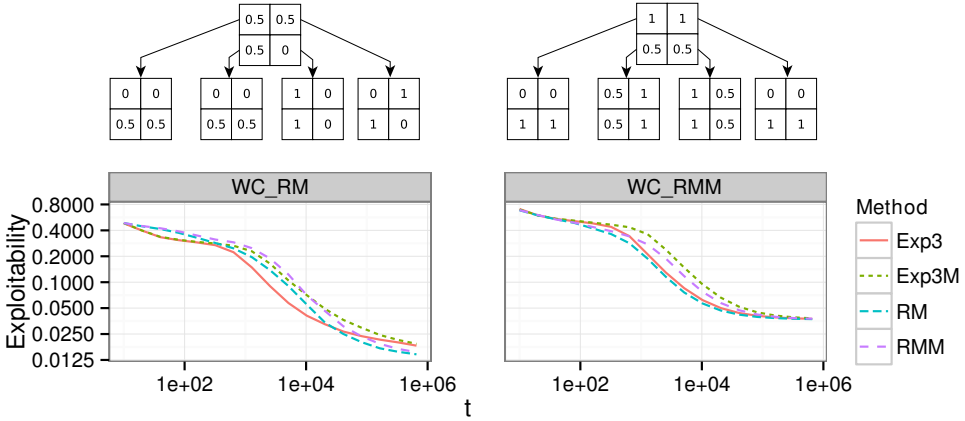
Lisy, Kovarik, Lanctot, Bosansky. Convergence of Monte Carlo Tree Search in Simultaneous Move Games. (2013)

SM-MCTS

```
1        (node h)
2   if h ∈ Z then return u_1(h)
3   else if h ∈ T and ∃(i,j) ∈ A_1(h) × A_2(h) not previously selected then
4       Choose one of the previously unselected (i,j) and h^0 ← T(h,i,j)
5       Add h^0 to T
6       u_1 ← Rollout(h^0)
7       X_{h^0} ← X_{h^0} + u_1; n_{h^0} ← n_{h^0} + 1
8       Update(h,i,j,u_1)
9       return RetVal(u_1, X_{h^0}, n_{h^0})
10  (i,j) ← Select(h)
11  h^0 ← T(h,i,j)
12  u_1 ← SM-MCTS(h^0)
13  X_h ← X_h + u_1; n_h ← n_h + 1
14  Update(h,i,j,u_1)
15  return RetVal(u_1, X_h, n_h)
```

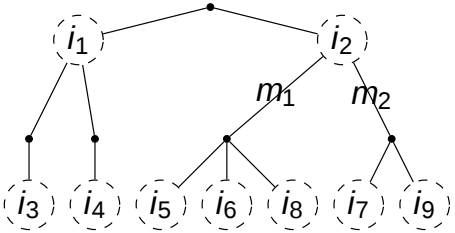
- Użycie UCB1 nie gwarantuje zbieżności do NE (Decoupled UCT)
- Update propaguje:
  - wartość z pojedynczej symulacji,
  - średnią z węzła (po uwzględnieniu wartości z dołu).

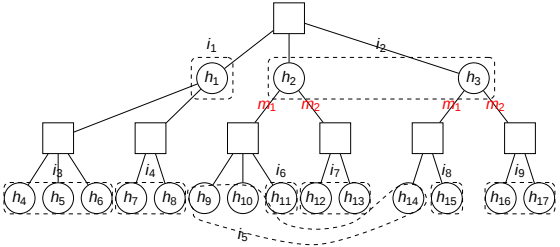
Wyniki



Lisy, Kovarik, Lanctot, Bosansky. Convergence of Monte Carlo Tree Search in Simultaneous Move Games. (2013)

Gry z niepełną informacją





Perfect Informaion MC (Determinized UCT)

W obecnym punkcie decyzyjnym

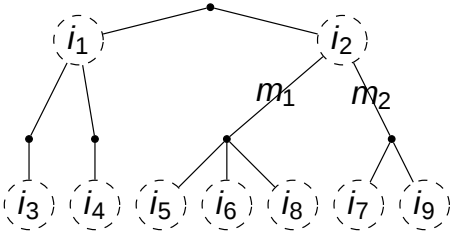
- wylosuj dopuszczalny pełny stan
  - wykonaj trening UCT z pełną informacją
- Powtórz losowanie stanu startowego wielokrotnie
- Uśrednij wszystkie drzewa i zwróć najlepszy ruch

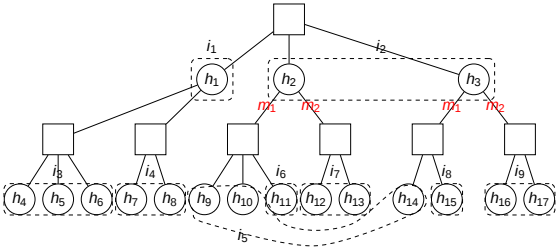
Wady

- decyzje w poddrzewach na podstawie pełnej informacji (założenie, że możemy podjąć różne decyzje w stanach, których nie rozróżniamy).
- przeciwnik może w rzeczywistej grze kierować grę do poddrzewa odpowiadającego jego preferencją, bez ujawniania nam tej informacji.

SO-ISMCTS

- Trening UCT na drzewie widzianym przez jednego gracza
  - jak trening gry dla dwóch graczy – przeciwnik wybiera ujawniana informację (nieujawniana jest losowana).





MO-ISMCTS

- Trening na zbiorze drzew, po jednym dla każdego gracza.
- W danym kroku selekcji używamy drzewa zadanego gracza.

Porównanie

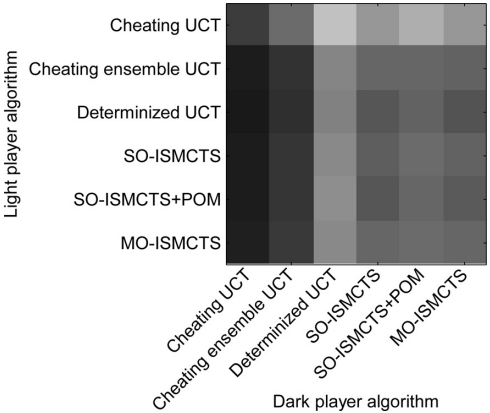


Fig. 7. Heat map showing the results of the *LOTR:C* playing strength experiment. A white square would indicate a 100% win rate for the specif ed Light player algorithm against the specif ed Dark player algorithm, while a black square would indicate a 100% win rate for Dark against Light. Shades of gray interpolate between these two extremes.

Cowling, Powley, Whitehouse. Information Set Monte Carlo Tree Search. (2012)

GIB: licytacje brydżowe z PIMC

Ginsberg. GIB: Imperfect information in a computationally challenging game. (2001)

Analiza PIMC

Long et al. Understanding the Success of Perfect Information Monte Carlo Sampling in Game Tree Search. (2010)

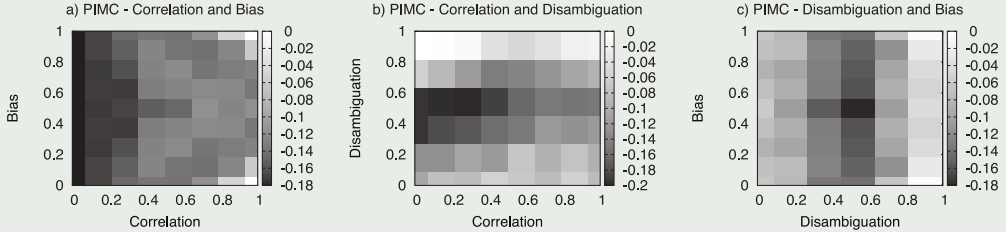


Figure 3: Performance of PIMC search against a Nash equilibrium. Darker regions indicate a greater average loss for PIMC. Disambiguation is f xed at 0.3, bias at 0.75 and correlation at 0.5 in f gures a, b and c respectively.

Inne kierunki

- MCTS + Funkcja oceny pozycji w fazie symulacji
- Agregacja stanów/ruchów
- Inne modelowanie przeciwnika
- Nie wymuszanie zagrania wszystkich ruchów przed selekcją w węźle

Podsumowanie

- przeszukiwanie bardziej obiecujących obszarów gry
- niezależność od heurystyk dziedzinowych
- modyfikacje do różnych gier:
  - jeden gracz
  - wielu graczy
  - niepełna informacja
  - ruchy równoczesne

Browne et al. A Survey of Monte Carlo Tree Search Methods. (2012)