

# The game of **NoGo**

## The state-of-art algorithms

Arkadiusz Nowakowski  
arkadiusz.nowakowski@us.edu.pl

Faculty of Computer Science and Materials Science  
University of Silesia, Poland

Sosnowiec 28.11.2016

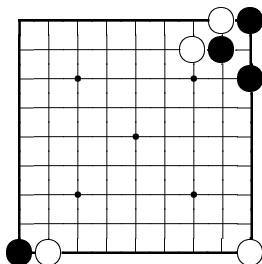


# Plan prezentacji

- ▶ Omówienie gry NoGo i jej zasad.
- ▶ Ocena aktualnego rozwoju algorytmów w kontekście NoGo.
- ▶ Przegląd niektórych algorytmów i modyfikacji.
- ▶ Podsumowanie.

## Ogólny opis gry

- ▶ Gra **NoGo** rozgrywana jest na planszy (zwanej **gobanem**) składającej się z przecinających się ze sobą linii; gracze naprzemiennie kładą białe i czarne pionki (tzw. **kamienie**) na przecięciach linii.
- ▶ Mimo podobieństw **strategia** i **taktyka** w NoGo znacznie różnią się od tej znanej z gry Go.
- ▶ Gra została opracowana na potrzeby warsztatów *Combinatorial Game Theory* w 2011r. Aczkolwiek pierwszy szkic zasad został opublikowany niezależnie w 1 kwietnia 2005r.



Przykładowa pozycja w NoGo.

# Zasady gry

- ▶ Dwaj gracze (czarny i biały) wykonują ruchy kładąc naprzemiennie po jednym kamieniu na planszy. Grę rozpoczynają **czarne**.
- ▶ Kamienie muszą być stawiane na przecięciu pionowych i poziomych linii.
- ▶ Gdy kamień jest położony, nie można go przemieścić czy usunąć z planszy.
- ▶ Pierwszy gracz, który popełni **samobójstwo, zbije grupę przeciwnika lub nie będzie miał możliwego ruchu przegrywa**.
- ▶ Sąsiedztwo kamieni wyznaczają linie na planszy; sąsiadujące ze sobą kamienie nazywamy **grupą**.
- ▶ Wolne pola sąsiadujące z grupą nazywane są **oddechami**; jeżeli grupa lub pojedynczy kamień nie mają oddechów uznaje je się za zbite (lub samobójstwem w przypadku gdy gracz pozbawia swoją grupę ostatniego oddechu).

# Pozycje w grze



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



(i)



(j)



(k)



(l)



(m)

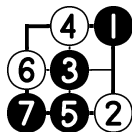
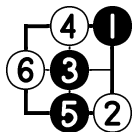
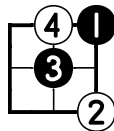
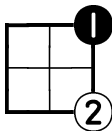
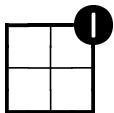


(n)



???

# Przykładowa gra



# Motywacja

- ▶ Rozwiązanie pozycji (zapisanej w formie grafu) w grze NoGo jest PSPACE-kompletne; wcześniej pokazano, że jest NP-trudne.
- ▶ NoGo było wybrane jako jedna z gier w Olimpiadzie Gier Komputerowych w roku 2011 oraz 2013. Co roku turniej NoGo jest rozgrywany w ramach TCGA Computer Game Tournament (kwiecień) oraz TAAI tournament (grudzień).
- ▶ Pomimo krótkiej historii NoGo jest chętnie wybierane do eksperymentów badających sztuczną inteligencję dla gier co potwierdzają publikowane każdego roku artykuły.

## Bibliografia

- ▶ Burke, Kyle, and Bob Hearn. "PSPACE-Complete Two-Color Placement Games." arXiv preprint arXiv:1602.06012 (2016).
- ▶ <http://combinatorialgametheory.blogspot.com/2011/02/graph-nogo-is-np-hard.html>

## Aktualny stan

- ▶ Programy grające w NoGo bazują głównie na **Monte-Carlo Tree Search** oraz jego modyfikacjach.
- ▶ Dwa najsilniejsze program **HappyNoGo** oraz **BobNoGo** bazują na połączeniu MCTS + RAVE + Opening Books + heurystyki w symulacjach. Programy różnicuje sposób wykonywania obliczeń równoległych (BobNoGo bazuje na **Fuego** a HappyNoGo na metodzie opracowanej przez Ting-Fu Liao).
- ▶ Po za wyżej wymienionymi przeprowadzono badania nad zastosowanie innych modyfikacji takich jak: *rozpoznawanie wzorców*, *optymalizację pamięciowo-czasową*, *zastosowanie logiki rozmytej* oraz *tradycyjne funkcje oceny*.

## Bibliografia

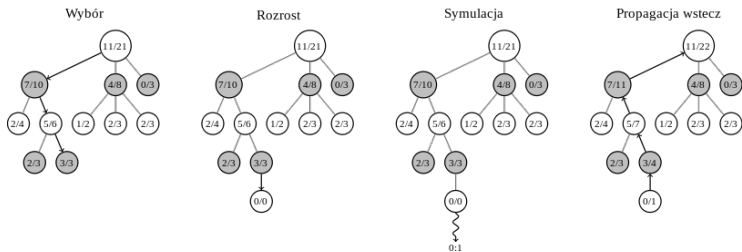
- ▶ Enzenberger, Markus, et al. "Fuego - an open-source framework for board games and Go engine based on Monte Carlo tree search." IEEE Transactions on Computational Intelligence and AI in Games 2.4 (2010): 259-270.
- ▶ Liao, Ting-Fu, et al. "A Study of Software Framework for Parallel Monte Carlo Tree Search".



# NoGo

- ▶ Wykorzystanie Kombinatorycznej Teorii Gier do rozwiązywania NoGo można znaleźć w pracy doktorskiej *Solving Games and Improving Search Performance with Embedded Combinatorial Game Knowledge* w roku 2013.
- ▶ Udowodnienie PSPACE-kompletności dla NoGo.
- ▶ Znamy rozwiązanie NoGo dla planszy 4x4 (biały gracz wygrywa).

# Monte-Carlo Tree Search (MCTS)



W grach takich jak NoGo czy Go, dla których brak jest funkcji oceny, MCTS jest standardową (uniwersalną) metodą budowania sztucznej inteligencji. Kluczowymi elementami metody są **zasady wyboru kolejnych węzłów** oraz **zasady symulacji**.

## Bibliografia

- ▶ Browne, Cameron B., et al. "A survey of monte carlo tree search methods." IEEE Transactions on Computational Intelligence and AI in Games 4.1 (2012): 1-43.

# Eksploatacja a eksploracja

Najczęściej wybieraną zasadą wyboru jest funkcja **UCB1** [1]. Razem z metodą MCTS stanowi ona najpopularniejszy algorytm *UCT*.

$$\operatorname{argmax}_{1 \leq i \leq K} \bar{x}_i + C \sqrt{\frac{2 \ln(n)}{n_i}} \quad \left( + \frac{b_i}{n_i} \right)$$

Kolejną obiecującą metodą wyboru jest tzw. sekwencyjne półwkowanie (*sequential halving*). Polega ona na odcinaniu połowy ruchów po wykonaniu pewnej liczby symulacji [2], liczbę symulacji określa wzór:

$$\left\lfloor \frac{\text{budget}}{|S| \lceil \log_2 |\text{possibleMoves}| \rceil} \right\rfloor$$

L.symulacji	SHOT vs. UCT
1000	75,6%
10000	75,8%
100000	66,8%

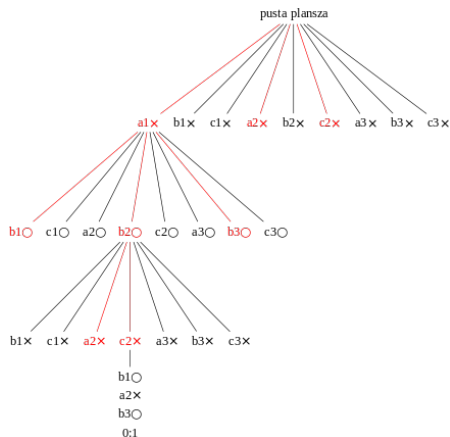
# All moves as first (AMAF) [3]

## $\alpha$ -AMAF

$$(1 - \alpha)\bar{x}_i + \alpha\bar{x}_i' + C\sqrt{\frac{2\ln(n)}{n_i}}$$

## RAVE

$$\alpha = \max\left\{0, \frac{V - n_i}{V}\right\}$$



L.symulacji	RAVE vs. UCT
50	56% $\pm$ 5%
500	64% $\pm$ 4%
1000	64% $\pm$ 2%
2000	70% $\pm$ 4%
4000	78% $\pm$ 3%
128000	82% $\pm$ 3%

# Modyfikacje symulacji [3]

## Decyzyjne i anty-decyzyjne ruchy

Gdy jeden z graczy ma ruch, który natychmiast kończy grę, zagraj go. W przeciwnym razie graj losowo.

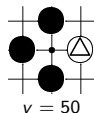
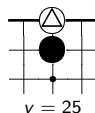
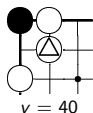
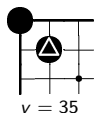
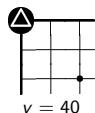
## Ogólna heurystyka dla gier kombinatorycznych

Przy wyborze kolejnego ruchu należy określić: jeżeli gracz tam zagra,  $a$  usuniętych możliwych ruchów przeciwnika,  $b$  usuniętych ruchów gracza; jeżeli zagra tam przeciwnik,  $a'$  usuniętych ruchów przeciwnika,  $b'$  usuniętych ruchów gracza. Wybierz ruch maksymalizujący funkcje:  $a - b - a' + b'$ .

L.symulacji	DM+ADM vs. UCT
100	60% ± 3%
1000	70% ± 3%
100000	86% ± 4%

L.symulacji	Funkcja vs. UCT
160	77,7% ± 4,9%
320	83,3% ± 5,3%
640	84,5% ± 5,1%

# Rozpoznawanie wzorców



L.sekund	Wzorce vs. UCT
5	69,4%
10	75,5%
20	81,7%

## Bibliografia

- ▶ Sun, Yuxia, Cheng Liu, and Hongkun Qiu. "The research on patterns and UCT algorithm in NoGo game." 2013 25th Chinese Control and Decision Conference (CCDC). IEEE, 2013.

# Podsumowanie

- ▶ Proste zasady NoGo kryją bardzo dużą głębie i złożoność, mimo, że gra jest uznawana jeszcze jako nowa, została już doceniona i jest przedmiotem badań wielu artykułów.
- ▶ Cykliczne turnieje pozwalają dwa razy do roku zweryfikować nowe pomysły w konfrontacji z najlepszymi programami.
- ▶ Badania nad NoGo są w większości przypadków pionierskie.
- ▶ W opinii autora warto skorzystać z *fali wznoszącej* stąd wybór NoGo jako gry na UMPGGK 2017.

# Bibliografia



Auer, Peter, Nicolo Cesa-Bianchi, and Paul Fischer  
*Finite-time analysis of the multiarmed bandit problem*  
Machine learning 47.2-3 (2002): 235-256



Cazenave, Tristan  
*Sequential halving applied to trees*  
IEEE Transactions on Computational Intelligence and AI in Games  
7.1 (2015): 102-105



Chou, C-W., Olivier Teytaud, and S-J. Yen  
*Revisiting Monte-Carlo tree search on a normal form game: NoGo*  
European Conference on the Applications of Evolutionary  
Computation. Springer Berlin Heidelberg, 2011