

Strategie gier dwuosobowych

Michał Słowikowski i Maciej Słaboń

Dlaczego są potrzebne?

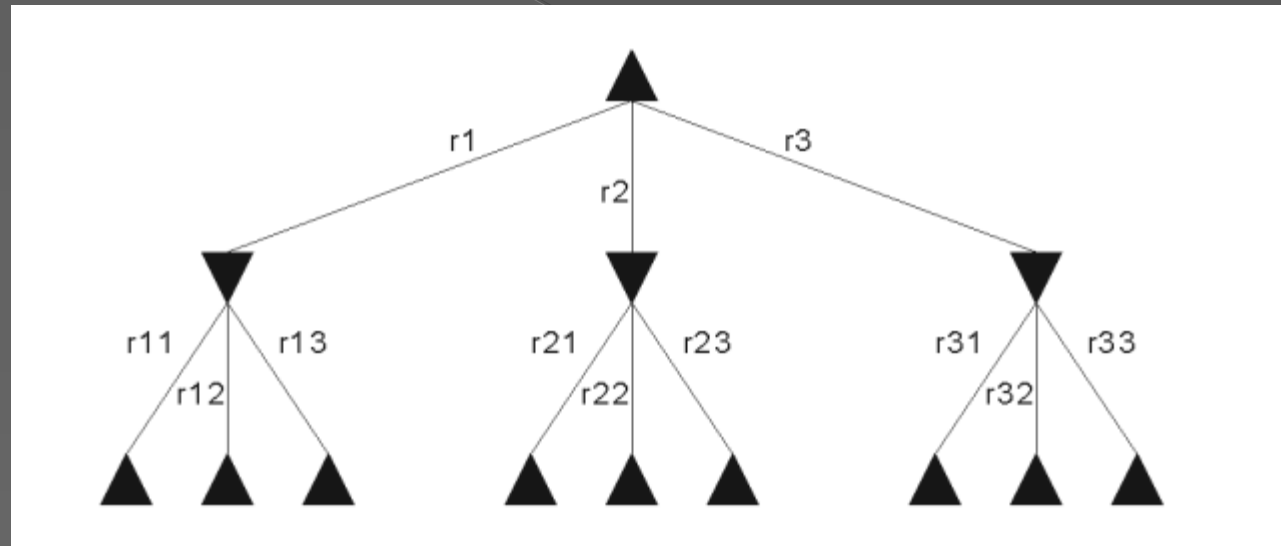
- Gry planszowe od zawsze były obszarem "testowania" inteligentnych programów
- Przy ruchach naprzemiennych należy sekwencyjnie podejmować decyzje o wyborze ruchu (jak największe szanse wygranej w każdym ruchu)
- Przy bardziej złożonych grach, maszyny mają za słabą moc obliczeniową do symulacji każdego scenariusza.
- Stąd zapotrzebowanie na strategie przeszukiwania w grach dwuosobowych

Podstawowy model gry

- W grze uczestniczy dwóch graczy.
- Gracze wykonują ruchy naprzemiennie.
- W każdej sytuacji na planszy jest skończona liczba możliwych do wykonania ruchów.
- Sytuacja na planszy i wykonany ruch jednoznacznie wyznaczają następną sytuację na planszy.
- Każda możliwa sytuacja na planszy może być jednoznacznie zaklasyfikowana do jednej z następujących kategorii:
 - > wygrana pierwszego gracza,
 - > wygrana drugiego gracza,
 - > remis,
 - > sytuacja nierozstrzygnięta.

Algorytm MinMax

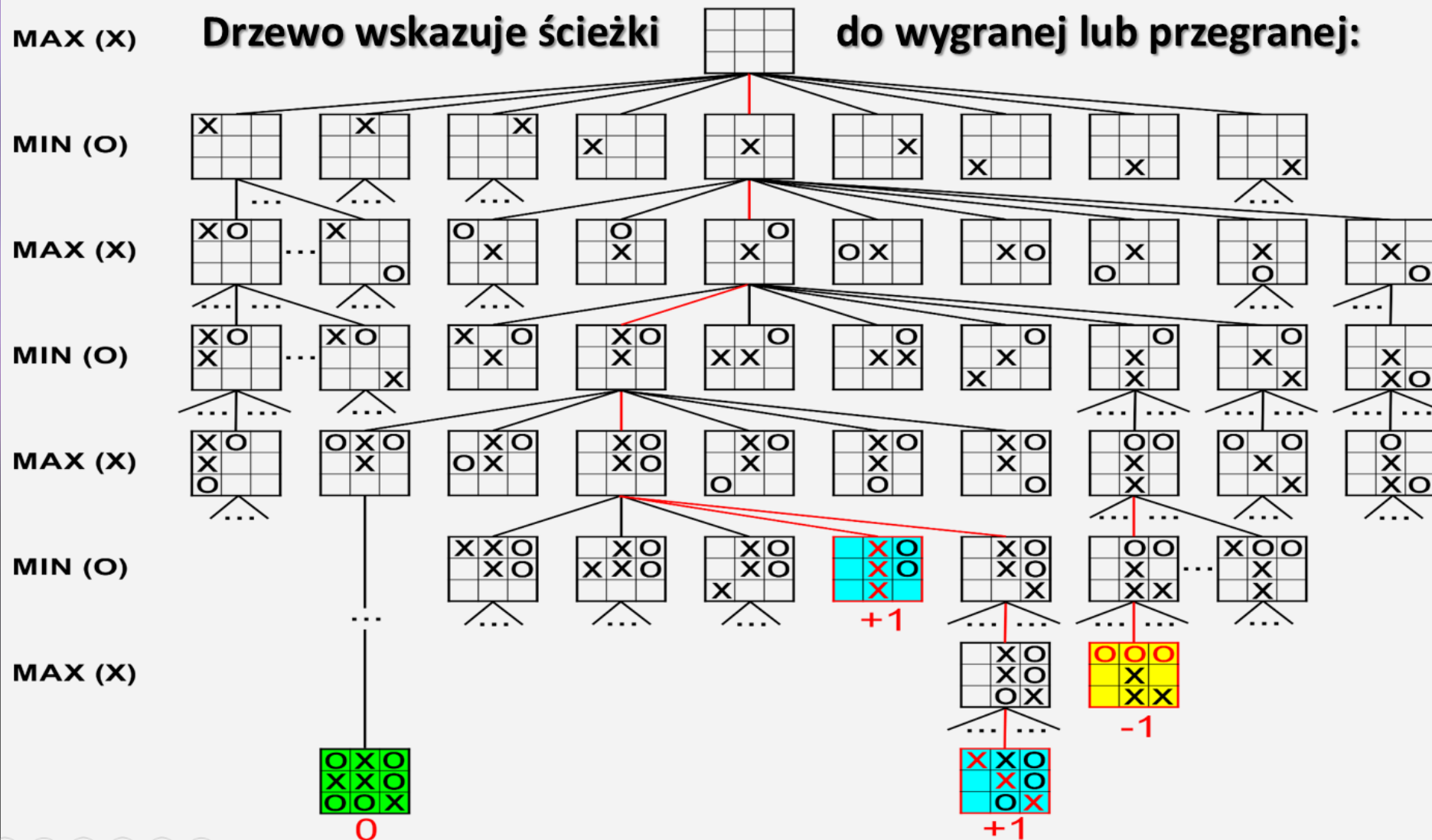
Pojedyncza partia gry może być w pełni opisana przez ciąg naprzemiennych ruchów obu graczy. Aby w dowolnym momencie w trakcie gry wybrać najbardziej odpowiedni ruch można rozważyć wszystkie możliwe scenariusze rozpoczynające się różnymi możliwymi do wybrania ruchami gracza, po każdym z których może nastąpić każdy możliwy ruch przeciwnika. Naturalną reprezentacją przestrzeni sytuacji, możliwych do osiągnięcia po kolejnych ruchach graczy jest **drzewo gry**.



Algorytm MinMax:

- Wywodzi się z teorii gier dwuosobowych, w której gracze wykonują ruch na zmianę.
- Polega na **minimalizacji maksymalnych korzyści przeciwnika** oraz **maksymalizacji minimalnych korzyści dla siebie**
- Algorytm zakłada że każdy z graczy będzie próbował zmaksymalizować szanse na wygraną oraz zminimalizować szanse na przegraną
- Nie zawsze to jest możliwe ze względu na potencjalnie dużą ilość możliwych posunięć i ich kombinacji w kolejnych ruchach, np. w grze w szachy czy go.
- Algorytm pomaga znaleźć najlepszy ruch w każdym momencie gry

Drzewo gry kółko-krzyżyk

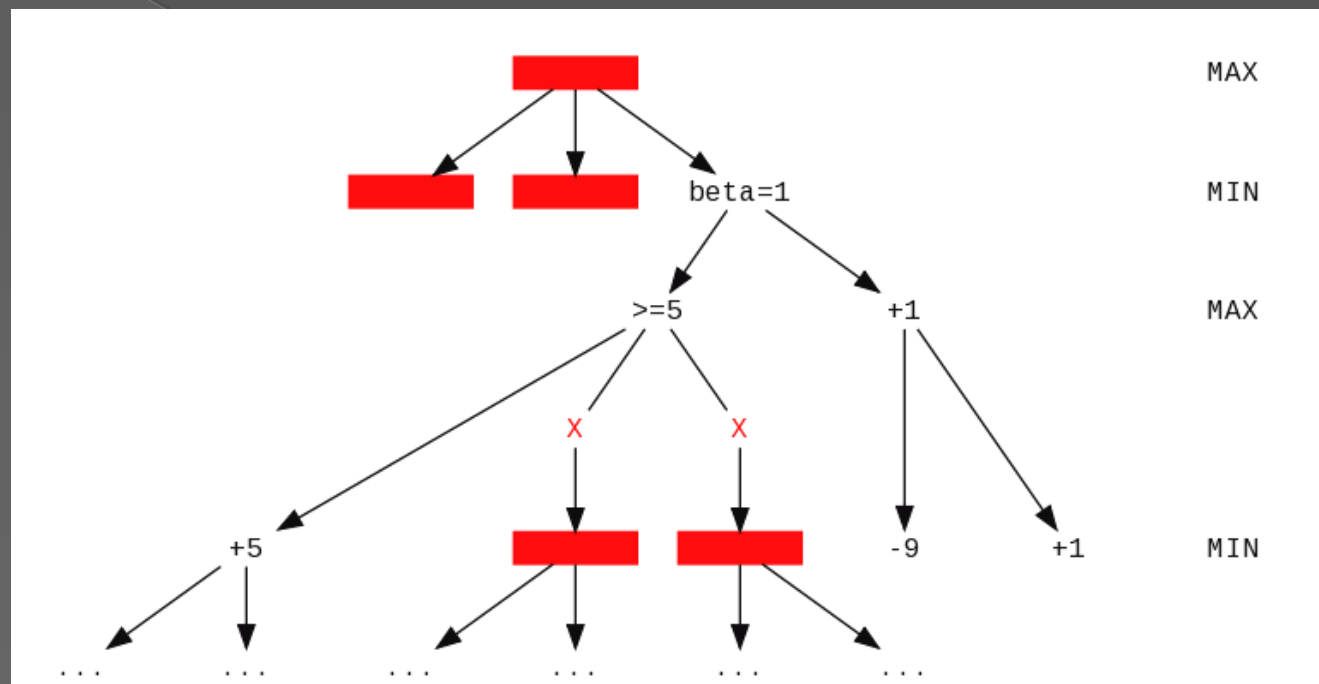


W drzewie może istnieć wiele ścieżek wygrywających, przegrywających lub remisujących. Przy wyborze konkretnej strategii należy wziąć pod uwagę takie ścieżki które:

- Zawierają możliwie dużą ilość ścieżek wygrywających
- Zawierają możliwie małą ilość ścieżek przegrywających
- Zawierają ścieżki na pewno wygrywające
- Nie zawierają ścieżki prowadzącej do pewnej przegranej

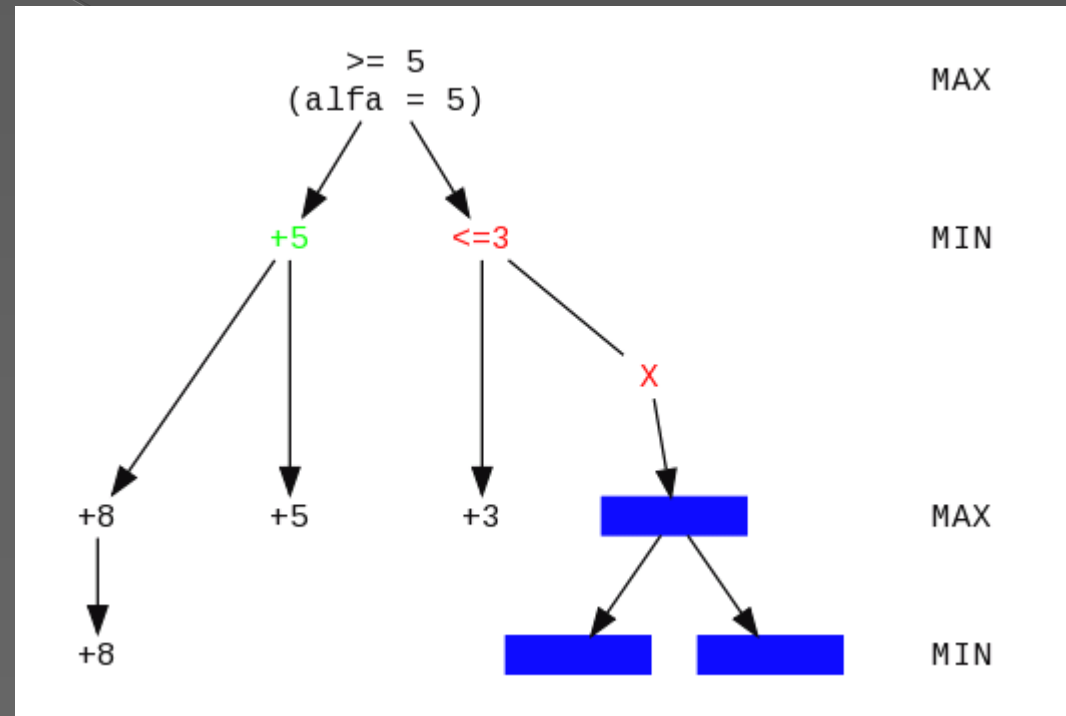
Algorytm alfa beta

- Alfa beta to tak naprawdę poprawiony algorytm min- max (wzbogacony o cięcia alfa- beta).
- Pozwala na „ucięcie” fragmentów drzewa gry.

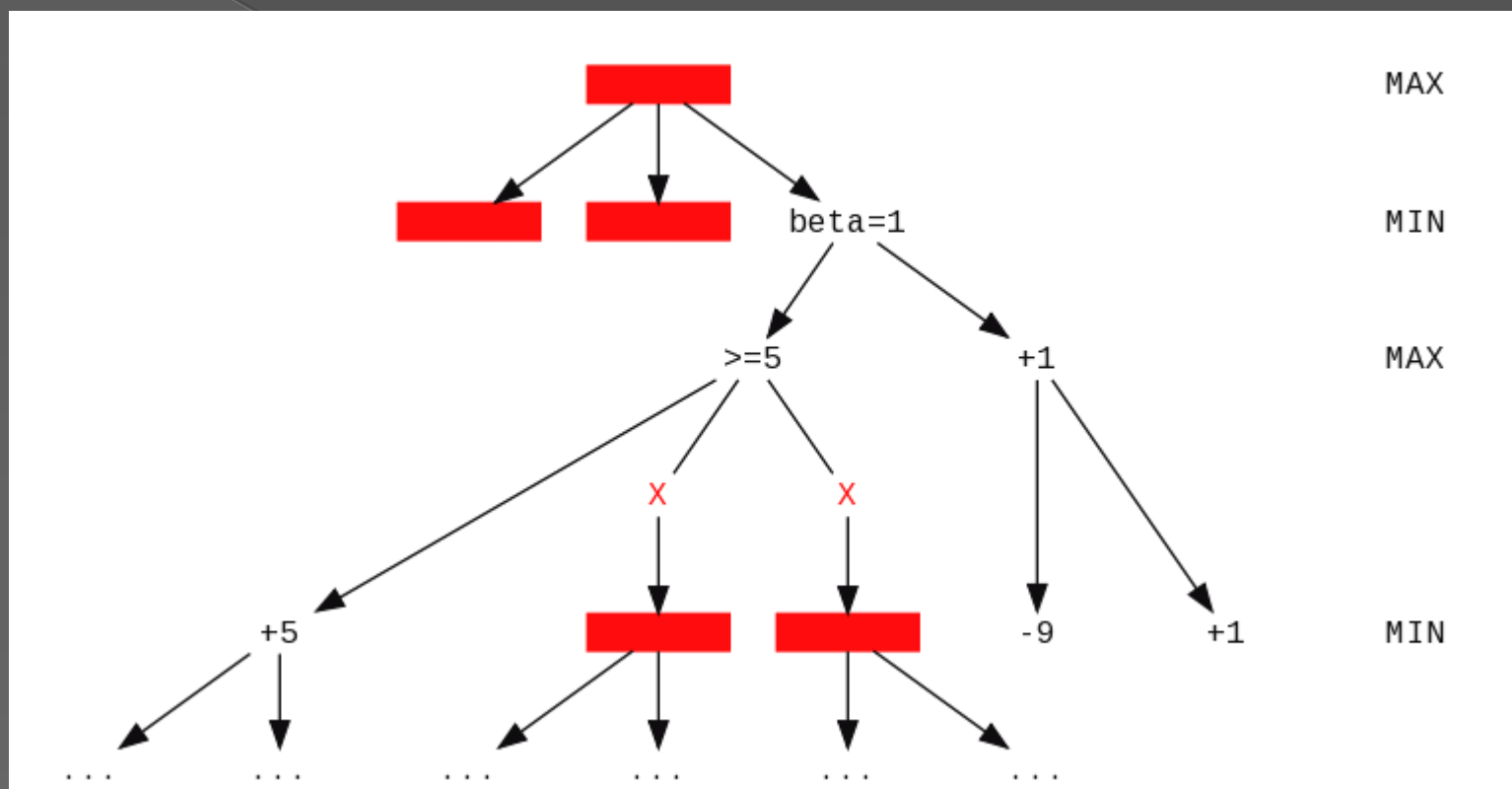


Jak wybieramy co uciąć?

- Cięcie alfa: Przeszukiwanie można zakończyć poniżej dowolnego wierzchołka typu MIN o wartości mniejszej lub równej wartości alfa dowolnego z jego poprzedników (typu MAX).



- Cięcie beta: Przeszukiwanie można zakończyć poniżej dowolnego wierzchołka typu MAX o wartości nie mniejszej od wartości beta dowolnego z jego poprzedników (typu MIN).



Pseudokod algorytmu alfa-beta:

- funkcja minimax(węzeł, głębokość)
- zwróć alfabet(węzeł, głębokość, $-\infty$, $+\infty$)
- funkcja alfabet(węzeł, głębokość, α , β)
- jeżeli węzeł jest końcowy lub głębokość = 0
- zwróć wartość heurystyczną węzła
- jeżeli przeciwnik ma zagrać w węźle
- dla każdego potomka węzła
- $\beta := \min(\beta, \text{alfabet}(\text{potomek}, \text{głębokość}-1, \alpha, \beta))$
- jeżeli $\alpha \geq \beta$
- przerwij przeszukiwanie {odcinamy gałąź Alfa}
- zwróć β
- w przeciwnym przypadku {my mamy zagrać w węźle}
- dla każdego potomka węzła
- $\alpha := \max(\alpha, \text{alfabet}(\text{potomek}, \text{głębokość}-1, \alpha, \beta))$
- jeżeli $\alpha \geq \beta$
- przerwij przeszukiwanie {odcinamy gałąź Beta}
- zwróć α

Źródła

<https://youtu.be/STjW3eH0Cik>

http://wazniak.mimuw.edu.pl/index.php?title=Sztuczna_inteligencja/SI_Modu%C5%82_8_-_Gry_dwuosobowe

<http://www.wszystko.aplus.pl/reszta/boo/magazynier.pdf>