

Heurystyczne metody przeszukiwania przestrzeni stanów

1/1

Plan wykładu

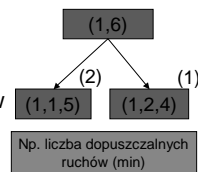
- Reprezentacja gry w postaci mini-max i nega-max
- Zasady implementacji algorytmu minimax
- Algorytm alfa-beta
- Algorytm B*
- Podsumowanie

Dr hab. inż. Joanna Józefowska, prof. PP

2

Implementacja przeszukiwania

- Reprezentacja aktualnego stanu w grze
 - NIM
 - Kółko i krzyżyk
 - Szachy
- Generowanie kolejnych stanów
 - ruchy odwrotne (undo)
- Funkcja oceny heurystycznej
 - szybkość (łożoność obliczeniowa)
 - dokładność



Dr hab. inż. Joanna Józefowska, prof. PP

3

Implementacja przeszukiwania przestrzeni stanów

pos	pozycja w grze
depth	liczba poziomów w grafie do przeszukiwania
Evaluate	funkcja obliczająca wartość gry z punktu widzenia gracza, który wykonuje ruch
best	najlepsza wartość znaleziona na kolejnym poziomie
Successors	funkcja generująca zbiór pozycji osiągalnych w jednym ruchu z aktualnej pozycji
succ	zbiór pozycji osiągalnych w jednym ruchu z aktualnej pozycji

Dr hab. inż. Joanna Józefowska, prof. PP

4

NegaMax

```

int NegaMax (pos, depth)
{
    if (depth == 0) return Evaluate(pos);
    best = -INFINITY;
    succ = Successors(pos);
    while (not Empty(succ))
    {
        pos = RemoveOne(succ);
        value = -NegaMax(pos, depth-1);
        if (value > best) best = value;
    }
    return best;
}
    
```

Dr hab. inż. Joanna Józefowska, prof. PP

5

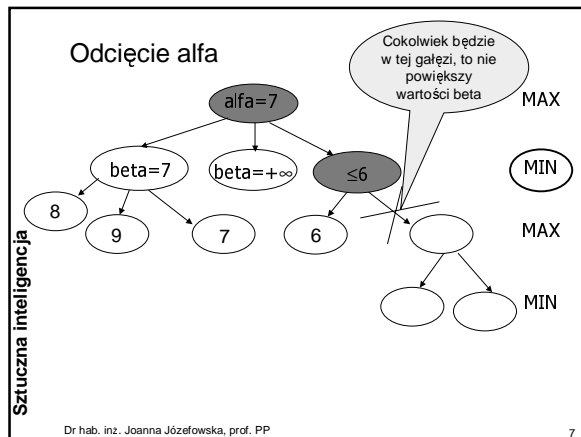
Procedura alfa-beta

Założenia:

1. Reguły gry nie pozwalają na wygenerowanie nieskończonej ścieżki.
2. Z każdego stanu można osiągnąć tylko skończoną liczbę stanów.
3. (Lemat nieskończoności): dla każdego stanu p istnieje liczba $N(p)$ taka, że żadna rozgrywka startująca ze stanu p nie będzie trwała dłużej niż $N(p)$ ruchów.

Dr hab. inż. Joanna Józefowska, prof. PP

6

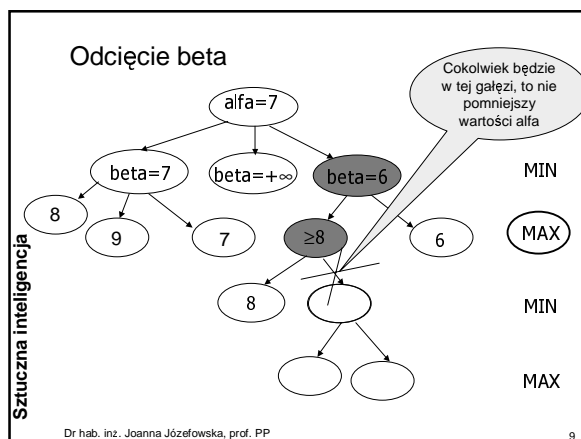


Odcięcie alfa

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

Dr hab. inż. Joanna Józefowska, prof. PP

8



Odcięcie beta

Przeszukiwanie można zakończyć poniżej dowolnego wierzchołka typu MAX o wartości większej lub równej wartości beta dowolnego z jego poprzedników (typu MIN).

Dr hab. inż. Joanna Józefowska, prof. PP

10

Sztuczna inteligencja

```

int AlphaBeta(pos, depth, alpha, beta)
{
    if (depth == 0) return Evaluate(pos);
    best = -INFINITY;
    succ = Successors(pos);
    while (not Empty(succ) && best < beta)
    {
        pos = RemoveOne(succ);
        if (best > alpha) alpha = best;
        value = -AlphaBeta(pos, depth-1, -beta, -alpha);
        if (value > best) best = value;
    }
    return best;
}

```

Dr hab. inż. Joanna Józefowska, prof. PP

11

Algorytm alfa-beta

- Jest algorytmem dokładnym
- W najgorszym razie przeszukuje całą przestrzeń (tak jak MINIMAX, $O(N^D)$)
- W najlepszym razie przeszukuje $N^{((D+1)/2)} + N^{(D/2)} - 1$ stanów
- W najlepszym razie pozwala dwukrotnie zwiększyć głębokość przeszukiwania

N – branching factor; D - głębokość

Dr hab. inż. Joanna Józefowska, prof. PP

12

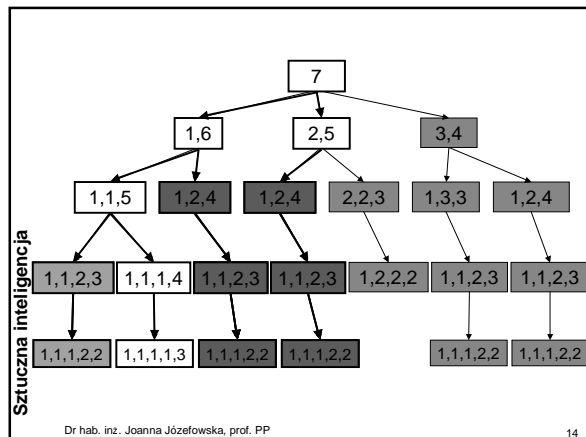
Modyfikacje algorytmu alfa-beta

• Kolejność ruchów (Move ordering)

- Rezultaty wcześniejszych przeszukiwań
- Dynamiczne porządkowanie ruchów
- Statyczne porządkowanie ruchów

Dr hab. inż. Joanna Józefowska, prof. PP

13



Dr hab. inż. Joanna Józefowska, prof. PP

14

Move ordering

• Transposition table (Hashtable)

Polega na haszowaniu identycznych stanów w różnych gałęziach. Może skrócić przeszukiwanie nawet czterokrotnie (co oznacza jeden poziom w grafie więcej w tym samym czasie).

Dr hab. inż. Joanna Józefowska, prof. PP

15

Move ordering

• Killer moves

Polega na zapamiętywaniu ruchów, które we wcześniejszych fazach przeszukiwania doprowadziły do odcięcia. Takie ruchy są wykonywane jako pierwsze podczas eksploracji stanu.

Po znalezieniu innego ruchu odcinającego, poprzedni jest zastępowany nowym.

Dr hab. inż. Joanna Józefowska, prof. PP

16

Move ordering

• History heuristics

Jest rozszerzeniem poprzedniej heurystyki, polegającym na zapamiętywaniu wszystkich „dobrych” ruchów. Każde skuteczne zastosowanie ruchu zwiększa jego priorytet. W ten sposób tworzy się pewne uporządkowanie ruchów.

Dr hab. inż. Joanna Józefowska, prof. PP

17

Move ordering

• Null move heuristics

Polega na zaprzestaniu przeszukiwania w tych obszarach, gdzie znaleziono zadowalająco dobrą pozycję w grze. Wykrycie takiej pozycji polega na wykonaniu pustego ruchu i powtórzeniu przeszukiwania poziom niżej. Jeżeli wynik jest większy niż beta, to przeszukiwanie kończy się, w przeciwnym razie jest kontynuowane normalnie.

Dr hab. inż. Joanna Józefowska, prof. PP

18

Move ordering

- Static ordering

Jest heurystyką zależną od gry. Mimo, że preferowane są strategie niezależne od gry, to jednak w praktyce często korzysta się z informacji charakterystycznej dla rodzaju gry, aby zwiększyć efektywność heurystyki.

Np. w warcabach można preferować ruchy, które „zbijają” więcej pionów przeciwnika itd.

Dr hab. inż. Joanna Józefowska, prof. PP

19

Modyfikacje algorytmu alfa-beta

- Aspiration search (windowing)

W korzeniu przeszukiwanego drzewa przyjmuje się ograniczony przedział [alfa, beta] nazywany aspiration window.

AlphaBeta (pos, depth, $-\infty$, $+\infty$)



AlphaBeta (pos, depth, -window, +window)

Początkowe wywołanie procedury przyjmuje postać

Jeżeli rozwiązanie leży poza oknem, to trzeba powtórzyć przeszukiwanie.

Dr hab. inż. Joanna Józefowska, prof. PP

20

Modyfikacje algorytmu alfa-beta

- Principal variation search (PVS)

Polega na przeszukiwaniu gałęzi poza principal variation z oknem $\beta - \alpha + 1$. Zakłada się, że jest mało prawdopodobne, aby ruchy poza principal variation były dobre (przy założeniu *perfect ordering*)

Dr hab. inż. Joanna Józefowska, prof. PP

21

Modyfikacje algorytmu alfa-beta

- MTD(f) (Memory enhanced test)

Polega na kilkakrotnym wywołaniu procedury alfa-beta ze zmieniającymi się oknami. Za każdym razem uzyskujemy dolne lub górne ograniczenie, które są zbieżne do prawdziwej wartości mini-max.

Dr hab. inż. Joanna Józefowska, prof. PP

22

```
int mtdf(struct position p, int firstguess, int depth)
{
    int g, lowerbound, upperbound;
    g = firstguess;
    upperbound = INFINITY;
    lowerbound = -INFINITY;
    while(lowerbound < upperbound)
    {
        if(g != lowerbound) beta = g + 1;
        else beta = g;
        g = alphabeta(p, depth, beta - 1, beta);
        if(g < beta) upperbound = g;
        else lowerbound = g;
    }
    return g;
}
```

wartość początkowa

Aktualizacja okna

Dr hab. inż. Joanna Józefowska, prof. PP

23

Modyfikacje algorytmu alfa-beta

- Enhanced transposition cutoffs (ETC)

W algorytmie alfa-beta z tablicą haszującą przegląda się wszystkie następni bieżącego stanu przed rekurencyjnym wywołaniem procedury alfa-beta.

W praktyce jest to koncepcja kosztowna i nakład zwraca się tylko w węzłach odległych od liści.

Dr hab. inż. Joanna Józefowska, prof. PP

24

Modyfikacje algorytmu alfa-beta

• Quiescence search

Jest to procedura wywoływana często zamiast funkcji Evaluate na głębokości $\text{depth}=0$. Ma to zapobiec tzw. efektowi horyzontu, który polega na ukryciu rychłej przegranej przez fakt, że nie przeszukujemy dalej przestrzeni. Quiescence search ma zapewnić, że Evaluate będzie wywoływana tylko w stabilnych stanach (w których nie ma bezpośredniego zagrożenia przegraną). Przeszukuje się tylko takie stany, które zwykle wiążą się z zagrożeniem (np. utrata piona w szachach).

Dr hab. inż. Joanna Józefowska, prof. PP

25

Modyfikacje algorytmu alfa-beta

• Iteracyjne pogłębianie

Polega na wielokrotnym wywoływaniu przeszukiwania na określoną głębokość, przy czym w kolejnych powtórzeniach głębokość przeszukiwania rośnie.

W kolejnych powtórzeniach można wykorzystać zdobytą wcześniej informację (zapisaną np. w tablicy transpozycji, czy w postaci okna aspiracji).

Dr hab. inż. Joanna Józefowska, prof. PP

26

Algorytm B*

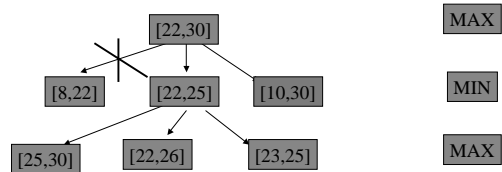
Strategia **PROVEBEST** polega na tym, że próbuje się zwiększać dolne ograniczenie najbardziej lewego wierzchołka tak, aby była ona nie gorsza od górnego ograniczenia dowolnego wierzchołka spośród rodzeństwa.

Strategia **DISPROVEREST** polega na tym, że próbuje się zmniejszać górne ograniczenie wszystkich wierzchołków na głębokości 1 tak, aby żadna z nich nie była lepsza od dolnego ograniczenia najbardziej lewego wierzchołka.

Dr hab. inż. Joanna Józefowska, prof. PP

27

Przykład algorytmu B*



Dr hab. inż. Joanna Józefowska, prof. PP

28

Deep Blue - podejście siłowe

Analizuje $2 \cdot 10^8$ stanów w ciągu sekundy.

Przestrzeń stanów dla szachów wynosi 10^{120} .

W czasie dozwolonym analizuje 6 ruchów do przodu.

Kasparow analizuje 3 ruchy do przodu.

Deep Junior

Analizuje mniej stanów, ale ma lepszą heurystykę.

Kasparow również nauczył się specyfiki gry z komputerem.

Dr hab. inż. Joanna Józefowska, prof. PP

29