

Wydział Elektroniki i Technik Informacyjnych  
Politechnika Warszawska

Wstęp do Sztucznej Inteligencji  
Semestr 24L  
Sprawozdanie z ćwiczenia nr 3

Algorytm MinMax dla gry "Kółko i krzyżyk"

Mikołaj Wewiór

Warszawa,  
16 IV 2024

## 1. Opis problemu

Celem ćwiczenia było zaimplementowanie algorytmu MinMax dla gry "Kółko i krzyżyk". Jest to jedna z metod przeszukiwania przestrzeni stanów gry, która pozwala znaleźć optymalne ruchy dla bieżącego stanu. Reprezentacją algorytmu jest drzewo stanów gry. W przeciwieństwie do algorytmu pełnego przeglądu, MinMax nie musi sprawdzać całego drzewa - jest to oszczędność czasu i zasobów. Dodatkowo algorytm może niezależnie i wielokrotnie analizować te same stany gry w różnych momentach. Modyfikacją algorytmu jest implementacja heurystyki *alpha pruning* (czasem nazywana  $\alpha - \beta$ ). Ta opcja pozwala przyspieszyć działanie i zmniejszyć liczbę przeszukiwanych stanów.

## 2. Implementacja logiki gry

Pierwszą częścią zadania była implementacja gry w "Kółko i krzyżyk". Zostało to wykonane jako gra terminalowa. Plansza została utworzona jako klasa `Board`, której jednym z głównych atrybutów była tablica ją reprezentująca. Pole, które miało być uzupełnione odpowiednim graczem (znakiem) wybierane było jako liczba od 1 do 9. Jeżeli wybrane pole było zajęte gra odpowiednio o tym informowała, a w przypadku algorytmu MinMax nie dozwolone było wybieranie zajętych pól. Po każdym wykonanym ruchu klasa sprawdza czy któraś z kombinacji nie jest wygrywająca.



(a) przed rozpoczęciem

(b) środek rozgrywki

(c) zakończenie rozgrywki

## 3. Implementacja MinMaxa i Alpha-pruningu

Oba algorytmy zostały zaimplementowane w jednej klasie `MinMaxSolver` jako jej metody. Są to funkcje rekurencyjne. Ich wyjściem jest ruch, który należy wykonać, aby zagranie było optymalne, a na wejście przyjmują obecny stan planszy, głębokość, którą mają przeszukać oraz informację czy maksymalizować wartość funkcji celu (lub minimalizować - zależnie od gracza). Atrybutem klasy jest również tablica wag, która następnie przekazuje wartości, dla funkcji nagrody, która odpowiednio punktuje wybrane pola wg heurystyki. Dodatkowe punkty są za wybranie pól, które wygrywają rozgrywkę.

## 4. Badanie algorytmów

Wszystkie z poniższych testów zostały wykonane po 15 razy dla danego stanu gry, z których wyciągnięto 10 środkowych wartości - z nich wyliczono średnią oraz odchylenie standardowe.

## 4.1. Liczba odwiedzonych węzłów

### 9 stanów początkowych

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	59704	19998	17001

Tab. 1: lewe górne pole (1)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	63905	19668	16992

Tab. 2: lewe środkowe pole (4)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	59704	19998	17001

Tab. 3: lewe dolne pole (7)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	63904	17838	15332

Tab. 4: środkowe górne pole (2)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	55504	25818	20686

Tab. 5: centralne pole (5)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	64904	23429	23307

Tab. 6: środkowe dolne pole (8)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	59704	21335	17282

Tab. 7: prawe górne pole (3)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	63904	22921	21260

Tab. 8: prawe środkowe pole (6)

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	59704	20200	18607

Tab. 9: prawe dolne pole (9)

Eksperymenty zostały przeprowadzone dla głębokości przeszukiwania 9. Spodziewanym efektem jest symetryczna względem położenia na planszy liczba odwiedzonych węzłów w przypadku zwykłego MinMaxa. Różne wartości w przypadku alpha-pruningu względem położenia startowego również są oczekiwanym efektem. Natomiast różnica między tym, czy zaczyna x, czy o nie jest oczekiwana. Według założeń nie powinny się one różnić.

### 3 stany ze środka (przykładowe)

x	2	3
4	o	6
x	8	9

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	o	
średnia	926	550	

Tab. 10: dla 3 zapełnionych pól

1	o	x
o	5	6
x	8	9

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	177	135	135

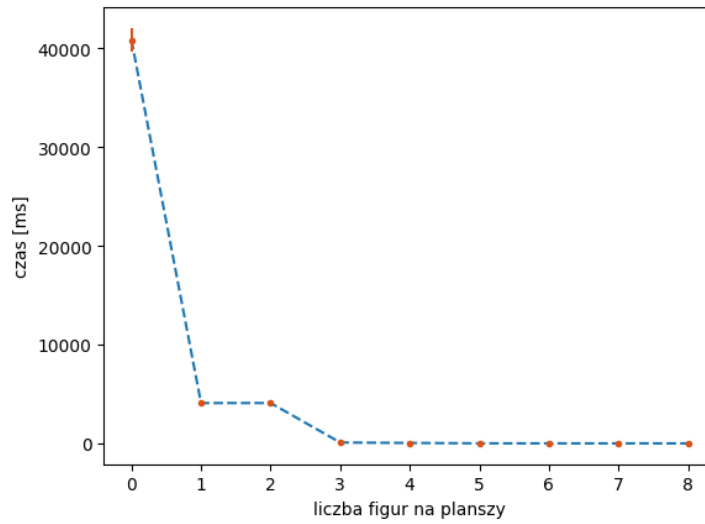
Tab. 11: dla 4 zapełnionych pól

1	o	x
o	o	6
x	x	9

tryb	MinMax	Alpha-Pruning	
rozpoczynający	-	x	o
średnia	10	10	10

Tab. 12: dla 6 zapełnionych pól

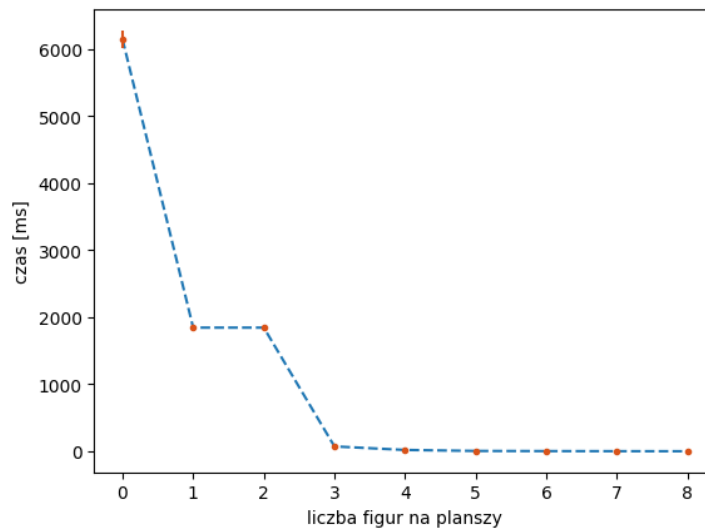
## 4.2. Czas wykonania ruchu



Rys. 2: czas wykonania dla MinMax

liczba figur	0	1	2	3	4	5	6	7	8
średnia	40852.047	4062.826	526.293	88.476	41.604	5.810	1.398	0.967	0.190
odchylenie standardowe	1212.624	91.339	24.900	9.749	6.937	2.241	0.342	0.847	0.154

Tab. 13: czas wykonania dla MinMax



Rys. 3: czas wykonania dla Alpha-pruning

liczba figur	0	1	2	3	4	5	6	7	8
średnia	6149.194	2073.538	185.880	71.897	19.400	4.525	2.278	1.041	0.196
odchylenie standardowe	131.369	44.721	13.229	5.728	4.676	1.130	1.061	0.389	0.119

Tab. 14: czas wykonania dla Alpha-pruning

---

## 5. Podsumowanie

Algorytm MinMax i Alpha-pruning w powyższych testach dawały te same wyjścia, które były najlepszymi do zagrania ruchami dla zastanej planszy. Po przeprowadzeniu badań widać, że Modyfikacja alpha-pruning daje dużo lepsze efekty - liczba przeszukanych węzłów jest mniejsza, dzięki czemu zużywamy mniej zasobów, a odpowiedź uzyskujemy szybciej. Zdziwieniem jest przypadek, kiedy dla alpha-pruningu wpływ na liczbę odwiedzonych węzłów ma to kto zaczyna grę. Istnieje możliwość, że jest to implikacja błędu implementacji algorytmu, której dotychczas niewykryto.