

# SPRAWOZDANIE

## z ćwiczenia nr 2

### “Dwuosobowe gry deterministyczne”

Przedmiot: Wprowadzenie do Sztucznej Inteligencji

Imię i nazwisko: Viktoriia Nowotka

---

Implementacja algorytmu minimax z obcinaniem  $\alpha - \beta$  wykonana na podstawie dostępnego repozytorium w instrukcji do ćwiczenia. Dla realizacji zadania dodano własne atrybuty oraz metody.

Do klasy State dodano metodę `ai_choose_move(moves)`, co jest ważną metodą dokonania wyboru najlepszego ruchu spośród możliwych.

Do klasy Player dodano atrybuty `ai` oraz `ai_depth`.

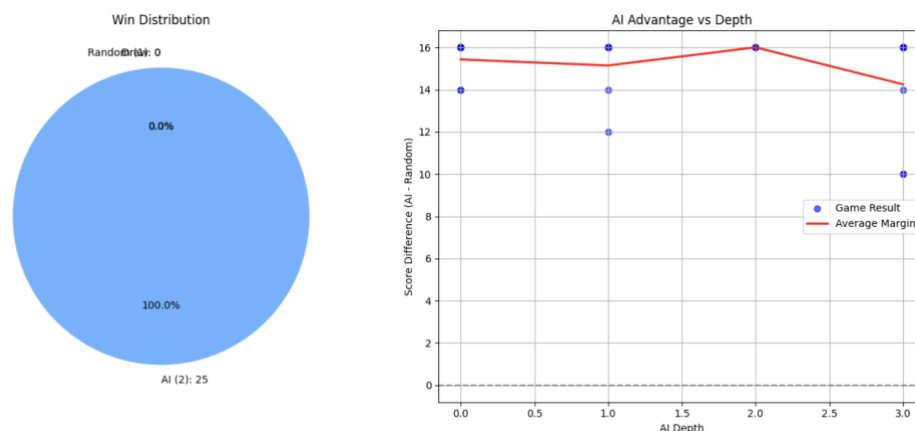
Główna klasa `DotsAndBoxes(Game)` została zmodyfikowana poprzez zmiany rozmiaru pola (dostępne są formy prostokątowe, a nie tylko kwadratowe). Wtedy jak klasa `DotsAndBoxesState(State)` zaznała największych zmian.

- `ai_choose_move(moves)`
- `_minimax(state, depth, player, a, b)`
- `_alphabeta(state, depth, a, b, player)`
- `_evaluate(state)`
- `_count_box_sides(row, col)`
- dodatkowo zmodyfikowano pomocniczy metod `_lines_row_to_str(row)` dla prawdziwej wizualizacji rzędów

W celu zebrania statystycznych danych ustalono następane hyperparametry:

- Rozmiar pola: 5x4
- Ilość iteracji: 25

Na początek przetestowano działanie AI gracza przeciwko „zwykłemu”. Zwycięstwo AI gracza wynosi 100%, co jest oczekiwanym wynikiem.



Rys. 1 – Wykres wyników 25 gier „AI vs zwykły gracz”

Zbadano wpływ głębokości algorytmu minimax z obcinaniem  $\alpha - \beta$  wykonaniem 6 eksperymentów na średniej z 25 partii. Wyniki przedstawiono w tabeli 1. Zielonym kolorem zaznaczono gracza zwycięzcy, wtedy jak grubą ścieżką zaznaczono gracza o większej głębokości AI.

Tabela 1 – Badanie wpływu głębokości algorytmu minimax na zwycięstwo.

Głębokość gracza 1	Głębokość gracza 2
3	<u>3</u>
<b>3</b>	1
<u>5</u>	2
3	<b>4</b>
4	<b>6</b>
1	<b>6</b>

Na podstawie uzyskanych danych można zauważyć dziwne zachowanie algorytmu, gdyż większa przewidywalność czyni algorytm głępszym. Tylko jeden z pięciu eksperymentów wykazał oczekiwany wynik, że większa głębia AI wygrywa przed mniejszą głębią. Z tego powodu można postawić tezę, że 5 – jest optymalną głębią algorytmu minimax dla pola 4x5. Sprawdzimy to kolejnym eksperymentem. Zwiększymy liczbę iteracji do 30.

Tabela 2 – Badanie wpływu głębokości algorytmu minimax na zwycięstwo 5 vs inne.

Głębokość gracza 1	Głębokość gracza 2
<u>5</u>	1
<u>5</u>	2
<u>5</u>	3
<u>5</u>	4
<u>5</u>	5
5	<b>6</b>

Na podstawie uzyskanych danych widzimy, że głębokość przeszukiwania 5 daje dobre wyniki zwycięstwa na polu 5x4. Przewaga nad słabszymi AI jest pełna. Można wnioskować, że głębokość przeszukiwania 5 jest optymalną głębokością na dany, polu. Potwierdza to dane z tabeli 3, gdzie głębokość przeszukiwania 4 wcale nie daje tak dobrych wyników zwycięstwa.

Tabela 3 – Badanie wpływu głębokości algorytmu minimax 4 vs mniejsze.

Głębokość gracza 1	Głębokość gracza 2
4	1
4	2
4	3
4	4

## Wnioski

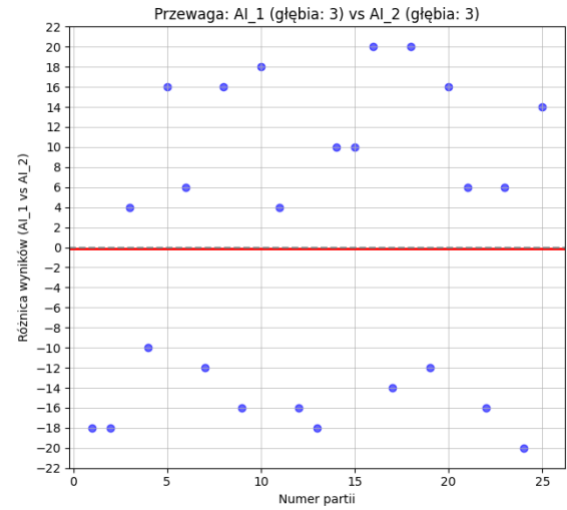
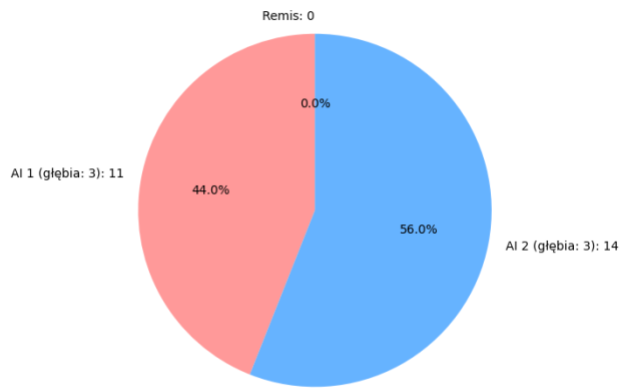
W trakcie wykonania ćwiczenia nr 2 zaimplementowano algorytm minimax z obcinaniem  $\alpha - \beta$ . Zbadano wpływ głębokości rekurencji w algorytmie minimax na skutek zwycięstwa.

Logika podpowiada, że większa głębokość przeszukiwania w algorytmie minimax powinna dawać większe szanse na zwycięstwo, jednak opierając się na uzyskane dane w trakcie eksperymentów nie można potwierdzić danej tezy. Czasami większa głębokość przeszukiwania przegrywa znacznie mniejszej.

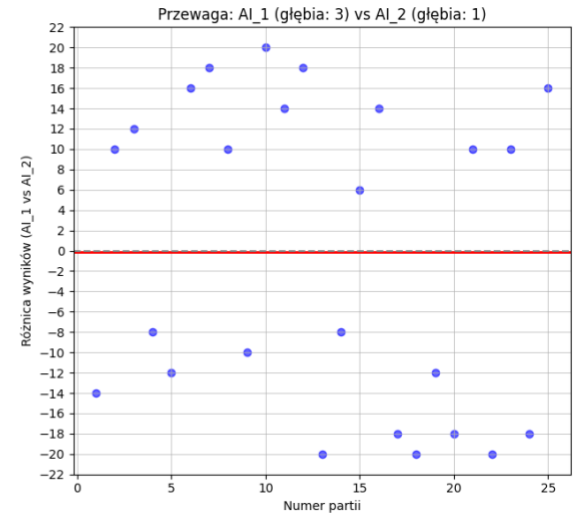
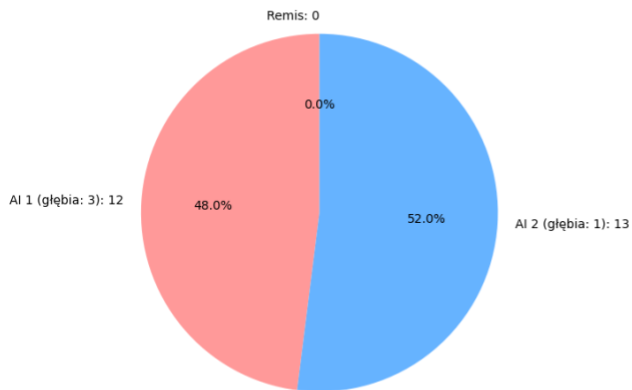
Jednak można stwierdzić, że dla pola 5x4 optymalną głębokością przeszukiwania jest 5.

# TESTY głębokości algorytmu minimax

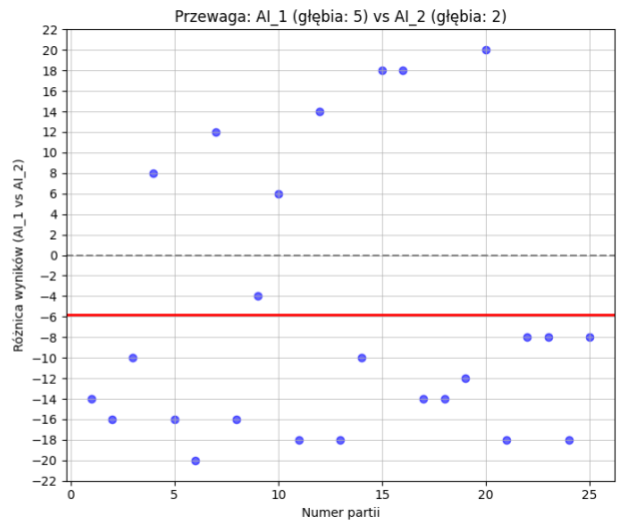
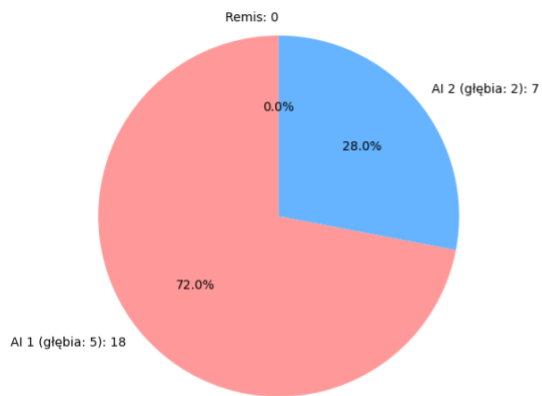
Dystrybucja wygranych. Partii: 25. Rozmiar: 5 x 4



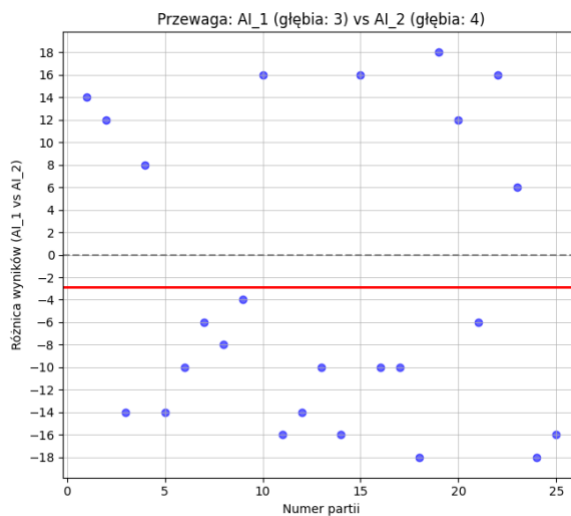
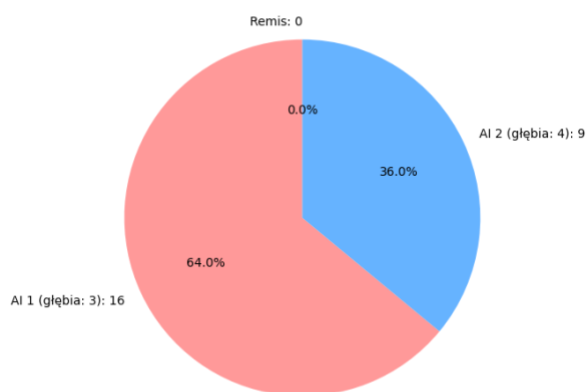
Dystrybucja wygranych. Partii: 25. Rozmiar: 5 x 4



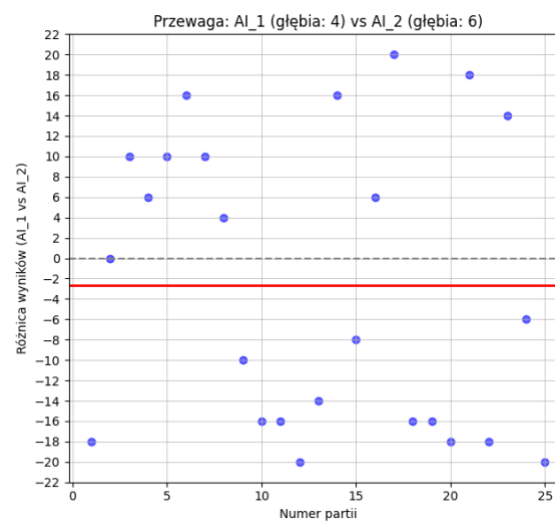
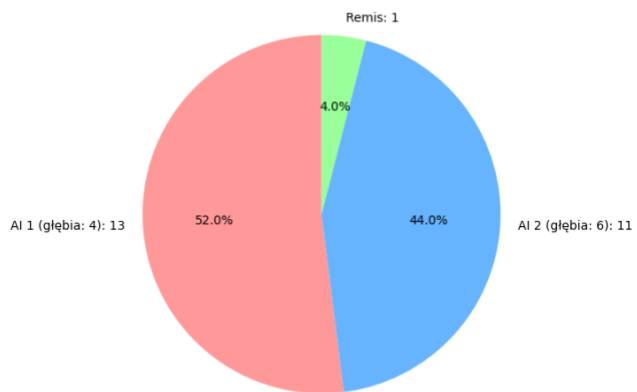
Dystrybucja wygranych. Partii: 25. Rozmiar: 5 x 4



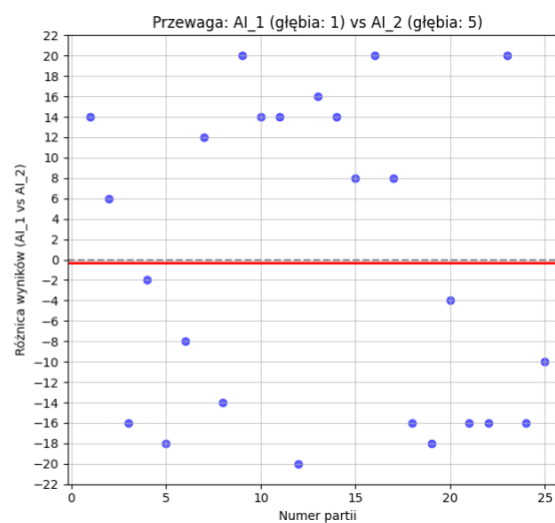
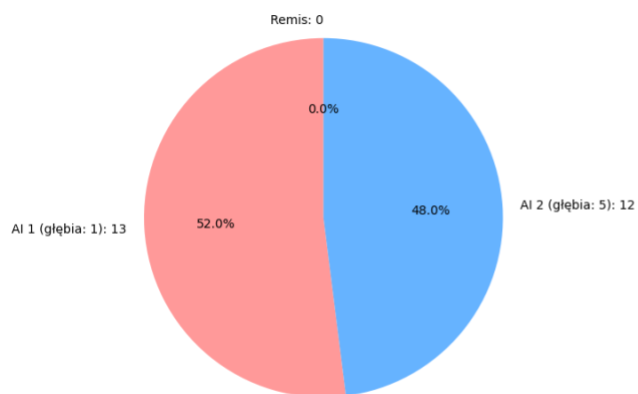
Dystrybucja wygranych. Partii: 25. Rozmiar: 5 x 4



Dystrybucja wygranych. Partii: 25. Rozmiar: 5 x 4



Dystrybucja wygranych. Partii: 25. Rozmiar: 5 x 4



Dystrybucja wygranych. Partii: 30. Rozmiar: 5 x 4

