Algorytm MiniMax z obcinaniem $\alpha - \beta$

Daria Bartkowiak

5 listopada 2025

Spis treści

1 Opis implementacji algorytmu

Zaimplementowałam algorytm **Minimax z obcinaniem alfa–beta** dla gry *Dots and Boxes*, którego celem działania jest wybór ruchu maksymalizującego wynik bieżącego gracza przy założeniu, że przeciwnik gra optymalnie.

1. Reprezentacja i interfejs:
Stan gry opisuje obiekt `DotsAndBoxesState` (z metodami m.in. `get_moves()`, `make_move()`, `is_finished()`, `get_scores()`). Gracz sterowany jest klasą `MinimaxPlayer(depth, rng)`, gdzie:

- `depth` – maksymalna głębokość przeszukiwania drzewa stanów,
- `ran` – generator losowy używany do tie–breaku między równie dobrymi ruchami.

2. Funkcja oceny (heurystyka):
Do oceny węzłów używam prostej, szybkiej heurystyki - **różnicy punktów:**

$$\mathrm{eval}(s) = \mathrm{score}_{\mathrm{player}}(s) - \mathrm{score}_{\mathrm{opponent}}(s),$$

co oznacza, że wartości dodatnie faworyzują bieżącego gracza, ujemne – przeciwnika, a zero odpowiada pozycji remisowej.
3. Przebieg algorytmu:
Dla aktualnego stanu zbierane są wszystkie ruchy `get_moves()`, a następnie rekurencyjnie eksplorowane do głębokości `depth`. Węzły dzielą się na:

- *maksymalizujące* – gdy na ruchu jest gracz, dla którego optymalizujemy (szukamy wartości maksymalnej),
- *minimalizujące* – gdy na ruchu jest przeciwnik (szukamy wartości minimalnej).

W obu przypadkach stosowane jest **obcinanie alfa–beta**: utrzymywane są granice α (najlepsza znana dolna granica dla gracza maksymalizującego) oraz β (najlepsza górna granica dla gracza minimalizującego). Gdy zachodzi $\beta \leq \alpha$, dalsze dzieci bieżącego węzła są **obcinane** (nie mogą zmienić decyzji na wyższym poziomie), co istotnie redukuje liczbę odwiedzanych węzłów.
4. Warunki stopu:
Rekursja kończy się, gdy:

1. osiągnięto zadaną głębokość (`depth == 0`),
2. stan jest terminalny (`is_finished()`).

5. Wybór ruchu i tie–break:
Na poziomie korzenia wybierane są wszystkie ruchy osiągające najlepszą wartość; jeśli jest ich kilka, wykorzystywany jest **losowy tie–break**.

2 Testowanie

Poniżej przedstawiam dwa eksperymenty weryfikujące działanie gracza **Minimax** w grze *Dots and Boxes*. We wszystkich testach stosowano walidację wielokrotną: dla każdej konfiguracji uruchamiano `no_seeds` różnych ziaren losowości oraz `no_games` gier na każde ziarno, a metryki agregowano po wszystkich powtórzeniach. Dodatkowo, aby zniwelować bias pierwszego ruchu, dla każdej konfiguracji rozgrywano **dwie symetryczne serie z różnym graczem zaczynającym** (A-start oraz B-start), a następnie *agregowano* wyniki z obu serii (czyli de facto w pierwszym teście, dla każdej pary (N,M) rozegrano po `no_games*2` gier dla każdego ziarna generatora).

Test 1: porównanie jakości N vs M

Cel: Sprawdzić, jak gracz z głębokością przeszukiwania N radzi sobie w starciu z graczem o głębokości M.

Ustawienia eksperymentu:

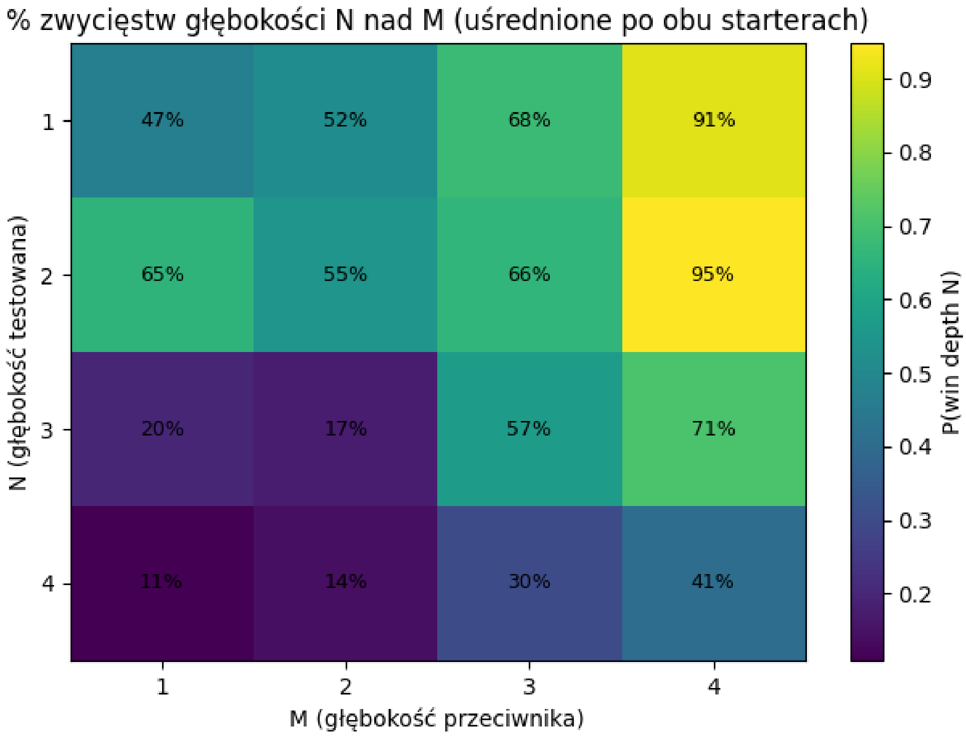
- Plansza: `size = 4`.
- Zakres głębokości: $N, M \in \{1, \ldots, 4\}$.
- Powtórzenia: `no_seeds = 10`, `no_games = 3` na seed.

Metryki: Dla każdej pary (N, M) raportujemy:

- % zwycięstw gracza A (zwycięstwa A / zwycięstwa A + zwycięstwa B) — rysowane jako heatmapa,
- Wyniki wszystkich rozegranych gier.

Tabela 1: Macierz % zwycięstw gracza z głebokością N dla par głębokości (N, M)

$N\backslash M$	1	2	3	4
1	0.471	0.519	0.681	0.909
2	0.654	0.547	0.660	0.948
3	0.200	0.173	0.571	0.709
4	0.109	0.143	0.304	0.407



Wnioski:

- **Brak monotoniczności:** niższe głębokości $(N = 1, 2)$ częściej wygrywają (co dziwne) niż wyższe $(N = 3, 4)$, np. $N = 2$ vs $M = 4 \approx 95\%$ wygranych dla $N = 2$.
- **Co to oznacza:** samo zwiększanie *depth nie poprawia* gry; model najprawdopodobniej zaczyna popełniać systematyczne błędy.
- **Dlaczego tak się dzieje?** używana przez nas prosta heurystyka (różnica punktów) nagradza szybkie, małe zyski. Przy większej głębokości algorytm częściej wybiera takie krótkie domknięcia, które w kolejnych ruchach pozwalają rywalowi przejąć długi łańcuch pól. W efekcie głębsze liczenie bywa gorsze.

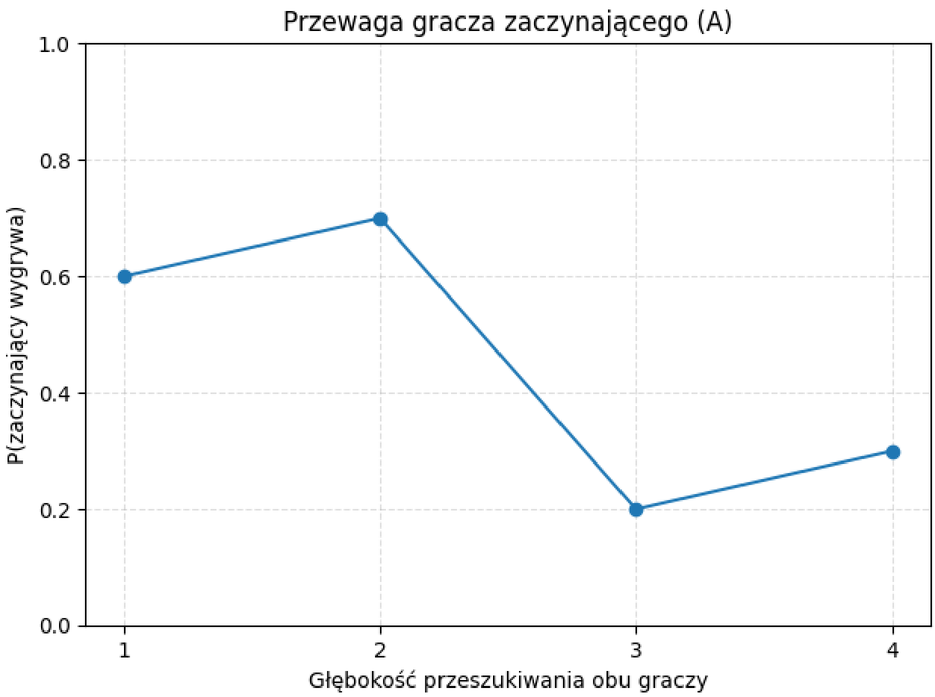
Test 2: czy zaczynanie ma znaczenie?

Cel: Ocenić, czy **zaczynający gracz** ma istotnie większą szansę wygranej, przy jednakowych głębokościach obu graczy ($N = M$).

Ustawienia:

- Plansza: `size = 4`.
- Zakres głębokości: $N, M \in \{1, \dots, 4\}$.
- Powtórzenia: `no_seeds = 10`, `no_games = 6` na seed.

Metryka: Dla każdego N raportujemy `P(startujący wygrywa)`, tj. odsetek gier, w których gracz zaczynający (A) zwycięża.



Wnioski:

- **Dla płytkich głębokości** start pomaga (≈ 0.6 dla $N = 1$, ≈ 0.7 dla $N = 2$).
- **Dla głębszych** przewaga **słabnie lub znika** (≈ 0.2 dla $N = 3$, ≈ 0.3 dla $N = 4$).
- Generalnie nie widać przewagi rozpoczynającego gracza, pomimo początkowych obaw.