

Wprowadzenie do sztucznej inteligencji – Laboratorium 3

1. Treść zadania

Celem zadania była implementacja algorytmu min-max w grze kółko i krzyżyk na planszy 3x3. Program powinien grać sam ze sobą i wizualizować kolejne stany gry w terminalu.

2. Struktura projektu

Projekt składa się z następujących plików:

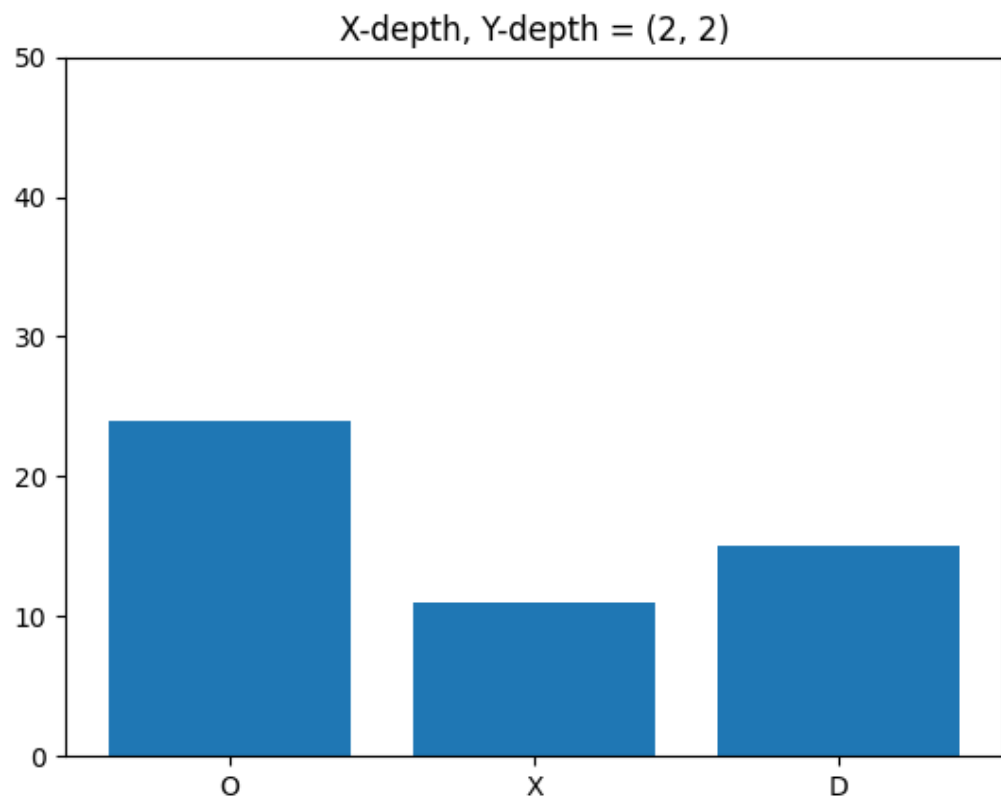
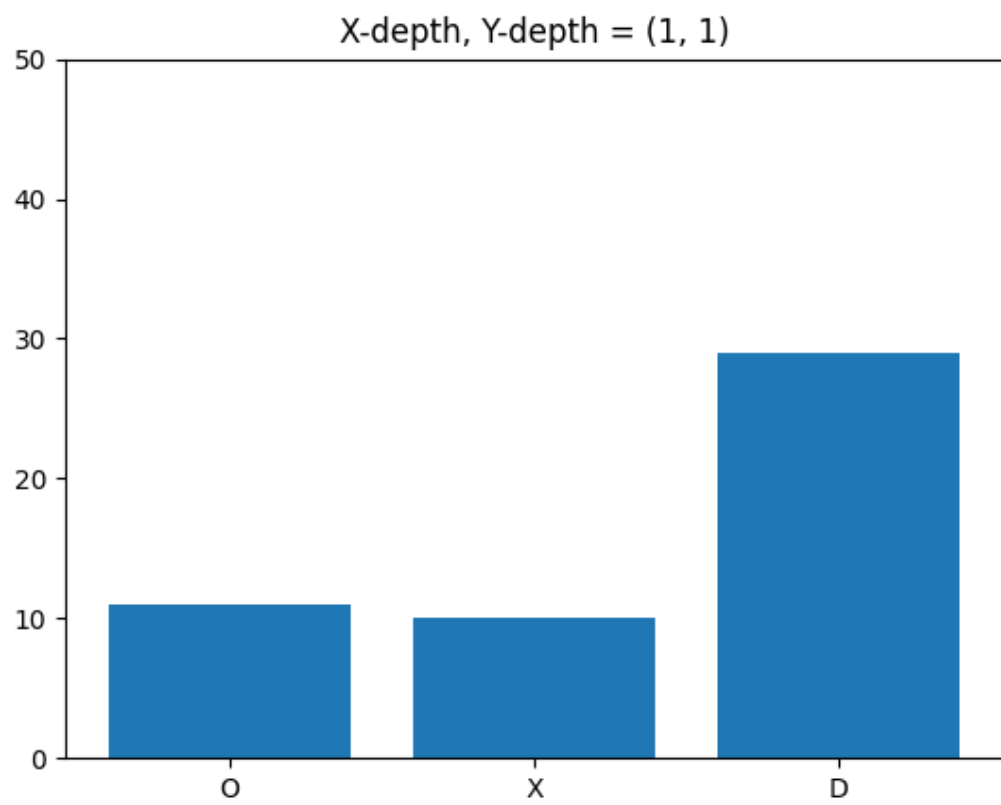
- board.py – implementacja planszy do gry
- min_max.py – implementacja algorytmu min-max
- main.py – algorytm gry i testowanie algorytmu
- player.py – implementacja gracza
- results.json – wyniki testowania algorytmu
- plots.py – rysowanie wykresów
- plots – katalog z wygenerowanymi wykresami

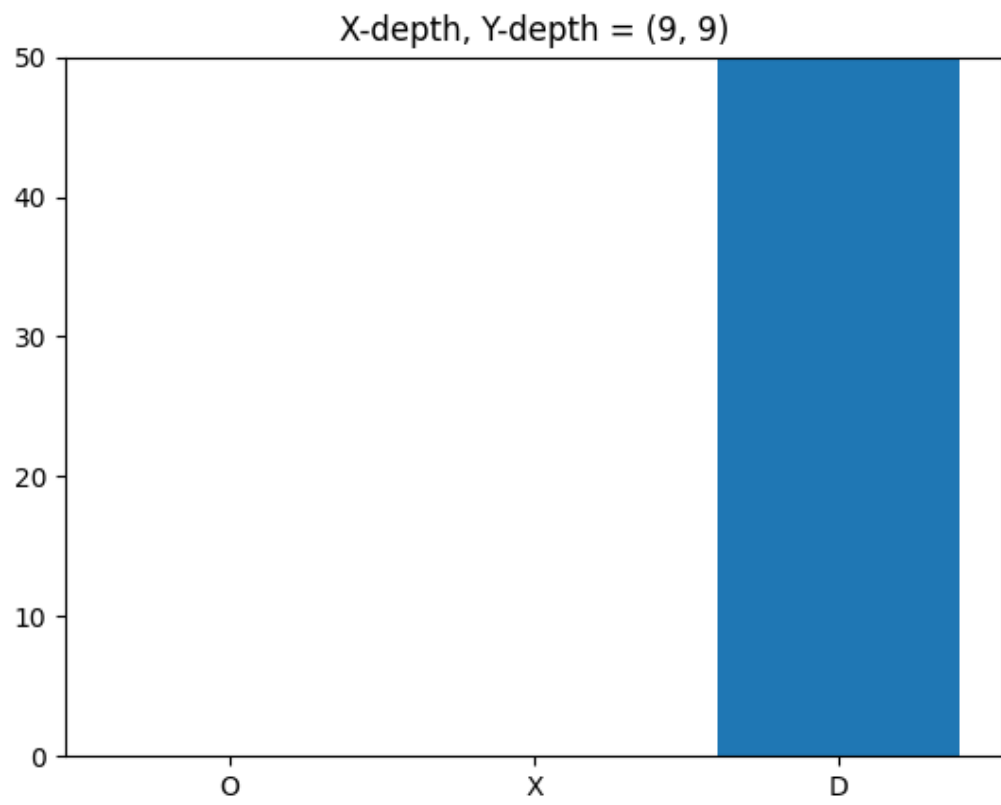
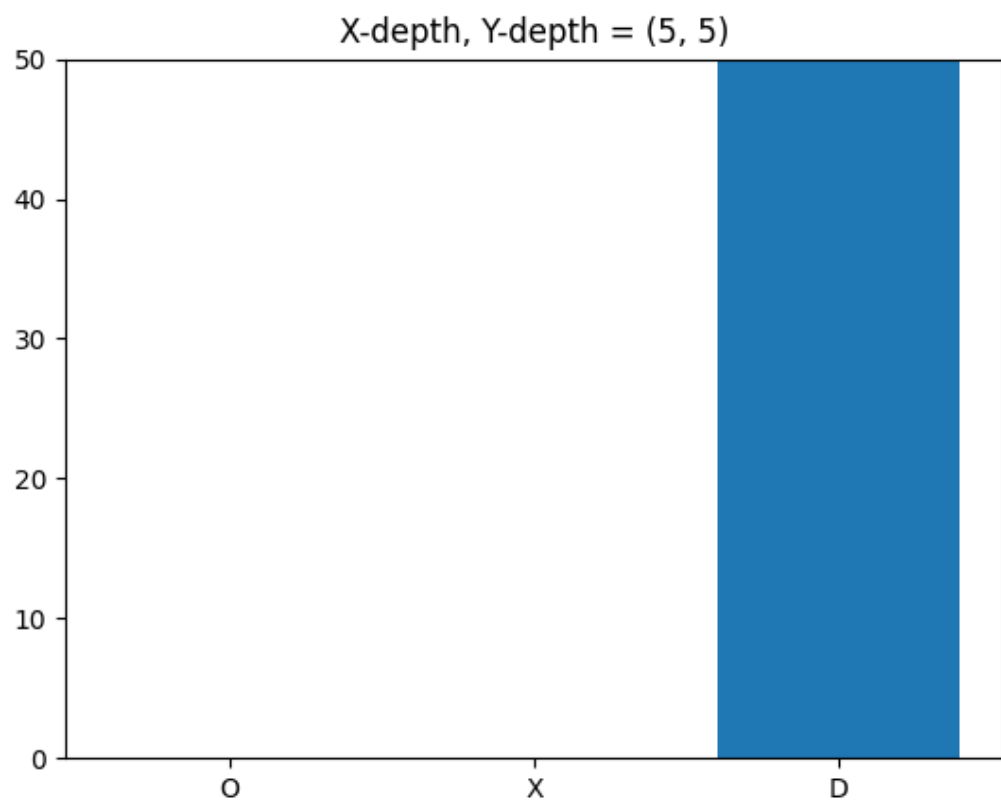
3. Implementacja algorytmu

Zaimplementowany przeze mnie algorytm jest algorytmem min-max z daną głębokością poszukiwań. W każdym ruchu algorytm tworzy wszystkie możliwe plansze, oraz rekurencyjnie wybiera najlepszą z nich. W zależności od tego, którego gracza jest obecny ruch, wybierana jest plansza o największej lub najmniejszej wartości. Funkcja oceniająca planszę zwracała 1 lub -1 gdy wygrał jeden z graczy lub 0 gdy gra nie była rozstrzygnięta. Po stworzeniu listy najlepszych plansz w danym momencie, losowo wybierana jest jedna z nich i staje się ona obecną planszą.

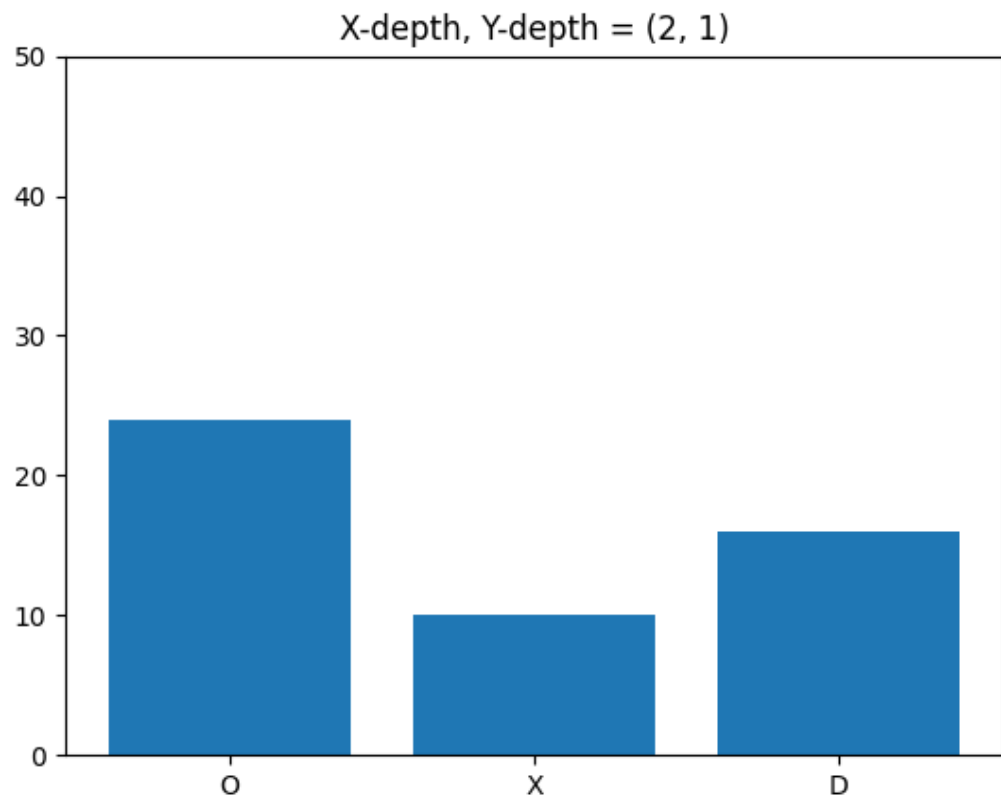
4. Testowanie

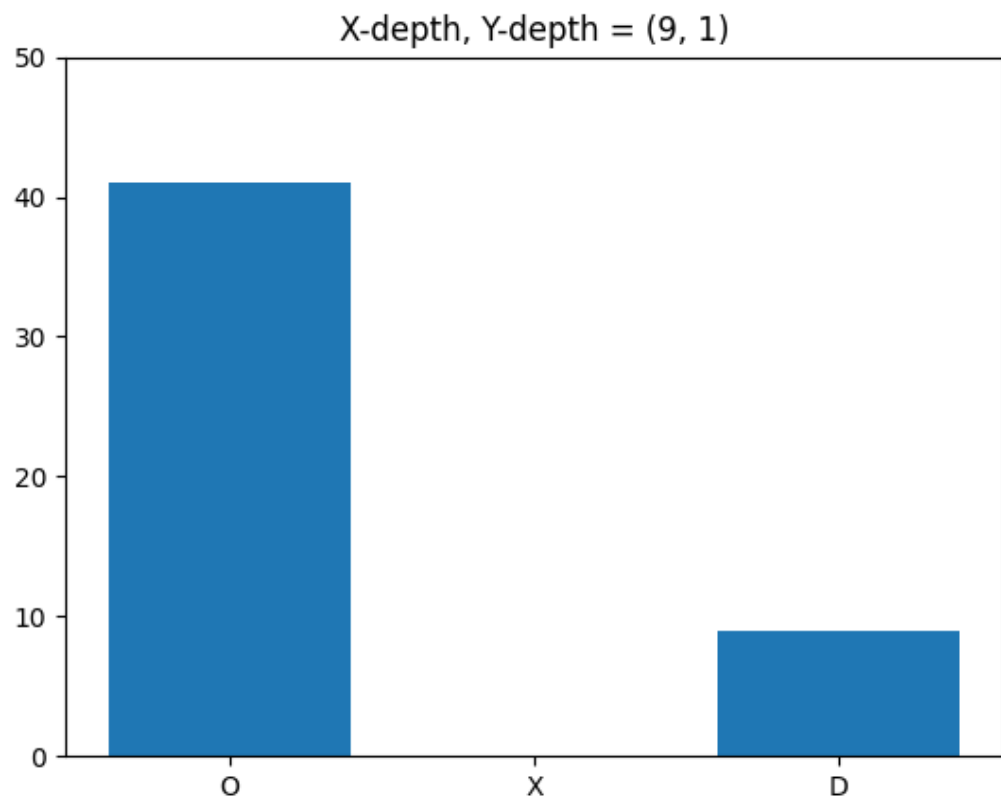
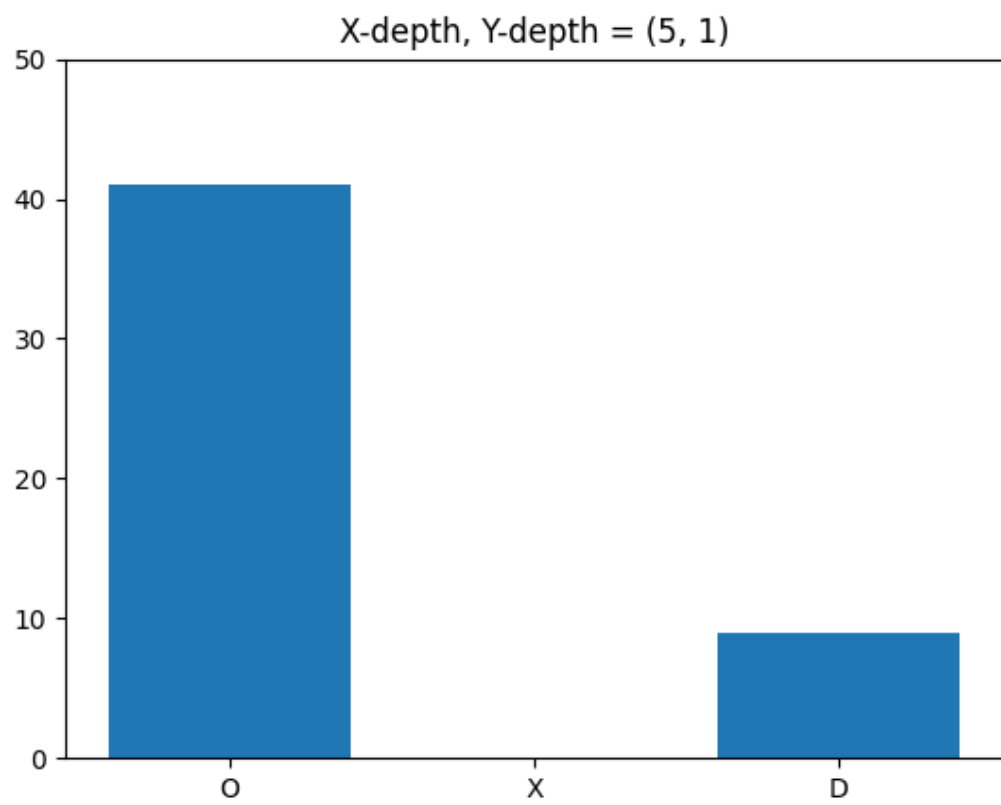
Przetestowałem wpływ głębokości przeszukiwań na przebieg gry dwóch graczy sterowanych przez komputer.



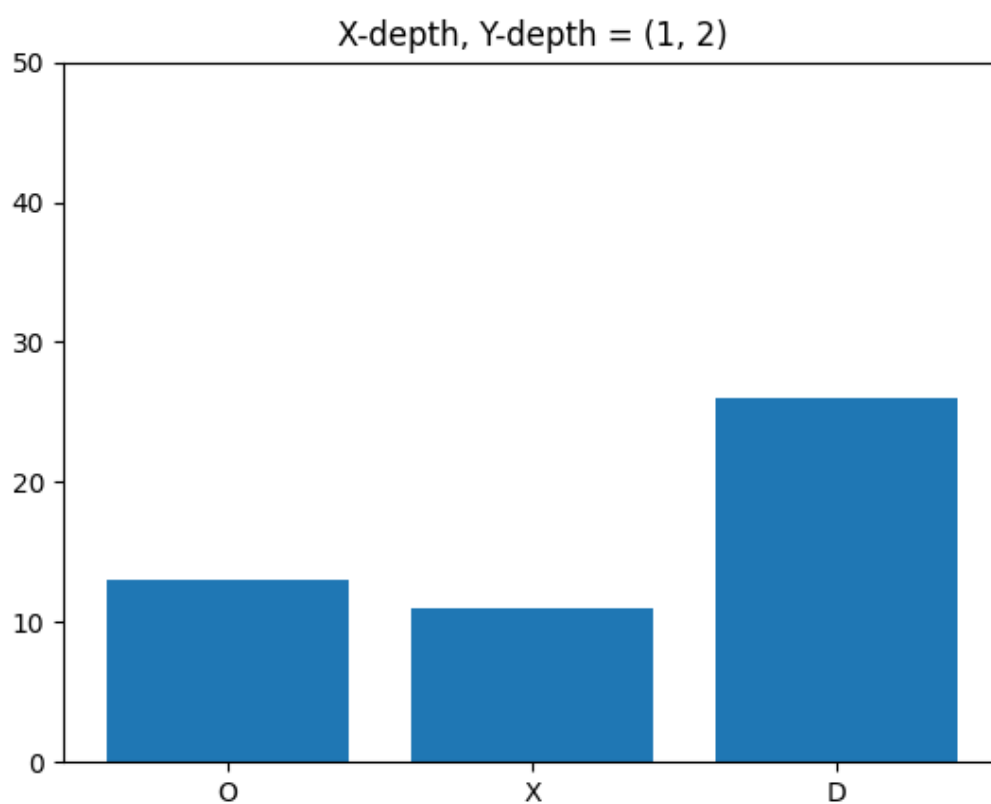


Gdy obaj gracze mają taką samą głębokość przeszukiwań przewagę ma gracz, który zaczyna. Dla niewielkich głębokości przeszukiwań osiąga on więcej zwycięstw niż gracz drugi. Gdy głębokości są odpowiednio duże gracze wykonują same optymalne ruchy. Przy takim przebiegu, gra nie może się skończyć w inny sposób niż remisem.

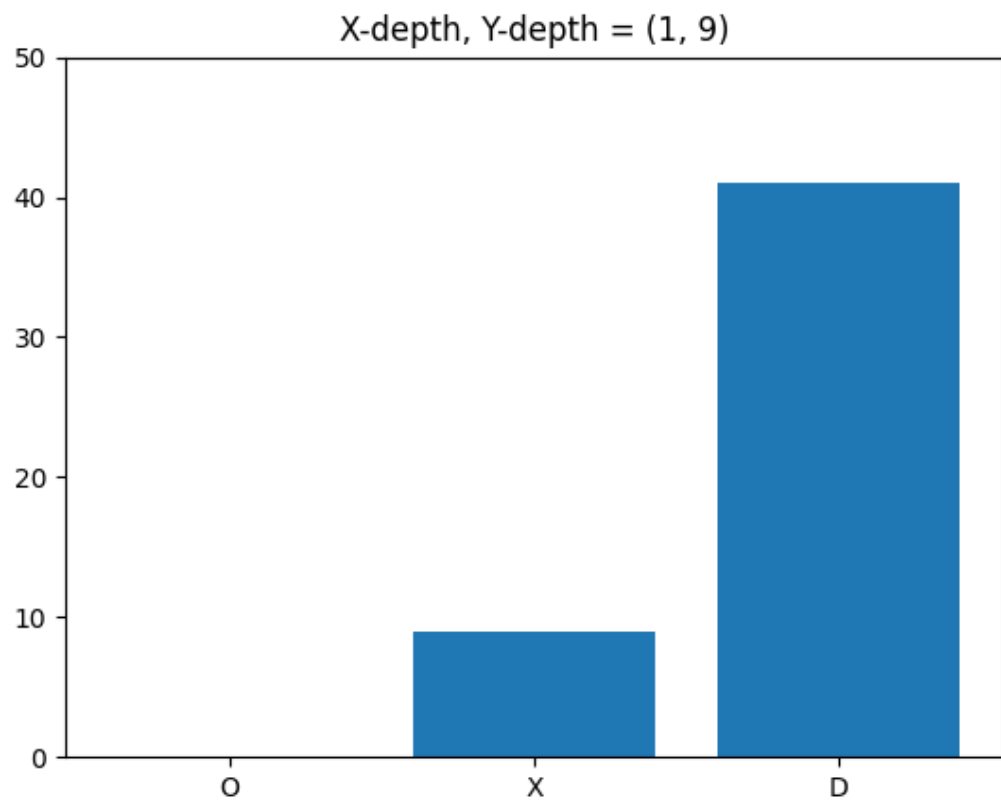
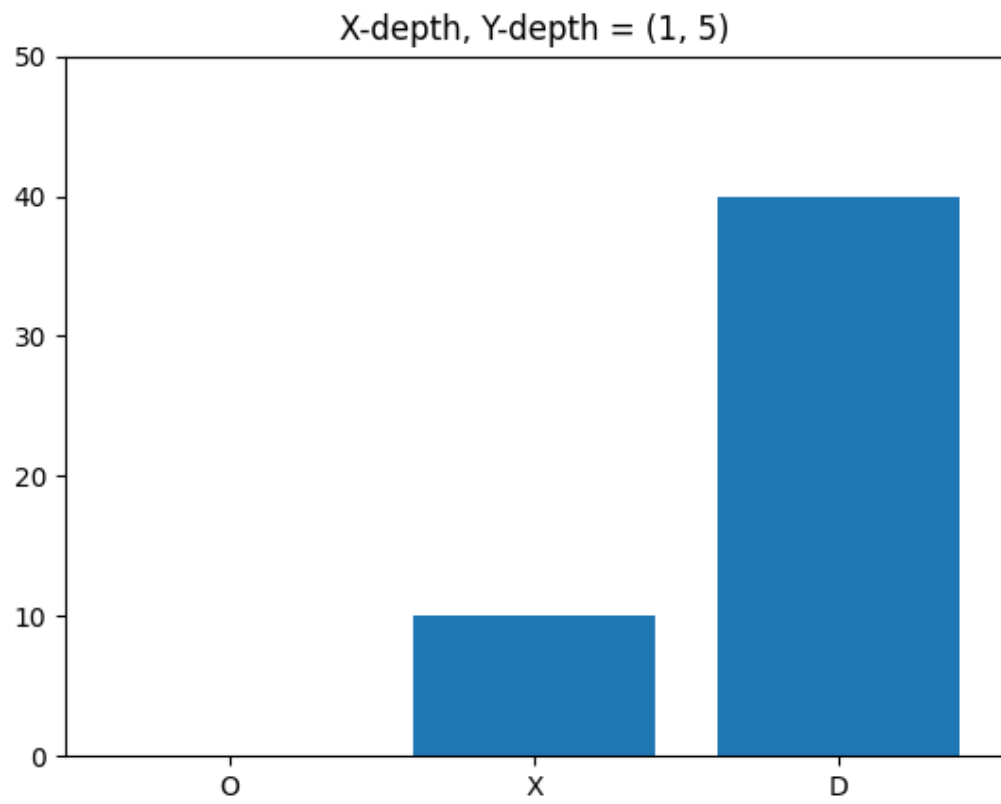




Gdy zwiększamy pierwszemu graczowi poziom przeszukiwań osiągnane przez niego wyniki znacząco się poprawiają. Nie przegrywa on ani razu. Oznacza to, że gracz 1. lepiej przewiduje rozwój wydarzeń na planszy i potrafi skuteczniej wybierać optymalne ruchy niż gracz 2.



*



Gdy zwiększamy głębokość przeszukiwań graczowi który zaczyna jako drugi jego osiągnięta przez niego liczba zwycięstw nie poprawia się. Rośnie natomiast liczba remisów w partii. Gracz, który zaczyna jako pierwszy nie wygrywa ani razu, gdy głębokość poszukiwań jest odpowiednio duża. Ok 85% partii kończy się remisem. Oznacza to, że algorytm skutecznie reaguje na ruchy gracza inicjującego rozgrywkę. Blokując jego ruchy doprowadza do remisu.

5. Wnioski końcowe

Dzięki algorytmowi min-max możemy stworzyć gracza komputerowego, który będzie grał optymalnie. Dzięki dobraniu odpowiedniej głębokości poszukiwań gracz będzie osiągał dobre rezultaty, a czas działania algorytmu i jego złożoność będą zadowalające. Zwiększenie głębokości poszukiwań skutkuje znacznym wydłużeniem działania algorytmu w początkowych ruchach. W grze kółko – krzyżyk maksymalna głębokość przeszukiwań wynosi 9. W kolejnym kroku nawet jeśli stworzony przez nas gracz ma taką głębokość, tworzy plansze o maksymalnej głębokości 8, ponieważ tyle zostało wolnych pól. Z tego wynika, że algorytm z każdym ruchem będzie coraz szybciej wyliczał kolejny optymalny ruch.