

Wprowadzenie do sztucznej inteligencji - ćwiczenie 3

Zadanie

Zaimplementować algorytm min-max z przycinaniem alfa-beta. Algorytm ten należy zastosować do gry w proste warcaby (checkers/draughts). Niech funkcja oceny planszy zwraca różnicę pomiędzy stanem planszy gracza a stanem przeciwnika. Za pion przyznajemy 1 punkt, za damkę 10 p. Przygotowałem dla Państwa kod, który powinien ułatwić wykonanie zadania. Zamiast mojego kodu, osoby chętne mogą napisać własny kod pomocniczy. Nie można używać kodu z Internetu, czy bardziej ogólnie, kodu, którego nie jest się autorem.

Wiem co jest dostępne w Internecie, większość dostępnych implementacji ma cechy szczególne, po których łatwo je rozpoznać.

Zasady gry (w skrócie: wszyscy ruszają się po 1 polu. Pionki tylko w kierunku wroga, damki w dowolnym) z następującymi modyfikacjami:

1. bicie nie jest wymagane
2. dozwolone jest tylko pojedyncze bicie (bez serii).

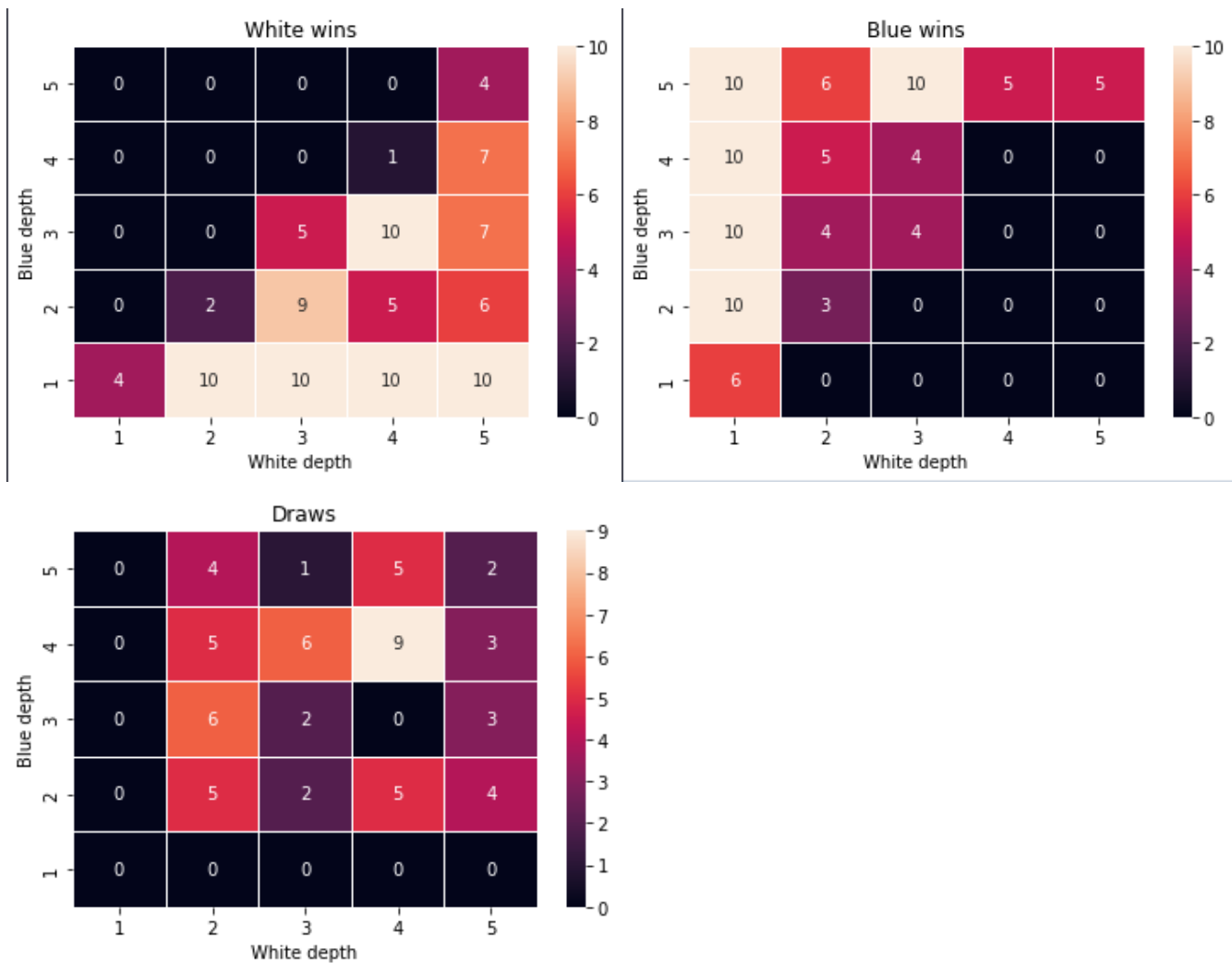
Pytania

Czy gracz sterowany przez AI zachowuje się rozsądnie z ludzkiego punktu widzenia? Jeśli nie to co jest nie tak?

Gracz sterowany przez AI myśli w sposób zgoła odmienny od ludzkiego. Zdarza mu się popełniać błędy, które nie przytrafiłyby się człowiekowi (np. zablokować wszystkie swoje pionki próbując zdobyć damkę) lub nie widzieć szans, które człowiek zobaczyłby na pierwszy rzut oka (np. prosta droga do damki, której długość przekracza jednak głębokość przeszukiwań algorytmu). W klasycznej wersji bardzo często zdarza mu się zapętlać w konkretnej sekwencji ruchu, co zniwelowałem poprzez wybieranie losowego z ruchów, jeśli więcej niż jeden został oceniony w najlepszy sposób.

Niech komputer gra z komputerem (bez wizualizacji), zmieniamy parametry jednego z oponentów, badamy jak zmiany te wpłyną na liczbę jego wygranych. Należy zbadać wpływ:

1. Głębokości drzewa przeszukiwań

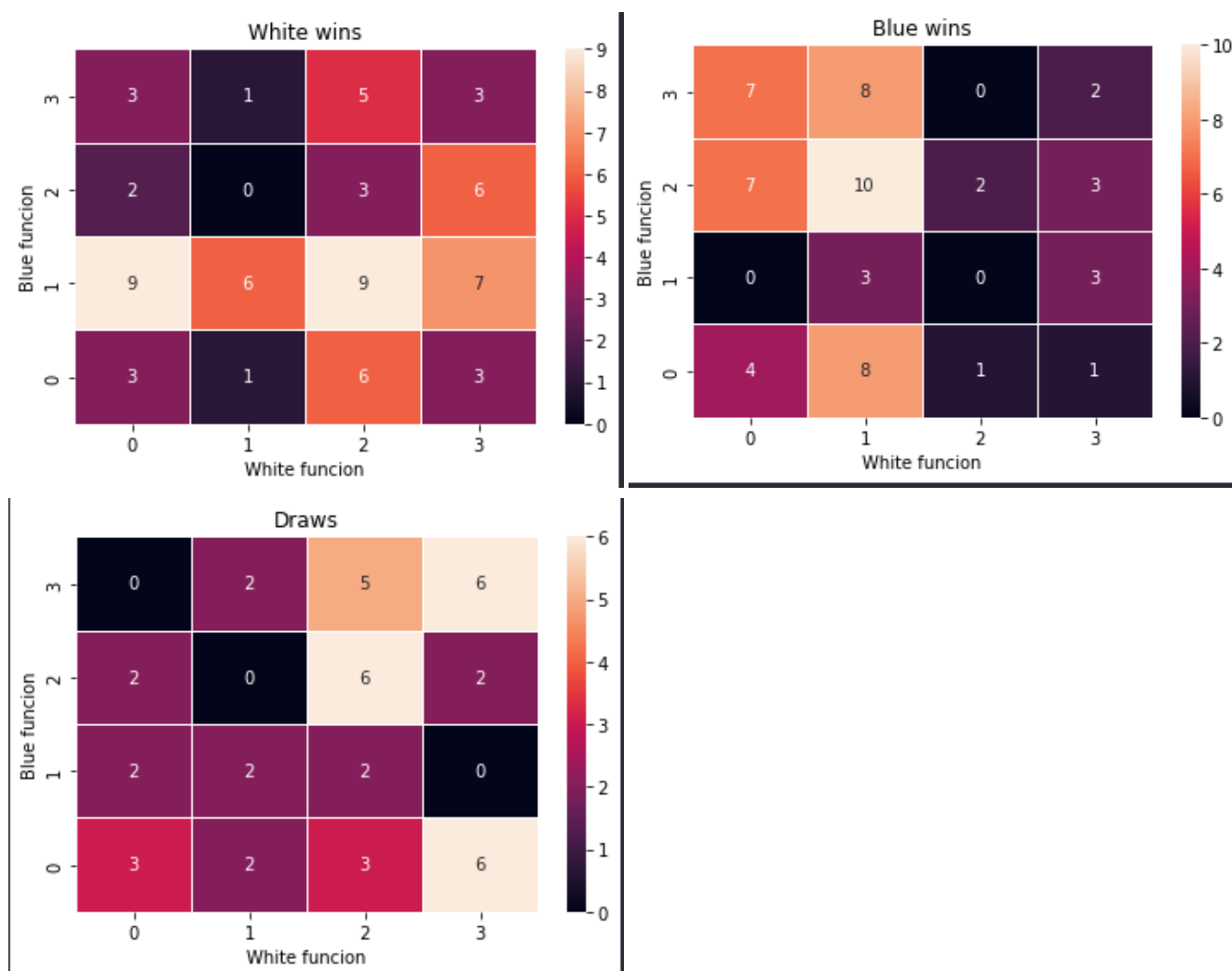


Już na pierwszy rzut oka widać, że im większa głębokość drzewa przeszukiwań, tym algorytm radzi sobie lepiej.

2. Alternatywnych funkcji oceny stanu:

0. Wersja podstawowa

1. Nagrody jak w wersji podstawowej + nagroda za stopień zwartości grupy (jak wszyscy blisko siebie to OK, no chyba, że da się coś zabrać przeciwnikowi)
2. Za każdy pion na własnej połowie planszy otrzymuje się 5 nagrody, na połowie przeciwnika 7, a za każdą damkę 10.
3. Za każdy nasz pion otrzymuje się nagrodę w wysokości: $(5 + \text{numer wiersza, na którym stoi pion})$ (im jest bliżej wroga tym lepiej), a za każdą damkę 10.



Można zauważyć, że funkcja oceny stanu także ma znaczenie na efektywność algorytmu. Najlepiej zachowały się algorytmy premiujące grę ofensywną - 2 i 3, które premiowały przekraczanie połowy i jak najszybsze dążenie do osiągnięcia damek. Znacznie gorzej wypadł algorytm numer 1, który zamiast na ekspansji planszy, skupiał się zbytnio na pozostawaniu w zwartej grupie. Pomiedzy nimi uplasował się algorytm podstawowy.