

# Raport do zadania 3

Algorytm MiniMax

Kacper Siemionek

Numer indeksu: 331430

## 1. Założenia algorytmu MiniMax

Algorytm MiniMax zaimplementowany w klasie MinMaxAgent decyduje o wyborze pierwszego lub ostatniego elementu zadanej listy przeszukując możliwe pozycje z określoną głębokością oraz dla każdej z nich wyznacza odpowiednią ocenę. Poza maksymalizacją własnego wyniku algorytm stara się zminimalizować wynik przeciwnika, tym samym dąży do wyboru optymalnego ruchu. W przypadku minimalizacji jako ocenę podajemy ujemne wartości sumy punktów.

## 2. Wpływ głębokości drzewa na wyniki, porównanie agentów

Aby sprawdzić wpływ głębokości drzewa przeszukiwań na wyniki algorytmu, zostały uruchomione testy sprawdzające średni wynik MinMaxAgent'a w pojedynku z GreedyAgent'em dla następujących głębokości: 1, 2, 3, 15. Dla każdej głębokości uruchomiono 1000 gier, każdy gracz rozpoczynał rozgrywkę tyle samo razy. Wyniki testów przedstawione są w poniższej tabeli:

Głębokość	Średni czas	Średni wynik Greedy	$\sigma$ Greedy	Średni wynik MinMax	$\sigma$ MinMax
1	0.000018s	-3.01	16.60	2.56	15.50
2	0.000030s	-4.31	15.96	3.63	15.28
3	0.000062s	-4.18	15.84	3.72	14.72
15	0.015068s	-3.47	16.29	4.81	14.74

Tabela 2.1 Wyniki testów dla GreedyAgent'a i MinMaxAgent'a

Powyższe dane ukazują nam istotny wpływ głębokości na wyniki rozgrywki. Większa głębokość oznacza, że MinMaxAgent może ujrzeć więcej pozycji, które zostaną poddane ocenie. W takiej sytuacji zwiększa się szansa na to, że zostanie podjęta bardziej trafna decyzja. Przy mniejszej głębokości MinMaxAgent nadal był w stanie wygrać, jednak podejmowane decyzje nie były tymi najlepszymi. Głębokość wpływa również na średni czas rozgrywek, przeszukanie każdego dodatkowego stopnia drzewa wymaga od algorytmu dokonania większej ilości obliczeń.

Poniżej znajdują się wyniki dla pojedynków NinjaAgent'a i MinMaxAgent'a oraz dwóch MinMaxAgent'ów.

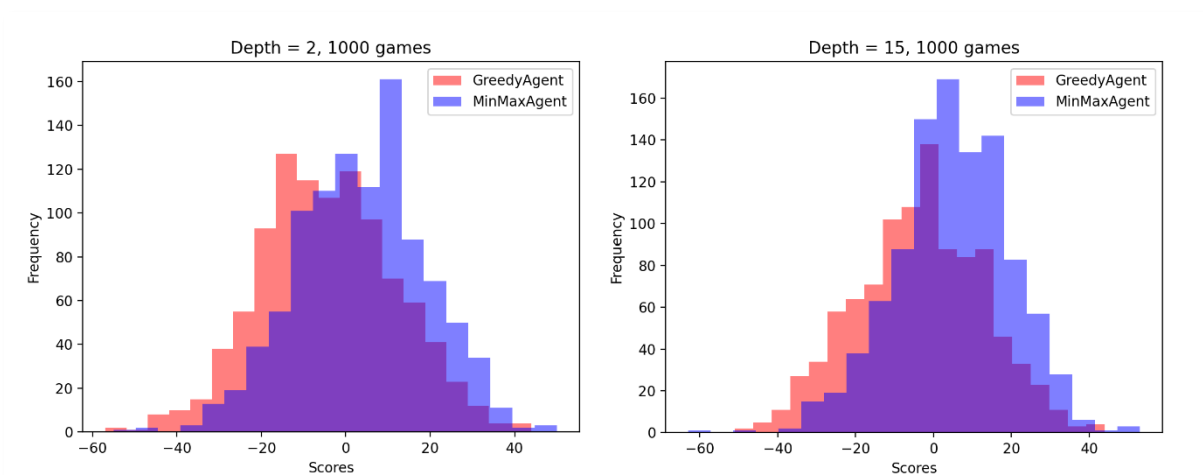
Głębokość	Średni czas	Średni wynik Ninja	$\sigma$ Ninja	Średni wynik MinMax	$\sigma$ MinMax
2	0.000051s	-1.02	16.83	0.56	16.54
15	0.028854s	-2.36	16.73	1.68	16.83

Tabela 2.2 Wyniki testów dla NinjaAgent'a i MinMaxAgent'a

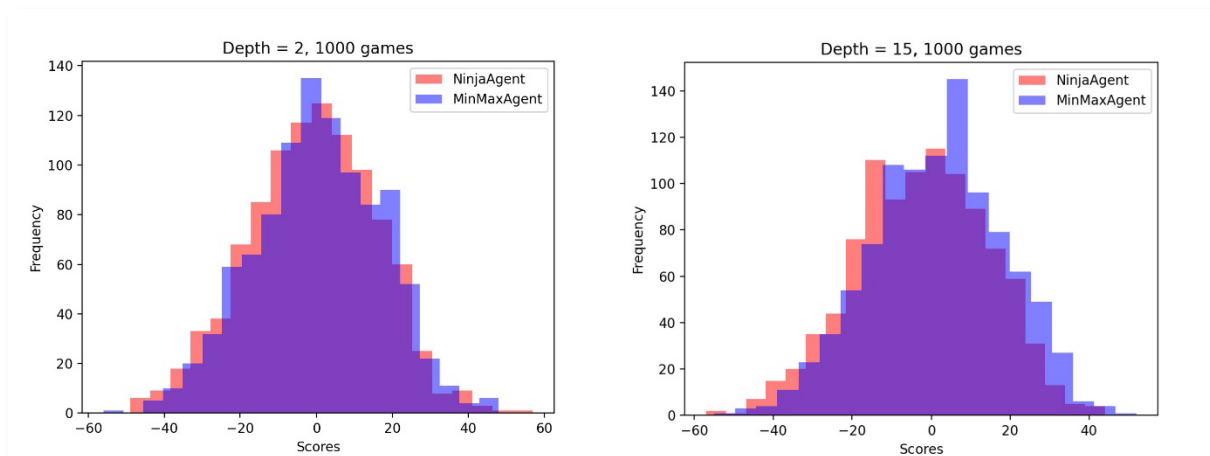
Głębokość	Średni czas	Średni wynik MinMax	$\sigma$ MinMax	Średni wynik MinMax(15)	$\sigma$ MinMax(15)
2	0.000051s	-1.47	16.23	1.01	16.02
15	0.028854s	-0.48	16.06	-0.20	16.17

Tabela 2.3 Wyniki testów dla dwóch MinMaxAgent'ów

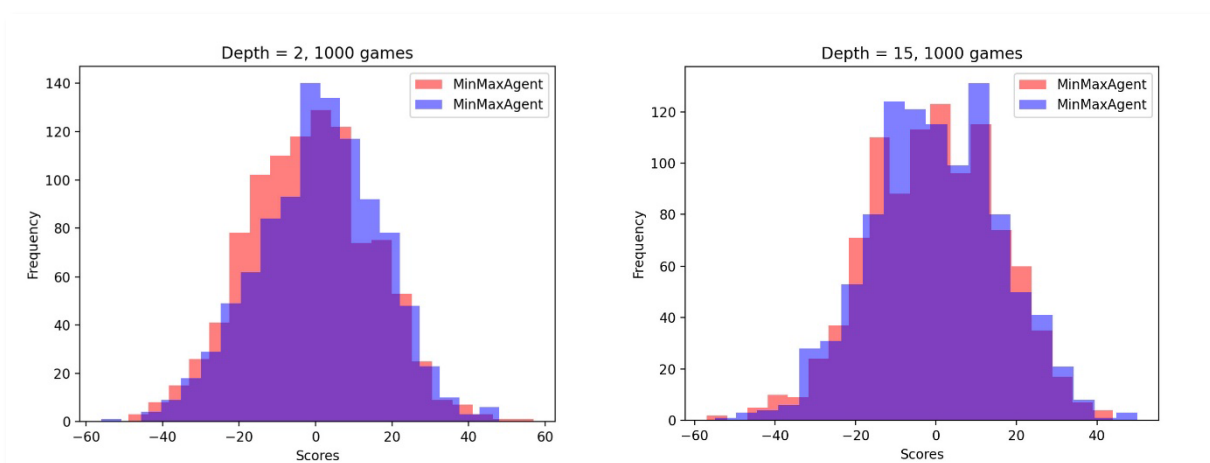
Średni wynik dla NinjaAgent'a w przypadku obu głębokości jest znacznie lepszy od GreedyAgent'a, mimo to nadal nie był w stanie wygrać. Natomiast średnie wyniki dwóch MinMaxAgent'ów różnią się tylko wtedy, gdy zastosowaliśmy różne głębokości, przy takich samych wynik jest bardzo zbliżony i nie zależy od graczy, a od wylosowanych liczb w rozgrywce.



Rysunek 2.1 Rozkład sumy punktów GreedyAgent'a i MinMaxAgent'a



Rysunek 2.2 Rozkład sumy punktów NinjaAgent'a i MinMaxAgent'a



Rysunek 2.3 Rozkład sumy punktów dwóch MinMaxAgent'ów

Jeżeli potraktujemy sumę punktów danego gracza w pojedynczej rozgrywce jako zmienną losową, możemy zauważyć, że ma ona rozkład normalny. Z histogramów rozgrywek jesteśmy w stanie bez problemu odczytać, który gracz posiadał większą średnią punktów w przeprowadzonym teście. Histogram przesunięty bardziej w prawą stronę wskazuje na zwycięzcę, ponieważ dla takiego rozkładu średni wynik jest większy. W przypadku pojedynku dwóch MinMaxAgent'ów jesteśmy w stanie wybrać lepszego gracza tylko przy głębokości równej 2, w większej różnicy są na tyle małe, że rozkłady się pokrywają.

### 3. Przyspieszenie oraz poprawienie działania algorytmu

W celu przyspieszenia działania algorytmu MiniMax możemy zastosować ograniczenie  $\alpha - \beta$ . Pozwala ono na eliminację części drzewa możliwych pozycji, które z góry można wykluczyć jako gorsze od dotychczas znalezionej najlepszego rozwiązania. Mniej rozgałęzione drzewo oznacza mniej pozycji do przeszukania, co przekłada się na czas potrzebny do znalezienia dobrego ruchu.

MinMaxAgent zainicjowany z maksymalną głębokością równą 1 jest w stanie ocenić tylko jeden ruch w przód. Aby uzyskać lepsze wyniki przy niższych wartościach głębokości, możemy modyfikować funkcję oceny w taki sposób, aby uwzględniała w zwracanej wartości nie tylko wybraną liczbę, ale także wpływ liczb, które zostały w grze po dokonanych wyborze.