

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Sprawozdanie

Wprowadzenie do sztucznej inteligencji

Ćwiczenie nr. 3

Mikołaj Bańkowski

Numer albumu 310408

prowadzący
Grzegorz Rypeś

Warszawa 2024

1. Temat ćwiczenia

Dany jest n-elementowy wektor liczb całkowitych z przedziału $[-10;10]$. Dwóch agentów gra w grę turową polegającą na tym, że w swojej turze agent może zabrać z wektora skrajną liczbę, tj. tę z początku wektora, bądź z końca. Gra kończy się, gdy wszystkie liczby zostaną zabrane z wektora. Celem każdego agenta jest wybrać liczby tak, aby suma zebranych przez niego liczb była jak największa. Przykładowo dla wektora $[2, 0, 3, 1]$ najlepsze możliwe rozegranie gry dla gracza rozpoczynającego to $[2, 3]$, a gracz drugi ma wtedy $[0, 1]$, lub $[1, 0]$.

Do ćwiczenia dostarczone są dwa pliki:

- `main.py`, zawierające implementacje agentów grające w grę i funkcje pomocnicze
- `test.py`, zawierający testy jednostkowe. Istniejących testów nie wolno zmieniać, ich nieuzasadniona edycja będzie skutkowała wyzerowaniem liczby punktów. Można natomiast dopisać swoje testy.

Polecenie:

Proszę dokończyć implementację agenta `MinMaxAgent` wykorzystującego algorytm `MiniMax`. Powinien on przechodzić wszystkie testy jednostkowe.

Następnie proszę przetestować go w starciu z agentem posługującym się heurystyką zachłanną (`GreedyAgent`) w 1000 rozgrywkach dla $n=15$ i następujących głębokości drzewa stanów (1, 2, 3, 15). Należy w tabelce raportować średni czas wykonania się gry, średnią i odchylenie standardowe sum punktów uzyskanych w rozgrywkach.

Odpowiedź na pytanie: Jak głębokość drzewa wpływa na wyniki uzyskiwane przez `MinMaxAgent`-a?

Następnie proszę dla głębokości drzewa 2 oraz 15 potraktować sumę punktów uzyskiwanych przez `MinMaxAgent`-a w każdej rozgrywce jako zmienną losową.

Odpowiedź na pytanie: Jaki jest jej rozkład? (Dobrym sposobem na wizualizację rozkładu jest stworzenie histogramu)

Proszę powtórzyć powyższe testy zamieniając `GreedyAgent`-a na `NinjaAgent`-a, oraz `GreedyAgent`-a na drugiego `MinMaxAgent`-a (głębokość 15).

Odpowiedź na pytanie: Jak na podstawie histogramu stwierdzić, który agent jest lepszy? Jak przyspieszyć działanie `MinMaxAgent`-a?

Odpowiedź na pytanie: Jak poprawić jego działanie dla drzewa przeszukiwań o głębokości 1?

Uzyskane tabelki, odpowiedzi na pytania i 6 podpisanych histogramów proszę zamieścić w raporcie w formacie pdf.

Uwagi:

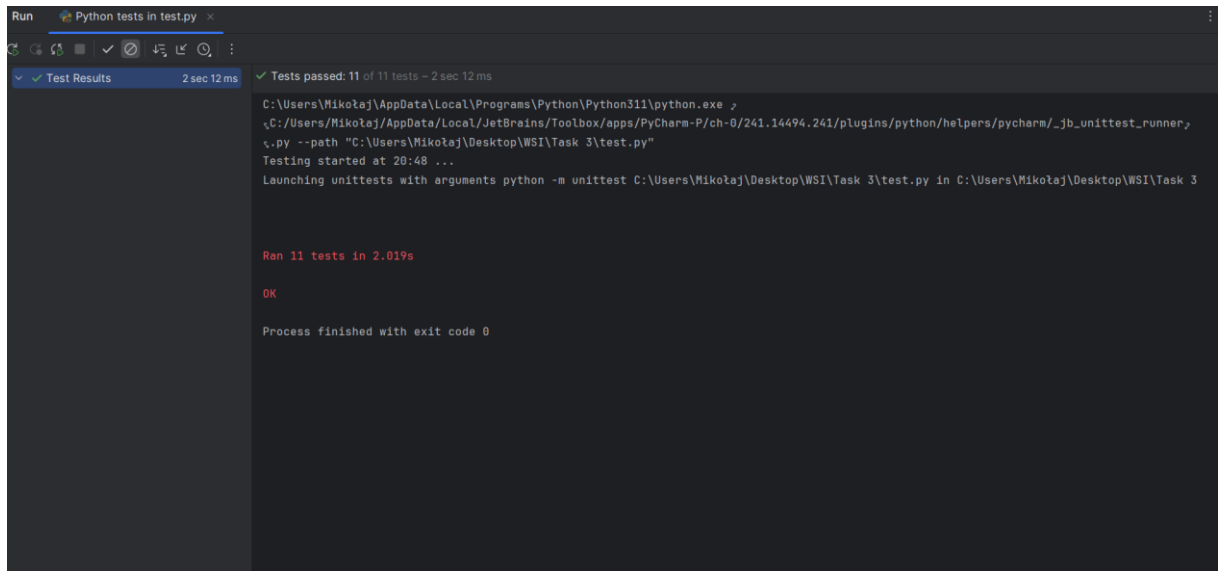
Proszę zauważyć, że aby sprawiedliwie móc porównać agenty, każdy z nich powinien rozpoczynać grę tę samą liczbą razy. W 3. linii pliku `main.py` proszę ustawić ziarno generatora liczb pseudolosowych zgodnie z komentarzem.

Szansa to nie jest prawdopodobieństwo [https://pl.wikipedia.org/wiki/Szansa_\(statystyka\)](https://pl.wikipedia.org/wiki/Szansa_(statystyka)).

Przebieg testów nie zawsze oznacza, że algorytm działa prawidłowo.

2. Testy jednostkowe

Zaimplementowany algorytm MinMax przechodzi pomyślnie wszystkie testy jednostkowe(11/11)



```
Run Python tests in test.py x
Test Results 2 sec 12 ms Tests passed: 11 of 11 tests - 2 sec 12 ms
C:\Users\Mikołaj\AppData\Local\Programs\Python\Python311\python.exe -c
c:\Users\Mikołaj\AppData\Local\JetBrains\Toolbox\apps\PyCharm-P\ch-0\241.14494.241\plugins\python\helpers\pycharm\_jb_unittest_runner.py
--path "C:\Users\Mikołaj\Desktop\WSI\Task 3\test.py"
Testing started at 20:48 ...
Launching unittests with arguments python -m unittest C:\Users\Mikołaj\Desktop\WSI\Task 3\test.py in C:\Users\Mikołaj\Desktop\WSI\Task 3

Ran 11 tests in 2.019s

OK

Process finished with exit code 0
```

3. Tabelki

1) MinMaxAgent vs GreedyAgent

```
Głębokość: 1, Średni czas: 0.00001201, Średnia suma punktów: -3.81, Odchylenie standardowe: 15.23
Głębokość: 2, Średni czas: 0.00002214, Średnia suma punktów: -3.82, Odchylenie standardowe: 15.65
Głębokość: 3, Średni czas: 0.00005000, Średnia suma punktów: -3.10, Odchylenie standardowe: 15.06
Głębokość: 15, Średni czas: 0.03833182, Średnia suma punktów: -1.71, Odchylenie standardowe: 15.12
```

2) MinMaxAgent vs NinjaAgent

```
Głębokość: 1, Średni czas: 0.00001600, Średnia suma punktów: -8.74, Odchylenie standardowe: 13.98
Głębokość: 2, Średni czas: 0.00002308, Średnia suma punktów: -8.79, Odchylenie standardowe: 14.24
Głębokość: 3, Średni czas: 0.00005045, Średnia suma punktów: -8.39, Odchylenie standardowe: 14.13
Głębokość: 15, Średni czas: 0.03970336, Średnia suma punktów: -8.36, Odchylenie standardowe: 14.29
```

3) MinMaxAgent vs MinMaxAgent(15)

```
Głębokość: 1, Średni czas: 0.01645031, Średnia suma punktów: -10.64, Odchylenie standardowe: 14.18
Głębokość: 2, Średni czas: 0.01767329, Średnia suma punktów: -10.79, Odchylenie standardowe: 14.47
Głębokość: 3, Średni czas: 0.01959232, Średnia suma punktów: -9.96, Odchylenie standardowe: 13.96
Głębokość: 15, Średni czas: 0.05827027, Średnia suma punktów: -9.34, Odchylenie standardowe: 14.08
```

Średni czas wykonania gry: Czas wykonania gry może również zależeć od strategii przeciwnika. Jeśli przeciwnik podejmuje szybkie decyzje lub ma mniejszą złożoność obliczeniową, czas wykonania gry może być krótszy. Z drugiej strony, bardziej zaawansowane strategie przeciwnika mogą wymagać dłuższego czasu obliczeń przez MinMaxAgent-a.

Średnia suma punktów: Wyniki uzyskiwane przez MinMaxAgent-a będą miały tendencję do zmniejszania się wraz z rosnącą trudnością przeciwnika. Jeśli przeciwnik jest bardziej wyrafinowany lub stosuje bardziej optymalną strategię, MinMaxAgent może mieć trudności w osiągnięciu wysokich sum punktów.

Odchylenie standardowe: Stabilność wyników MinMaxAgent-a może być również uzależniona od strategii przeciwnika. Jeśli przeciwnik jest bardziej przewidywalny lub stosuje podobne strategie w różnych rozgrywkach, odchylenie standardowe wyników MinMaxAgent-a może być niższe. Z drugiej strony, bardziej zróżnicowane strategie przeciwnika mogą prowadzić do większej zmienności wyników.

Pytanie 1. Jak głębokość drzewa wpływa na wyniki uzyskiwane przez MinMaxAgent-a?

Głębokość drzewa stanów, która jest parametrem algorytmu Minimax, ma istotny wpływ na wyniki uzyskiwane przez agenta w różnych sytuacjach. Wraz ze wzrostem głębokości drzewa, czyli liczby poziomów, na których algorytm rozważa możliwe ruchy, agent może podejmować bardziej przemyślane decyzje, co potencjalnie prowadzi do lepszych wyników w grze. Jednakże wpływ głębokości drzewa nie jest jednoznaczny i może zależeć od wielu czynników, takich jak strategia przeciwnika, złożoność gry, a nawet losowość czy losowe elementy w grze.

Analizując wyniki uzyskane przez MinMaxAgent-a w starciach z różnymi przeciwnikami, można zauważyć kilka trendów. W przypadku rozgrywek z GreedyAgent-em, który stosuje prostą heurystykę i zawsze wybiera ruch prowadzący do maksymalnej bieżącej korzyści, MinMaxAgent może nie mieć potrzeby głębokiego przeszukiwania drzewa stanów. W rezultacie, nawet dla niewielkich głębokości drzewa, MinMaxAgent może uzyskiwać porównywalne wyniki do tych uzyskiwanych dla większych głębokości. Z kolei w przypadku bardziej zaawansowanych przeciwników, takich jak NinjaAgent, który może stosować bardziej wyrafinowane strategie, głębokość drzewa może mieć większe znaczenie. MinMaxAgent potrzebuje bardziej dogłębnej analizy możliwych ruchów, aby unikać pułapek i prowadzić grę w kierunku optymalnego rozwiązania.

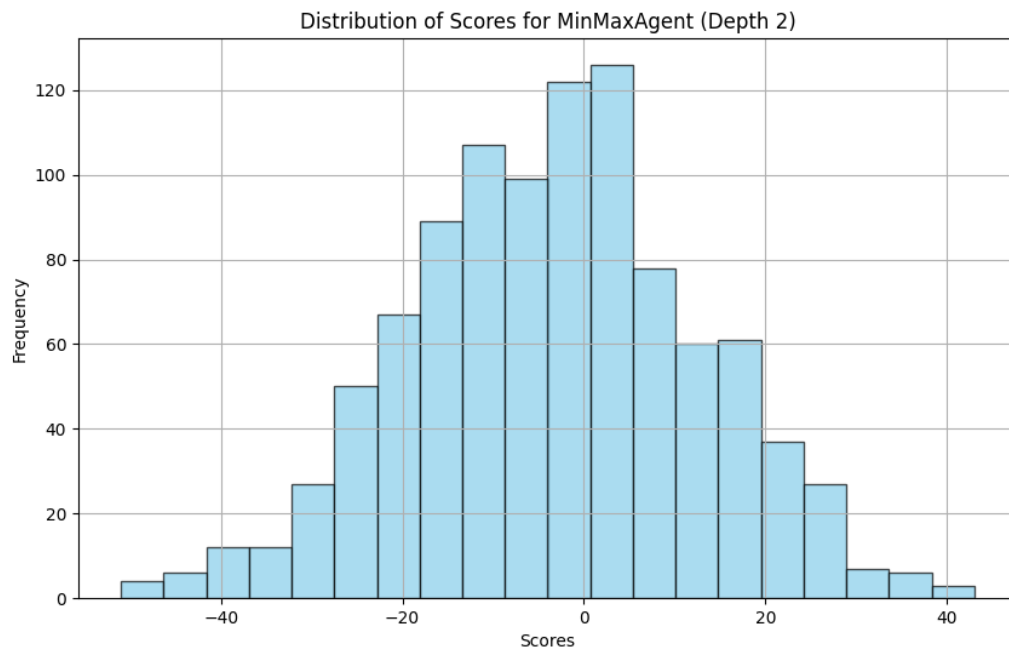
Odchylenie standardowe uzyskanych wyników stanowi ważny wskaźnik stabilności strategii MinMaxAgent-a. Im niższe odchylenie standardowe, tym bardziej stabilne są wyniki uzyskiwane przez agenta w różnych rozgrywkach. Jednakże warto zauważyć, że stabilność wyników może również zależeć od strategii przeciwnika. Jeśli przeciwnik stosuje bardziej zróżnicowane strategie lub podejmuje decyzje w sposób mniej przewidywalny, to odchylenie standardowe wyników może wzrosnąć, nawet dla tego samego poziomu głębokości drzewa.

Średni czas wykonania gry przez MinMaxAgent-a również może być uzależniony od głębokości drzewa oraz od strategii przeciwnika. W przypadku bardziej złożonych strategii przeciwnika, potrzebna jest większa liczba obliczeń, co może prowadzić do dłuższego czasu wykonania gry. Jednakże nawet dla tego samego przeciwnika, zwiększenie głębokości drzewa może znacznie zwiększyć czas obliczeń, co może być niepożądane w praktycznych zastosowaniach.

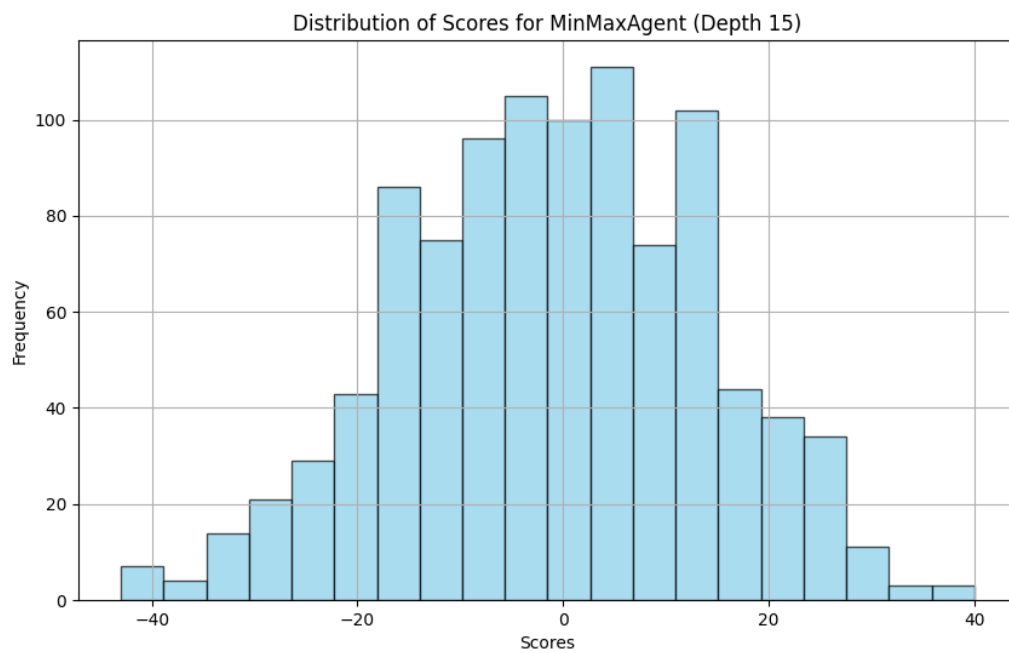
Podsumowując, wpływ głębokości drzewa na wyniki uzyskiwane przez MinMaxAgent-a jest złożony i zależy od wielu czynników, takich jak strategia przeciwnika, stabilność wyników oraz czas obliczeń. Dlatego też dobranie optymalnej głębokości drzewa wymaga uwzględnienia tych czynników i dostosowania się do konkretnych warunków gry.

4. Histogramy

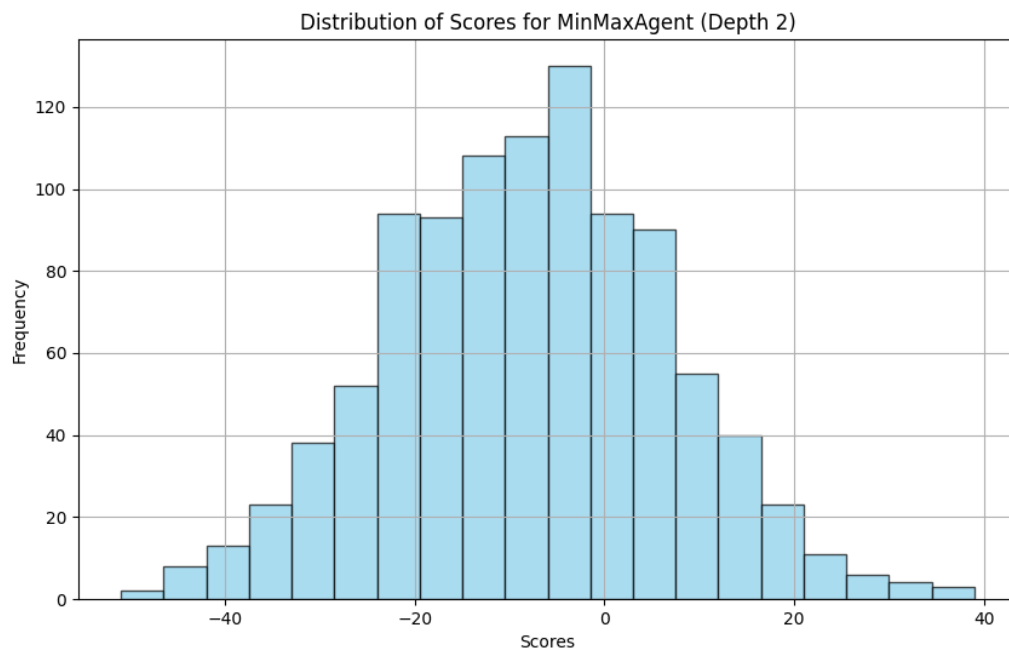
1) Test dla głębokości drzewa 2 MinMaxAgent vs GreedyAgent



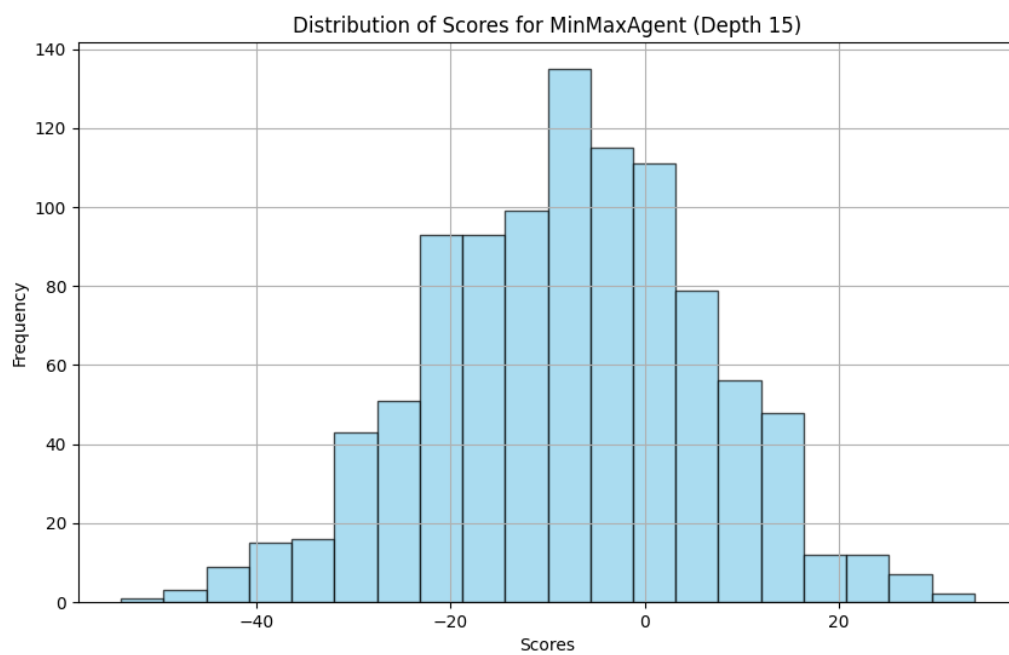
2) Test dla głębokości drzewa 15 MinMaxAgent vs GreedyAgent



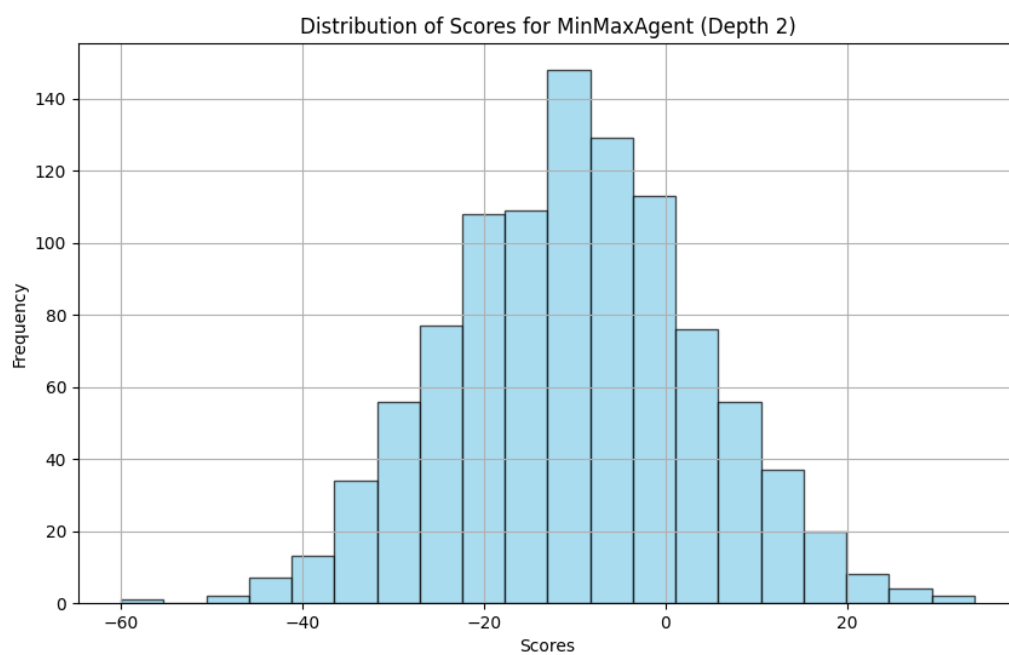
3) Test dla głębokości drzewa 2 MinMaxAgent vs NinjaAgent



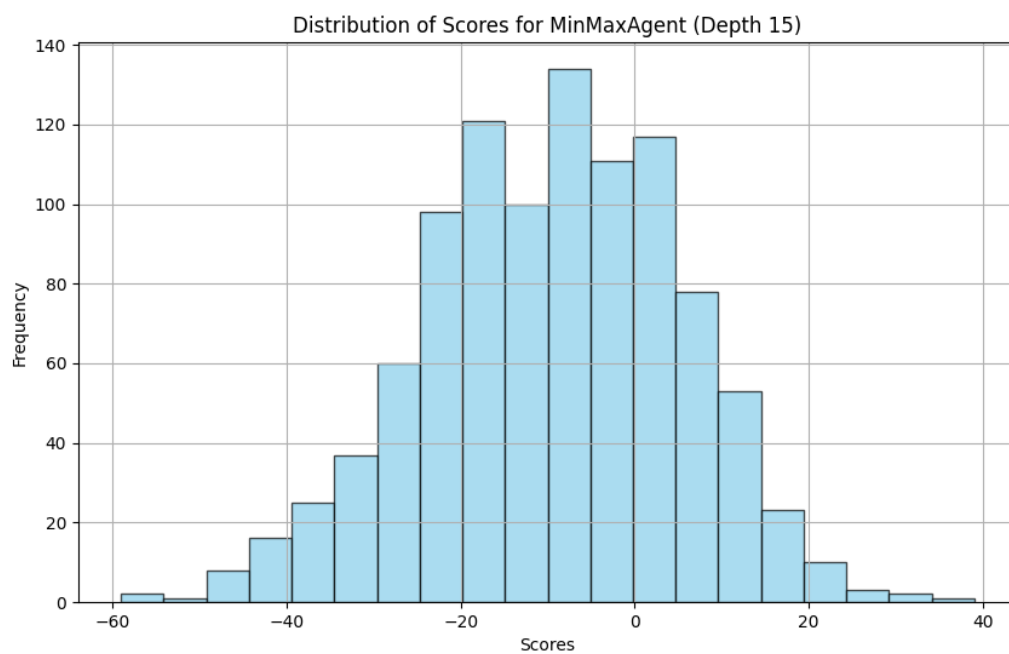
4) Test dla głębokości drzewa 15 MinMaxAgent vs NinjaAgent



5) Test dla głębokości drzewa 2 MinMaxAgent vs MinMaxAgent(15)



6) Test dla głębokości drzewa 15 MinMaxAgent vs MinMaxAgent(15)



5. Odpowiedzi na pytania

Pytanie 2. Jaki jest jej rozkład? (Dobrym sposobem na wizualizację rozkładu jest stworzenie histogramu)

Rozkład sum punktów uzyskiwanych przez MinMaxAgent-a dla głębokości drzewa 2 oraz 15 można uznać za przybliżony do rozkładu normalnego. W przypadku obu głębokości drzewa, większość wyników koncentruje się wokół pewnej średniej wartości, a odchylenia od tej wartości są coraz rzadsze, co odpowiada charakterystyce rozkładu normalnego.

Dla głębokości drzewa 2, histogram sum punktów wykazuje rozkład zbliżony do normalnego, gdzie większość wyników koncentruje się wokół pewnej średniej wartości, a odchylenia od tej wartości są symetryczne.

Natomiast dla głębokości drzewa 15 również można zauważyć, że histogram sum punktów ma kształt zbliżony do rozkładu normalnego, gdzie większość wyników koncentruje się wokół pewnej średniej wartości, a odchylenia od tej wartości są coraz rzadsze w obu kierunkach.

Pytanie 3. Jak na podstawie histogramu stwierdzić, który agent jest lepszy? Jak przyspieszyć działanie MinMaxAgent-a?

Na podstawie histogramu możemy ocenić, który agent jest lepszy, analizując średnie wartości oraz rozrzut wyników dla każdego agenta.

Jeśli histogram sum punktów dla jednego z agentów jest przesunięty w kierunku wyższych wartości i ma mniejszy rozrzut niż histogram dla drugiego agenta, możemy uznać ten agent za lepszego. Oznacza to, że ten agent osiąga średnio wyższe wyniki oraz jest bardziej skuteczny w maksymalizacji sumy punktów.

Jeśli chodzi o przyspieszenie działania MinMaxAgent-a, istnieje kilka metod:

Przycinanie alfa-beta: Implementacja algorytmu przycinania alfa-beta pozwala na przyspieszenie przeszukiwania drzewa gry poprzez eliminację gałęzi, które nie będą miały wpływu na wynik końcowy. Dzięki temu można uniknąć zbędnych obliczeń i skrócić czas działania algorytmu.

Heurystyki: Zastosowanie heurystyk do oceny stanów gry może pomóc w przyspieszeniu MinMaxAgent-a poprzez zmniejszenie głębokości przeszukiwania drzewa. Heurystyki pozwalają na szybsze podejmowanie decyzji na podstawie oceny stanu gry bez konieczności przeszukiwania wszystkich możliwych ruchów.

Ograniczenie głębokości przeszukiwania: Ograniczenie maksymalnej głębokości przeszukiwania drzewa gry może znacząco skrócić czas działania MinMaxAgent-a kosztem pewnej straty dokładności w ocenie stanów gry. W praktyce może okazać się, że ograniczenie głębokości nie wpłynie znacząco na skuteczność agenta, a jednocześnie pozwoli znacznie przyspieszyć jego działanie.

Pytanie 4. Jak poprawić jego działanie dla drzewa przeszukiwań o głębokości 1?

Dynamiczne dostosowywanie strategii: Agent może być wyposażony w zdolność do adaptacji swojej strategii w zależności od aktualnej sytuacji w grze. Na przykład, jeśli agent zdobywa przewagę, może zmienić strategię na bardziej agresywną lub zachować ostrożność w przypadku przewagi przeciwnika.

Uczenie maszynowe: Zamiast polegać wyłącznie na z góry zdefiniowanych regułach oceny pozycji, można wykorzystać uczenie maszynowe do trenowania modelu na podstawie danych historycznych lub symulacji gier. Model taki może nauczyć się oceniać pozycje na podstawie doświadczenia z gier i dostosowywać się do różnych sytuacji.