Zadanie 6 - Uczenie się ze wzmocnieniem

1. Implementacja

a. Algorytm Q-learning

Powyższy algorytm zaimplementowany jest jako osobna metoda przyjmująca podaną listę parametrów:

- *environment* - środowisko, eksperymenty przeprowadzane są w środowisku Taxi
- *action_type* - sposób wyboru akcji

oraz hiper parametrów:

- *e_max, gamma, exploration_rate, learning_rate*

których wpływ na wyniki działania algorytmu jest celem zadania

b. Sposób wyboru akcji

Za wybór akcji odpowiedzialna jest metoda *choose_action*. Dostępne są dwie strategie wyboru:

- e-zachłanna - z prawdopodobieństwem e (zmienna exploration_rate) wybieramy akcję losową, z 1-e akcję zachłanną, czyli tą z największą wartością Q
- strategia oparta na rozkładzie Boltzmanna - wybieramy daną akcję z prawdopodobieństwem $\pi(x, a) = \frac{exp(Q(x,a)/T)}{\sum_{B=1}^{N} exp(Q(x,b)/T)}$, gdzie $T = exp(-t)$

c. Testowanie

Do testowania działania algorytmu wykorzystywana jest metoda test, która przyjmuje trzy parametry:

- *Q* - macierz wartości zwrócona w wyniku działania metody q_learning
- *t_max* - maksymalna ilość iteracji
- *environment* - środowisko

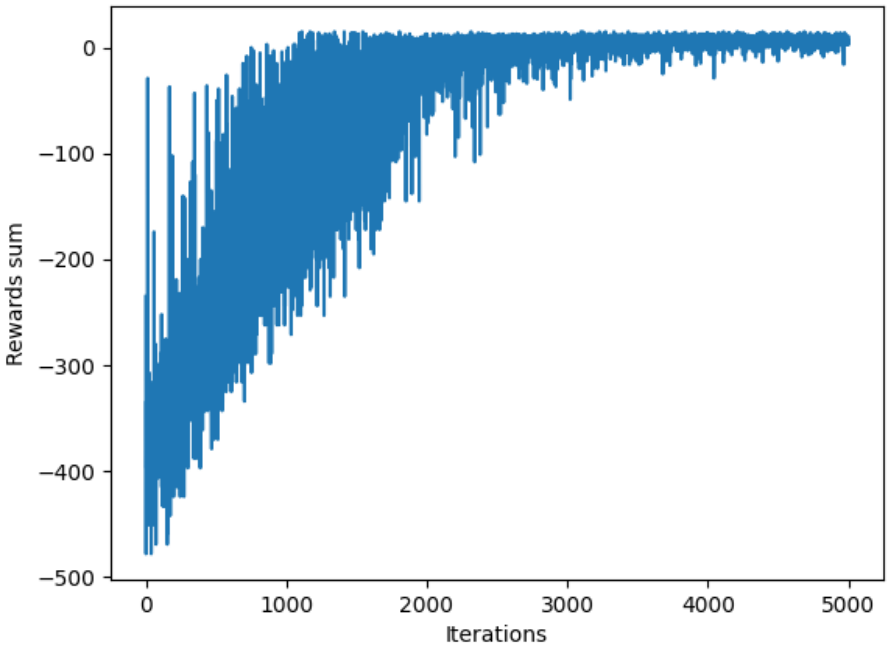
Metoda przeprowadza symulację “przejazdu taksówki”. W momencie zakończenia przejazdu zwracana jest przez nią wartość nagrody

2. Testowanie

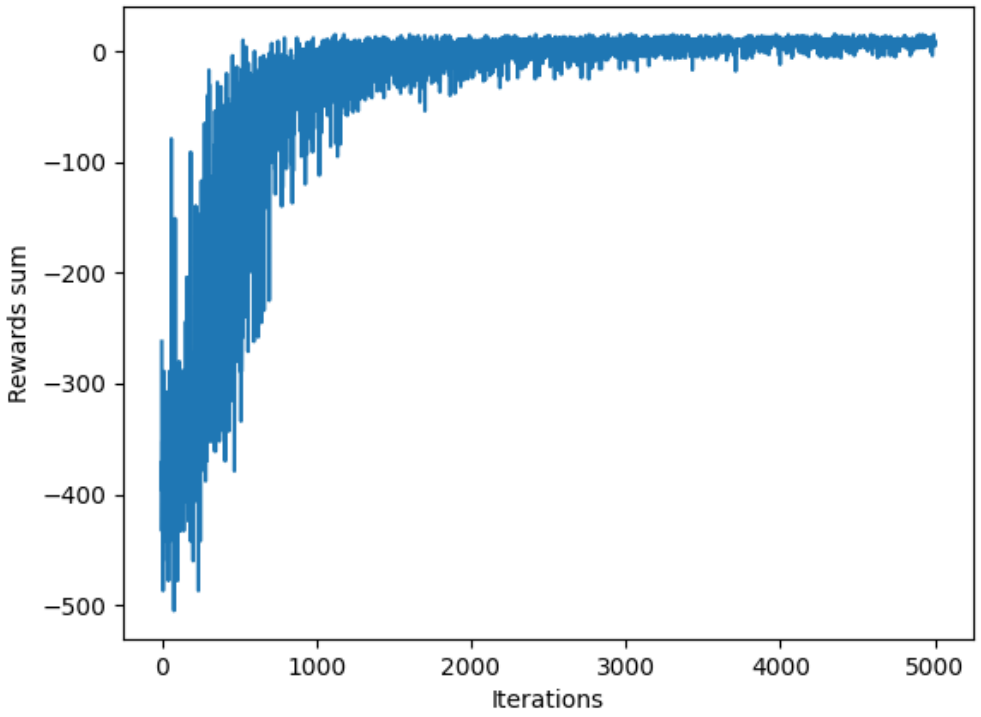
Jakość działania algorytmu mierzona jest przez wartość sumy nagrody dla danego epizodu

a. Strategia e-zachłanna

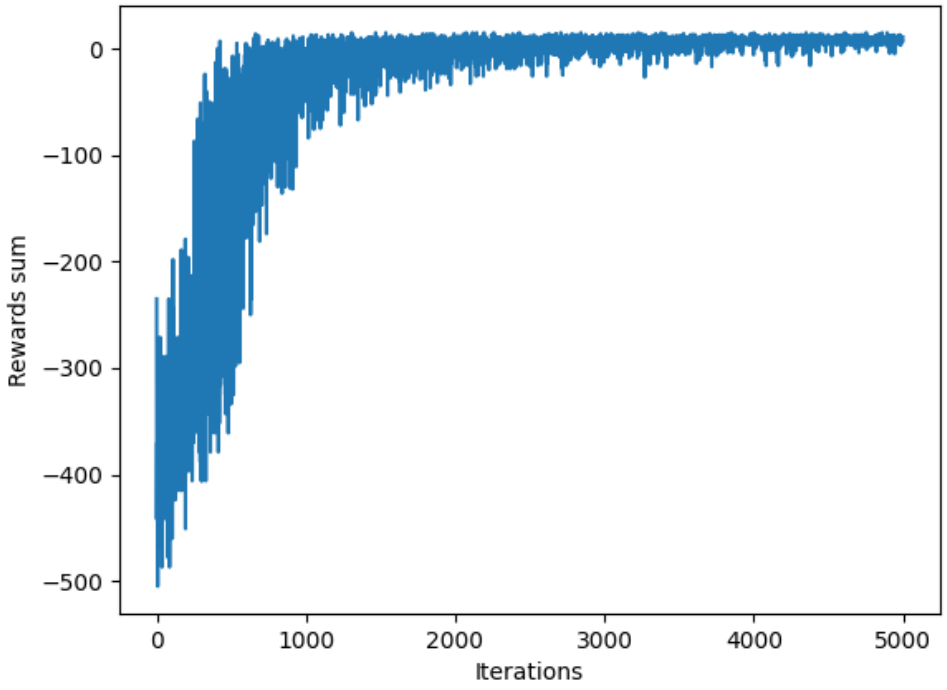
e_max = 5000, gamma = 0.7, exploration_rate = 0.1, learning_rate = 0.1



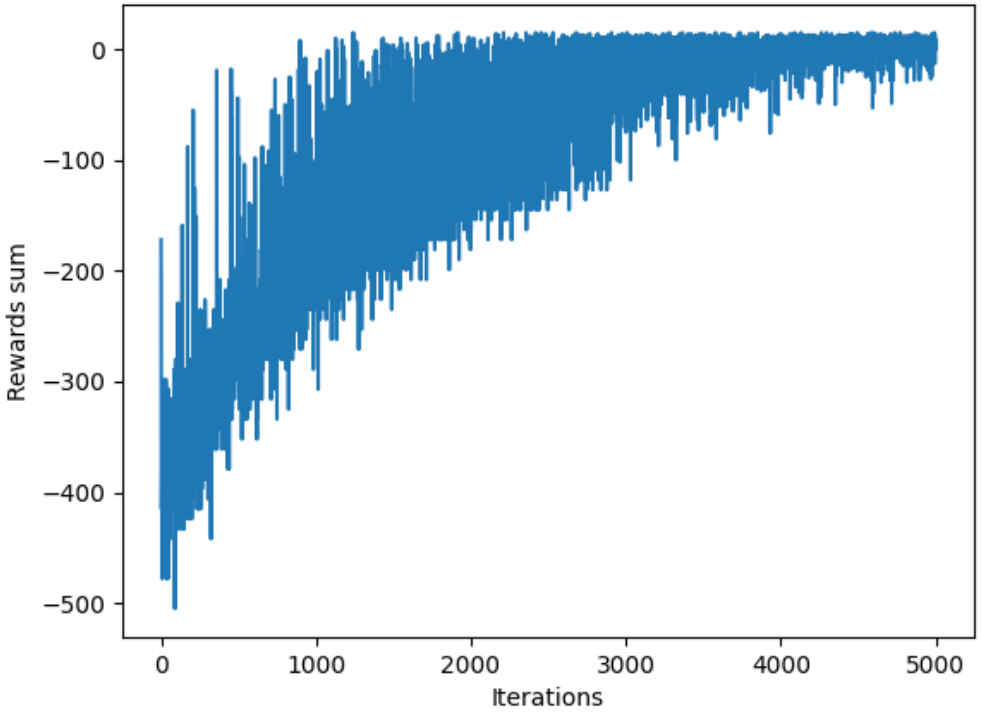
e_max = 5000, gamma = 0.7, exploration_rate = 0.1, learning_rate = 0.5



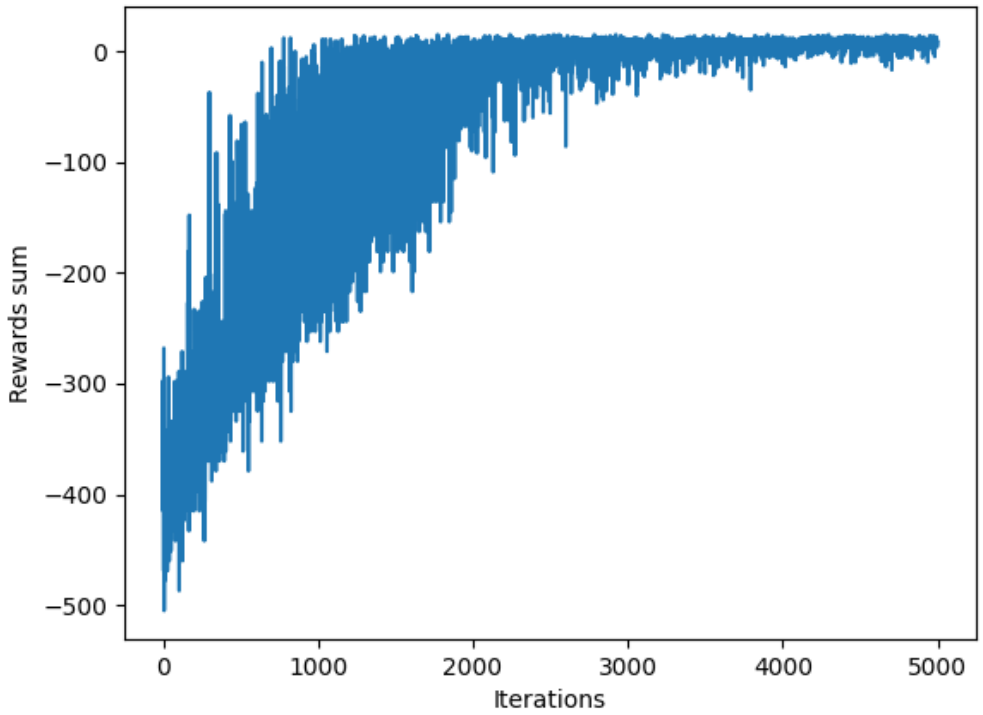
e_max = 5000, gamma = 0.7, exploration_rate = 0.5, learning_rate = 0.5



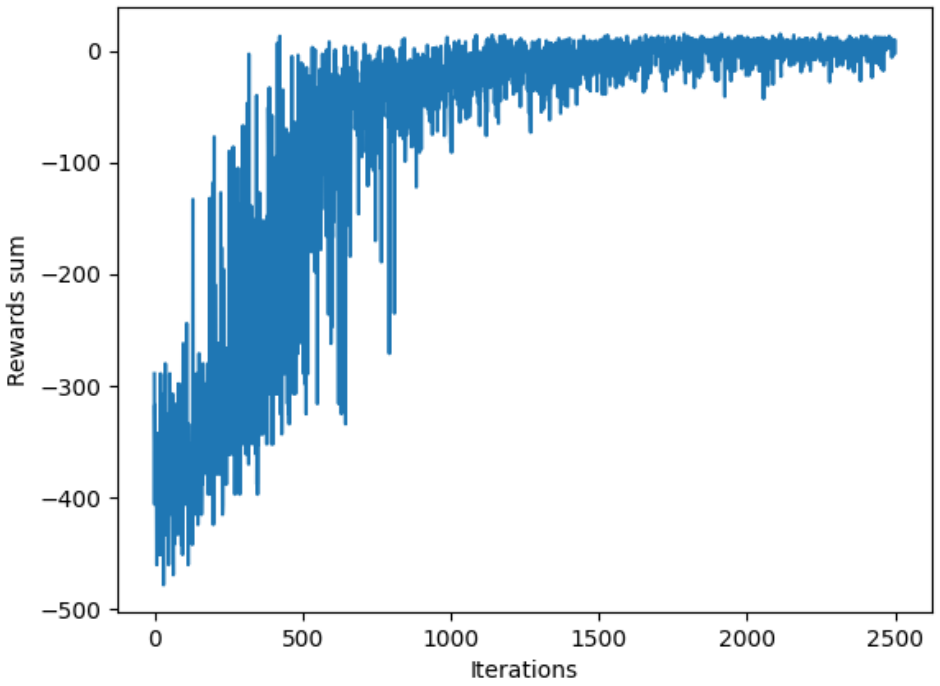
e_max = 5000, gamma = 0.3, exploration_rate = 0.1, learning_rate = 0.1



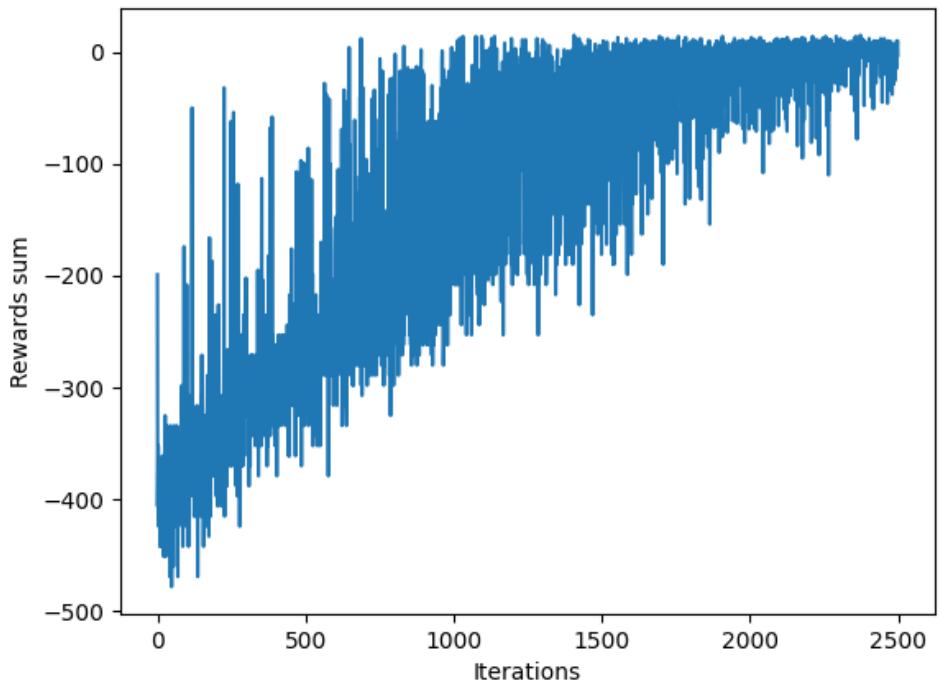
e_max = 5000, gamma = 0.7, exploration_rate = 0.5, learning_rate = 0.1



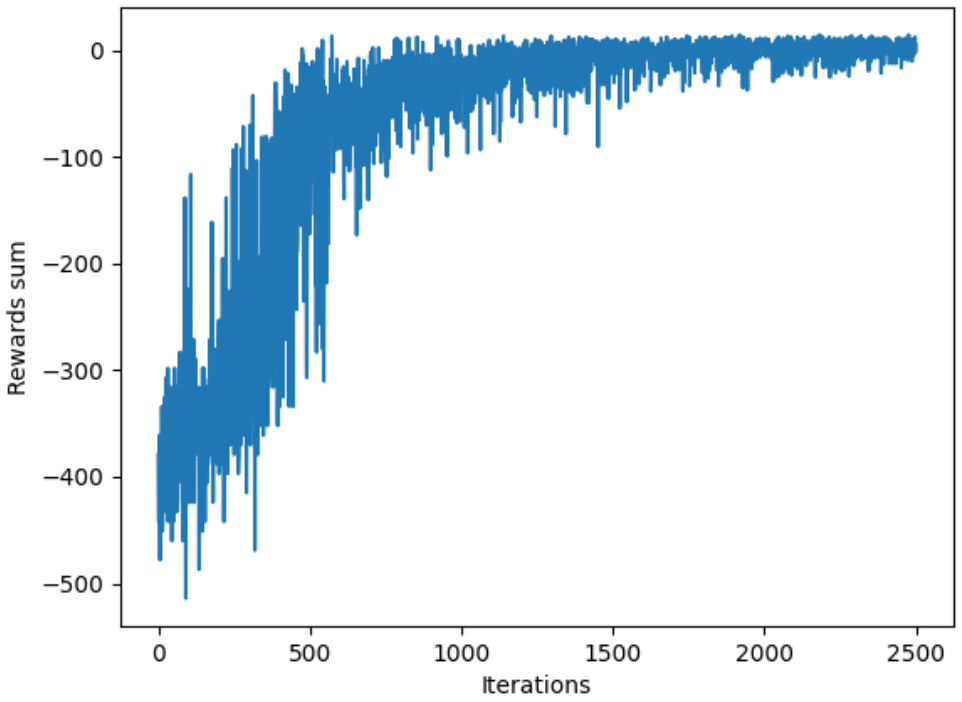
e_max = 2500, gamma = 0.7, exploration_rate = 0.5, learning_rate = 0.5



e_max = 2500, gamma = 0.7, exploration_rate = 0.1, learning_rate = 0.1



e_max = 2500, gamma = 0.7, exploration_rate = 1.0, learning_rate = 1.0



Wnioski:

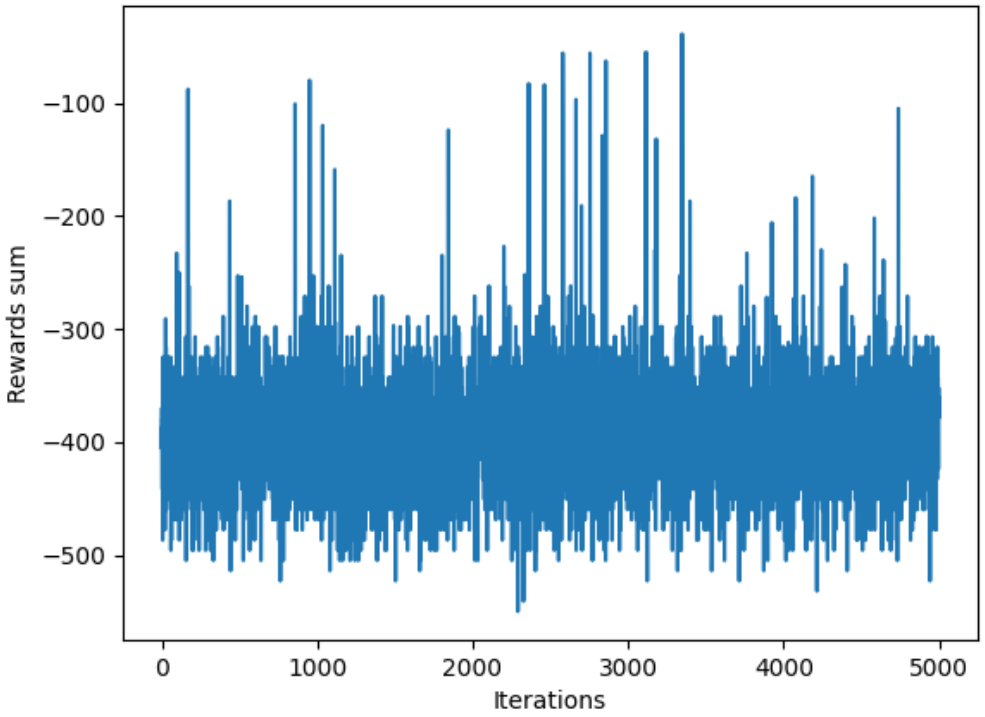
Najlepsze wyniki algorytm osiągnął dla parametrów:
e_max = 5000, gamma = 0.7, exploration_rate = 0.5, learning_rate = 0.5
Wykres dla powyższych danych nie ma zbyt dużych wahań, zdecydowanie najrzadziej występuje w jego przypadku sytuacja, gdy dodatnie wartości sumy mieszają się często z wartościami ujemnymi
Natomiast najgorsze dla parametrów:
e_max = 2500, gamma = 0.7, exploration_rate = 0.1, learning_rate = 0.1
Wyniki przedstawione na wykresie są skrajnie nieprzewidywalne, ujemne wartości sumy mieszają się z dodatnimi praktycznie na całej długości osi X (liczby iteracji)

Analizując pozostałe wykresy można zauważyć, że zwiększanie parametrów ma pozytywny wpływ na wyniki algorytmu. Największy wpływ wywiera zmiana zmiennej e_max. Kiedy zwiększymy exploration_rate oraz learning_rate do wartości 1, algorytm szybko zacznie przyjmować wartości bliskie zeru, jednak jego wahania są zauważalnie większe niż dla wyżej wymienionych najlepszych parametrów. Nasuwa to następujący wniosek - dobranie

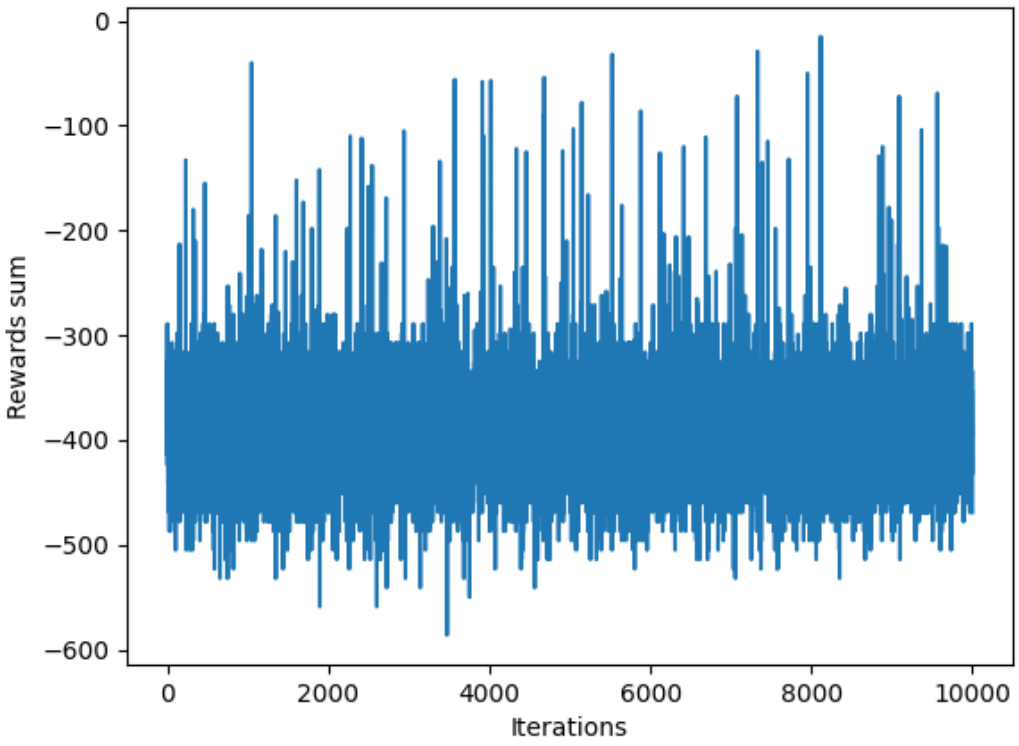
maksymalnej wartości tych parametrów wcale nie gwarantuje najbardziej zadowalających wyników, optymalne wartości oscylują przy 0.5

b. Strategia Boltzmanna

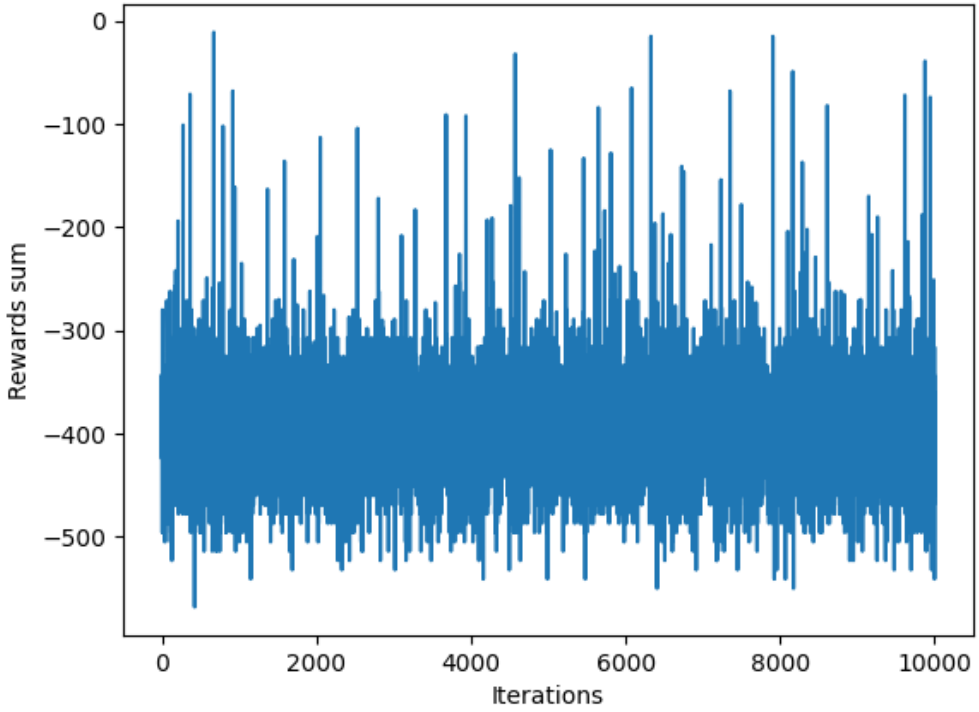
e_max = 5000, gamma = 0.7, learning_rate = 0.5



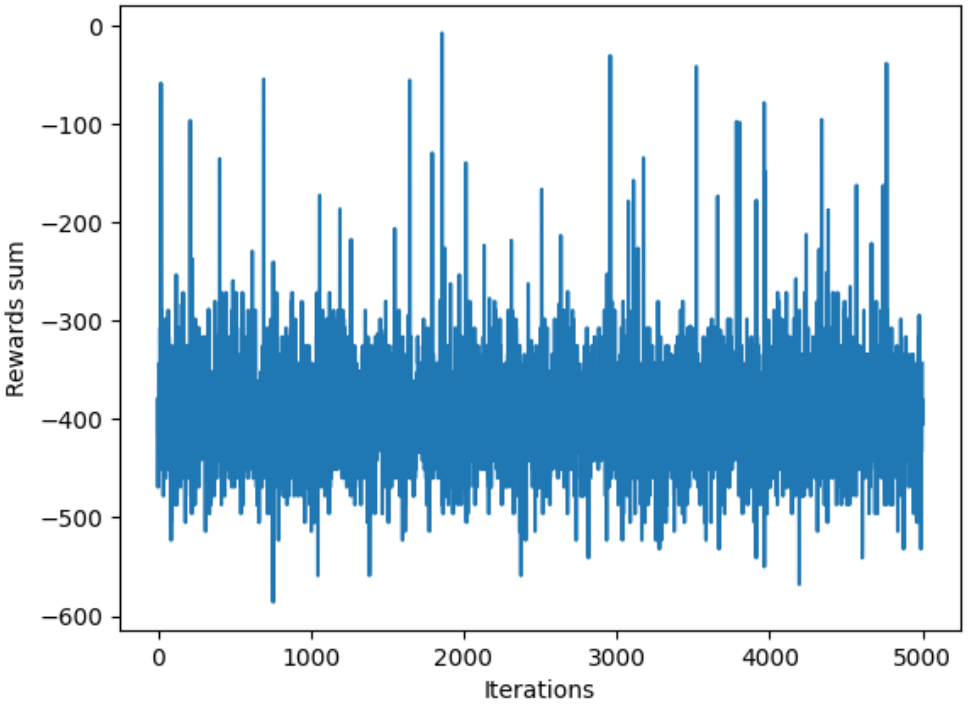
e_max = 10000, gamma = 0.7, learning_rate = 0.5



e_max = 10000, gamma = 0.7, learning_rate = 1.0



e_max = 5000, gamma = 0.3, learning_rate = 0.5



Wyniki działania algorytmu przy sposobie Boltzmanna są nieregularne. W każdym przypadku doboru parametrów suma oscyluje w okolicy 400, zależnie od wielkości hiper parametrów zmienia się częstotliwość rezultatów, w których suma jest bliska lub większa od zera. Zapewne wynika to z błędu implementacyjnego w strategii wyboru akcji

Paweł Rogóż, 318714