WSI - Q-learning

Jan Szymczak

Maj 2024

1 Cel ćwiczenia

Cel ćwiczenia to implementacja algorytmu Q-learning. Następnie wykorzystany on został do nauczenia agenta w problemie Taxi: `https://gymnasium.farama.org/environments/toy_text/taxi/`. Implementacja algorytmu jest uniwersalna, metody uczące i ukazujące wyniki przyjmują jako argumenty odpowiednie funkcje.

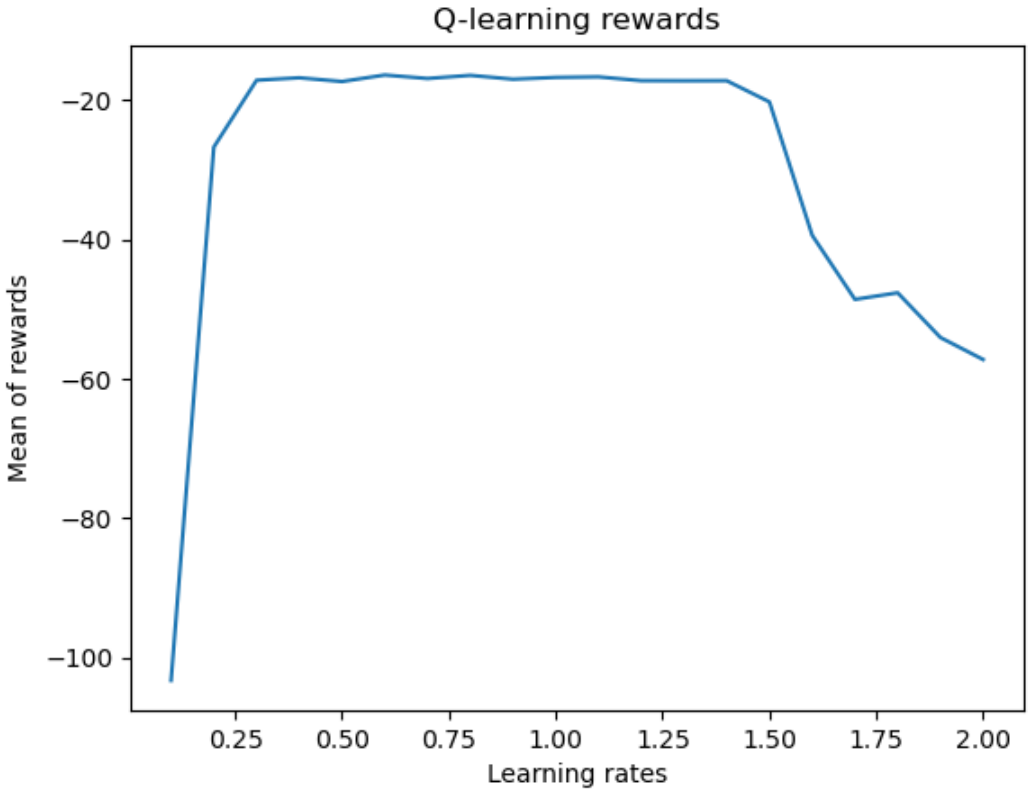
1.1 Parametry

- Learning rate: wpływa na szybkość uczenia się algorytmu. Podobnie jak w poprzednich zadanaich - zbyt niska wartość powoduje, że algorytm do nauczenia się potrzebuje więcej iteracji, a zbyt wysoka powoduje brak poprawnego działania programu i brak znalezienia żadnego dobrego rozwiązania. Innymi słowy, parametr ten określa, jak nowe informacje napływające do agenta mają zastąpić stare. Przy wartości parametru 1 agent bierze pod uwagę tylko najnowsze informacje.
- Discount rate: wpływa na wagę możliwych nagród w przyszłości. Czym wyższy parametr, tym agent będzie bardziej priorytetyzował akcje, które mają przynieść większe nagrody w przyszłości. Zbyt duża wartość parametru wypacza działanie programu, zbyt mała z kolei powoduje, że agent nie będzie w stanie "dostrzec drogi" do optymalnego rozwiązania.
- Epsilon i decay: epsilon wpływa na to jak długo agent ma eksplorować - wykonywać losowe ruchy, aby "poznać" środowisko. Parametr ten maleje z każdym epizodem, wykładniczo zgodnie z parametrem decay. Parametry te przyspieszają proces uczenia, ponieważ pozwalają agentowi na początku działania programu zebrać jak najwięcej informacji, jednak nie są one kluczowe w kontekście całego działania tak jak wyżej wymienione.
- Epizodes: ilość epizodów - pełnych prób rozwiązania problemu przez agenta. Ich ilość nie może być zbyt mała, aby agent zdążył nauczyć się.
- Max steps: maksymalna ilość ruchów jakie agent może wykonać w trakcie jednego epizodu. Zbyt mała wartość może prowadzić, że agent nawet nie będzie miał okazji do dojścia do optymalnego rozwiązania. Zwiększenie tej wartości naturalnie wydłuża działanie algorytmu.

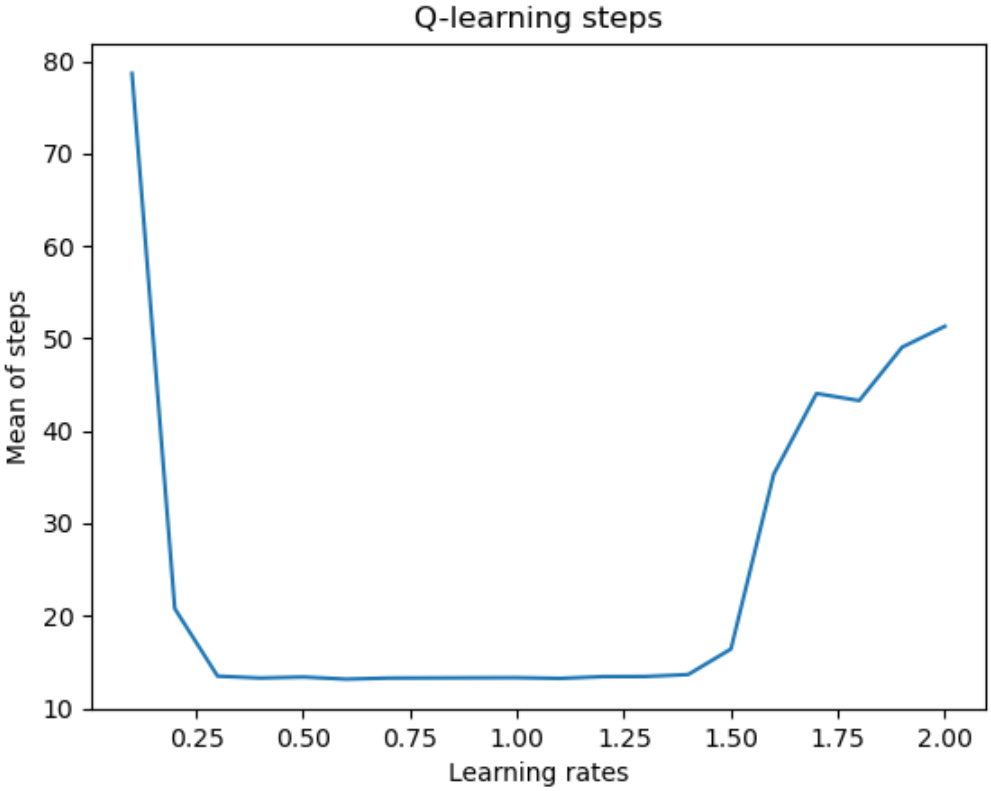
2 Wyniki eksperymentów

2.1 Wpływ learning rate

Aby zbadać wpływ tego parametru trenowany był agent z każdą wartością learning rate z przedziału $\langle 0.1; 2.0 \rangle$ ze skokiem 0.1. Wartość pozostałych parametrów to: discount rate = 0.8, epsilon = 1, decay = 0.005, episodes = 1000, max steps = 100. Tak prezentują się wyniki:



Rysunek 1: Wpływ learning rate na średnią z nagród uzyskanych przez agenta

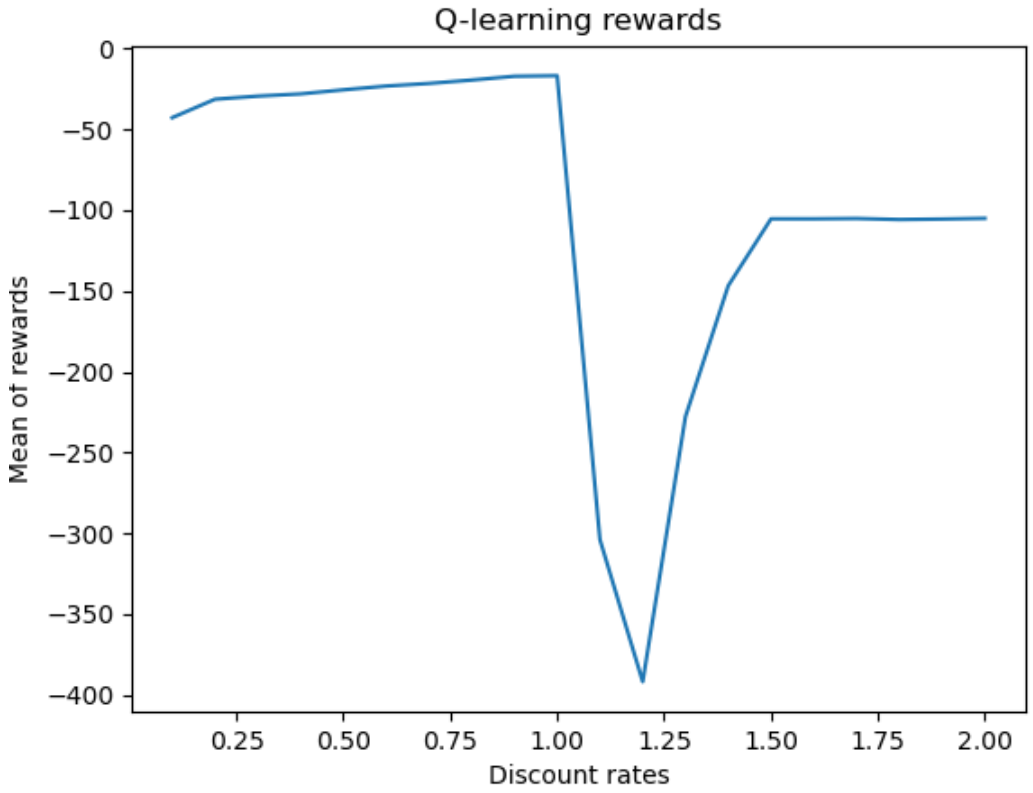


Rysunek 2: Wpływ learning rate na średnią z całkowitych kroków wykonanych przez agenta

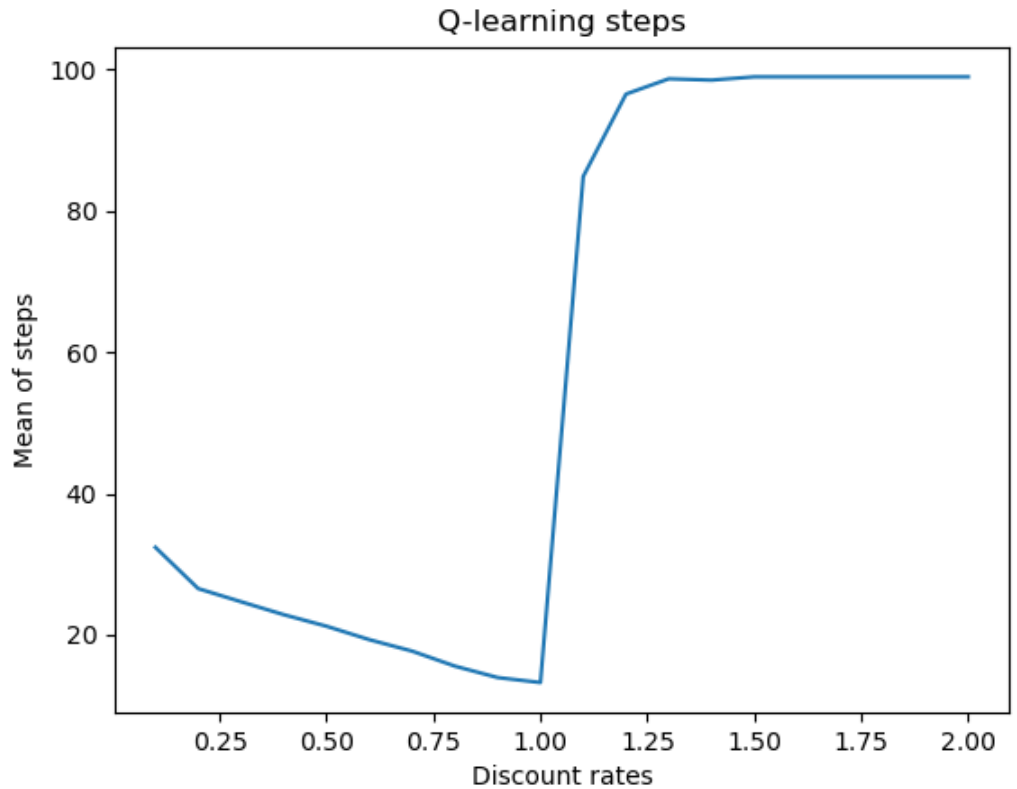
Co prawda, wartość parametru 1 już oznacza, że agent bierze pod uwagę tylko najnowsze informacje, jednak widać, że akurat w problemie Taxi i tak możliwe jest wtedy znalezienie optymalnego rozwiązania. Tak naprawdę każda wartość parametru z przedziału $\langle 0.3; 1.2\rangle$ daje bardzo podobne, dobre rozwiązania - średnia z nagród jest wysoka, a średnia ilość wykonanych kroków niska (czyli agent zdołał szybko się nauczyć i wykonywał już dobre, optymalne ruchy).

2.2 Wpływ discount rate

Podobnie jak w poprzedniej sekcji, badany parametr był brany z tego samego przedziału co learning rate, przy tych samych parametrach, a wartość learnin rate wynosiła 0.9.



Rysunek 3: Wpływ discount rate na średnią z nagród uzyskanych przez agenta



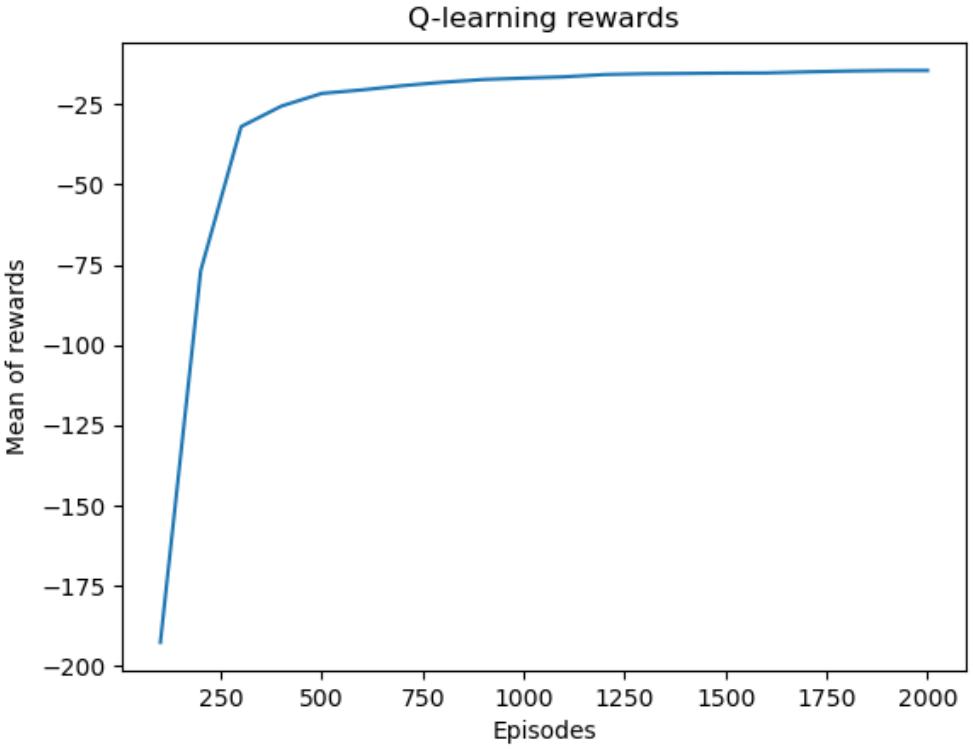
Rysunek 4: Wpływ discount rate na średnią z całkowitych kroków wykonanych przez agenta

Zgodnie z oczekiwaniami, skuteczność algorytmu rośnie wraz ze wzrostem wartości parametru, aż do wartości 1 - wartość powyżej jedności wypacza działanie algorytmu. Co ciekawe, optymalną wartością parametru jest

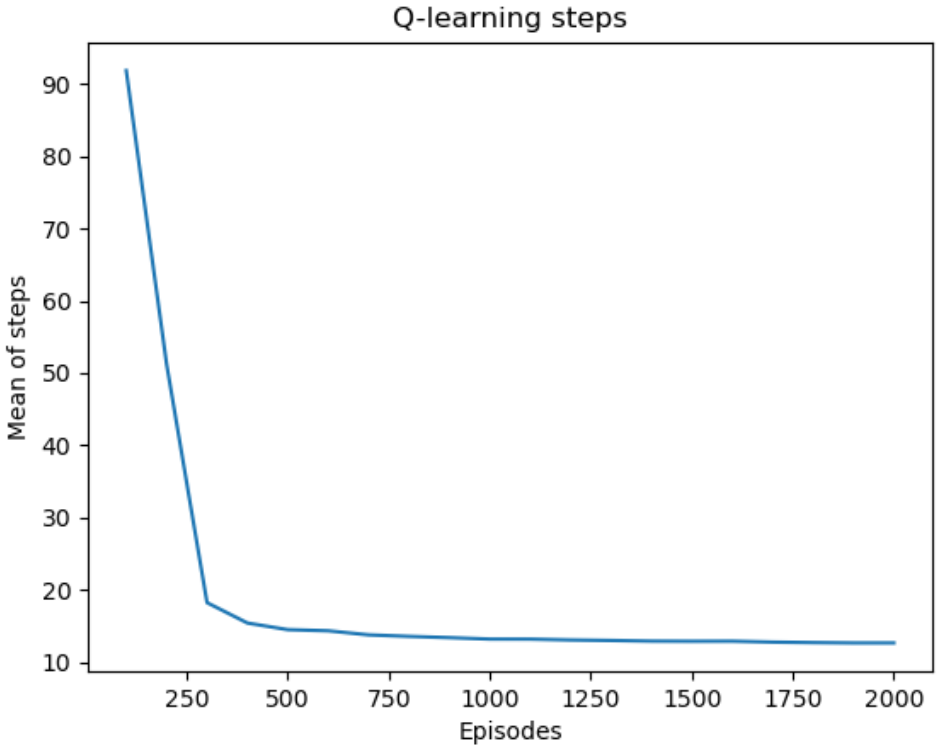
aż 1, co oznacza, że agent będzie bardzo priorytetyzował potencjalne przyszłe nagrody. Prawdopodobnie jest to w porządku dla tak prostego problemu, jakim jest problem Taxi - w bardziej skomplikowanych problemach ta wartość byłaby zbyt duża.

2.3 Wpływ episodes

Wartość parametru zmieniała się z $\langle 100, 2000 \rangle$ ze skokiem 100, discount rate został ustalony na 0.8, reszta parametrów bez zmian.



Rysunek 5: Wpływ episodes na średnią z nagród uzyskanych przez agenta



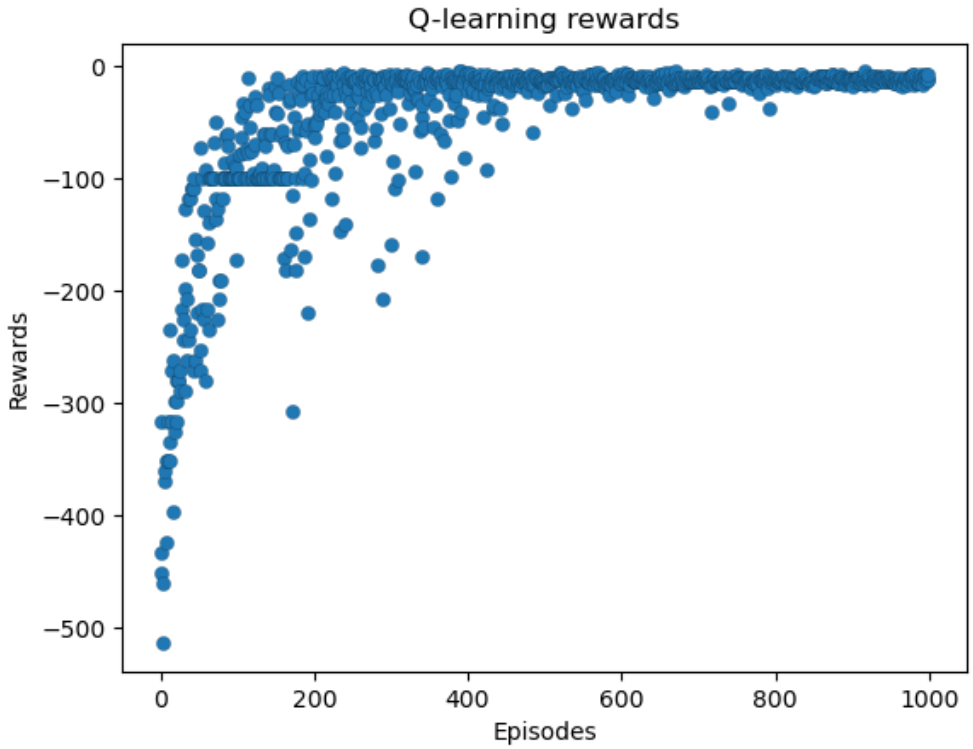
Rysunek 6: Wpływ episodes na średnią z całkowitych kroków wykonanych przez agenta

Czym większa wartość parametru tym lepsze rozwiązania, jednak na dobrą sprawę dobre wyniki zaczynają

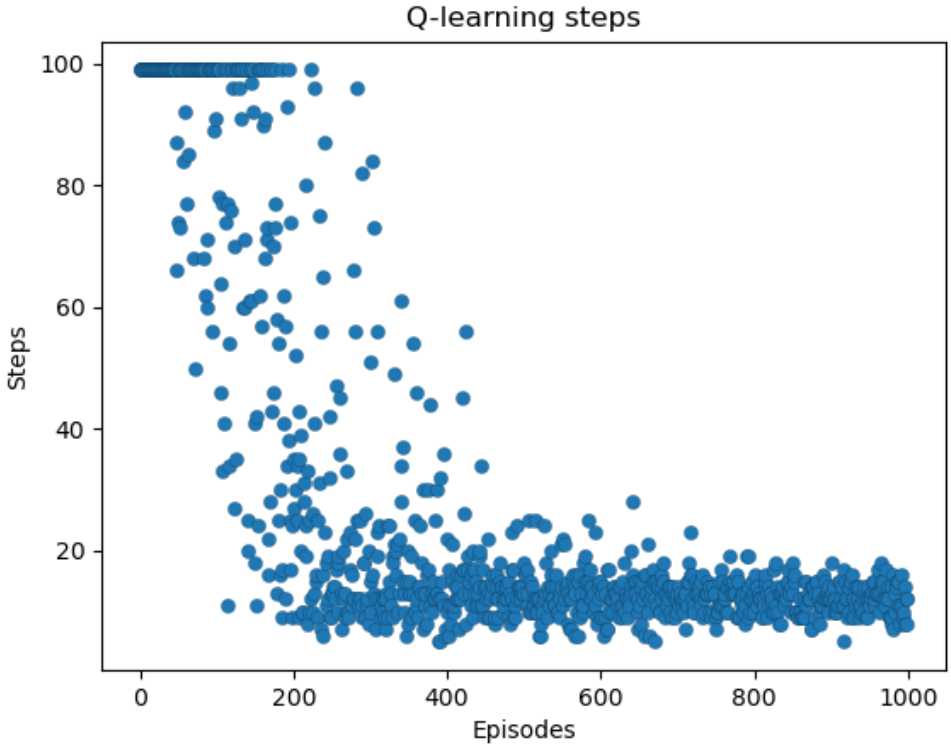
się już od wartości 400, jak widać przy takich parametrach agent już zdąży się nauczyć problemu. Zwykle minimalna wartość jest większa, ale ponownie - problem Taxi nie jest skomplikowany.

2.4 Ostateczny test

Ostateczny test został przeprowadzony z parametrami: learning rate = 0.9, discount rate = 1.0, epsilon = 1.0, decay = 0.005, episodes = 1000, max steps = 100. Tak wyglądają wykresy zmian kolejno nagród oraz ilości kroków w trakcie kolejnych epizodów:



Rysunek 7: Zmiany nagród względem kolejnych epizodów dla ostatecznego testu



Rysunek 8: Zmiany ilości kroków względem kolejnych epizodów dla ostatecznego testu

Widać, że w trakcie pierwszych 200 epizodów agent głównie się uczył, wykonywał losowe ruchy, zbierał

informacje. Wtedy też wartość epsilon była wysoka. Już po około 500 epizodach wartość końcowej nagrody utrzymywała się wysoko, a ilość kroków zmalała poniżej 20. Przy końcu działania programu agent osiągał już tylko najwyższe nagrody przy najmniejszej liczbie kroków. Tak nauczony agent faktycznie osiąga ostateczną nagrodę na poziomie ok. 8, 9, 10. W tym problemie każda dodatnia wartość tak naprawdę oznacza dobre rozwiązanie, to wynika z natury przyznawania nagród: -1 za każdy ruch, -10 za "nielegalny ruch" i +20 za poprawne dostarczenie pasażera. Wartość ostatecznej nagrody więc zmienia się zależnie od początkowego stanu - początkowej pozycji pasażera, taksówki i hotelu. Niemniej jednak agent z pewnością się uczy, a wizualizację nauczonego agenta można zobaczyć wywołując metodę single_run() z pliku z testami (aby wizualizacja ruszyła należy wpisać "y" w terminalu - napisałem taką prostą walidację, ponieważ okienko z nią wyskakiwało w połowie na jednym monitorze, w połowie na drugim i często nie dało się nawet zobaczyć poprawnego rozwiązania zanim nie przeciągnąłem okienka w wygodne miejsce).