



WYŻSZA SZKOŁA ZARZĄDZANIA I BANKOWOŚCI W KRAKOWIE

Wydział Nauk Stosowanych

KIERUNEK: Informatyka

ZAKRES KSZTAŁCENIA: Bazy danych

PRACA DYPLOMOWA

Robert Jan Smoter

*Symulacja ruchu drogowego z zastosowaniem algorytmów
optymalizacji sterowania sygnalizacją świetlną*

Promotor: dr hab. inż. Rafał Dreżewski

Recenzent: dr inż. Robert Marcjan

WSTĘP, MOTYWACJA, CEL I ZAKRES PRACY

WSTĘP I MOTYWACJA

Coraz większe natężenie ruchu drogowego prowadzi do:

- wydłużenia czasów przejazdu,
- wzrostu emisji spalin i hałasu,
- zatorów oraz strat ekonomicznych i społecznych.

Tradycyjne systemy sygnalizacji (stałe cykle, prognozy) nie radzą sobie ze zmiennością warunków drogowych.

CEL PRACY

Zapoznanie się z działaniem algorytmu Actor-Critic i jego praktyczna implementacja w środowisku symulacyjnym SUMO.

Sprawdzenie, czy agent uczący się ze wzmocnieniem może skutecznie sterować sygnalizacją świetlną – w warunkach laboratoryjnych.

ZAKRES METODYCZNY

Przegląd literatury – analiza istniejących rozwiązań z zakresu ITS i RL.

Implementacja algorytmu – opracowanie modelu Actor-Critic w Pythonie z wykorzystaniem TensorFlow/Keras.

Opracowanie środowiska – integracja z symulatorem SUMO przez interfejs TraCI.

Analiza danych i porównanie wyników – ocena skuteczności agenta względem podejścia statoczasowego (czas oczekiwania, przepustowość, zatrzymania).

OCZEKIWANE REZULTATY I KORZYŚCI

Pokazanie, że agent RL potrafi autonomicznie dostosowywać plan świateł do aktualnego natężenia ruchu.

Wskazanie potencjału uczenia maszynowego w sterowaniu ruchem – nawet w prostej, nieskomercjalizowanej formie.



INTELIGENTNE SYSTEMY TRANSPORTOWE (ITS)

ITS to nowoczesne rozwiązania technologiczne wspierające zarządzanie ruchem drogowym oraz poprawę bezpieczeństwa i efektywności transportu.

Ich działanie opiera się na integracji technologii informatycznych, komunikacyjnych i automatyki, co umożliwia dynamiczne reagowanie na zmieniające się warunki ruchu w czasie rzeczywistym.

Kluczowe elementy ITS to:

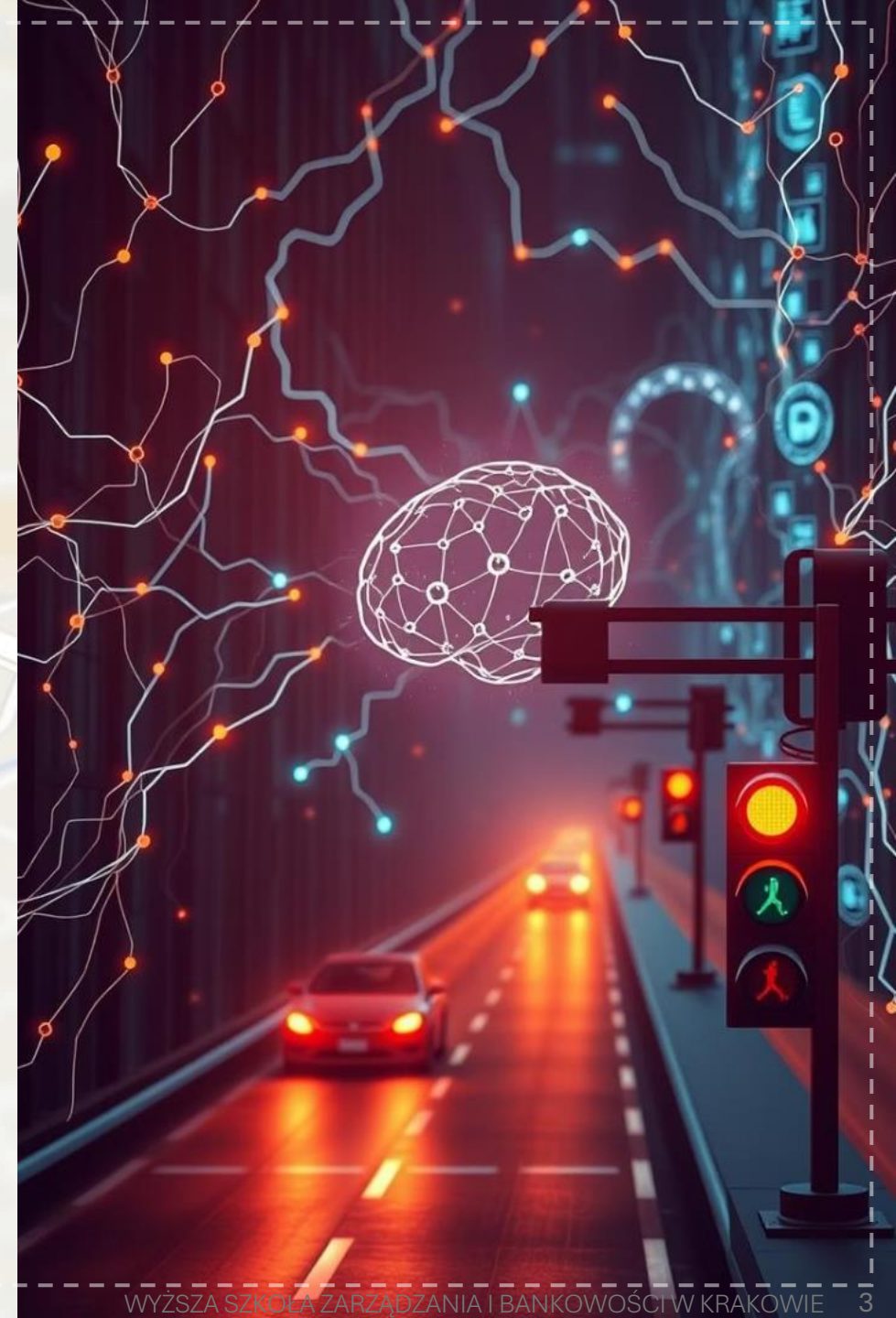
- czujniki i detektory ruchu,
- systemy monitoringu i zarządzania ruchem,
- komunikacja między pojazdami (V2V) i infrastrukturą (V2I).

Przykłady istniejących systemów adaptacyjnych:

- SCATS (Sydney Coordinated Adaptive Traffic System) – stosowany m.in. w Australii i Azji,
- SCOOT (Split Cycle Offset Optimization Technique) – rozwijany w Wielkiej Brytanii,
- RHODES – dynamiczne sterowanie światłami oparte na predykcji przepływu pojazdów.

Mimo zaawansowania technicznego, systemy te mają ograniczenia – np. wysokie koszty wdrożenia, trudności w skalowaniu oraz ograniczoną elastyczność wobec nagłych zmian w ruchu.

Dlatego coraz większe znaczenie mają rozwiązania oparte na sztucznej inteligencji, takie jak uczenie ze wzmocnieniem, które mogą uczyć się i adaptować bez potrzeby ręcznego modelowania warunków ruchu.



PODSTAWY UCZENIA ZE WZMOCNIENIEM

Uczenie ze wzmocnieniem (RL) to dziedzina sztucznej inteligencji, w której agent uczy się poprzez interakcję ze środowiskiem, otrzymując nagrody lub kary za swoje działania.

Duży rozgłos RL zdobyło dzięki sukcesowi agenta DeepMind, który samodzielnie nauczył się grać w klasyczne gry Atari (np. *Breakout*), a także dzięki projektom takim jak **AlphaGo (2016)** i **AlphaZero (2017)** – osiągając wyniki lepsze od ludzi.

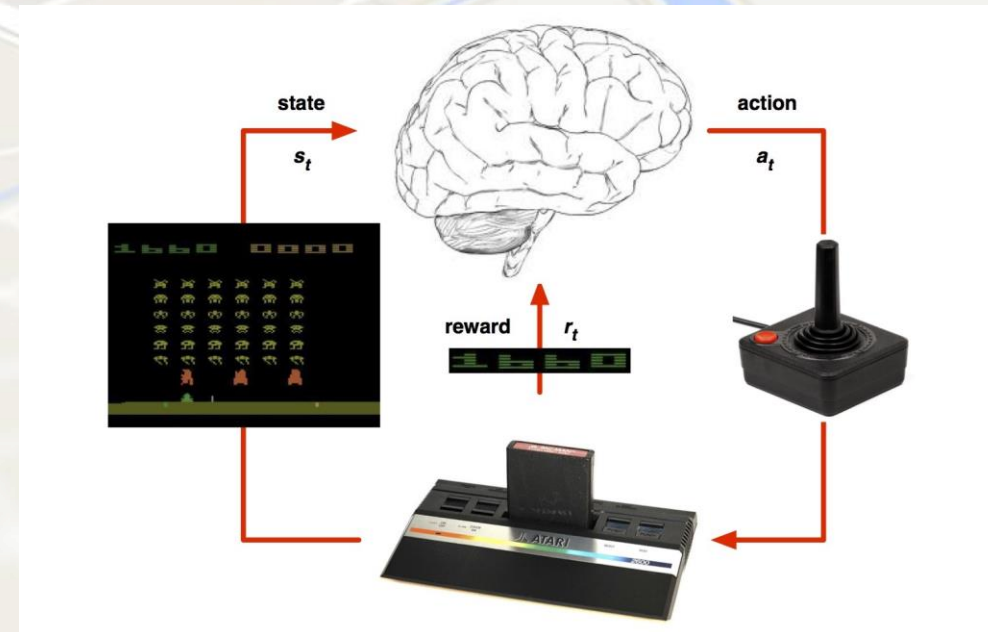
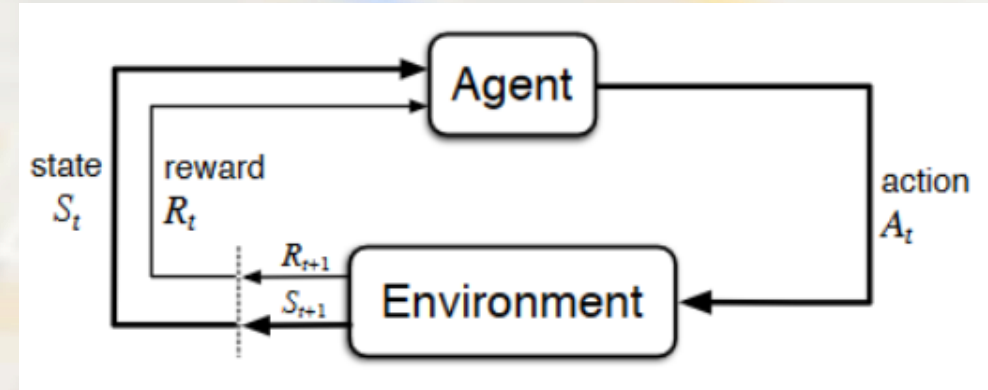
Model RL opiera się na tzw. **procesie decyzyjnym Markowa (MDP)**, który składa się z:

- **stanów (S)** – opisujących bieżącą sytuację (np. długości kolejek),
- **akcji (A)** – możliwych decyzji agenta (np. zmiana światła),
- **nagrody (R)** – informacji zwrotnej za wykonanie danej akcji,
- **funkcji przejścia (P)** – określającej prawdopodobieństwo przejścia ze stanu do stanu.

Celem agenta jest **maksymalizacja skumulowanej nagrody w czasie**.

RL może być stosowane w środowiskach, gdzie model świata nie jest w pełni znany – agent sam uczy się optymalnej polityki działania.

Polityka ta może być deterministyczna (zawsze ta sama akcja w danym stanie) **lub stochastyczna** (akcja wybierana losowo z rozkładu prawdopodobieństwa) – co pozwala na lepszą eksplorację i elastyczność w dynamicznych środowiskach.



ALGORYTM ACTOR-CRITIC

Actor-Critic to architektura uczenia ze wzmocnieniem, która łączy:

- Aktor (ang. Actor) – decyduje, jaką akcję podjąć w danym stanie (polityka działania),
- Krytyk (ang. Critic) – ocenia, jak dobra była ta decyzja, obliczając wartość stanu i błąd predykcji.

Działa na zasadzie współpracy dwóch sieci neuronowych:

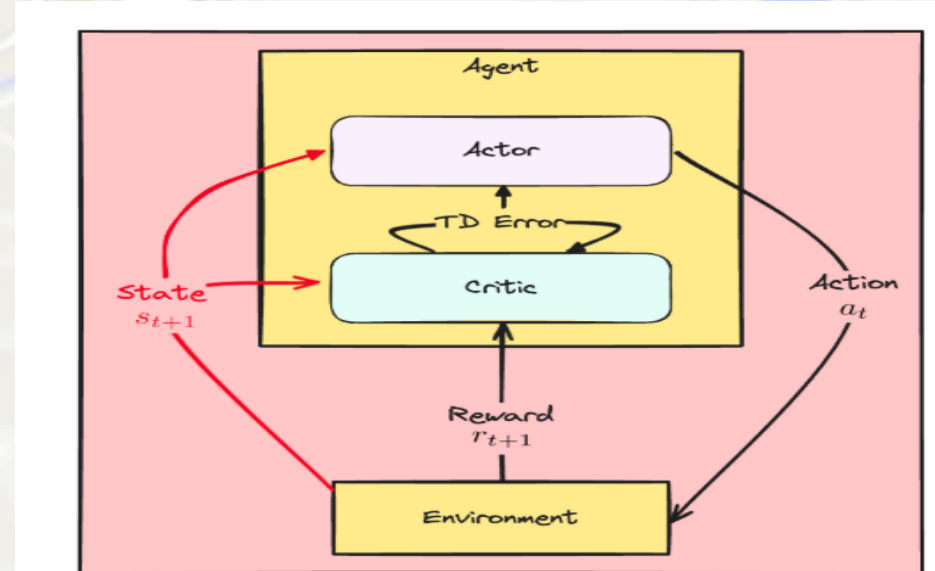
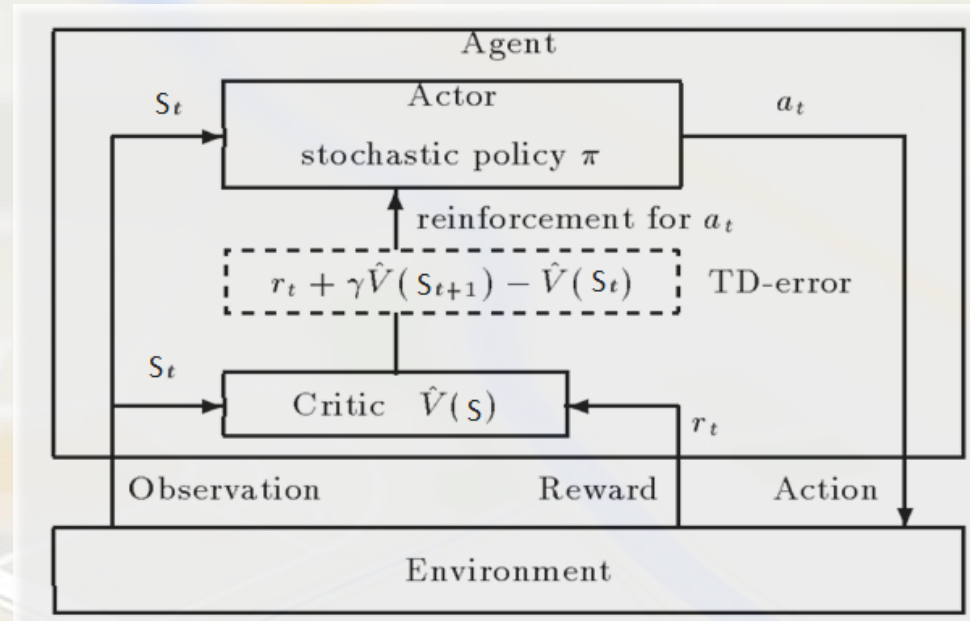
- Aktor aktualizuje strategię na podstawie informacji zwrotnej od Krytyka,
- Krytyk uczy się na podstawie różnicy czasowej (TD error) między przewidywaną a rzeczywistą nagrodą.

Algorytm potrafi generować deterministyczne lub stochastyczne polityki – w zależności od zastosowanej funkcji wyjściowej (np. softmax vs argmax).

Actor-Critic może być wykorzystywany zarówno w przestrzeniach dyskretnych, jak i ciągłych, co czyni go elastycznym w zastosowaniach takich jak sterowanie ruchem.

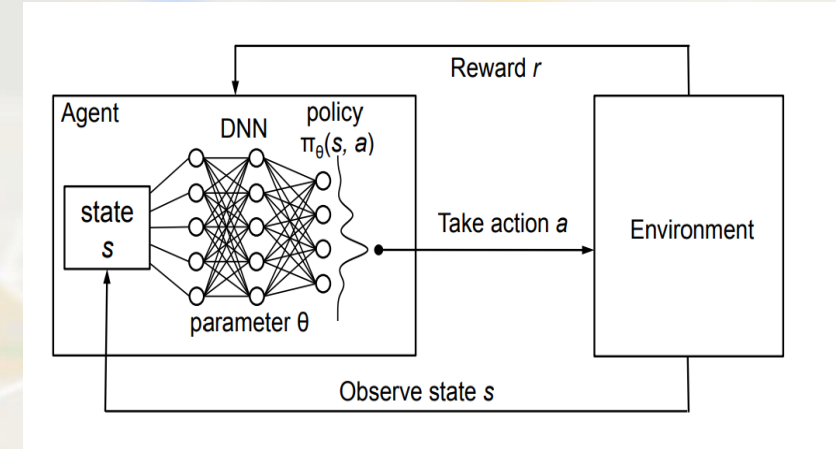
Zalety:

- szybsza konwergencja niż w metodach opartych wyłącznie na wartości,
- mniejsza wariancja gradientów niż w klasycznym policy gradient,
- możliwość działania w czasie rzeczywistym.



UCZENIE GŁĘBOKIE I ARCHITEKTURA SIECI NEURONOWEJ

- Uczenie głębokie (Deep Learning) wykorzystuje wielowarstwowe sieci neuronowe do aproksymacji funkcji i podejmowania decyzji.
- W kontekście sterowania ruchem drogowym znacząco upraszcza procesy decyzyjne, ponieważ:
 - eliminuje konieczność ręcznego definiowania reguł i tabel Q,
 - zmniejsza zasoby pamięci i obliczeniowe – nie trzeba przechowywać i aktualizować dużych macierzy wartości dla każdej kombinacji stan-akcja,
 - sieć automatycznie uczy się reprezentacji istotnych cech na podstawie surowych danych (np. długości kolejek).
- W algorytmie Actor-Critic sieć pełni podwójną rolę:
 - Aktor – generuje politykę (rozkład prawdopodobieństw akcji),
 - Krytyk – ocenia wartość stanu (funkcja V).



Środowisko symulacyjne SUMO + TraCI

SUMO (Simulation of Urban MObility) to otwartoźródłowy mikrosymulator ruchu drogowego, który umożliwia dokładne modelowanie zachowań pojedynczych pojazdów.

Pozwala tworzyć realistyczne sieci dróg, definiować skrzyżowania, sygnalizację świetlną, źródła ruchu oraz pojazdy komunikacji publicznej.

W pracy wykorzystano sieć składającą się z czterech skrzyżowań i ośmiu wlotów – zróżnicowany, realistyczny model ruchu.

TraCI (Traffic Control Interface)

Interfejs umożliwiający dwukierunkową komunikację między Pythonem a SUMO w czasie rzeczywistym.

Dzięki TraCI agent RL może:

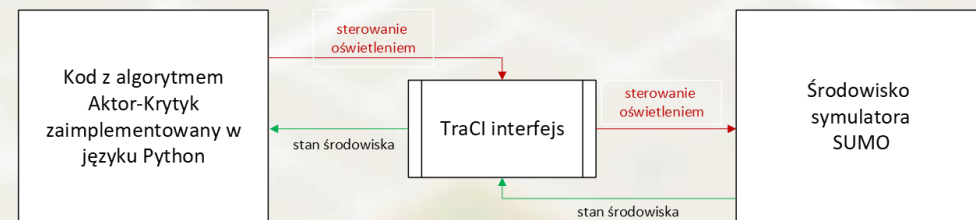
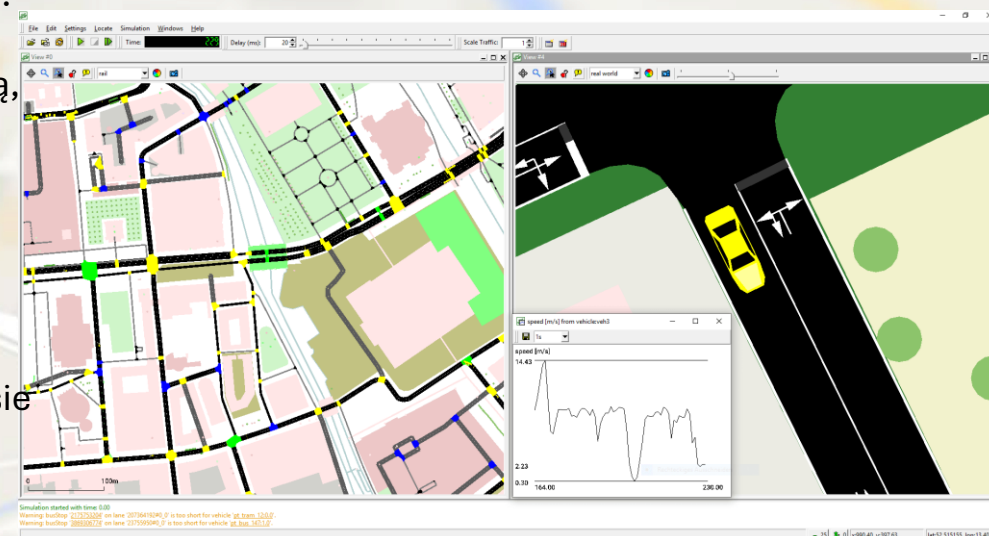
- odczytywać dane o stanie ruchu (kolejki, czas oczekiwania, pojazdy),
- sterować światłami: `traci.trafficlight.setRedYellowGreenState()`
- wykonywać akcje co krok symulacji i na bieżąco reagować na sytuację na skrzyżowaniu.

Dlaczego SUMO + TraCI?

Łatwa integracja z Pythonem,

Niskie wymagania sprzętowe,

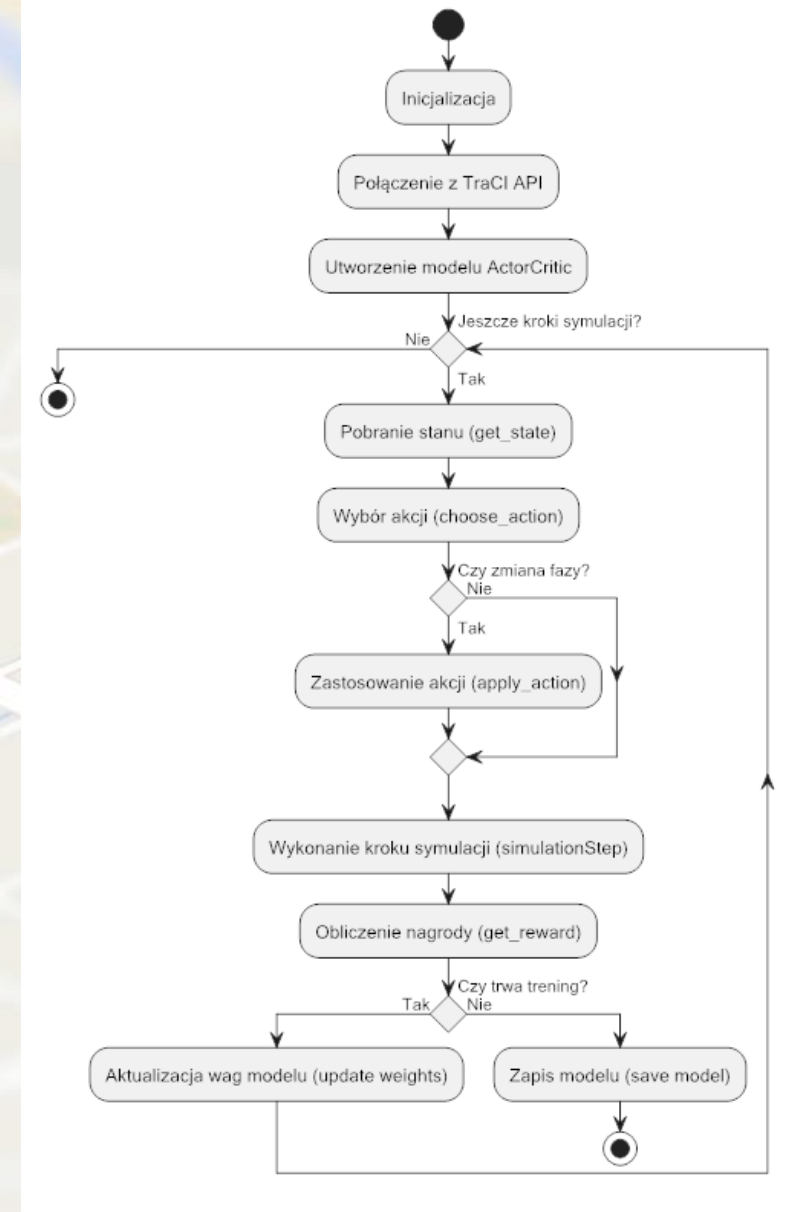
Możliwość pełnej kontroli środowiska do testów algorytmu RL,



Implementacja w Python + TraC

- Inicjalizacja i połączenie z TraCI API
 - Uruchomienie symulacji SUMO oraz nawiązanie połączenia z Pythonem.
- Utworzenie modelu Actor-Critic
 - Inicjalizacja architektury sieci neuronowej aktora i krytyka.
- Pętla działania (dla każdego kroku symulacji):
 - Pobranie aktualnego stanu (get_state) – np. długości kolejek.
 - Wybór akcji (choose_action) – np. zmiana fazy świateł.
 - Zastosowanie akcji (apply_action), jeśli decyzja tego wymaga.
 - Wykonanie kroku symulacji (simulationStep).
 - Obliczenie nagrody (get_reward) – np. na podstawie łącznego czasu oczekiwania.
- Uczenie i zapis modelu:
 - Jeśli trwa trening, następuje aktualizacja wag sieci (update weights).
 - Po zakończeniu treningu model zostaje zapisany (save model).

```
import traci                # Interfejs komunikacji z SUMO
import numpy as np          # Operacje na danych wejściowych
import tensorflow            # Uczenie głębokie (sieci neuronowe)
```



SIEĆ NEURONOWA I PROCES UCZENIA

Struktura zastosowanej sieci (Actor-Critic)

Wejście: dane ze środowiska (kolejki, czasy oczekiwania)

Warstwa ukryta:

- 64 neurony, aktywacja ReLU

Dwie głowy wyjściowe:

- Aktor: warstwa softmax – zwraca rozkład prawdopodobieństw akcji
- Krytyk: pojedynczy neuron liniowy – zwraca wartość stanu

Optymalizator: Adam

Funkcja straty: błąd TD (Temporal Difference)

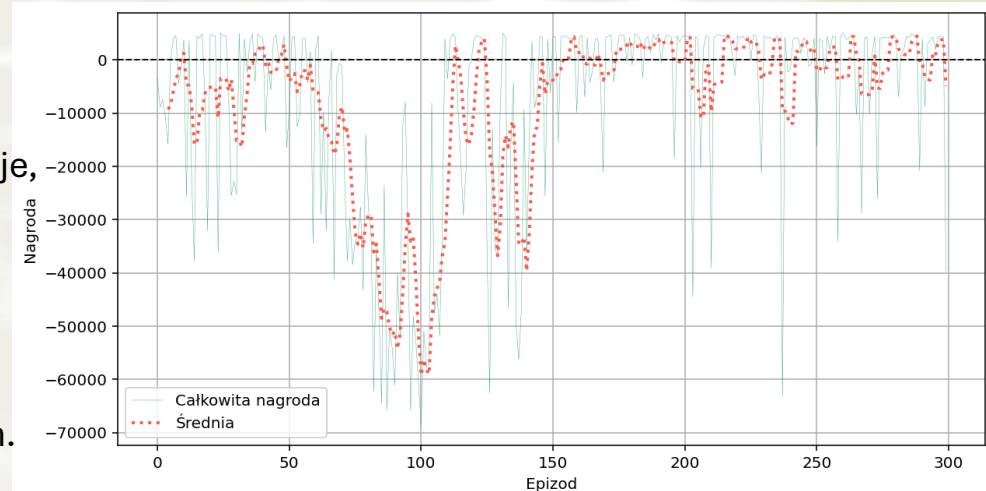
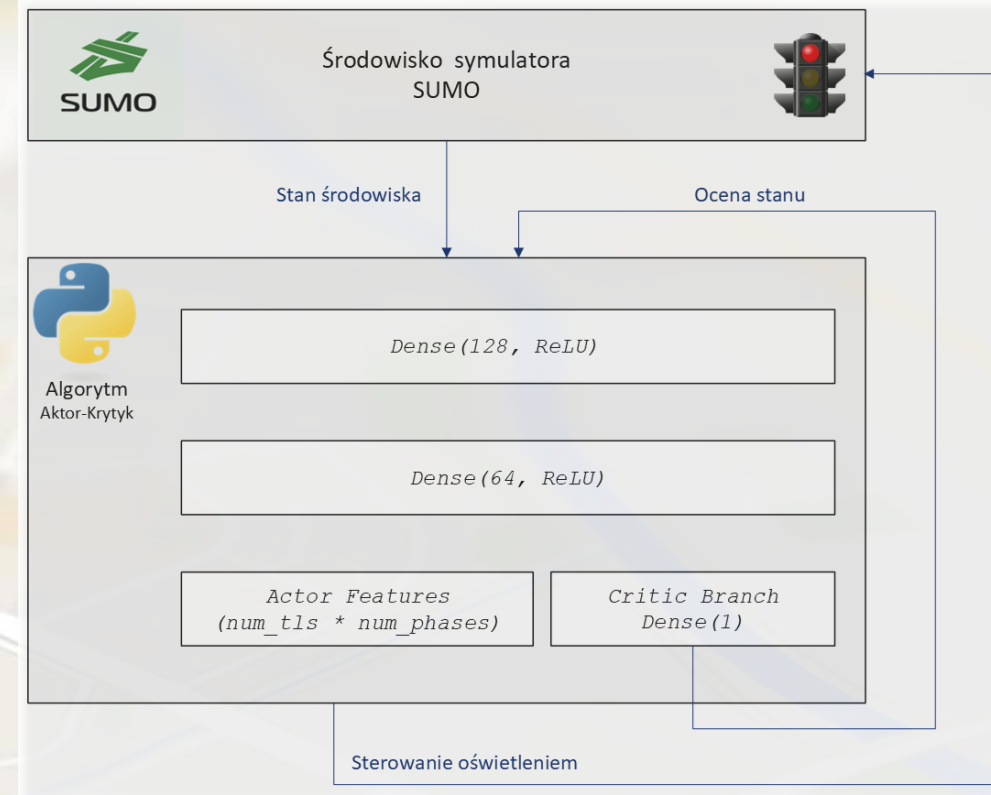
Eksploracja: ϵ -greedy lub wybór na podstawie softmaxa

Proces uczenia

Agent uczył się przez wiele epok, cyklicznie odczytując stan środowiska, wybierając akcje, otrzymując nagrodę i aktualizując model.

Uczenie nadzorowane przez sygnał nagrody: skracanie kolejek i zmniejszanie czasu oczekiwania.

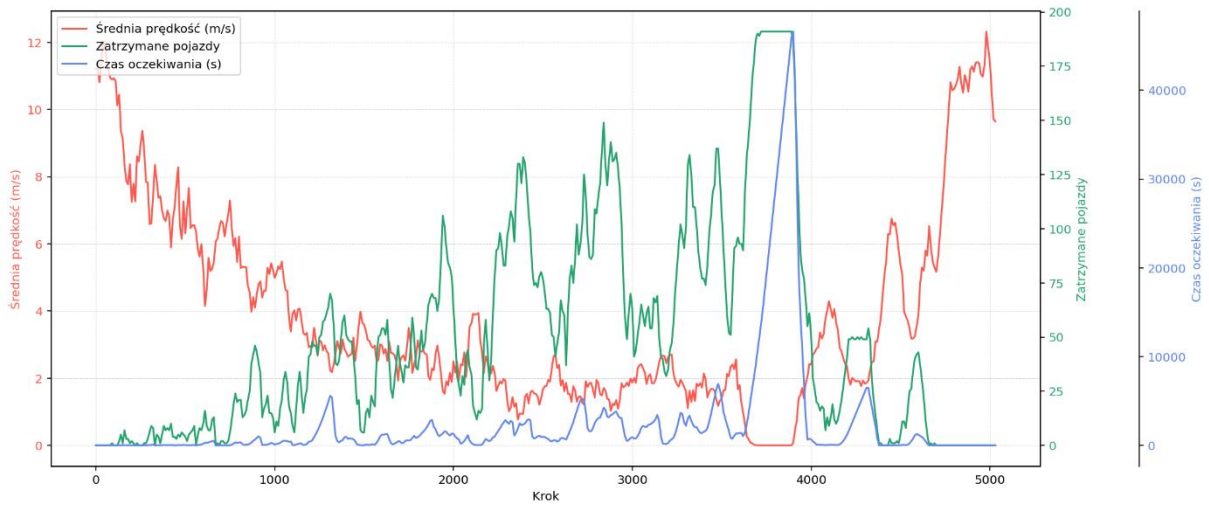
Trening podzielony na epoki, z możliwością zapisu modelu po zakończeniu każdej z nich.



Wyniki symulacji

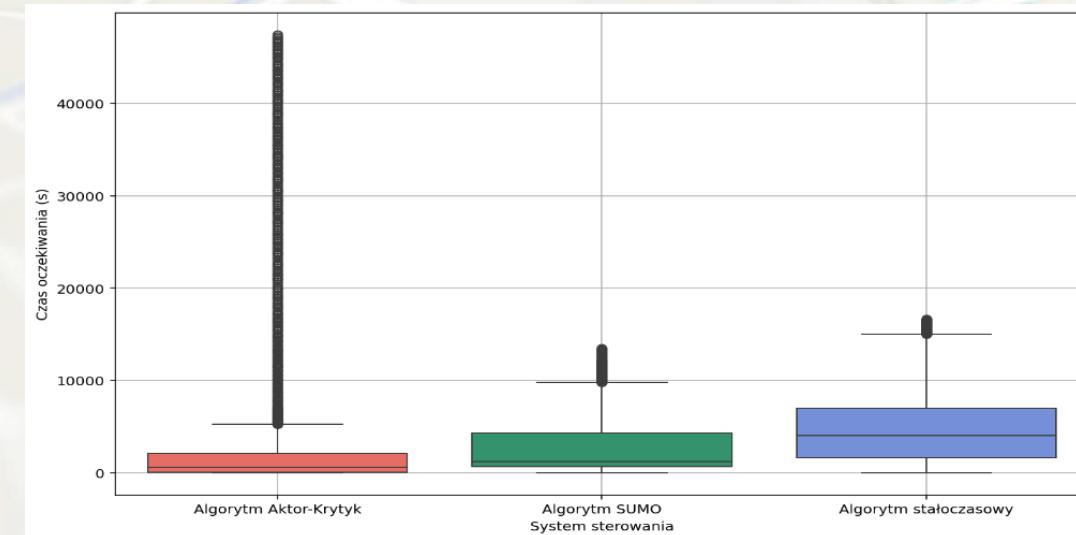
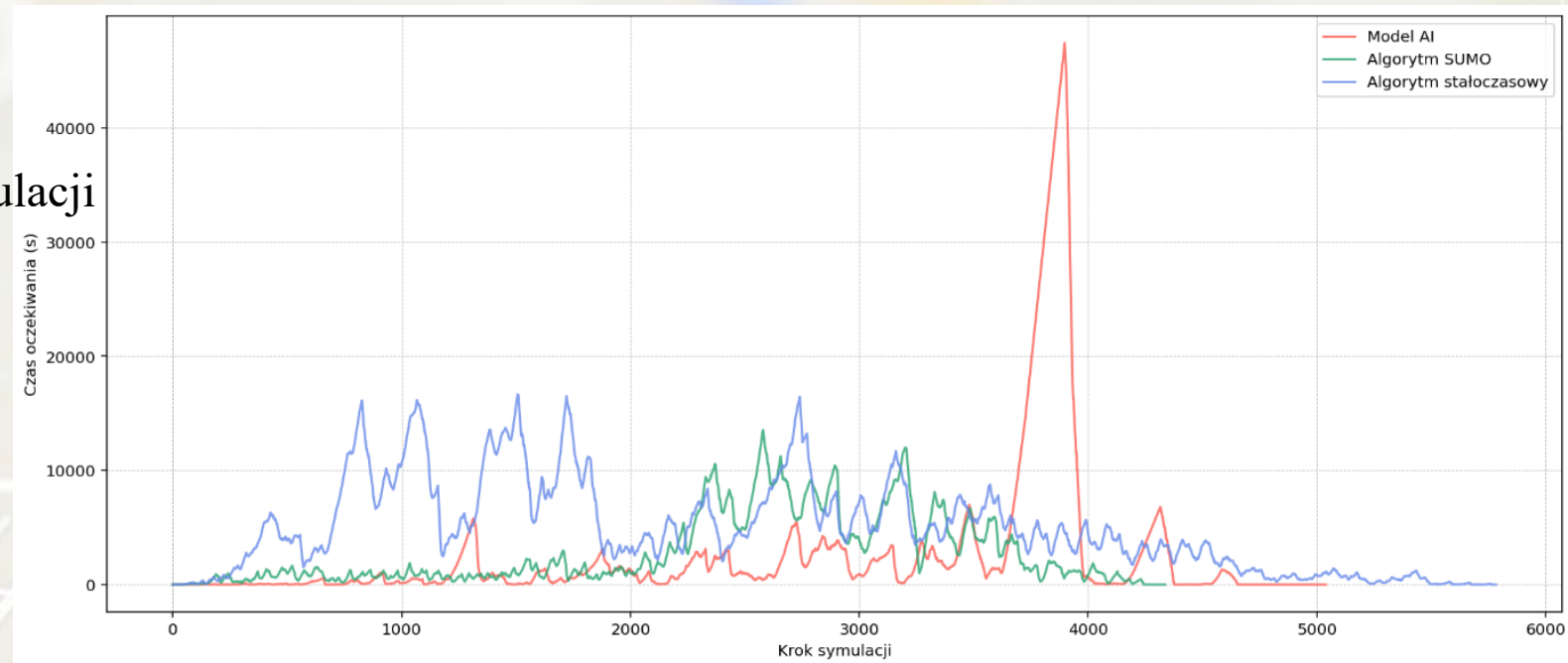
Analiza wskaźników jakości sterowania ruchem

Wskaźnik	Średni a	Max. wartość	Odchylenie standardowe
Średnia prędkość (m/s)	3,86	12,80	2,94
Zatrzymane pojazdy	50,40	191,00	49,35
Czas oczekiwania wszystkich pojazdów (s)	2536,3	47 445,00	6594,54



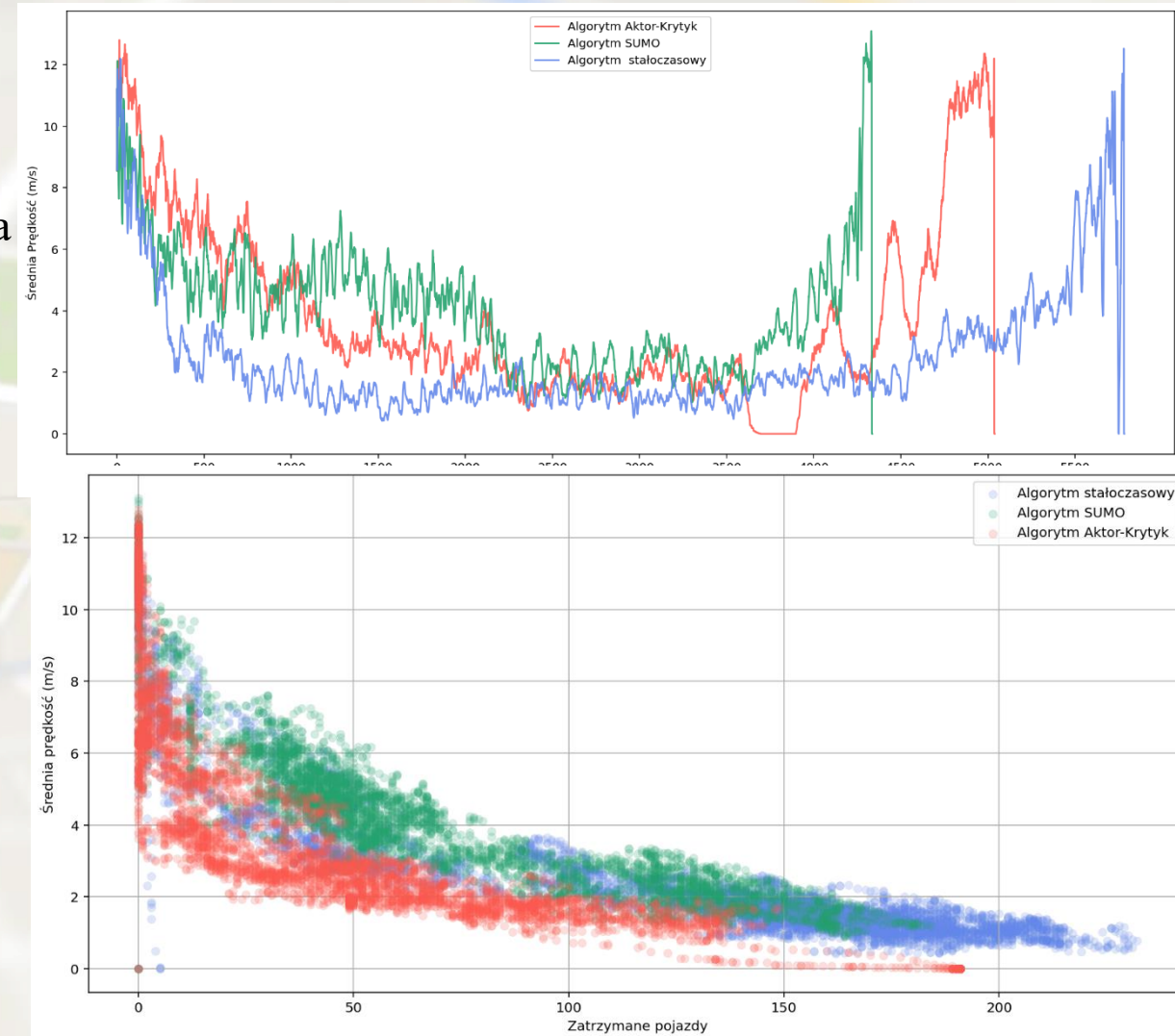
Wyniki symulacji

- Czas zatrzymania pojazdów w czasie symulacji



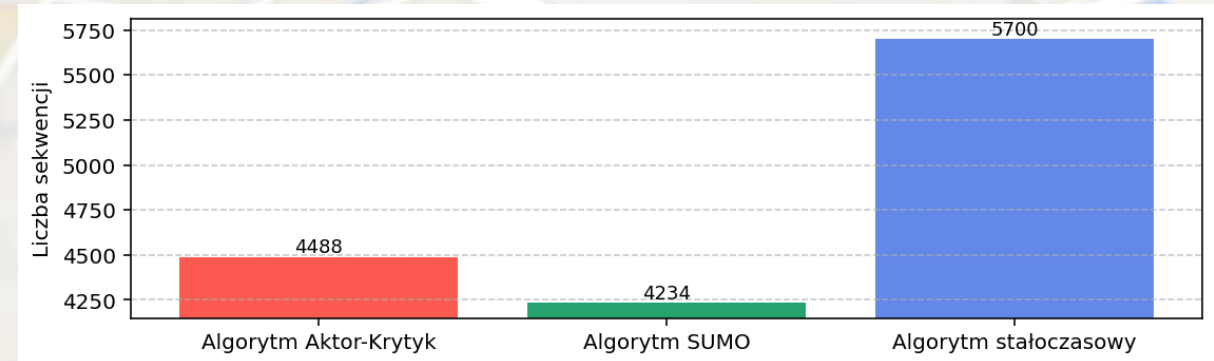
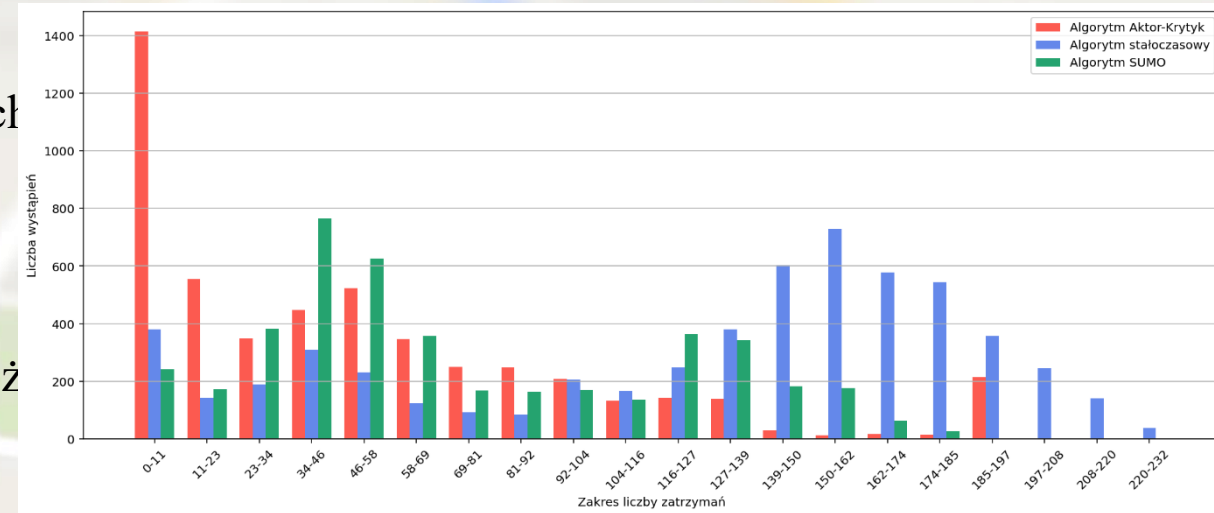
Wyniki symulacji

- Średnią prędkość pojazdów podczas symulacji
- Analiza zależności między liczbą zatrzymanych pojazdów a prędkością



Wyniki symulacji

- Rozkład liczby zatrzymanych pojazdów w różnych systemach sterowania
- Liczba sekwencji symulacji potrzebna do całkowitego opróżnienia skrzyżowań z pojazdów.



Rezultaty wnioski

- Algorytm Actor-Critic znacząco przewyższył sterowanie stałoczasowe:
 - krótsze czasy oczekiwania,
 - mniejsze kolejki i liczba zatrzymań,
 - lepsza adaptacja do zmiennego natężenia ruchu.
- Połączenie aktora i krytyka z błędem TD pozwoliło agentowi szybciej korygować strategię i równomiernie rozkładać ruch.
- Wyzwania i możliwości rozwoju:
 - Priorytetyzacja pojazdów uprzywilejowanych i komunikacji zbiorowej
 - Wieloagentowe sterowanie ruchem (agent dla każdego skrzyżowania)
 - Wielokryterialna optymalizacja – np. emisja, zużycie paliwa, koszty
 - Ujęcie pieszych i rowerzystów – poprawa bezpieczeństwa i płynności
 - Integracja z danymi zewnętrznymi – np. z pojazdów autonomicznych, sieci GSM





KONIEC