

Optymalizacja systemu sygnalizacji świetlnej w oparciu o przepływowy model ruchu pojazdów.

Michał Lis

10 września 2019

Spis treści

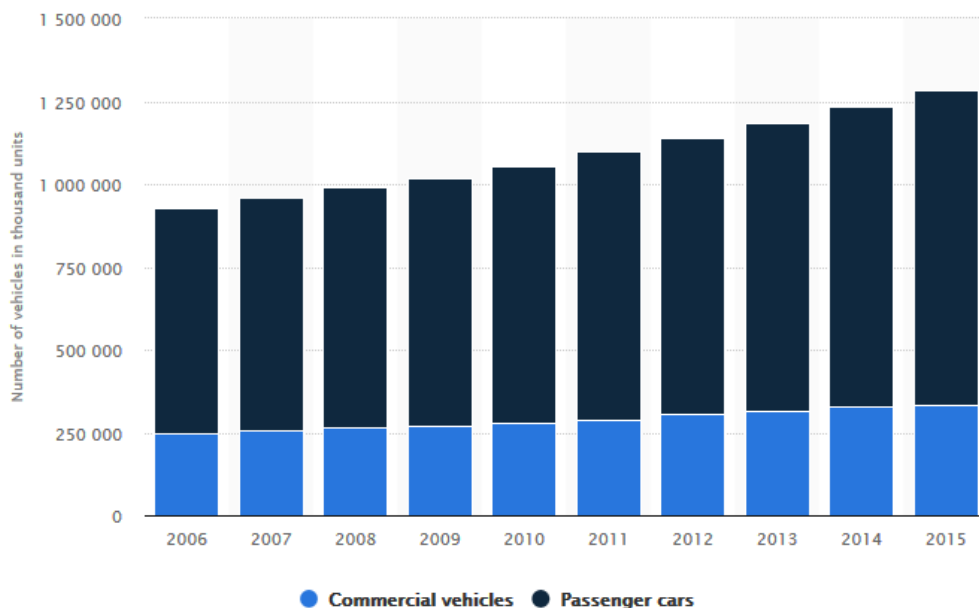
1	Wprowadzenie	5
2	Cel i zakres pracy	7
3	Uczenie ze wzmocnieniem	9
3.1	Kategorie uczenia maszynowego	9
3.2	Uczenie ze wzmocnieniem	9
3.3	Nomenklatura uczenia ze wzmocnieniem	12
3.4	Metoda Q-learning	14
3.5	Metoda Q-learning z użyciem sieci neuronowej	15
3.5.1	Wygenerowane dane podczas symulacji treningowych	15
3.5.2	Trening sieci neuronowej	16
3.5.3	Modyfikacja wzoru Q dla fazy żółtych światel	16
4	Makroskopowy model ruchu	17
4.1	Klasyfikacja modeli ruchu drogowego	17
4.2	Wstęp	17
4.3	Rozwój gęstości ruchu na drodze	18
4.4	CTM - Cell Transmission Model	18
4.4.1	Wprowadzenie	18
4.4.2	Model CTM dla pojedynczej drogi	19
4.4.3	Model CTM dla sieci dróg	21
4.4.4	Przepływ pojazdów na skrzyżowaniu	22
4.4.5	Przepływ pojazdów na skrzyżowaniu z sygnalizacją świetlną	24
5	Model sieci dróg	25
5.1	Wstęp	25
5.2	Wektor stanu drogi	25
5.3	Rozwój wektora stanu drogi i wprowadzenie macierzy systemu	26
5.4	Wektor stanu sieci dróg	27
5.5	Rozwój wektora stanu sieci dróg w oparciu o CTM	28
5.5.1	Macierz prawdopodobieństwa	28
5.5.2	Macierz systemu	30
5.6	Wprowadzenie sygnalizacji świetlnej	32
5.7	Wprowadzenie źródeł ruchu	35

6	Środowiska symulacyjne i ich nauka	37
6.1	Środowisko 1	37
6.1.1	Sygnalizacje świetlne	38
6.1.2	Przestrzeń decyzyjna	38
6.1.3	Model ruchu	38
6.1.4	Uczenie	38
6.2	Środowisko 2	40
6.2.1	Sygnalizacje świetlne	40
6.2.2	Przestrzeń decyzyjna	40
6.2.3	Model ruchu	41
6.2.4	Uczenie	41
6.3	Środowisko 3	44
6.3.1	Sygnalizacje świetlne	44
6.3.2	Przestrzeń decyzyjna	45
6.3.3	Model ruchu	46
6.3.4	Uczenie	47
6.4	Środowisko 4 - kampus B Politechniki Łódzkiej	49
6.4.1	Sygnalizacje świetlne	50
6.4.2	Przestrzeń decyzyjna	52
6.4.3	Model ruchu	52
6.4.4	Uczenie	52
6.4.5	Modyfikacja środowiska	53
7	Dokumentacja programu	55
7.1	Wstęp	55
7.2	Warstwa logiki	55
7.2.1	Wymagania	55
7.2.2	Instrukcje uruchomienia	56
7.3	Warstwa prezentacyjna	56
7.3.1	Wymagania	56
7.3.2	Instrukcje uruchomienia	56

Rozdział 1

Wprowadzenie

Problem zatłoczonych ulic staje się coraz bardziej powszechny na całym świecie [9]. W ostatnich latach obserwowany jest ogromny wzrost ilości pojazdów na drogach. Według danych statystycznych firmy *Statista* liczba zarejestrowanych pojazdów na świecie w roku 2006 wynosiła 947 tysięcy [25]. W 2015 roku na świecie jeździło już 1282 tysięcy pojazdów. Wzrost przez te 9 lat był niemalże liniowy. Co roku rejestrowano około 40 tysięcy nowych samochodów rocznie, co wyznacza stopę wzrostu liczby pojazdów na poziomie 3,7% w skali roku.



Rysunek 1.1: Liczba pojazdów na świecie

W Polsce wzrost ilości pojazdów w latach 2006 - 2015 był jeszcze większy [22]. W 2006 roku według GUS w Polsce było zarejestrowanych 13,4 miliona samochodów osobowych. W 2015 roku ich liczba wynosiła już 20,7 miliona, co wyznacza 5 procentowy roczny wzrost ilości pojazdów. Najbardziej zatłoczonym polskim miastem jest Łódź. Według rankingu firmy *TomTom* Łódź zajmuje bardzo wysokie 5 miejsce na świecie i 1 w Europie pod względem

zatłoczenia dróg [27]. Oprócz Łodzi w pierwszej setce najbardziej zatłoczonych miast świata są inne polskie miasta: Lublin(34), Kraków(48), Warszawa(50), Wrocław(63), Poznań(69), Bydgoszcz(83). Zatory drogowe są jednak problem całej Europy. Spośród 100 najbardziej zatłoczonych miast świata aż 45 znajduje się w Europie. W 2008 roku Unia Europejska oszacowała, iż koszty zatłoczenia dróg kształtują się na poziomie 0,9% – 1,5% PKB unijnego [1]. Następny raport z 2017 roku może napawać optymizmem, gdyż przedstawione w nim wyliczenia określiły jedynie 0,77% straty całkowitego PKB wspólnoty [2]. Ten sam raport ocenia koszty zatorów komunikacyjnych w Polsce na poziomie 1,2% polskiego PKB. Problemy zatorów komunikacyjnych w miastach są o tyle trudniejsze do rozwiązania niż poza miastem, ponieważ na terenach zurbanizowanych często brakuje przestrzeni do wybudowania dróg o większej przepustowości. Przystępnym rozwiązaniem wydaje się być wprowadzenie sygnalizacji świetlnej na zatłoczonych skrzyżowaniach. Niepodważalnie istotną kwestią jest wtedy optymalizacja ustawień sygnalizacji świetlnej. Praca jest poświęcona temu aspektowi.

Rozdział 2

Cel i zakres pracy

Celem pracy jest przedstawienie metod uczenia maszynowego, których zadaniem będzie ustalenie strategii doboru faz świetlnych tak, aby zapewnić możliwie płynny przepływ ruchu drogowego. W rozdziale 3 przedstawiona zostanie metoda uczenia ze wzmocnieniem wykorzystana w optymalizacji sygnalizacji świetlnej. Rozdział 4 opisuje przepływowy model ruchu pojazdów, który będzie obowiązywał w symulacjach drogowych. W rozdziale 5 przedstawiony zostanie model sieci dróg. Początkowy model będzie bardzo prosty z pominięciem większości aspektów. W każdej kolejnej sekcji rozdziału będzie stopniowo rozwijany. Rozdział 6 przedstawia środowiska symulacyjne dla których zostanie przeprowadzony proces optymalizacji sygnalizacji świetlnych. Ostatnie środowisko testowe w pracy przedstawia sieć dróg wokół kampusu B Politechniki Łódzkiej i to głównie na optymalizacji tego środowiska będzie skupiona uwaga. Wszystkie poprzednie sieci dróg są mniej skomplikowane i służą głównie przy przedstawieniu modelu i procesu optymalizacji. Stworzony przez autora pracy został program symulujący ruch drogowy środowisk z rozdziału 6. Specyfikacja technologiczna, wymagania i instrukcje uruchomienia programu zostały przedstawione w rozdziale 7.

Rozdział 3

Uczenie ze wzmocnieniem

3.1 Kategorie uczenia maszynowego

Uczenie maszynowe to dziedzina wchodząca w skład nauk zajmujących się sztuczną inteligencją [16]. Samo uczenie maszynowe w najprostszym kształcie może być rozumiane jako dobór optymalnych parametrów na podstawie dostępnych danych. Uczenie maszynowe jest powszechnie dzielone na 3 kategorie nauki [13].

1. Nadzorowane

Dane używane do uczenia nazywane są zbiorem treningowym. Każdy pojedynczy element zbioru treningowego ma informacje wejściowe oraz pewną pożądaną wartość wyjściową. W trakcie uczenia algorytm dopasowuje swoje parametry tak aby na podstawie danych wejściowych mógł przewidzieć wartość wyjściową. Przykładami uczenia nadzorowanego jest np. rozpoznawanie tekstu pisanego, detekcja obiektów na zdjęciach.

2. Nienadzorowane

Uczenie nienadzorowane różni się od nadzorowanego tym, że nie są znane pożądane wartości wyjściowe. Celem nauki jest na podstawie podobieństw poszczególnych elementów zgrupowanie ich do odpowiednich klas. Przykładem uczenia nienadzorowanego może być np. klasyfikacja gatunków drzew na podstawie danych jedynie o ich wysokościach i szerokości korony drzew.

3. Wzmocnione

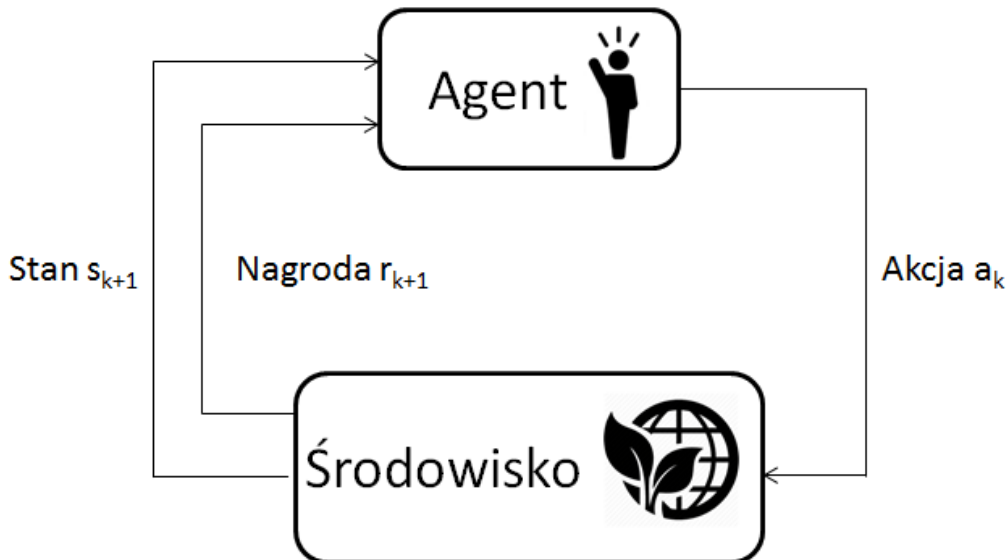
W środowisku uczenia ze wzmocnieniem istnieje agent(lub kilku), który jest odpowiedzialny za podejmowanie pewnych akcji. Każda akcja ma wpływ na zmianę stanu środowiska, które w międzyczasie zwraca agentowi nagrodę. Uczenie ze wzmocnieniem jest oparte o strategię maksymalizującą skumulowaną wartość nagród.

Spośród tych trzech kategorii uczenie ze wzmocnieniem odpowiada problemowi optymalizacji sygnalizacji świetlnej.

3.2 Uczenie ze wzmocnieniem

Schemat uczenia ze wzmocnieniem składa się z następujących elementów:

1. **Agent** jest odpowiedzialny za podejmowanie decyzji odnośnie wykonania pewnych akcji. Ma on wiedzę na temat obecnego stanu środowiska i otrzymuje w każdym kroku czasowym sygnał nagrody. Agentów może być więcej niż jeden. Wtedy mówimy, że system jest wieloagentowy. Akcje podjęte przez agenta wpływają na zmianę stanu środowiska.
2. **Środowisko** jest przestrzenią posiadającą dynamiczny stan widoczny dla agenta. Choć agent podejmuje akcje, to środowisko ma zdefiniowany model zmiany stanu. Model zmiany stanu może być stochastyczny oraz niewidoczny dla agenta. Oznacza to, że dwie te same akcje podjęte w tym samym stanie nie zawsze przyniosą identyczny następny stan. Innymi słowy agent nie może być stuprocentowo pewny rezultatów swoich akcji. Środowiska rozważane w rozdziale 6 są jednak w pełni deterministyczne.
3. **Strategia** definiuje sposób doboru akcji przez agenta. Jest to funkcja, która przyjmuje stan środowiska i zwraca akcję, która powinna być przeprowadzona.
4. **Sygnał nagrody** jest liczbą rzeczywistą, którą otrzymuje agent po podjęciu akcji. Określa ona jak dużą natychmiastową korzyść przyniosła ta akcja. Wartości nagród są czynnikiem wpływającym na zmianę strategii, gdyż zadaniem agenta jest maksymalizacja nagród. Wartość nagród zatem definiuje, które zdarzenia są dobre, a które złe dla agenta. Biologicznym odpowiednikiem dodatniej nagrody jest przyjemność, a ujemnej - ból [4].
5. **Funkcja wartości** zwraca wartość stanu czyli oczekiwaną sumę nagród jakie agent osiągnie w przyszłości będąc aktualnie w tym stanie.



Rysunek 3.1: Interakcje pomiędzy agentem a środowiskiem.

Proces Markowa

Algorytmy uczenia ze wzmocnieniem zazwyczaj stosuje się do rozwiązywania problemu procesu decyzyjnego Markowa [26]. Sam **proces decyzyjny Markowa** jest zdefiniowany jako uporządkowana czwórka (S, A, P^a, R^a) , gdzie:

1. S to zbiór wszystkich stanów.
2. A to zbiór akcji. Notacją A_s oznaczane są możliwe akcje dla stanu s .
3. $P^a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ to prawdopodobieństwo, że akcja a wykonana w stanie s w chwili t doprowadzi do stanu s' w chwili $t + 1$.
4. $R^a(s, s')$ to oczekiwana nagroda otrzymana w wyniku akcji podjętej w stanie s prowadzącej do stanu s' . W dalszych rozważaniach R_k będzie oznaczało nagrodę za akcję podjętą w chwili k .

Problem procesu Markowa

Problemem procesu decyzyjnego Markowa jest odnalezienie optymalnej strategii. Strategia określona jest jako funkcja $\pi(s)$ przyjmująca jako argument stan, a zwracająca podejmowaną akcję. Celem optymalizacji jest odnalezienie strategii maksymalizującej następującą wartość:

$$G = \sum_{k=0}^K \gamma^k R_k \quad (3.1)$$

Chociaż we wzorze (3.1) nie ma strategii $\pi(s)$, to wpływa ona na otrzymywane nagrody R_k , gdyż określa podejmowane akcje. Parametr $\gamma \in (0, 1]$ jest czynnikiem dyskontującym.

Idea dyskonta

Idea dyskontowania nagród została zaczerpnięta z rachunku finansowego. Przykładowo wpływ na konto kapitału 1000 złotych po upływie roku czasu jest z pewnością bardziej wartościowy niż za 20 lat. Pieniądze w matematyce finansowej są prawie zawsze liczone w czasie i tak samo należy postępować z nagrodami. Im wartość γ jest bliższa 0 tym bardziej istotne są początkowe nagrody. Dla $\gamma = 1$ wszystkie nagrody są równie istotne - bez względu na czas ich otrzymania.

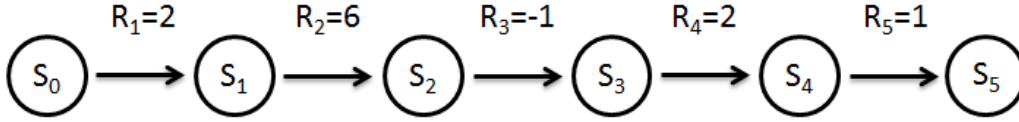
Wartość stanu

Analogicznie do (3.1) jest ustalona funkcja wartości stanu. Wartość stanu s , który miał miejsce w chwili t jest określona jako:

$$G_t = \sum_{k=t}^K \gamma^k R_k = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots + \gamma^K R_K \quad (3.2)$$

Przykład obliczeniowy

Zostanie przedstawiony teraz przykład obliczeniowy. Agent podejmuje decyzje na których podstawie otrzymuje ciąg nagród $R_0 = 0, R_1 = 2, R_2 = 6, R_3 = -1, R_4 = 2, R_5 = 1$. Czynniki dyskontujący γ jest równy 0,9. Jaka jest wartość sumy zdyskontowanych nagród G oraz kolejnych stanów t.j. G_1, G_2, G_3, G_4, G_5 ?



Rysunek 3.2: Ciąg nagród i stanów.

Najłatwiej obliczenia rozpocząć od G_5 i zakończyć na G_1 .

$$G_5 = R_5. \quad (3.3)$$

Wzór (3.2) można przekształcić do następującej postaci dla $t \in \{1, 2, 3, 4\}$:

$$G_t = R_t + \gamma G_{t+1}. \quad (3.4)$$

$$G_5 = R_5 = 1.$$

$$G_4 = R_4 + \gamma G_5 = 2 + 0,9 \cdot 1 = 2,9.$$

$$G_3 = R_3 + \gamma G_4 = -1 + 0,9 \cdot 2,9 = 1,61.$$

$$G_2 = R_2 + \gamma G_3 = 6 + 0,9 \cdot 1,61 \approx 7,45.$$

$$G_1 = R_1 + \gamma G_2 = 2 + 0,9 \cdot 7,45 \approx 8,70.$$

$$G = R_0 + \gamma G_1 = 0 + 0,9 \cdot 8,70 \approx 7,83.$$

3.3 Nomenklatura uczenia ze wzmocnieniem

Algorytmy model-free i model-based

Algorytmami nie wymagającymi modelu środowiska (model-free) nazywane są algorytmy, które nie wykorzystują prawdopodobieństw określających możliwe konsekwencje akcji (punkt 3 wcześniej opisanego procesu Markowa). Algorytmy wymagające modelu środowiska (model-based) z kolei bazują na nich [26].

Epizody treningowe i testowe

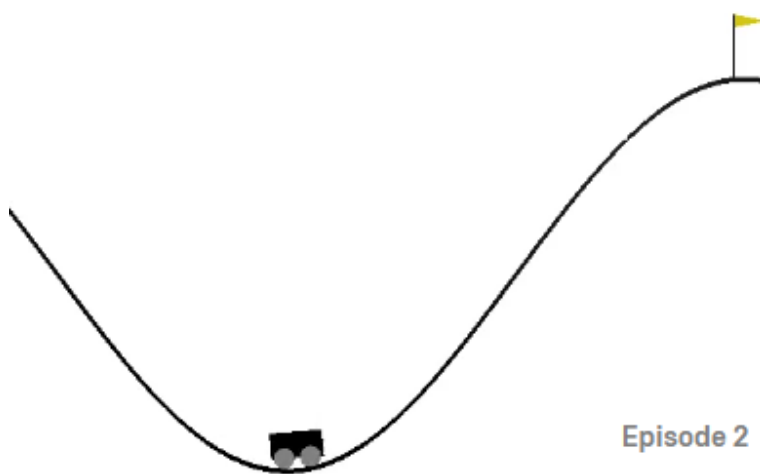
Jako epizod rozumiana jest pełna symulacja środowiska. Środowiska epizodyczne posiadają pewien warunek zakończenia symulacji (np. limit czasowy, osiągnięcie pewnego celu). Epizody treningowe mają na celu jedynie generowanie danych do nauki agentów. Mogą one być przeprowadzane wedle dowolnej strategii ustalonej przez programistę. Z kolei epizody testowe mają na celu przetestować działanie obecnie wyuczonej strategii.

Optimal policy i behaviour policy

Pierwszy termin - optimal policy odnosi się do strategii podejmowania jedynie najlepszych możliwych akcji wedle dotychczas wyuczonej strategii π . Jeśli epizod jest przeprowadzany wedle optymalnej strategii, to jest on w pełni zachłanny. Epizody testowe są zawsze w pełni zachłanne. Kolejny termin - behaviour policy odnosi się do strategii zachowania podczas epizodów treningowych. Sposób wyboru akcji podczas treningu jest bezpośrednio związany z poniższym rozważaniem.

Exploration or Exploitation - eksploracja czy eksploatacja?

Exploration or Exploitation jest bardzo często pojawiającym się terminem w artykułach tematyki uczenia ze wzmocnieniem [24], gdyż dotyczy on właściwie każdego problemu uczenia ze wzmocnieniem. Termin ten odnosi się do dylematu jakie agent powinien wybierać akcje podczas treningu. Agent może wybierać akcje losowe (eksploracja) albo najlepsze według wyuczonej polityki (zachłanne - eksploatujące). Mogłoby się wydawać, że pełna eksploracja jest dobrym rozwiązaniem podczas epizodów treningowych. Zazwyczaj tak jednak nie jest. Podejmowanie jedynie losowych akcji może prowadzić środowisko do stanów, które nie będą miały miejsca w epizodach testowych. Powstaje przez to szum informacyjny, który zaburza i spowalnia uczenie. Może też dojść do odwrotnej sytuacji. Akcje w pełni losowe mogą skutkować nie zbadaniem pewnych stanów. Doskonałym środowiskiem przedstawiającym ten problem jest 'MountainCar-v0' biblioteki OpenAi Gym [6]. Przedstawia ono pojazd w dolinie, którego celem jest podjechanie pod górę z flagą. Nie może tego zrobić jadąc w prawo. Pojazd musi się rozpędzić wjeżdżając pierw na pagórek z lewej strony. Agent dokonując same losowe akcje prawdopodobnie nigdy nie podjedzie pod górę z flagą. Musi on podjąć szereg zarówno zachłannych jak i losowych akcji, które doprowadzą pojazd wyżej, w przeciwnym razie wiele stanów nigdy nie pojawi się w epizodach treningowych.



Rysunek 3.3: Środowisko MountainCar-v0 biblioteki OpenAi Gym

Epizody ϵ - zachłanne

Parametrem wyznaczającym proporcje akcji eksploracyjnych i eksploatacyjnych w trakcie epizodów treningowych oznacza się $\epsilon \in [0, 1]$. Zanim zostanie podjęta akcja losowana jest pewna liczba u z przedziału $[0, 1]$ wedle rozkładu jednostajnego. Jeżeli $u < \epsilon$ agent podejmuje losową akcję (eksploracja). W przeciwnym przypadku gdy $u \geq \epsilon$ podejmowana jest najlepsza akcja wedle dotychczas wyuczonej strategii (eksploatacja). Zazwyczaj na początku nauki $\epsilon = 1$, aby za pomocą losowych akcji dokonać eksploracji środowiska. W dalszych etapach często opłacalnym jest zmniejszenie wartości ϵ [21].

3.4 Metoda Q-learning

Q learning jest prostym algorytmem uczenia ze wzmocnieniem rozwiązujący problem Markowa [30]. Jest to algorytm nie wymagający modelu środowiska (model-free). Wykazane zostało [20], że dla skończonych procesów Markowa algorytm Q learning odnajduje strategię, która jest optymalna w sensie maksymalizacji wartości oczekiwanej nagród (3.1). Nazwa metody wzięła się od wartości $Q(s, a)$, która określa opłacalność podjęcia akcji a w stanie s [19]. Algorytm przedstawiony jest poniższym pseudokodem.

1. Zainicjuj słownik $Q(s, a)$ przypisany do jednego agenta. Jego wartości określają opłacalność wyboru akcji a w stanie s .
2. Zasymuluj ϵ - zachłanny epizod treningowy. Obserwuj jakie zostały podjęte akcje a w stanie s i do jakiego stanu s' one doprowadziły oraz jaka została przyznana nagroda r .
3. Dla każdej obserwacji (s, a, s', r) , która pojawiła się w epizodzie przypisz nowe wartości $Q(s, a)$:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a' \in A_{s'}} Q(s', a')] \quad (3.5)$$

Zalety

Warto zauważyć następujące zalety metody Q-learningu:

1. Brak potrzeby wiedzy na temat modelu środowiska - t.j. rozkładu prawdopodobieństwa konsekwencji akcji.
2. Nie wymaga często nierealnego założenia, iż stan środowiska może zostać sztywno ustalony przez programistę. Wymagana jest jedynie możliwość przeprowadzania pełnych epizodów, co jest nieporównywalnie słabszym założeniem.
3. W przypadku dobrze wybranego parametru ϵ dostarczane są adekwatne dane do nauki agenta. Czas treningowy jest poświęcony dla stanów, które są często odwiedzane. Z kolei te, które nie pojawiają się prawie wcale - nie marnują czasu w procesie nauki.
4. Daje pewność maksymalizacji zdyskontowanej wartości oczekiwanej nagród.

Skalowanie metody

Dla bardziej złożonych środowisk - z dużą przestrzenią stanu i akcji powyższy algorytm często okazuje się niewystarczający [11]. Należy zauważyć, że bardzo podobne stany nie mają ze sobą w tym algorytmie żadnej korelacji. Gdy przestrzeń jest bardzo duża, ograniczenia czasowe mogą nie pozwolić na eksplorację każdego stanu, co skutkuje dużą ilością wartości $Q(s, a)$ niezmienionych od momentu inicjacji słownika. Jako przykład problemu niech stanem środowiska będzie kolejka na 3 odcinkach przed skrzyżowaniem. Wtedy dwa bardzo podobne stany, na pozór wręcz identyczne [3, 22, 72] oraz [3, 22, 73] są od siebie różne i w trakcie treningu należy prześledzić oddzielnie każdy z tych stanów. Zakładając, że wartości każdej zmiennej należą do zbioru $\{0, 1, \dots, 100\}$ liczność przestrzeni stanu jest równa aż 1 000 000. Lepiej skalującym rozwiązaniem jest zastosowanie sieci neuronowych zamiast tablic. Zostanie ono przedstawione w następnej sekcji.

3.5 Metoda Q-learning z użyciem sieci neuronowej

Rozwiązaniem problemu skalowalności Q-learningu jest zastosowanie sieci neuronowych do aproksymacji wartości Q zamiast słowników. Sieć neuronowa tworzy funkcje rzeczywistą, która ma za zadanie aproksymację wartości Q . W przeciwieństwie do słowników funkcja ta może przybliżać też wartości $Q(s, a)$ dla stanów i akcji, które nie pojawiły się w trakcie symulacji treningowych. Algorytm uczenia jest wtedy następujący:

1. Stwórz sieć neuronową dla wartości Q dla każdego agenta (chyba, że już została wcześniej utworzona)
2. Zasymuluj ϵ - zachłanne epizody treningowe. Obserwuj jakie zostały podjęte przez agentów akcje a w stanie s i do jakiego stanu s' one doprowadziły oraz jaka została przyznana nagroda r .
3. Trenuj sieć neuronową każdego agenta na podstawie danych wygenerowanych w trakcie symulacji treningowych.
4. Jeśli nie jest spełniony warunek stopu zacznij nową iterację wracając do kroku 2, w przeciwnym razie zakończ algorytm.

Krok 3 powyższego algorytmu wymaga dokładnego wyjaśnienia.

3.5.1 Wygenerowane dane podczas symulacji treningowych

Dla każdej z obserwacji (s, a^*, s', r) wygenerowany zostaje element (\mathbf{x}, \mathbf{y}) , gdzie \mathbf{x} jest wektorem utworzonym na podstawie stanu s i nazywany będzie **sygnałem wejściowym**. Powinien on zawierać jedynie istotne dla agenta informacje na temat stanu środowiska. Z kolei \mathbf{y} jest wektorem przedstawiającym wartość Q dla rozważanego stanu s i wszystkich możliwych akcji a . Nazywany on będzie **sygnałem wyjściowym**. Dla akcji z obserwacji, czyli $a = a^*$ wartość $y[a]$ jest wyliczana ze znanego już wcześniej wzoru (3.5):

$$y[a] = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a' \in A'} Q(s', a')] \quad (3.6)$$

Gdzie wartości $Q(s, a), Q(s', a')$ są aproksymacjami uzyskanymi przez sieć neuronową.

Dla pozostałych akcji $a \neq a^*$ wartość $y[a]$ jest uzyskana z dotychczasowej aproksymacji sieci neuronowej dla stanu s i akcji a . Tak utworzone elementy (x, y) ze wszystkich obserwacji tworzą zbiór na podstawie którego przebiega trening.

3.5.2 Trening sieci neuronowej

Wygenerowany wcześniej zbiór jest dzielony na dwie części - treningową i walidacyjną. Nie wgłębiając się szczegółowo w strukturę sieci neuronowej - posiada ona parametry nazywane wagami. Wagi te definiują funkcję, która jako argument przyjmuje wartości x . Wagi są zmieniane w trakcie treningu tak aby funkcja najlepiej przybliżała wartości y dla odpowiadających im argumentom x ze zbioru treningowego. Sposoby zmian wag są zazwyczaj oparte na metodach gradientowych [23]. W cytowanym artykule znajduje się przegląd najczęściej używanych metod gradientowych. Warto zauważyć, że zmiana wag jest podyktowana poprzez elementy zbioru treningowego, a nie zbioru walidacyjnego. Jednym z przeznaczeń zbioru walidacyjnego jest sprawdzenie jak dobrze sieć neuronowa prognozuje wartości z poza zbioru treningowego (ekstrapolacja). Często praktyką jest przerwanie trenowania sieci neuronowej (optymalizacji wag) w chwili gdy dla danych zbioru walidacyjnego otrzymuje się coraz gorsze aproksymacje. Jest to znak, że sieć neuronowa przestaje generalizować, a zaczyna zapamiętywać wartości treningowe. Jest to nieporządane zachowanie znane jako nadmierne dopasowanie.

3.5.3 Modyfikacja wzoru Q dla fazy żółtych świateł

W rozdziale 5 przedstawione środowiska sieci dróg oraz model działania sygnalizacji świetlnych. W kontekście algorytmu Q-learningu rozważony zostanie przypadek, gdy faza sygnalizacji świetlnej przypisanej do pewnego agenta to 'żółte'. Agent wtedy jest zobowiązany do podjęcia akcji 'żółte'. Nie potrzebne jest zatem wyliczanie wartości Q dla tej wymuszonej akcji. Aby nie obciążać sieci neuronowej do aproksymacji nieistotnych wartości zostaną podjęte następujące kroki. Jeśli dana jest wiedza na temat długości fazy żółtych świateł należy zastąpić wzór (3.5) poniższymi formułami. Niech faza żółtych świateł trwa k interwałów czasowych, wtedy dla akcji a utrzymania obecnej fazy:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a' \in A'} Q(s', a')], \quad (3.7)$$

dla akcji a zmiany obecnej fazy:

$$Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha[\gamma^k r + \gamma^{k+1} \max_{a' \in A_{s'}} Q(s', a')]. \quad (3.8)$$

Rozdział 4

Makroskopowy model ruchu

4.1 Klasyfikacja modeli ruchu drogowego

Modele ruchu drogowego mają na celu ukazanie rzeczywistego przepływu pojazdów w sposób czysto matematyczny. Ważnym kryterium doboru modelu jest przystępność jego implementacji informatycznej. Powszechnie klasyfikuje się 3 podejścia modelowe dla omawianego problemu [5] - makroskopowy, mezoskopowy oraz mikroskopowy. Czasem [15] wyróżnia się także czwarte podejście - submikroskopowe. Jest to podział ze względu na poziom modelu. Najniższy poziom i najbardziej dokładny model gwarantuje podejście mikroskopowe. Rozważa ono pojazdy indywidualnie w czasoprzestrzeni. Przyspieszenie pojazdu jest wyliczane na podstawie dynamiki (prędkości, przyspieszenia) i pozycji pojazdu bezpośrednio przed nim. Model mezoskopowy zapewnia indywidualne rozróżnienie pojazdów, jednak ich zachowanie jest wyliczane na danych zagregowanych [7]. Przykładowo pojazdy są podzielone na grupy podróżujące od pewnego punktu startowego do punktu końcowego. Inne modele [3] mezoskopowe wyliczają dynamikę ruchu na podstawie aktualnego zatłoczenia drogi. Poziom mezoskopowy jest obliczeniowo bardziej opłacalny od mikroskopowego. Wiele symulatorów stosujących model mezoskopowy oferuje symulację w czasie rzeczywistym dla sieci dróg całego miasta [28]. Ideą modelu makroskopowego jest traktowanie ruchu ulicznego identycznie jak ruchu cieczy lub gazów. Po raz pierwszy w roku 1956 M. J. Lighthill i G. B. Whitham [17] przedstawili pomysł przyrównania ruchu ulicznego na zatłoczonych drogach do przepływu wody w rzekach. Z tego powodu nie uznajemy w nim pojazdów jako niepodzielne jednostki. Pojazd można przyrównać do pewnej ilości wody w rzece. Jest to najmniej kosztowny obliczeniowo model. Właśnie w modelu makroskopowym zostało stworzone środowisko symulacyjne i dlatego zostanie on dokładnie przedstawiony w tym rozdziale.

4.2 Wstęp

Jako zmienną stanu makroskopowego modelu ruchu zazwyczaj wybierana jest gęstość ruchu [5, 12]. Formalnie gęstość można rozumieć jako czynnik definiujący dynamikę ruchu. Im większa jest gęstość tym mniejsza prędkość ruchu. Gęstość ruchu [14] jest przedstawiona jako iloraz ilości pojazdów znajdujących się na pewnym odcinku i długości tego odcinka drogi. Prawidłowe jest także uznanie ilości pojazdów na pojedynczym odcinku drogi jako zmienną

stanu. Należy jednak pamiętać, że pojazdy w ruchu makroskopowym nie są nierozłączną jednostką. Pojazd w analogii do przepływu wody w rzece jest pewną jednostką wody. Wielokrotnie w symulacjach zdarza się, że na odcinku jest zatem niecałkowita liczba pojazdów.

4.3 Rozwój gęstości ruchu na drodze

Makroskopowy model ruchu jest oparty o równanie różniczkowe (4.2) wraz z warunkiem początkowym (4.1) [12]. W analogii do przepływu wody w rzece gęstość ruchu można utożsamiać z polem powierzchni przekroju poprzecznego rzeki, co dla stałej szerokości rzeki - upraszcza się do wysokości wody w rzece.

Dla ustalonej pojedynczej drogi zmianę gęstości ruchu definiuje następujący układ równań:

$$\begin{cases} p(x, 0) = p_0(x) & (4.1) \\ \frac{\partial p(x, t)}{\partial t} + \frac{\partial f(p(x, t))}{\partial x} = 0 & (4.2) \end{cases}$$

Gdzie $p(x, t)$ to gęstość ruchu w punkcie x i czasie t . Wartość funkcji gęstości należy do przedziału $[0, p^{max}]$.

Równanie (4.1) zakłada istnienie pewnej z góry nałożonej początkowej gęstości drogi $p_0(x)$. Równanie (4.2) określa wedle założeń modelu makroskopowego [17] rozwój gęstości ruchu na drodze. Funkcja płynności ruchu f powinna być wklęsła [12].

4.4 CTM - Cell Transmission Model

4.4.1 Wprowadzenie

W tym rozdziale zostanie przedstawiony model CTM będący dyskretyzacją pierwotnego modelu makroskopowego przepływu [17]. Jego zmienną stanu jest ilość pojazdów. Model został opisany w artykule Carlosa Daganzo [8] w 1992. Ta pozycja jest podstawą wyliczeń w tym rozdziale.

Siatka czasowa i przestrzenna drogi

Dyskretny charakter modelu CTM obliguje do określenia siatki czasowej i przestrzennej drogi. Dla par czasu i miejsc należących do tych dwóch siatek będą określone zmienne stanu.

Siatka czasowa jest zdefiniowana jako skończony ciąg liczb naturalnych:

$$(0, 1, \dots, T). \quad (4.3)$$

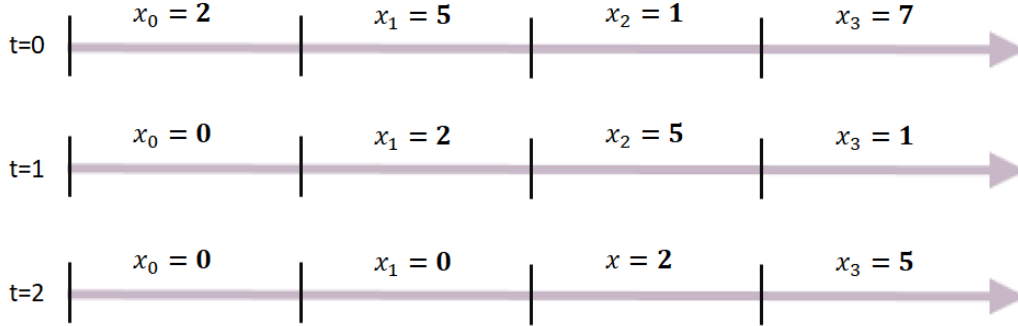
Następnie zdefiniowana zostanie siatka przestrzenna drogi e . Ustalona droga e to odcinek $[0, L]$. Zostaje ona podzielona na L odcinków o równej długości. **Siatka przestrzenna** drogi to ciąg odcinków:

$$([0, 1), [1, 2), \dots, [L - 1, L]).$$

4.4.2 Model CTM dla pojedynczej drogi

Przepływ pojazdów na drodze

Niech dana będzie droga e i pewien odcinek j na tej drodze. Zmienna stanu w modelu CMT jest oznaczana jako $x_j(t)$ i określa ona ilość pojazdów na odcinku j w chwili t . Przepływ pojazdów opiera się o założeniu, iż przy braku zatorów pojazdy pokonują w jednym interwale czasowym jeden odcinek drogi. Przykładowy przepływ jest następujący:



Rysunek 4.1: Przykładowy przepływ bez zatorów w modelu CTM

Łatwo zauważyć, że w tym przypadku

$$x_{j+1}(t+1) = x_j(t) \quad (4.4)$$

Wprowadzenie zatorów

Poprzedni przykład ruchu pojazdów opierał się o założenie, iż wszystkie pojazdy przemieszczają się o jeden odcinek w jednym interwale czasowym. Często jednak część pojazdów będzie musiała pozostać na dotychczasowym odcinku z powodu zbyt dużego natężenia ruchu. Taka sytuacja będzie nazywana **zatorem**. Model CMT opiera się o następujące parametry definiujące sposób formowania się zatorów:

- $N_j(t)$ - maksymalna ilość pojazdów, które mogą znajdować się na odcinku j w chwili t .
- $Q_j(t)$ - maksymalna ilość pojazdów, które mogą napłynąć do odcinka j w momencie $t+1$.

Rozwój zmiennej stanu jest zadany poprzez następujące równanie:

$$x_j(t+1) = x_j(t) + y_j(t) - y_{j+1}(t) \quad (4.5)$$

Gdzie:

- $x_j(t+1)$ to ilość pojazdów na odcinku j w chwili $t+1$
- $x_j(t)$ to ilość pojazdów na odcinku j w chwili t

- $y_j(t)$ to ilość pojazdów napływających do odcinka j w chwili $t + 1$.
- $y_{j+1}(t)$ to ilość pojazdów napływających do odcinka $j + 1$ w chwili $t + 1$. Jest to zatem także ilość pojazdów opuszczających odcinek j w interwale $[t, t + 1)$.

Przepływ pojazdów $y_j(t)$ jest zdefiniowany jako minimum z następujących trzech wartości:

- $x_{j-1}(t)$ - ilość pojazdów na poprzedzającym odcinku (t.j. $j - 1$)
- $Q_i(t)$ - maksymalna ilość pojazdów, które mogą napłynąć do odcinka i w chwili $t + 1$.
- $N_i(t) - x_j(t)$ - ilość wolnych miejsc na odcinku j w chwili t

Co wyraża się wzorem:

$$y_j(t) = \text{Min}\{x_{j-1}(t), Q_i(t), N_i(t) - x_i(t)\} \quad (4.6)$$

Źródło i ujście ruchu

Powyższe wzory potrzebują ustaleń odnośnie źródła i ujścia ruchu. Bez nich wzór (4.6) nie ma sensu dla krańcowych odcinków $j = 0$ oraz $j = L - 1$. Dla ułatwienia pogrubione zostały ustalone w tej sekcji wartości zmiennych źródła i ujścia ruchu.

Źródło ruchu może być traktowane jako odcinek o indeksie $j = -1$. Jediną wartością, która musi być ustalona dla źródła jest ilość pojazdów w nim, czyli $x_{j-1}(t)$. Przyjęte zostaje, że $\mathbf{x}_{j-1}(\mathbf{t}) = \infty$ - wtedy napływ pojazdów do odcinka $j = 0$ jest ograniczony już tylko przez $Q_0(t)$ oraz $N_0(t) - x_0(t)$.

Ujście ruchu z kolei może być traktowane jako odcinek $i = L$. Istotne w kontekście wzoru (4.6) są następujące wartości:

- $Q_L(t)$ - ilość pojazdów, które mogą opuścić odcinek $L - 1$ w chwili t . Ustalono jest, że jest to dowolna ilość, wtedy $\mathbf{Q}_L(\mathbf{t}) = \infty$
- $N_L(t)$ - ilość pojazdów, które może pomieścić ujście. Ustalono jest, że $\mathbf{N}_L(\mathbf{t}) = \infty$, gdyż ujście może zawierać nieograniczoną ilość pojazdów.
- $x_L(t)$ - ilość pojazdów w ujściu. Służy jako licznik pojazdów, które opuściły układ.

Przykład przepływu na pojedynczej drodze

Zostają ustalone następujące parametry implementacji przykładowego modelu

- $L = 9$ - liczba odcinków drogi
- $[N_i(t) = 15$ dla dowolnego i, t - odcinki drogi mogą pomieścić maksymalnie 15 pojazdów
- Maksymalne przepływy są następujące:

$$Q_i(t) = \begin{cases} 1 & \text{dla } i = 5, t \in \{0, \dots, 7\} \\ 5 & \text{dla } i = 5, t \in \{7, \dots, 20\} \\ 4 & \text{dla pozostałych przypadków} \end{cases} \quad (4.7)$$

$$(4.8)$$

$$(4.9)$$

Podsumowując powyższe parametry - droga posiada 9 odcinków (ponumerowanych od 0 do 8). Każdy odcinek może pomieścić maksymalnie 15 pojazdów. Maksymalny przepływ z jednego odcinka na drugi to 4. Jedynym wyjątkiem od tej zasady jest odcinek nr. 5. Początkowo może napływać na niego tylko po 1 pojeździe. Zmienia się to dopiero w chwili $t = 7$ - by rozładować powstałe zatory ustalone zostaje, że może co interwał wjeżdżać 5 pojazdów na odcinek 5. Przeprowadzona zostanie symulacja przy warunku początkowym określającym liczebność pojazdów na poszczególnych odcinkach. Niech zatem $x_i(0) = 3$ dla $i \in \{0, \dots, 8\}$. Wtedy wyniki symulacji przedstawia poniższa tabela.

		Odcinek													
		-1	0	1	2	3	4	5	6	7	8	9	Q5		
Legenda Źródło ruchu Odcinek drogi Ujście ruchu	Czas	0	999	3	3	3	3	3	3	3	3	3	0	1	
		1	999	4	3	3	3	5	1	3	3	3	3	1	
		2	999	4	4	3	3	7	1	1	3	3	6	1	
		3	999	4	4	4	3	9	1	1	1	3	9	1	
		4	999	4	4	4	4	11	1	1	1	1	12	1	
		5	999	4	4	4	4	14	1	1	1	1	13	1	
		6	999	4	4	4	7	14	1	1	1	1	14	1	
		7	999	4	4	4	10	14	1	1	1	1	15	5	
		8	999	4	4	4	13	10	5	1	1	1	16	5	
		9	999	4	4	6	11	9	6	4	1	1	17	5	
		10	999	4	4	6	11	8	7	4	4	1	18	5	
		11	999	4	4	6	11	7	8	4	4	4	19	5	
		12	999	4	4	6	11	6	9	4	4	4	23	5	
		13	999	4	4	6	11	5	10	4	4	4	27	5	
		14	999	4	4	6	11	4	11	4	4	4	31	5	
		15	999	4	4	6	11	4	11	4	4	4	35	5	
		16	999	4	4	6	11	4	11	4	4	4	39	5	
		17	999	4	4	6	11	4	11	4	4	4	43	5	
		18	999	4	4	6	11	4	11	4	4	4	47	5	
		19	999	4	4	6	11	4	11	4	4	4	51	5	
		20	999	4	4	6	11	4	11	4	4	4	55	5	

Rysunek 4.2: Tabela ilości pojazdów

4.4.3 Model CTM dla sieci dróg

Poprzednio przedstawiony wariant modelu CTM określał przepływ jedynie dla jednej drogi. W tej sekcji zostanie rozważony przypadek modelu CTM dla sieci dróg. **Zbiór wszystkich odcinków** w środowisku będzie oznaczany jako Ω .

Źródła ruchu

W przypadku większej ilości dróg niż jedna należy rozważyć inne podejście do ustalenia źródeł ruchu. Dla każdego odcinka $j \in \Omega$ zostanie ustalona wartość $u_j(t)$ **napływu ze źródeł**. Określa ona ile pojazdów ze źródła może maksymalnie napłynąć do odcinka j w chwili $t + 1$. Dla odcinków j do których nie wjeżdżają żadne pojazdy ze źródła ruchu wartość $u_j(t)$ jest równa 0. Wartości $u_j(t)$ tworzą **wektor źródła** ruchu:

$$\mathbf{u}(t) = [u_0(t), \dots, u_{|\Omega|-1}(t)]$$

Źródła ruchu są uwzględnione we wzorze ilości napływających pojazdów (4.12)

Ujścia ruchu

Niech zbiór odcinków znajdujących się bezpośrednio przed punktami ujścia ruchu będzie oznaczony jako Ω_{out} . Zgodnie z poprzednimi założeniami pojazdy będące w chwili t na odcinku bezpośrednio przed ujściem ruchu opuszczają układ w chwili $t + 1$. Zostanie to uwzględnione we wzorze (4.13).

4.4.4 Przepływ pojazdów na skrzyżowaniu

Wprowadzona zostanie modyfikacja funkcji $y_j(t)$ określającej ilość pojazdów wjeżdżających na odcinek j w chwili $t + 1$. Do tej pory dotyczyła ona jedynie ilości pojazdów które napływają do odcinka j z odcinka $j - 1$. W przypadku sieci dróg należy rozważyć możliwość wjazdu z innych odcinków niż $j - 1$. Niech zatem:

$$y_{ij}(t) = \text{Min}\{p_{ij}x_i(t), Q_j(t), N_j(t) - n_j(t)\} \quad (4.10)$$

określa ilość pojazdów, które wjeżdżają z odcinka i do odcinka j w chwili $t + 1$. Wartości p_{ij} określają jaka część pojazdów z odcinka i ma zamiar wjechać na odcinek j . Tworzą one macierz prawdopodobieństwa \mathbf{P} , która została szczegółowo opisana w sekcji (5.5.1).

Suma pojazdów opuszczających odcinek i w chwili $t + 1$ jest następująca:

$$y_{i*}(t) = \sum_{j \in \Omega/\{i\}} y_{ij}(t) \quad (4.11)$$

Z kolei napływ pojazdów do odcinka j w chwili $t + 1$ jest równy:

$$y_{*j}(t) = \sum_{i \in \Omega/\{j\}} y_{ij}(t) + u_j(t) \quad (4.12)$$

Rozwój zmiennej stanu jest zdefiniowany przez równanie:

$$x_j(t + 1) = \begin{cases} y_{*j}(t) & \text{dla } j \in \Omega_{out} \\ x_j(t) + y_{*j}(t) - y_{j*}(t) & \text{dla } j \notin \Omega_{out} \end{cases} \quad (4.13)$$

$$(4.14)$$

Dla przypomnienia wartości z powyższego równania oznaczają:

- $x_j(t + 1)$ - ilość pojazdów na odcinku j w chwili $t + 1$
- $x_j(t)$ - ilość pojazdów na odcinku j w chwili t
- $y_{*j}(t)$ - ilość pojazdów napływających do odcinka j w chwili $t + 1$
- $y_{j*}(t)$ ilość pojazdów opuszczających odcinek j w chwili $t + 1$

Przykład

Przykład przepływu pojazdów na skrzyżowaniu został zaprezentowany w sekcji (5.5.2) z zastosowaniem macierzy systemu. Wszelkie warunki początkowe i parametry środowiska z

(5.5.2) zostaną wykorzystane w tej sekcji. Jako przykład wykorzystania wzoru (4.13 - 4.14) zostanie wyliczona wartość wektora $\mathbf{x}(1)$.

$$x_0(1) = x_0(0) + y_{*0}(0) - y_{0*}(0) = 7 + 0 - 3 = 4$$

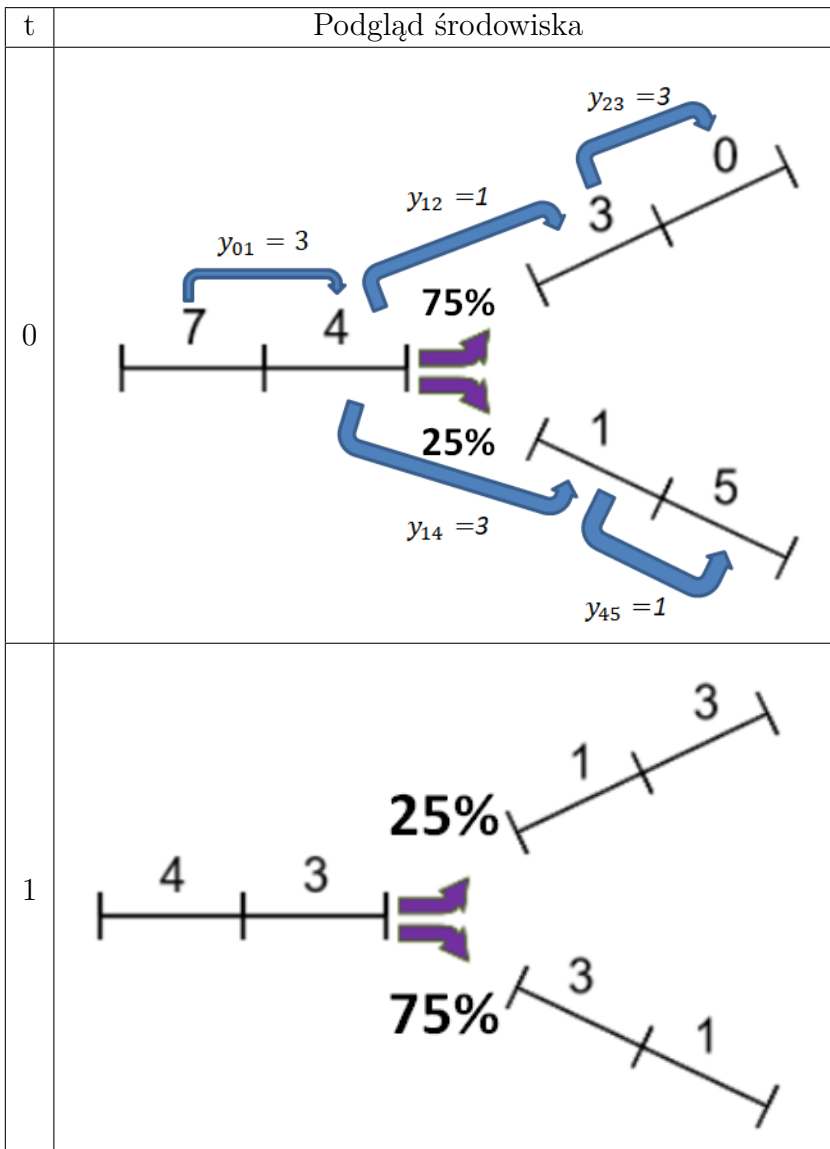
$$x_1(1) = x_1(0) + y_{*1}(0) - y_{1*}(0) = 4 + 3 - 4 = 3$$

$$x_2(1) = x_2(0) + y_{*2}(0) - y_{2*}(0) = 3 + 1 - 3 = 1$$

$$x_3(1) = y_{*3}(0) = 3$$

$$x_4(1) = x_4(0) + y_{*4}(0) - y_{4*}(0) = 3 + 3 - 3 = 3$$

$$x_5(1) = y_{*5}(0) = 1$$



4.4.5 Przepływ pojazdów na skrzyżowaniu z sygnalizacją świetlną

W tej sekcji przedstawiony zostanie przypadek przepływu na skrzyżowaniu posiadającym sygnalizację świetlną. Nie różni się on mocno od poprzednio omawianego modelu ruchu na skrzyżowaniu bez sygnalizacji świetlnej (4.4.4). Jediną różnicą jest zmiana wzoru (4.10). Zostanie on zastąpiony następującą formułą:

$$y_{ij}(t) = \text{Min}\{p_{ij}s_{ij}(t)x_i(t), Q_j(t), N_j(t) - n_j(t)\} \quad (4.15)$$

Gdzie: $s_{ij}(t)$ jest wartością określoną przez macierz sygnalizacji świetlnej $\mathbf{S}[\mathbf{j}, \mathbf{i}]$, która została opisana w sekcji (5.6).

Rozdział 5

Model sieci dróg

5.1 Wstęp

Ze względu na dużą złożoność końcowego modelu zostanie przedstawiony najpierw bardzo prosty, podstawowy model. W każdej kolejnej sekcji dodawane będą zmiany przybliżające do ostatecznej postaci. Jest to podejście pozwalające na proste przedstawienie modelu, który zawiera bardzo wiele aspektów m.in: sygnalizacji świetlnej, wielu dróg, zatoru drogowego, źródła i ujścia ruchu. Zestawienie w jednej sekcji wszystkich tych kwestii byłoby bardzo przytłaczające.

5.2 Wektor stanu drogi

Wektor stanu jest strukturą przedstawiającą stan środowiska. Dla każdego odcinka drogi składa się on wartości zmiennych stanu. Zmienna stanu jest identyfikowana jako ilość pojazdów na danym odcinku drogi.

Przykład

Niech będzie dana droga z wydzielonymi czterema odcinkami. Przykładowy wektor stanu takiego środowiska to

$$\mathbf{x}(t) = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} \quad (5.1)$$

Zawiera on w sobie następujące informacje dla chwili t :

- Są 2 pojazdy na zerowym odcinku
- Są 4 pojazdy na pierwszym odcinku
- Są 3 pojazdy na drugim odcinku
- Nie ma żadnego pojazdu na trzecim odcinku



Rysunek 5.1: Droga z ilością pojazdów na poszczególnych odcinkach

5.3 Rozwój wektora stanu drogi i wprowadzenie macierzy systemu

Formalnym wzorem definiującym rozwój wektora stanu jest:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) \quad (5.2)$$

Gdzie \mathbf{A} jest **macierzą systemu**. Wartość macierzy \mathbf{A} na przecięciu kolumny i i wiersza j określa jaka część pojazdów z odcinka i przejeżdża do odcinka j . W dalszych rozważaniach **manewr** przejazdu z odcinka i do j będzie oznaczany jako $[j, i]$.

Przykład ruchu bez zatorów

Dla przykładu przedstawionego w (5.2) zostanie przedstawiony rozwój wektora stanu. Niech zatem

$$\mathbf{x}(0) = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} \quad (5.3)$$

Macierzą systemu zapewniającą przejazd bez zatorów jest:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.4)$$

Co można zdefiniować także jako:

$$A[j, i] = \begin{cases} 1 & \text{dla manewrów } [j, i] = \{[1, 0], [2, 1], [3, 2]\} \\ 0 & \text{dla pozostałych manewrów} \end{cases} \quad (5.5)$$

$$(5.6)$$

Wedle wzoru (5.2) wyliczone zostają kolejne wartości wektora stanu.

$$\mathbf{x}(1) = \mathbf{A}\mathbf{x}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 3 \end{bmatrix} \quad (5.7)$$

$$\mathbf{x}(2) = \mathbf{Ax}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 4 \end{bmatrix} \quad (5.8)$$

$$\mathbf{x}(3) = \mathbf{Ax}(2) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} \quad (5.9)$$

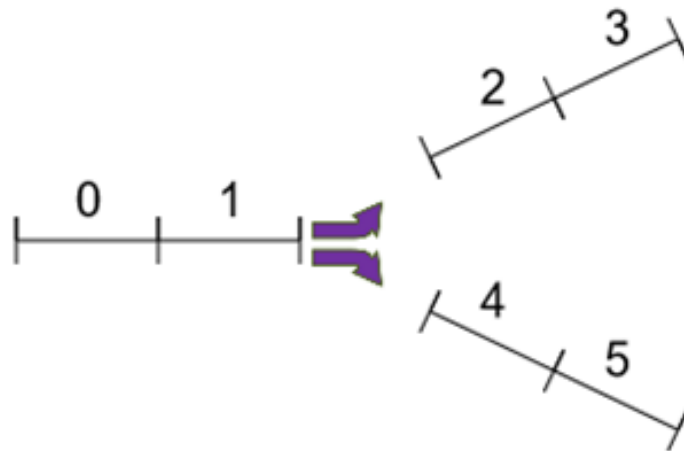
$$\mathbf{x}(4) = \mathbf{Ax}(3) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.10)$$

5.4 Wektor stanu sieci dróg

W rozdziale (5.2) przedstawiony został wektor stanu dla pojedynczej drogi. W tym rozdziale zostanie sformułowany wektor stanu dla bardziej ogólnego przypadku - sieci dróg. Zbiór wszystkich odcinków w układzie jest oznaczany jako Ω . Niejednokrotnie dwa odcinki $i, j \in \Omega$ będą leżały na innych drogach. Sposób przedstawienia wartości stanu jednak jest bardzo podobny do tego z poprzedniej sekcji. Wektor stanu jest utworzony z wartości stanu wszystkich odcinków należących do układu.

Przykład

Niech będzie dana sieć składająca się z trzech dróg. Dla każdej drogi zostaną wydzielone 2 odcinki. W sumie środowisko posiada 6 odcinków. Odcinki są ponumerowane od 0 co przedstawia poniższy rysunek.

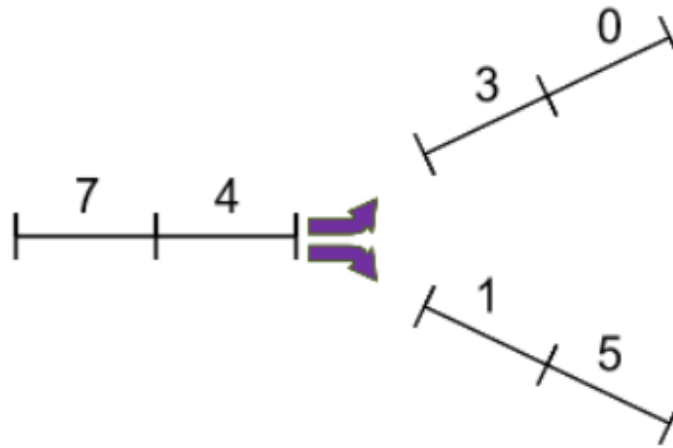


Rysunek 5.2: Numeracja odcinków środowiska

Niech przykładowym wektorem stanu będzie:

$$\mathbf{x}(t) = \begin{bmatrix} 7 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix} \quad (5.11)$$

Zawiera on w sobie informacje dotyczące ilości pojazdów na poszczególnych odcinkach w chwili t . Poniższy obraz przedstawia ilości pojazdów na poszczególnych odcinkach dla stanu zadanego przez wektor $\mathbf{x}(t)$.



Rysunek 5.3: Sieć dróg z ilościami pojazdów na poszczególnych odcinkach

5.5 Rozwój wektora stanu sieci dróg w oparciu o CTM

Przedstawiony zostanie rozwój stanu środowiska sieci dróg. Jako model ruchu wykorzystany zostanie model CTM opisany w sekcji (4.4). W przypadku sieci dróg należy uwzględnić przypadek gdy pojazdy mogą obrać różne kierunki ruchu (np. na skrzyżowaniu). W tym celu przedstawiona zostanie macierz prawdopodobieństwa.

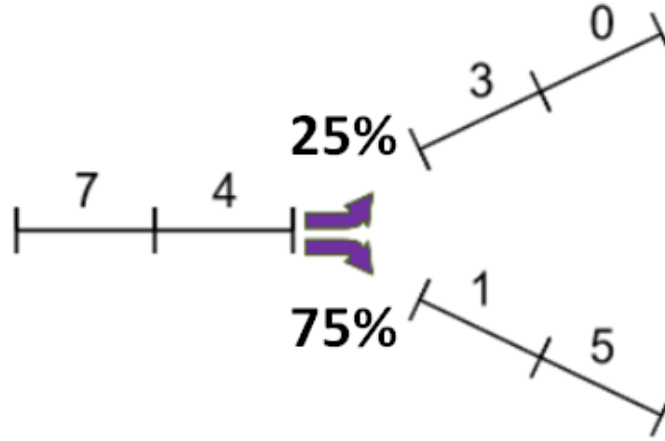
5.5.1 Macierz prawdopodobieństwa

Macierz prawdopodobieństwa \mathbf{P} jest niezmienną w czasie rzadką macierz. Wartości macierzy $P[j, i]$ określają jaką część pojazdów będących na odcinku i ma zamiar przejazdu do odcinka j . Dla niemożliwych manewrów $[j, i]$ wartości macierzy P są równe 0.

Przykład macierzy prawdopodobieństwa

Rozważmy poprzednią sieć dróg z dodatkowym założeniem, że 25 procent pojazdów ma zamiar dokonać manewru skrętu w lewo. Pozostałe 75 procent pojazdów z tego odcinka

wybiera jazdę w prawo.



Rysunek 5.4: Sieć dróg z prawdopodobieństwami przejazdów

Wtedy macierz prawdopodobieństwa jest następująca:

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.12)$$

Dodatknie wartości macierzy P są następujące:

- $P[1, 0] = 1$ - manewr jazdy prosto z odcinka 0 do 1
- $P[3, 2] = 1$ - manewr jazdy prosto z odcinka 2 do 3
- $P[5, 4] = 1$ - manewr jazdy prosto z odcinka 4 do 5
- $P[2, 1] = 0.25$ - manewr jazdy w lewo z odcinka 1 do 2
- $P[4, 1] = 0.75$ - manewr jazdy w prawo z odcinka 1 do 4

Wartości macierzy prawdopodobieństwa dla pierwszych trzech manewrów są równe 1, gdyż każdy pojazd na odcinku 0 ma zamiar przejechać na odcinek 1 (analogicznie z 2 na 3 oraz z 4 na 5). 25 procent pojazdów z odcinka 1 ma zamiar jazdy w lewo do odcinka 2. Z kolei pozostałe 75 procent chce obrać kierunek jazdy w prawo do odcinka 4. Kolumny odpowiadające odcinkom 3 i 5 mają wartości zerowe. Jest to uzasadnione tym, że pojazdy będące na tych odcinkach opuszczają układ.

5.5.2 Macierz systemu

Przedstawiona w tej sekcji macierz systemu \mathbf{A} powinna uwzględnić przepływy pojazdów na skrzyżowaniach oraz model ruchu CTM. Opisy modelu ruchu CTM oraz oznaczeń użytych w poniższej definicji są zawarte w rozdziale (4.4). Macierz systemu z następującymi wartościami gwarantuje przepływ zgodny z modelem ruchu CTM:

$$A[j, i](t) = \begin{cases} \frac{y_{ij}(t)}{x_i(t)} & \text{dla } i \neq j, i \notin \Omega_{out} \\ \frac{x_i(t) - y_{i*}(t)}{x_i(t)} & \text{dla } i = j, i \notin \Omega_{out} \\ 0 & \text{dla } i \in \Omega_{out} \end{cases} \quad (5.13)$$

$$(5.14)$$

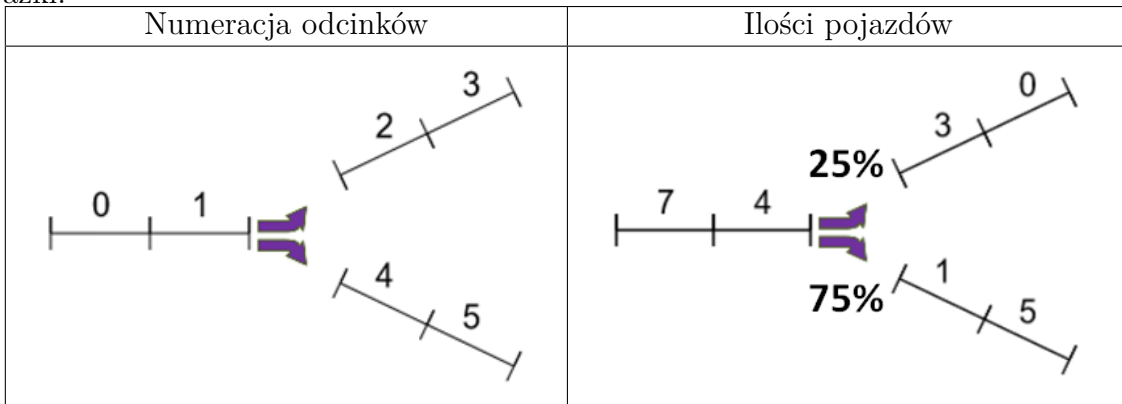
$$(5.15)$$

Powyższy wzór wymaga dokładnego omówienia. Jest on zgodny z początkową definicją macierzy systemu (5.3). Dla przypomnienia wartości macierzy systemu są określone jako część pojazdów będących na odcinku i dokonujących manewru $[j, i]$. Zgodność wszystkich 3 przypadków jest omówiona poniżej:

- (5.13) - Wartość $\frac{y_{ij}(t)}{x_i(t)}$ to stosunek pojazdów, które przejechały z odcinka i do odcinka j (t.j. $y_{ij}(t)$) do całkowitej liczby pojazdów na odcinku i (t.j. $x_i(t)$).
- (5.14) - Wartość $y_{i*}(t)$ określa ilość pojazdów, które opuściły odcinek i . Zatem licznik $x_i(t) - y_{i*}(t)$ to ilość pojazdów, które pozostały na odcinku i .
- (5.15) - Wszystkie pojazdy z odcinka i bezpośrednio przed ujściem przejeżdżają do ujścia. Zatem nie wjeżdżają do żadnego innego odcinka $j \in \Omega$,

Przykład

Rozważone zostanie środowisko (5.4) z założeniem, że 75 procent pojazdów ma zamiar skręcić w prawo. Pozostałe 25 procent wybiera skręt w lewo. Środowisko przedstawiają poniższe obrazki.



Powyższy sposób przepływu przez skrzyżowanie definiuje macierz prawdopodobieństwa \mathbf{P} określoną w przykładzie (5.12). Ustalono, że do sieci nie napływają żadne pojazdy, zatem wartości wektora źródła są równe 0 co jest zapisane jako $\mathbf{u}(t) = \mathbf{0}$ dla dowolnego t . Ustalono, że maksymalna ilość pojazdów będąca na odcinku to $N_j(t) = 7$ dla dowolnego i, t . Przepływ pojazdów w jednym interwale jest ograniczony przez $Q_j(t) = 4$. Wtedy następująca tabela przedstawia rozwój środowiska:

t	Równanie stanu	Podgląd środowiska
0	$\mathbf{x}(0) = \begin{bmatrix} 7 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix}$	
1	$\mathbf{x}(1) = \mathbf{A}(0)\mathbf{x}(0) = \begin{bmatrix} \frac{4}{7} & 0 & 0 & 0 & 0 & 0 \\ \frac{3}{7} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 7 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 3 \\ 3 \\ 1 \end{bmatrix}$	
2	$\mathbf{x}(2) = \mathbf{A}(1)\mathbf{x}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 1 \\ 3 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ \frac{3}{4} \\ 1 \\ 2\frac{3}{4} \\ 3 \end{bmatrix}$	
3	$\mathbf{x}(3) = \mathbf{A}(2)\mathbf{x}(2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 4 \\ \frac{3}{4} \\ 1 \\ 2\frac{3}{4} \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 6 \\ 0 \\ 0 \\ 2 \end{bmatrix}$	

5.6 Wprowadzenie sygnalizacji świetlnej

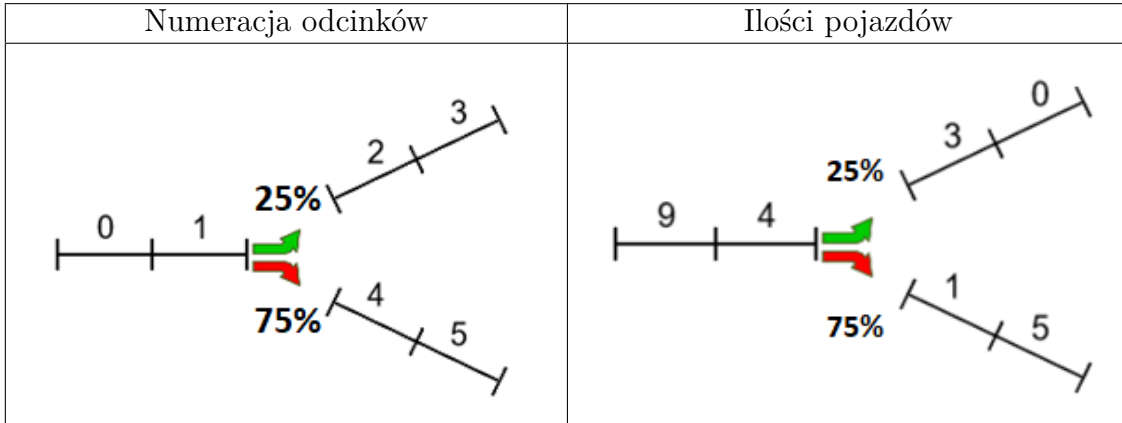
Kolejnym etapem rozwoju modelu jest wprowadzenie sygnalizacji świetlnej. Warto zauważyć, że do tej pory rozważane układy były pozbawione jakiegokolwiek sterowania. Zostanie teraz zdefiniowana macierz sygnalizacji świetlnej $\mathbf{S}(t)$. Jest ona zmienna w czasie, gdyż odpowiada za sterowanie sygnalizacją świetlną. Wartości macierzy sygnalizacji $S[j, i](t)$ określają, czy manewr $[j, i]$ jest wykonalny w chwili t .

$$S[j, i](t) = \begin{cases} 1 & \text{dla manewru zezwolonego przez zielone światło} & (5.16) \\ 1 & \text{dla manewru nie wymagającego sygnalizacji świetlnej} & (5.17) \\ 0 & \text{dla manewru wstrzymanego przez czerwone światło} & (5.18) \\ 0 & \text{dla niemożliwego manewru} & (5.19) \end{cases}$$

Sygnalizacja świetlna zwykle posiada pewną fazę, która określa dozwolone (5.16) oraz nie-dozwolone (5.18) manewry. Chociaż macierz sygnalizacji świetlnej jest zmienna w czasie to wartości macierzy \mathbf{S} dla manewrów niemożliwych (5.19) oraz nie wymagających sygnalizacji świetlnej (5.17) są niezmiennie dla ustalonego środowiska.

Przykład macierzy sygnalizacji świetlnej

Rozważone będzie znane z poprzednich przykładów skrzyżowanie - tym razem mające sygnalizację świetlną z aktywną fazą zezwalającą na manewr $[2, 1]$. Ta sama faza zabrania przejazdu z 1 odcinka do 4.



Poszczególne manewry są następujące:

- (5.16) - Manewrem zezwolonym przez zielone światło dla fazy 0 jest $[2, 1]$.
- (5.17) - Prawidłowym manewrem bez sygnalizacji świetlnej są manewry $[1, 0]$, $[3, 2]$, $[5, 4]$.
- (5.18) - Dla fazy świetlnej 0 manewrem zatrzymanym przez czerwone światło jest $[4, 1]$.
- (5.19) - Pozostałe manewry są niemożliwe. Przykładem jest $[2, 0]$, gdyż nie ma możliwości bezpośredniego przejazdu z odcinka 0 do 2.

Macierz sygnalizacji świetlnej dla tego przykładu jest następująca:

$$\mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \mathbf{0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

W dalszych rozważaniach **zielony kolor** wartości macierzy sygnalizacji świetlnej odnosi się do zielonego światła dla manewru zadanego przez wiersz j i kolumnę i . Analogicznie **czerwony kolor** oznacza czerwone światło.

Przykład rozwoju wektora stanu

Przepływ ruchu odbywa się zgodnie z opisanym w (4.4.5) przypadku modelu CTM z sygnalizacją świetlną. Jako parametry $s_{ij}(t)$ przypisane są wartości macierzy sygnalizacji świetlnej $S(t)[j, i]$. Formuła macierzy systemu przedstawiona w (5.13-5.15) pozostaje niezmienna. Niech będzie dany układ z przykładu (5.6) z zielonym światłem dla lewoskrętu. W chwili $t=2$ zostanie zmieniona faza świetlna. Od tego momentu obydwa manewry na skrzyżowaniu są dozwolone.

t	Równanie stanu	Podgląd środowiska
0	$\mathbf{x}(0) = \begin{bmatrix} 9 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix}$	
1	$\mathbf{x}(1) = \mathbf{A}(0)\mathbf{x}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{3}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 9 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 12 \\ 3 \\ 0 \\ 1 \\ 1 \end{bmatrix}$	
2	$\mathbf{x}(2) = \mathbf{A}(1)\mathbf{x}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & \frac{3}{4} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 12 \\ 3 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 9 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix}$	
3	$\mathbf{x}(3) = \mathbf{A}(2)\mathbf{x}(2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{3}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 9 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2\frac{1}{4} \\ 3 \\ 7\frac{3}{4} \\ 0 \end{bmatrix}$	

W chwili $t=3$ wartości stanu są niecałkowite, co nie jest sprzeczne z modelem makroscopowym. Rozwój wektora stanu został przedstawiony w poniższej tabeli. Macierz sygnalizacji

światłej dla $t = 0, 1$ jest następująca:

$$S(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \mathbf{0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.20)$$

Z kolei dla $t = 2, 3$:

$$S(t) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.21)$$

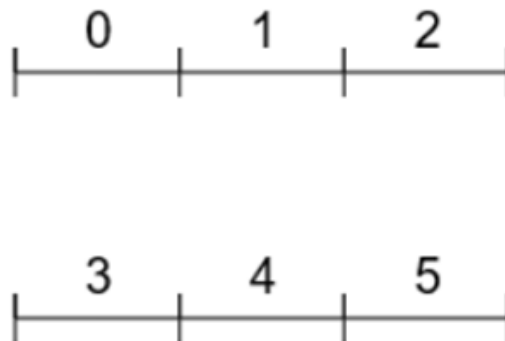
5.7 Wprowadzenie źródeł ruchu

Wszystkie poprzednie przykłady układów ruchu drogowego szybko kończyły się stanem w którym nie było już żadnych pojazdów na drogach. W tej sekcji zostanie przedstawiony sposób napływania nowych pojazdów do układu. Wprowadzony zostanie wektor źródła $\mathbf{u}(t)$. Jest on zmienny w czasie, a jego wartości określają ile pojazdów pojawia się w chwili $t + 1$ na poszczególnych odcinkach układu. Równanie systemu uwzględniające źródła ruchu to:

$$\mathbf{x}(t) = \mathbf{A}(t-1)\mathbf{x}(t-1) + \mathbf{u}(t-1) \quad (5.22)$$

Przykład

Rozważony zostanie prosty przykład, który dla ułatwienia nie uwzględnia wcześniej wprowadzonych pojęć sygnalizacji świetlnej oraz zatoru. Przedstawione zostanie środowisko składające się z dwóch dróg, które nie są ze sobą w żaden sposób połączone.



Rysunek 5.5: Numeracja odcinków środowiska

Pojazdy poruszają się z odcinków 0 (3) kończąc swój bieg po przejechaniu przez odcinki 5 (2) co przedstawia macierz systemu A :

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Dla ułatwienia powyższa macierz przedstawia środowisko bez zatorów. Pragnąc by do odcinków 0 i 3 napływało odpowiednio po 4, 8, 20 oraz 3, 7, 5 pojazdów należy zdefiniować ciąg wektorów $\mathbf{u}(t)$ w sposób przedstawiony w poniższej tabeli.

t	Równanie stanu	Podgląd środowiska	$\mathbf{u}(t)$
0	$\mathbf{x}(0) = \begin{bmatrix} 0 \\ 8 \\ 1 \\ 3 \\ 3 \\ 1 \end{bmatrix}$		$\begin{bmatrix} 4 \\ 0 \\ 0 \\ 3 \\ 0 \\ 0 \end{bmatrix}$
1	$\mathbf{x}(1) = \mathbf{Ax}(0) + \mathbf{u}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 8 \\ 1 \\ 3 \\ 3 \\ 1 \end{bmatrix} + \begin{bmatrix} 4 \\ 0 \\ 0 \\ 3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 8 \\ 3 \\ 3 \\ 3 \end{bmatrix}$		$\begin{bmatrix} 8 \\ 0 \\ 0 \\ 7 \\ 0 \\ 0 \end{bmatrix}$
2	$\mathbf{x}(2) = \mathbf{Ax}(1) + \mathbf{u}(2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \\ 8 \\ 3 \\ 3 \\ 3 \end{bmatrix} + \begin{bmatrix} 8 \\ 0 \\ 0 \\ 7 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \\ 7 \\ 3 \\ 3 \end{bmatrix}$		$\begin{bmatrix} 20 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \end{bmatrix}$
3	$\mathbf{x}(3) = \mathbf{Ax}(2) + \mathbf{u}(3) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 8 \\ 4 \\ 0 \\ 7 \\ 3 \\ 3 \end{bmatrix} + \begin{bmatrix} 20 \\ 0 \\ 0 \\ 5 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 20 \\ 8 \\ 4 \\ 5 \\ 7 \\ 3 \end{bmatrix}$		

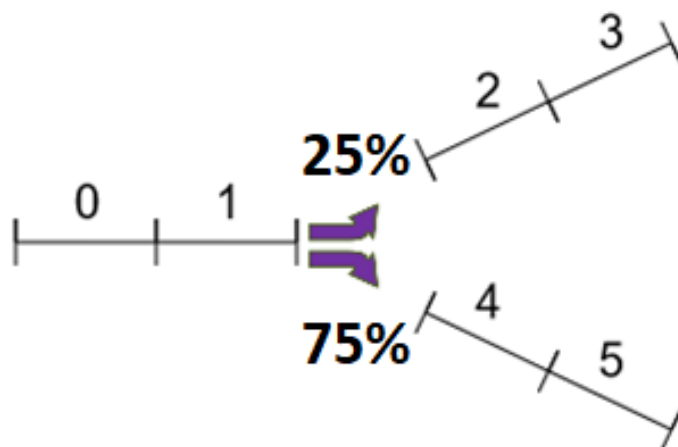
Rozdział 6

Środowiska symulacyjne i ich nauka

W tym rozdziale przedstawione zostaną środowiska symulacyjne. Głównym ich elementem jest sieć dróg. Podstawowe informacje na temat działania modelu sieci dróg i ruchu obowiązującego na nich zostały przedstawione w rozdziale 5. Każde środowisko posiada dokładną specyfikację możliwych faz świetlnych. Do każdego ze skrzyżowań jest przyporządkowany agent, który ma za zadanie podejmować akcje prowadzące do dobrej przepustowości ruchu. Dokładnie zostaną przedstawione przestrzenie decyzyjne i konsekwencje każdej z akcji. Każde środowisko będzie posiadało swój problem optymalizacyjny i przedstawiony zostanie sposób uczenia, który prowadzi do rozwiązania tego problemu.

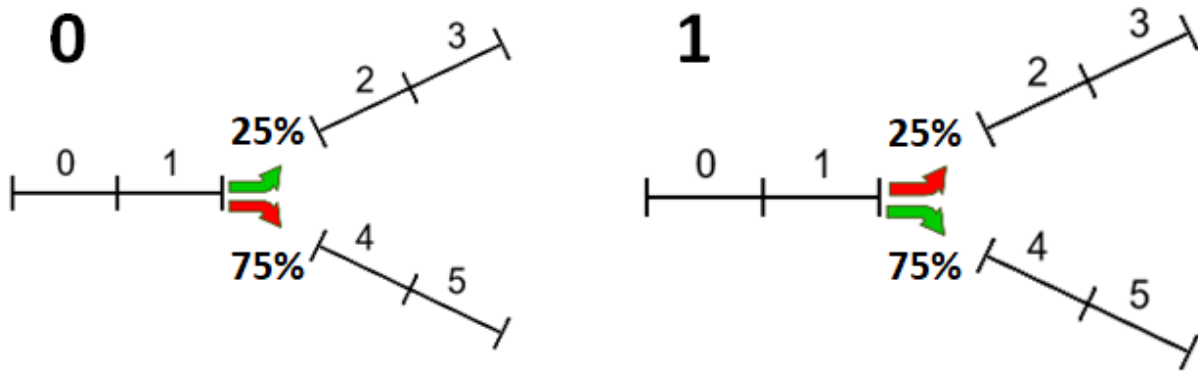
6.1 Środowisko 1

Elementy sieci dróg pierwszego środowiska były niejednokrotnie przedstawiane w rozdziale (5), jednak dopiero teraz nastąpi przedstawienie pierwszego pełnego środowiska symulacyjnego. Na model sieci składają się 3 drogi - każda ma po 2 odcinki. Jest jedno skrzyżowanie na którym 75 procent pojazdów skręca w prawo, a pozostałe obierają kierunek w lewo.



Rysunek 6.1: Środowisko 1

6.1.1 Sygnalizacje świetlne



Rysunek 6.2: Środowisko 1 - fazy świateł

Skrzyżowanie posiada 2 fazy świetlne przedstawione powyżej. Faza 0 umożliwia skręt w lewo (manewr $[1,2]$) z kolei faza 1 umożliwia skręt w prawo (manewr $[1,4]$).

6.1.2 Przestrzeń decyzyjna

Agent ma dwie możliwe akcje do podjęcia - 0 oraz 1. Obydwie powodują aktywację odpowiadającej im fazy świetlnej.

6.1.3 Model ruchu

Jako model ruchu zostanie wykorzystany przepływ CTM z następującymi parametrami:

- Ilość pojazdów na odcinku jest nieograniczona t.j. $N_j(t) = \infty$ dla dowolnych j, t .
- Maksymalna ilość pojazdów, które mogą napłynąć do odcinka to $Q_j(t) = 10$ dla dowolnego j, t .
- Ze źródła ruchu napływa w każdym interwale po 7 pojazdów do odcinka 0.
- Macierze prawdopodobieństwa P i sygnalizacji S są analogiczne do tych przedstawionych w (5.5.1) i (5.6).

6.1.4 Uczenie

Cel nauki

Celem nauki jest uzyskanie jak największej liczby pojazdów, które opuszczą rozwidlenie dróg.

Pożądane zachowanie

Pożądanym zachowaniem jest utrzymywanie fazy 1. Jest ona w każdym przypadku lepsza lub równa fazie 0. Spowodowane jest to tym, że więcej pojazdów skręca w prawo. W przypadku gdy przed rozwidleniem drogi nie ma pojazdów lub jest ich co najmniej 40 poprawną akcją jest także wybór fazy 0.

Sygnał wejściowy

Sygnał wejściowy \mathbf{x} to ilości pojazdów na odcinkach przed skrzyżowaniem.

Przykład sygnału wejściowego

Sygnał wejściowy $\mathbf{x} = [4, 7]$ zawiera w sobie informację, iż na odcinku zerowym są 4 pojazdy oraz na pierwszym - jest ich 7.

Sygnał nagrody

Nagrodą jest ilość pojazdów, które właśnie przejechały przez rozwidlenie dróg.

Sposób uczenia

Zastosowana została metoda uczenia przedstawiona w (3.5). Szczegóły dotyczące sposobu uczenia są następujące:

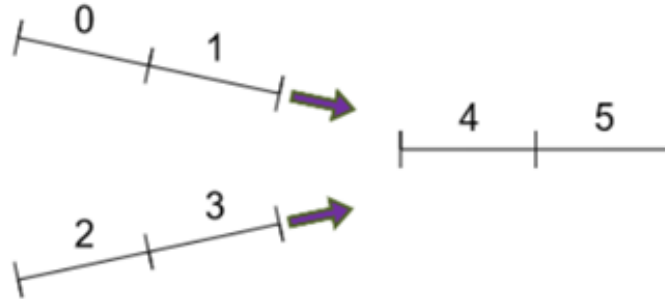
- $\gamma = 0.9$ - stopa dyskontowa
- $\alpha = 1$ - stała szybkości uczenia (ang. learning rate)
- $\epsilon = 1$ - akcje są wybierane w pełni losowy sposób podczas epizodów treningowych.
- Generowanych jest jednorazowo 50 epizodów treningowych. Każdy ma 90 interwałów czasowych.
- Wygenerowane zbiory danych są dzielone w proporcjach 80 i 20 procent dla odpowiednio treningu i walidacji.
- Warunkiem stopu jest 60 sekund działania algorytmu.
- Za uczenie odpowiada sieć neuronowa biblioteki Keras. Posiada ona 2 ukryte warstwy, każda z 5 neuronami i funkcją aktywacji 'relu'. Optymalizacja jest przeprowadzana przez metodę Adam z funkcją straty błędu średniokwadratowego i parametrem $learning_rate = 0,01$.

Wyniki uczenia

Algorytm osiąga pożądane zachowanie. Dla każdego stanu agent wybiera akcję 1. Zdarza się, że agent wybiera akcję 0 na samym początku epizodu, gdy nie ma żadnych pojazdów, co nie sprawia istotnej różnicy. Nie tworzą się zatory i wszystkie pojazdy opuszczają skrzyżowanie (oczywiście z wyjątkiem tych, które pojawiły się w dwóch ostatnich interwałach czasowych).

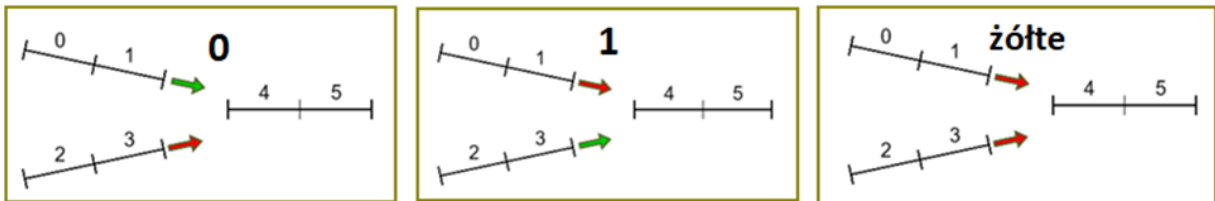
6.2 Środowisko 2

Na model sieci składają się 3 drogi - każda ma po 2 odcinki. Istnieje jedno skrzyżowanie z dwiema drogami wlotowymi i jedną wylotową.



Rysunek 6.3: Środowisko 2

6.2.1 Sygnalizacje świetlne

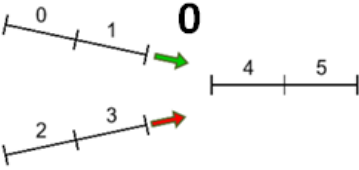
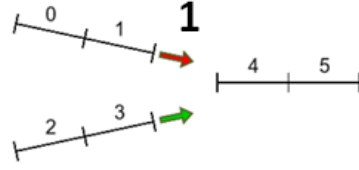
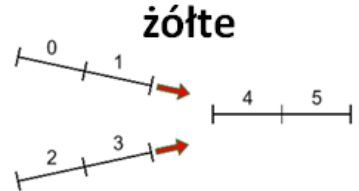


Rysunek 6.4: środowisko 2 - fazy świateł

Skrzyżowanie posiada 3 fazy świetlne przedstawione powyżej. Faza 0 umożliwia przejazd przez skrzyżowanie z górnej drogi (manewr $[1,4]$) z kolei faza 1 umożliwia wjazd z drogi dolnej (manewr $[3,4]$). Faza 'żółte' blokuje przejazd przez skrzyżowanie.

6.2.2 Przestrzeń decyzyjna

Przestrzeń decyzyjną oraz konsekwencje akcji podjętych przez agenta przedstawia poniższa tabela:

Obecna faza	Możliwe akcje	Konsekwencje akcji
	0	Podtrzymanie fazy 0
	1	2 interwały fazy żółtych świateł, a następnie faza 1
	0	2 interwały fazy żółtych świateł, a następnie faza 0
	1	Podtrzymanie fazy 1
	żółte	Podtrzymanie fazy żółtych świateł. Jeśli żółta faza trwa dłużej niż 2 interwały czasowe – włączenie oczekiwanej fazy 0 lub 1.

6.2.3 Model ruchu

Jako model ruchu zostanie wykorzystany przepływ CTM z następującymi parametrami:

- Ilość pojazdów na odcinku jest nieograniczona t.j. $N_j(t) = \infty$ dla dowolnych odcinków j i czasu t .
- Maksymalna ilość pojazdów, które mogą napłynąć do odcinka to $Q_j(t) = 10$ dla dowolnego j, t .

6.2.4 Uczenie

Ogólny zarys

Początkowo do każdej z dróg wjeżdżają po 2 pojazdy w jednym interwale czasowym. Przeprowadzona zostaje jedna iteracja nauki (3.5), po której następuje epizod testowy. Agent ma za zadanie wykonywać w trakcie niego akcje, które pozwolą na opuszczenie skrzyżowania przez co najmniej 98 procent pojazdów. W przypadku spełnienia tego warunku następuje zwiększenie ilości wjeżdżających do układu pojazdów. Następnie przeprowadzana jest kolejna iteracja uczenia (3.5). Ten proces jest powtarzany, aż do momentu gdy w trakcie 10

kolejnych epizodów testowych nie uzyskany zostanie warunek 98 procent pojazdów opuszczających skrzyżowanie.

Cel nauki

Bezpośrednim celem nauki jest uzyskanie jak największej ilości pojazdów wjeżdżających do sieci. Aby to się stało agenci muszą podejmować decyzje, które pozwalają na opuszczenie sieci jak największej ilości pojazdów.

Spodziewany wynik

Koniec nauki spodziewany jest na chwilę, gdy na każdą z dróg będzie wjeżdżało około 5 pojazdów (w sumie 10 pojazdów). Zgodnie z opisem modelu ruchu - maksymalnie 10 pojazdów może przejechać w jednej chwili przez skrzyżowanie, zatem muszą się tworzyć zatory przy większej ilości pojazdów.

Pożądane zachowanie agenta

Pożądane zachowanie agenta w tym przypadku nie jest jednoznaczne do określenia. Z pewnością jeśli na drodze na której jest zielone światło jest co najmniej 10 pojazdów, to warto podtrzymać fazę świetlną. Zmiana sygnalizacji powoduje 2 interwały karencji, podczas których żaden pojazd nie przejedzie i może to być czynnikiem decydującym o pozostaniu przy obecnej fazie. Jaka powinna być akcja w przypadku gdy zielone światło jest dla drogi z mniejszą niż 10 ilością pojazdów? Wtedy agent powinien odnaleźć złoty środek i jeśli na drugiej drodze jest dużo pojazdów - może zmienić światło.

Sygnal nagrody

Nagroda jest ilością pojazdów, które właśnie przejechały przez skrzyżowanie.

Sygnal wejściowy

Sygnal wejściowy posiada 4 wartości. Pierwsze dwie z nich to ilości pojazdów przed skrzyżowaniem. Kolejne określają aktualną fazę świetlną.

Przykłady sygnałów wejściowych

- Sygnal wejściowy $\mathbf{x} = [8, 2, 0, 1]$ zawiera w sobie informację, iż na odcinku zerowym jest 8 pojazdów oraz na pierwszym - jest ich 2. Aktywna jest faza 1.
- Sygnal wejściowy $\mathbf{x} = [7, 5, 1, 0]$ zawiera w sobie informację, iż na odcinku zerowym jest 7 pojazdów oraz na pierwszym - jest ich 5. Aktywna jest faza 0.

Sposób uczenia

Następujące stałe oraz ustalenia określają szczegóły uczenia:

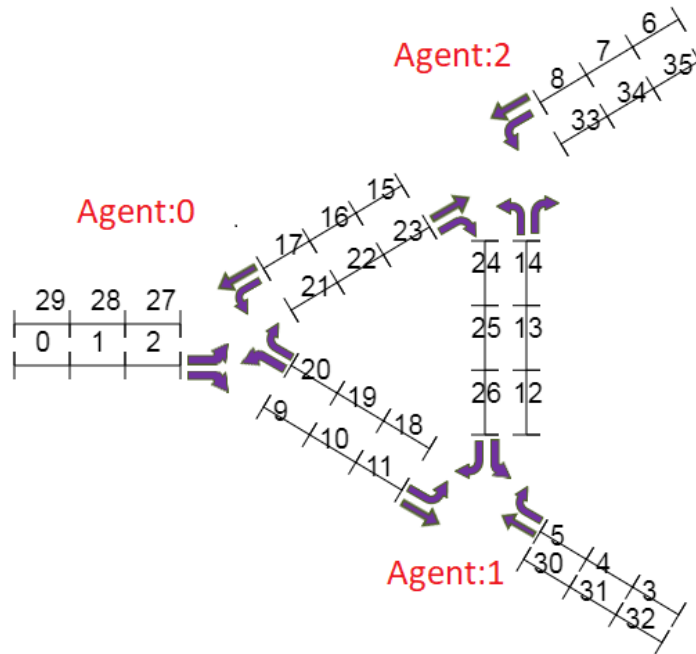
- $\gamma = 0.9$ - stopa dyskontowa

- $\alpha = 1$ - stała szybkości uczenia (ang. learning rate)
- Parametr ϵ jest początkowo równy 1 (w pełni losowe epizody). Z każdą iteracją jest zmniejszany o 0,01 aż do wartości $\epsilon = 0,2$.
- Generowanych jest jednorazowo 10 epizodów treningowych. Każdy z nich trwa 1000 interwałów czasowych.
- Wygenerowane zbiory danych są dzielone w proporcjach 80 i 20 procent dla odpowiednio treningu i walidacji.
- Za uczenie odpowiada sieć neuronowa biblioteki Keras. Posiada ona 3 ukryte warstwy (10,14,10 neuronów) z funkcjami aktywacji relu. Optymalizacja jest przeprowadzana przez metodę Adam z funkcją straty błędu średniokwadratowego i parametrem $learning_rate = 0,01$.

Wyniki uczenia

Algorytm szybko uzyskuje kolejne warunki 98 procent pojazdów przejeżdżających przez skrzyżowanie. Dopiero przy 4,97 wjeżdżających pojazdach na drogę widoczny jest zastój, który powoduje warunek stopu. Najlepszy wynik to średnio 9,48 pojazdów przejeżdżających w jednej chwili przez skrzyżowanie. Jest to dobry wynik zważając na to, iż potencjalnie najwyższy możliwy wynik do osiągnięcia jest w granicy 9,94. Z 9940 pojazdów 450 nie opuściło skrzyżowania. Zatem przy niemalże 5 pojazdach napływających do układu powstają korki, co jest naturalnym zjawiskiem - wynikającym z ograniczeń ruchu, a nie nieoptymalnej sygnalizacji świetlnej.

6.3 Środowisko 3

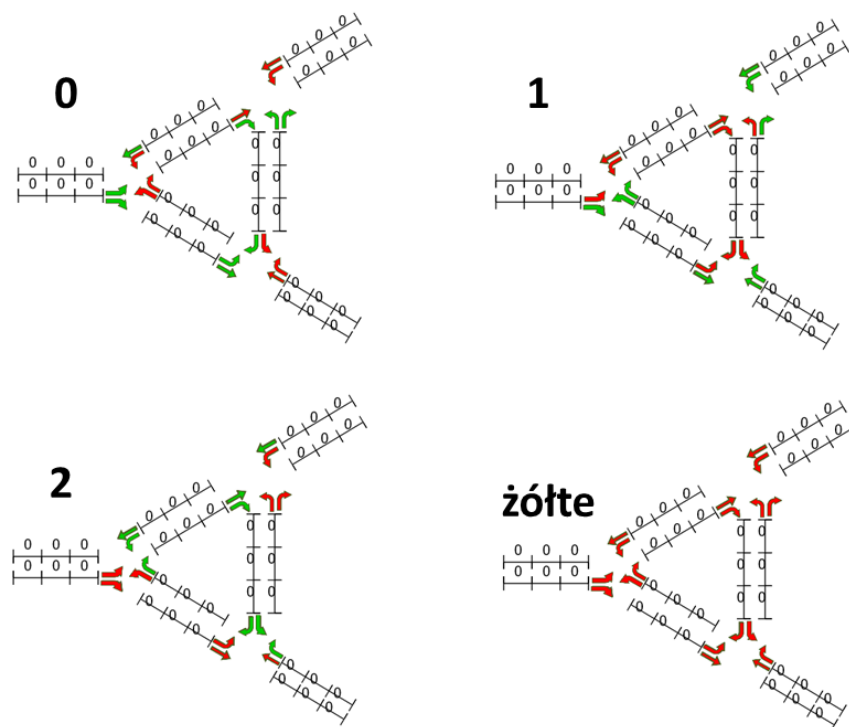


Rysunek 6.5: Środowisko 3

Środowisko posiada 12 jednokierunkowych dróg. Każda droga ma 3 odcinki co daje w sumie 36 odcinków (są numerowane od 0 co widać na rysunku 6.3). W sieci dróg znajdują się 3 skrzyżowania. Do każdego z nich jest przypisany agent, który odpowiada za sterowanie sygnalizacją świetlną.

6.3.1 Sygnalizacje świetlne

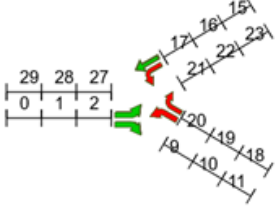
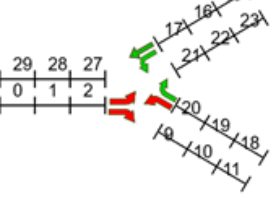
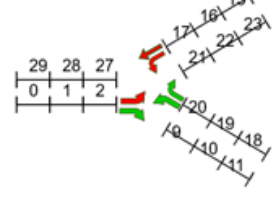
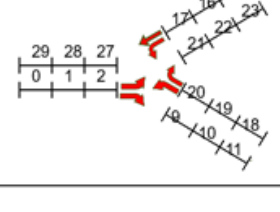
Każde skrzyżowanie posiada 4 fazy świetlne przedstawione poniżej. Fazy 0,1 i 2 są fazami, które posiadają pewne zielone światła. Zmiana pomiędzy tymi fazami nie jest natychmiastowa i następuje dopiero po 2 interwałach fazy żółtych światel.



Rysunek 6.6: Środowisko 3 - fazy świateł

6.3.2 Przestrzeń decyzyjna

Przedstawiona zostanie przestrzeń decyzyjna jedynie dla agenta 0, jednak dla pozostałych dwóch agentów przestrzenie decyzyjne są analogiczne. Schemat zmiany faz świetlnych jest identyczny, jedynie manewry faz są inne.

Obecna faza	Możliwe akcje	Konsekwencje akcji
	0	Podtrzymanie fazy 0
	1	2 interwały fazy żółtych świateł, a następnie faza 1
	2	2 interwały fazy żółtych świateł, a następnie faza 2
	0	2 interwały fazy żółtych świateł, a następnie faza 0
	1	Podtrzymanie fazy 1
	2	2 interwały fazy żółtych świateł, a następnie faza 2
	0	2 interwały fazy żółtych świateł, a następnie faza 0
	1	2 interwały fazy żółtych świateł, a następnie faza 1
	2	Podtrzymanie fazy 2
	żółte	Podtrzymanie fazy żółtych świateł. Jeśli żółta faza trwa dłużej niż 2 interwały czasowe – włączenie oczekiwanej fazy 0 lub 1.

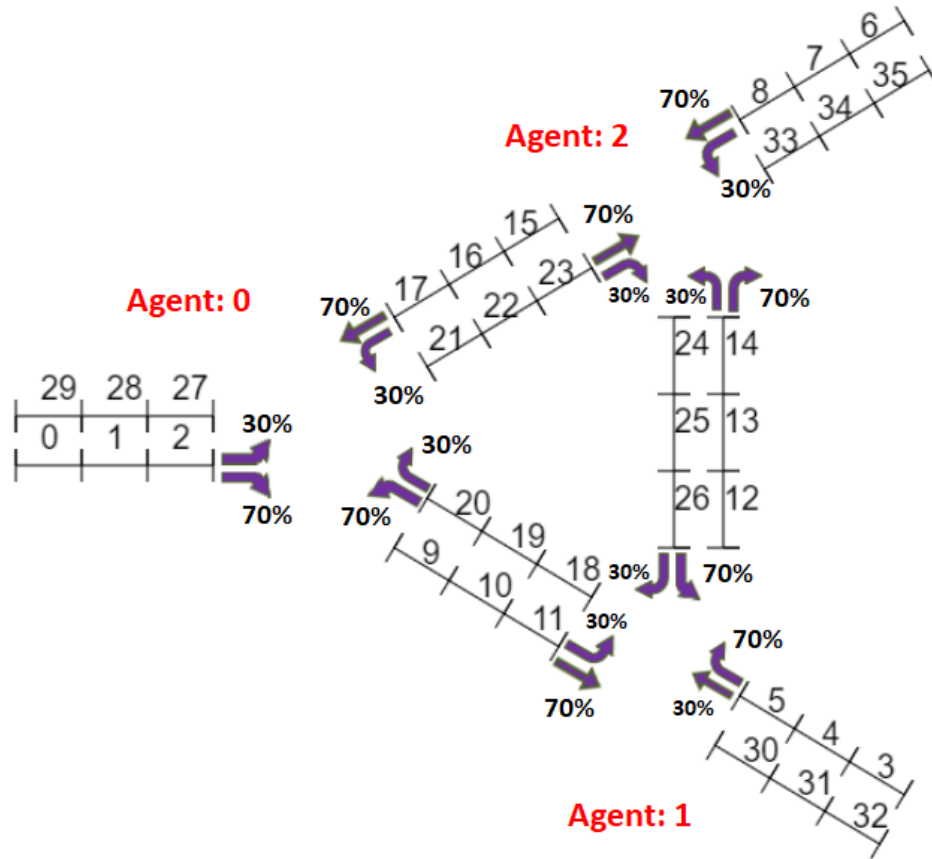
Rysunek 6.7: Środowisko 3 - fazy świateł

6.3.3 Model ruchu

Jako model ruchu zostanie wykorzystany przepływ CTM z następującymi parametrami:

- Ilość pojazdów na odcinku jest nieograniczona t.j. $N_j(t) = \infty$ dla dowolnego odcinka j i czasu t .
- Maksymalna ilość pojazdów, które mogą napłynąć do odcinka to $Q_j(t) = 10$ dla dowolnego j, t .
- Macierze sygnalizacji świetlnej \mathbf{S} są zgodne z fazami przedstawionymi w tabeli (6.3.2)

- Macierze prawdopodobieństwa przejazdów P są zgodne z przepływami przedstawionymi poniżej (6.3.3). Jeśli na skrzyżowaniu jest dostępny manewr wyjazdu z sieci dróg, to 70 procent pojazdów obiera go. Jeśli nie ma takiego manewru, to 70 procent pojazdów skręca w prawo. Jedyne manewr [15,8] jazdy prosto obiera 70 procent pojazdów. Reszta manewrów jest wybierana przez pozostałe 30 procent pojazdów.



Rysunek 6.8: Środowisko 3 - przepływy na skrzyżowaniach

6.3.4 Uczenie

Sygnal wejściowy

Sieć neuronowa przypisana do każdego z agentów przyjmuje jako sygnał wejściowy 6 elementową tablicę. Pierwsze 3 elementy to ilości pojazdów na odcinkach będących przed skrzyżowaniem przypisanym do agenta. Pozostałe 3 elementy mają wartości z zakresu $\{0,1\}$. Określają one aktualną fazę świetlną.

Przykład sygnału wejściowego

Sygnał wejściowy $x = [5, 17, 2, 0, 0, 1]$ przedstawia informację, iż przed skrzyżowaniem są kolejki odpowiednio 5, 17 i 2 pojazdów. Ostatnie 3 elementy oznaczają, że aktualna faza

światlna to 2.

Sygnal nagrody

Nagrodą dla każdego z agentów jest ilość pojazdów, które opuściły skrzyżowanie przyporządkowane do agenta.

Kooperacja agentów

W pracy [29] zostały przedstawione zasady systemu wieloagentowego. Pierwszą z nich jest realizacja założonego celu globalnego poprzez kooperacje agentów. Nagrody przyznawane agentom powinny służyć uzyskaniu celu globalnego. W tym przypadku jest to maksymalizacja ilości pojazdów opuszczających środowisko. Nagrody jednak nie muszą być modelowane bezpośrednio jako ilość pojazdów opuszczających układ. Natomiast powinny być one tak ukształtowane, aby osiągnąć cel globalny.

Ogólny zarys

Początkowo do każdej z dróg wjeżdża po jednym pojeździe. Na koniec każdej iteracji nauki (3.5) przeprowadzany jest epizod testowy. Agent ma za zadanie wybierać akcje, które pozwolą na opuszczenie sieci dróg przez co najmniej 93 procent pojazdów. Jeśli ten warunek zostanie spełniony następuje zwiększenie ilości wjeżdżających pojazdów o 20 procent dotychczasowej wartości. W przypadku gdy ten warunek nie zostanie spełniony w trakcie 15 kolejnych epizodów testowych następuje koniec nauki.

Cel nauki i spodziewany wynik

Bezpośrednim celem nauki jest uzyskanie jak największej ilości pojazdów wjeżdżających do sieci. Aby to się stało agenci muszą podejmować decyzje, które pozwalają na opuszczenie sieci jak największej ilości pojazdów. Moment stopu spodziewany jest na chwilę, gdy na każdą z dróg będzie wjeżdżało niecałe 10 pojazdów. Zgodnie z opisem modelu ruchu - maksymalnie 10 pojazdów może wjechać w jednej chwili na dany odcinek, zatem muszą się tworzyć zatory przy większej ilości pojazdów.

Sposób uczenia

Następujące stałe oraz ustalenia określają szczegóły uczenia:

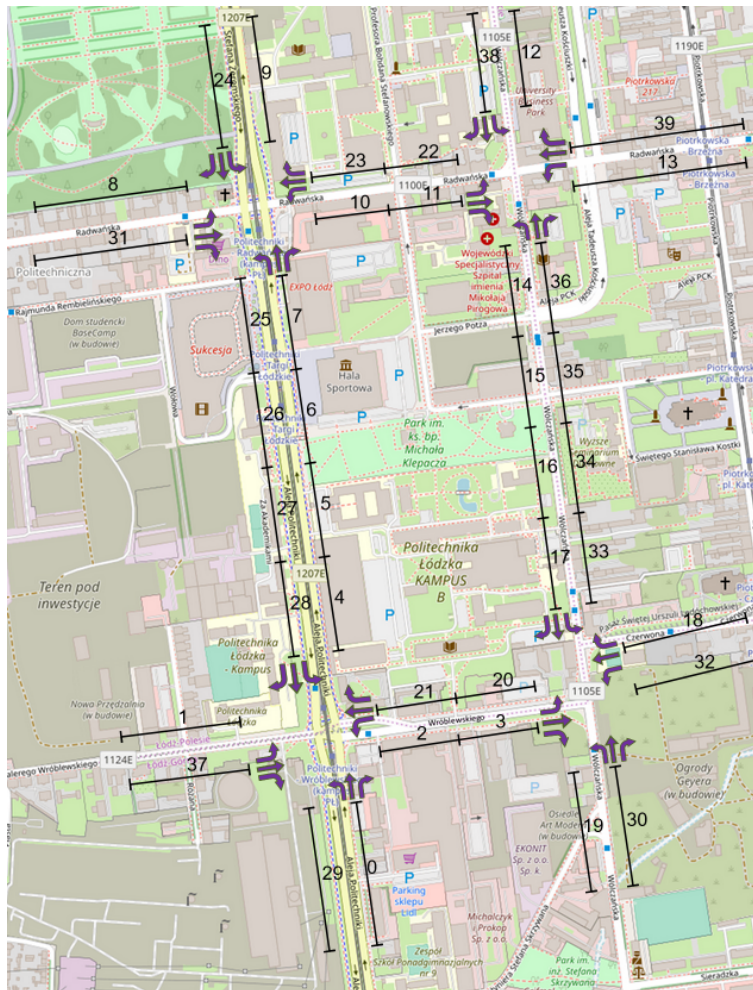
- $\gamma = 0.9$ - stopa dyskontowa
- Parametr zachłanności ϵ jest początkowo równy 1 (w pełni losowe epizody). Z każdą iteracją jest zmniejszany o 0,04 aż do wartości $\epsilon = 0,2$.
- Generowanych jest jednorazowo 20 epizodów treningowych. Każdy z nich trwa 300 interwałów czasowych.
- Wygenerowane zbiory danych są dzielone w proporcjach 80 i 20 procent dla odpowiednio treningu i walidacji.

- Za uczenie odpowiada sieć neuronowa biblioteki Keras. Posiada ona 3 ukryte warstwy (10,14,10 neuronów) z funkcjami aktywacji relu. Optymalizacja jest przeprowadzana przez metodę Adam z funkcją straty błędu średniokwadratowego, parametrem $learning_rate = 0,01$ oraz $bias_initializer = 15$ dla warstwy wyjściowej.
- Epizod testowy trwa 1000 interwałów czasowych.

Wyniki uczenia

Moment stopu jest osiągnięty przy 8.9 pojazdów napływających do sieci ruchu. Najwyższy osiągnięty wyniki 20 epizodów z takim napływem ruchu to 90 procent. Żadna z akcji nie jest zaniechana - agenci conajmniej kilkadziesiąt razy podejmują każdą z dostępnych akcji w zachłannym epizodzie.

6.4 Środowisko 4 - kampus B Politechniki Łódzkiej

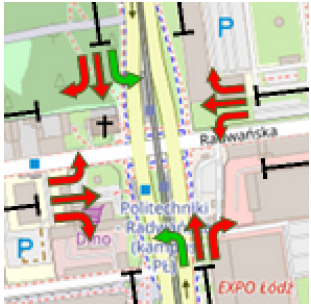
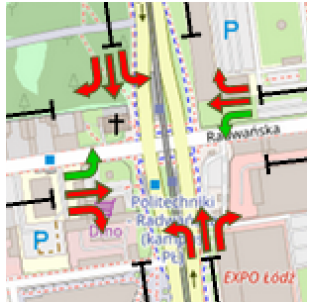
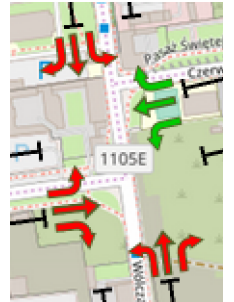
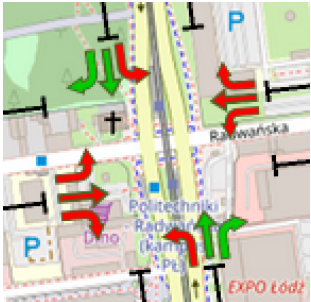
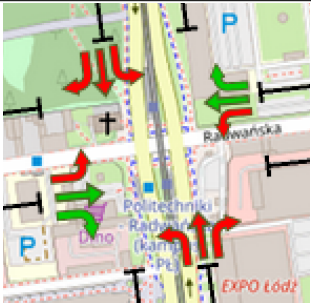




Rysunek 6.9: Środowisko 5 - Skrzyżowania wokół kampusu B Politechniki Łódzkiej. Mapy Open Street Map

Struktura środowiska bazuje na rzeczywistej sieci dróg znajdującej się w Łodzi. Sieć wyznaczają ulice Wróblewskiego, Wólczańska, Radwańska i aleja Politechniki Łódzkiej. Środowisko posiada 40 odcinków w tym 8 odcinków źródłowych (0,18,24,30,31,37,38,39) i 8 wylotowych (1,8,9,12,13,19,29,32). Drogi krzyżują się w 4 miejscach. Na dowolnym skrzyżowaniu kierowcy mogą wykonać manewry skrętu w lewo, w prawo oraz jazdy prosto.

6.4.1 Sygnalizacje świetlne

Każde skrzyżowanie posiada 5 faz świetlnych. Fazy 0,1,2 i 3 są fazami, które posiadają pewne zielone światła. Zmiana pomiędzy tymi fazami nie jest natychmiastowa i następuje dopiero po 2 interwałach fazy żółtych światel. Skrzyżowanie ulic Wróblewskiego, Wólczańskiej i Czerwonej posiada inne od pozostałych trzech skrzyżowań fazy świetlne. Wynika to z kolizyjności lewoskrętów z ulic Wróblewskiego i Czerwonej. Poniższa tabela przedstawia możliwe fazy świetlne.

Faza	Wólczańska/ Wróblewskiego/ Czerwona	Pozostałe 3 skrzyżowania
0		
1		
2		
3		
Żółte		

6.4.2 Przestrzeń decyzyjna

Analogicznie do poprzednich przestrzeni decyzyjnych z fazą żółtych świateł - przykładowo (6.3.2).

6.4.3 Model ruchu

Jako model ruchu zostanie wykorzystany przepływ CTM z następującymi parametrami:

- Ilość pojazdów na odcinku jest nieograniczona t.j. $N_j(t) = \infty$ dla dowolnego odcinka j i czasu t .
- Maksymalna ilość pojazdów, które mogą napłynąć do odcinka to $Q_j(t) = 10$ dla dowolnego j, t .
- Macierze sygnalizacji świetlnej \mathbf{S} są zgodne z fazami przedstawionymi w powyższej tabeli z sekcji (6.4.1)
- Macierze prawdopodobieństwa przejazdów \mathbf{P} uwzględniają następujące proporcje obierania kierunków. Jazdę prosto obiera połowa pojazdów. Manewr skrętu w prawo dokonuj 30 procent pojazdów, a w lewo ma zamiar skrócić pozostałe 20 procent.

6.4.4 Uczenie

Sygnal wejściowy

Sygnal wejściowy jest ustalony na podstawie ilości pojazdów na odcinkach przed skrzyżowaniem oraz aktualnej fazy świetlnej.

Sygnal nagrody

Nagrodą dla każdego z agentów jest ilość pojazdów, które opuściły skrzyżowanie przyporządkowane do agenta.

Ilość wjeżdżających pojazdów

Poprzednie środowiska zakładały, że w każdym interwale napływa stała ilość pojazdów do dróg. Jest to mało realne założenie, gdyż zawsze są pewne wahania co do ruchu pojazdów. Do każdej z dróg źródłowych wjeżdża zatem pewna ilość pojazdów zadana przez zmienną losową X o rozkładzie log-normalnym. Jest to rozkład o podobnych właściwościach i kształcie wykresu gęstości do rozkładu normalnego. Wykorzystywany jest on zamiast rozkładu normalnego przeważnie gdy realizacje zmiennej losowej nie mogą mieć wartości ujemnych. Przykładem mogą być chociażby wyniki zawodów skoku w dal albo zarobki pracowników pewnej firmy. Podobnie ilość pojazdów w model makroskopowym powinna być nieujemna. Funkcja gęstości rozkładu log-normalnego jest zadana wzorem:

$$f(x) = \frac{1}{x\delta\sqrt{2\pi}} e^{-\frac{\ln(x-\mu)^2}{2\delta^2}} \quad (6.1)$$

Wartość oczekiwana tej zmiennej losowej to

$$EX = e^{\mu + \frac{\delta^2}{2}} \quad (6.2)$$

Parametr δ w obliczeniach zostaje ustalony jako wartość $\delta = 0,4$. Początkowo do układu ma napływać średnio po 1 pojeździe w trakcie interwału czasowego, czyli należy dobrać taki parametr μ aby $EX = 1$. Parametr¹ μ rozkładu będzie, więc ustalany na podstawie wartości oczekiwanej. Przekształcając (6.2)

$$\mu = \ln(EX) - \frac{\delta^2}{2}. \quad (6.3)$$

Początkowo zatem

$$\mu = \ln(1) - \frac{0,4^2}{2} = 0,08. \quad (6.4)$$

Sposób uczenia

Na koniec każdej iteracji nauki (3.5) z uwzględnieniem modyfikacji wzoru (3.5.3) przeprowadzane są epizody testowe. Agent ma za zadanie wykonywać akcje, które w trakcie symulacji pozwolą na opuszczenie skrzyżowania przez co najmniej 93 procent pojazdów. W przypadku spełnienia tego warunku następuje zwiększenie średniej ilości napływających pojazdów EX o 20 procent dotychczasowej wartości. Parametr μ jest zwiększany wedle przedstawionego wyżej wzoru (6.3). Ten proces jest powtarzany, aż do momentu gdy przez 15 kolejnych iteracji uczenia nie uzyskany zostanie warunek 95 procent pojazdów opuszczających skrzyżowanie.

Cel nauki

Bezpośrednim celem nauki jest uzyskanie jak największej wartości ilości napływających pojazdów wedle powyższego schematu. Aby to się stało agenci muszą podejmować decyzje, które pozwalają na opuszczenie sieci możliwie jak największej ilości pojazdów.

Wyniki uczenia

Koniec uczenia przypadł na moment z wartością $EX = 7,43$. Okazuje się, że proces uczenia doprowadził do zaniechania przez agentów manewrów lewoskrętów (akcje 0 i 1). Jest to z pewnością niepożądane zachowanie, jednak można było się go spodziewać, gdyż nagrody zostały wymodelowane tak, by gratyfikować jedynie przepływ - nieistotne z jakiego manewru.

6.4.5 Modyfikacja środowiska

Zostaje nałożona dodatkowa restrykcja związana z przestrzenią akcji. W przypadku nie pojawienia się pewnej fazy przez 100 interwałów czasowych zostaje podjęta akcja mająca na celu włączenie tej fazy.

¹Dla rozkładu lognormalnego powszechnie przyjmuje się oznaczenie μ dla tego parametru. Może to być mylące, gdyż często μ jest utożsamiane z wartością oczekiwaną zmiennej losowej, a tak nie jest w tym przypadku.

Sposób nauki

Oprócz powyżej przedstawionej metody uczenia zostanie przeprowadzony drugi wariant nauki. Poza elementami sygnału poprzednio opisanego zostaną wprowadzone 4 wartości określające jak długo nie została wybrana dana akcja. Wartości te zostaną znormalizowane dzieląc rzeczywistą ilość interwałów bez danej akcji przez 100. Jest to powszechny zabieg, gdy dane mogą przyjmować bardzo rozległe wartości, co może zaburzyć trening sieci neuronowej. Dodatkowo z powodu wyższego wymiaru sygnału ustalone są większe ilości neuronów (14,18,10) na kolejnych warstwach sieci neuronowej.

Wyniki uczenia zmodyfikowanego środowiska

Wyniki uczenia obydwu sposobów są identyczne do wariantu bez restrykcji wystąpienia wszystkich faz. Jedynie wariant nauki z sygnałem zawierającym informację odnośnie ilości interwałów bez akcji potrzebuje więcej czasu do nauki. Warunek stopu jednak w obydwu przypadkach jest osiągany przy tej samej wartości $EX = 7,43$. Wyeliminowany został zatem niepożądany efekt zaniechania lewoskrętów.

Rozdział 7

Dokumentacja programu

7.1 Wstęp

Utworzony program jest podzielony na dwie warstwy - prezentacyjną oraz logiki. Jest to podejście powszechnie spotykane przy tworzeniu aplikacji internetowych [18]. Warstwy te są zwyczajowo określane jako front-end i back-end. Warstwa prezentacyjna programu ma za zadanie przedstawić dane wygenerowane w trakcie symulacji. Innymi słowy cała logika symulacji jest przeprowadzana w warstwie logiki, ale obraz przedstawiający te symulacje jest widoczny w warstwie prezentacyjnej.

7.2 Warstwa logiki

7.2.1 Wymagania

Warstwa logiki została napisana w języku Python (wersja 3.6.7). Aby w pełni korzystać z programu należy mieć zainstalowany interpreter pythona dla tej wersji. W programie zostały wykorzystane następujące biblioteki, które także są niezbędne do uruchomienia programu:

Biblioteka	Wersja	Opis
Mypy	0.701	Umożliwia statyczne typowanie w pythonie
Numpy	1.16.4	Struktury obliczeniowe
Tensorflow	2.0.0b1	Biblioteka wykorzystywana w uczeniu maszynowym odpowiedzialna za działanie sieci neuronowych
Attrs	19.1.0	Umożliwia zwięzłe pisanie kodu bez tzw. boilerplate'u
Matplotlib	3.0.3	Tworzenie wykresów

Interpreter z powyższymi bibliotekami jest załączony w dodatku do pracy w folderze py-

thon_interpreter.

7.2.2 Instrukcje uruchomienia

W folderze warstwa_logiki znajduje się 5 folderów odpowiadających przedstawionym środowiskom i sposobom ich uczenia z rozdziału 6. Każde ze środowisk można uruchomić za pomocą skryptu run.py. Następuje wtedy proces nauki w trakcie którego są wyświetlane komunikaty odnośnie zachłannych epizodów. Są one także zapisywane w formacie JSON w katalogu src/assets warstwy prezentacyjnej. Nazwa zapisanych epizodów jest istotna w kontekście warstwy prezentacyjnej. Z tego powodu nazwa zapisanego pliku jest logowana w konsoli. Schemat nazewnictwa to 'env[x]_learnt_[y]', gdzie [x] to numer środowiska oraz [y] to numer epizodu testowego.

7.3 Warstwa prezentacyjna

7.3.1 Wymagania

Warstwa wizualizacyjna została napisana w typescriptowym frameworku Angular w wersji 6. Grafiki przedstawiające środowiska drogowe są narysowane przy pomocy tagów canvas, które są dostępne w większości przeglądarek internetowych [10]. Do uruchomienia warstwy prezentacyjnej niezbędny jest jednak zainstalowany framework Angular. W tym celu należy pierw zainstalować środowisko node.js. Wraz z nim dostępny jest domyślny system zarządzania pakietami - node package manager skrótowo nazywany jako npm. Warto podczas instalacji zaznaczyć opcję dodania npm do zmiennych środowiskowych. Po zainstalowaniu npm należy wykonać komendę: 'npm install -g @angular/cli@6.1.10', aby zainstalować wiersz polecenia angulara (zbiór angularowych komend dostępnych pod nazwą ng). Operacja ta wymaga dostępu do internetu. Końcowym etapem jest instalacja pakietów wykorzystanych w samej aplikacji. Należy z poziomu folderu warstwa_prezentacyjna wpisać komendę 'npm install'. Wywołanie tego polecenia może zająć do kilku minut w zależności od szybkości łącza internetowego.

7.3.2 Instrukcje uruchomienia

Należy z poziomu folderu warstwa_prezentacyjna wpisać komendę 'ng serve'. Uruchamia ona aplikację Angulara na domyślnym hoście. Jeśli nie było żadnych komunikatów o błędach w konsoli strona powinna wyświetlić się po wejściu na adres <http://localhost:4200>. Aby wyświetlić symulację należy wpisać w lewym górnym rogu jej nazwę i kliknąć przycisk 'Wczytaj sieć'. Jeśli wczytywanie się powiodło zostanie wyświetlony zielony komunikat potwierdzający powodzenie operacji. Czas dla którego ma zostać wyświetlony stan sieci dróg można zmieniać na dwa sposoby - suwakiem lub przyciskami '+/-'.

Bibliografia

- [1] Plan działania na rzecz wdrażania inteligentnych systemów transportowych w europie. *Komisja Wspólnot Europejskich, KOM*, 886, 2008.
- [2] Study on urban mobility – assessing and improving the accessibility of urban areas. *Komisja Wspólnot Europejskich, KOM*, 886, 2017.
- [3] Moshe Ben-Akiva, Michel Bierlaire, Didier Burton, Haris N Koutsopoulos, and Rabi Mishalani. Network state estimation and prediction for real-time traffic management. *Networks and spatial economics*, 1(3-4):293–318, 2001.
- [4] Kent C Berridge. Reward learning: Reinforcement, incentives, and expectations. In *Psychology of learning and motivation*, volume 40, pages 223–278. Elsevier, 2000.
- [5] Samia Boubaker, Ferid Rehim, and Adel Kalboussi. Comparative analysis of microscopic models of road traffic data. In *Logistics (LOGISTIQUA), 2011 4th International Conference on*, pages 474–478. IEEE, 2011.
- [6] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.
- [7] Wilco Burghout, Haris N Koutsopoulos, and Ingmar Andreasson. A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation. In *Intelligent Transportation Systems Conference, 2006. ITSC’06. IEEE*, pages 1102–1107. IEEE, 2006.
- [8] Carlos F Daganzo. The cell transmission model, part ii: network traffic. *Transportation Research Part B: Methodological*, 29(2):79–93, 1995.
- [9] Tim Lomax David Schrank, Bill Eisele. 2019 urban mobility report. 2019.
- [10] Alexis Deveria. Can i use... support tables for html5, css3, etc. <https://caniuse.com/#feat=canvas>, 2019 (accessed April 4, 2019).
- [11] Gabriel Dulac-Arnold, Richard Evans, Hado van Hasselt, Peter Sunehag, Timothy Lillicrap, Jonathan Hunt, Timothy Mann, Theophane Weber, Thomas Degris, and Ben Coppin. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.
- [12] Simone Göttlich, Michael Herty, and Ute Ziegler. Modeling and optimizing traffic light settings in road networks. *Computers & operations research*, 55:36–51, 2015.

- [13] Zhenyu Guan, Liangxu Bian, Tao Shang, and Jianwei Liu. When machine learning meets security issues: A survey. In *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, pages 158–165. IEEE, 2018.
- [14] Dirk Helbing, Ansgar Hennecke, Vladimir Shvetsov, and Martin Treiber. Master: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model. *Transportation Research Part B: Methodological*, 35(2):183–211, 2001.
- [15] Pushpendra Kumar, Rochdi Merzouki, Blaise Conrard, Vincent Coelen, and Belkacem Ould Bouamama. Multilevel modeling of the traffic dynamic. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):1066–1082, 2014.
- [16] Ted G Lewis and Peter J Denning. Learning machine learning. *Communications of the ACM*, 61(12):24–27, 2018.
- [17] Michael James Lighthill and Gerald Beresford Whitham. On kinematic waves ii. a theory of traffic flow on long crowded roads. *Proc. R. Soc. Lond. A*, 229(1178):317–345, 1955.
- [18] Kun Liu, Jinmin Jiang, Xiaohan Ding, and Hui Sun. Design and development of management information system for research project process based on front-end and back-end separation. In *2017 International Conference on Computing Intelligence and Information System (CIIS)*, pages 338–342. IEEE, 2017.
- [19] Tambet Matiisen. <https://neuro.cs.ut.ee/demystifying-deep-reinforcement-learning/>. <https://neuro.cs.ut.ee/demystifying-deep-reinforcement-learning/>, 2019 (accessed September 1, 2019).
- [20] Francisco S Melo. Convergence of q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.
- [21] Thijs Nieuwdorp. Dare to discover: The effect of the exploration strategy on an agent’s performance. 2017.
- [22] polskawliczbach.pl. Samochody osobowe w polsce w latach 2003-2016 (źródło: Gus). <http://www.polskawliczbach.pl/#transport-i-komunikacja>, 2019 (accessed March 3, 2019).
- [23] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv pre-print arXiv:1609.04747*, 2016.
- [24] Isaac J Sledge and José C Príncipe. Balancing exploration and exploitation in reinforcement learning using a value of information criterion. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2816–2820. IEEE, 2017.
- [25] Statista. Number of passenger cars and commercial vehicles in use worldwide from 2006 to 2015 in (1,000 units). <https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide/>, 2019 (accessed March 3, 2019).

- [26] Richard S Sutton, Andrew G Barto, and Ronald J Williams. Reinforcement learning is direct adaptive optimal control. *IEEE Control Systems Magazine*, 12(2):19–22, 1992.
- [27] TomTom. Tomtom traffic index measuring congestion worldwide). https://www.tomtom.com/en_gb/trafficindex/list?citySize=ALL&continent=ALL&country=ALL, 2019 (accessed March 3, 2019).
- [28] Vinh An Vu and Gary Tan. High-performance mesoscopic traffic simulation with gpu for large scale networks. In *Proceedings of the 21st International Symposium on Distributed Simulation and Real Time Applications*, pages 127–135. IEEE Press, 2017.
- [29] Bohui Wang, Xinpeng Fang, and Bin Zhang. Cooperative control for industrial multi-agent systems: Framework and problems. In *2018 37th Chinese Control Conference (CCC)*, pages 6971–6976. IEEE, 2018.
- [30] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.