

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

Michał Lis

9 sierpnia 2019

Spis treści

1	Wprowadzenie	5
2	Cel i zakres pracy	7
3	Siatka czasowa i przestrzenna	9
4	Makroskopowy model ruchu	11
4.1	Klasyfikacja modeli ruchu drogowego	11
4.2	Wstęp	11
4.3	Rozwój gęstości ruchu na drodze	12
4.4	Dyskretyzacja makroskopowego modelu ruchu	12
5	Model sieci dróg	15
5.1	Wstęp	15
5.2	Wektor stanu drogi	15
5.2.1	Przykład	15
5.3	Rozwój wektora stanu jednej drogi	16
5.3.1	Przykład	16
5.4	Wektor stanu sieci dróg	17
5.4.1	Przykład	17
5.5	Rozwój wektora stanu sieci dróg	18
5.5.1	Przykład	18
5.6	Wprowadzenie sygnalizacji świetlnej	21
5.6.1	Przykład macierzy sygnalizacji świetlnej	21
5.6.2	Macierz systemu uwzględniająca sygnalizację świetlną	22
5.6.3	Przykład rozwoju wektora stanowego	22
5.7	Wprowadzenie źródeł ruchu	23
5.7.1	Przykład	24
5.7.2	Zatory drogowe	25
5.7.3	Przykład zatoru na pojedynczej drodze	26
5.7.4	Przykład zatoru na skrzyżowaniu	27
6	środowiska symulacyjne	31
6.1	Środowisko 1(11 na froncie)	31
6.1.1	Sygnalizacje świetlne	32
6.1.2	Przestrzeń decyzyjna	32

6.1.3	Zatory drogowe	32
6.1.4	Uczenie agenta	32
6.1.5	Pożądane zachowanie	32
6.1.6	Sposób uczenia	33
6.1.7	Wyniki uczenia	33
6.2	Srodowisko 2(14 na froncie)	33
6.2.1	Sygnalizacje świetlne	34
6.2.2	Przestrzeń decyzyjna	34
6.2.3	Zatory drogowe	35
6.2.4	Cel nauki i pożądane zachowanie agenta	35
6.2.5	Sposób uczenia	36
6.2.6	Wyniki uczenia	36
6.3	Srodowisko 4	37
7	Przegląd metod optymalizacyjnych	39
7.1	Kategorie uczenia maszynowego	39
7.2	Uczenie ze wzmocnieniem	39
7.3	Programowanie dynamiczne	42
7.4	Metoda Monte Carlo On-Policy	43
7.5	Mój algorytm który potrzebuje nazwy ale proponuję: Monte Carlo DQN full exploration	44
7.6	Mój algorytm który potrzebuje nazwy ale proponuję: Monte Carlo DQN	45

Rozdział 1

Wprowadzenie

Problem zatłoczonych ulic staje się coraz bardziej powszechny na całym świecie. W ogromnym tempie wzrasta ilość pojazdów na drogach. Według danych firmy gromadzącej dane statystyczne *Statista* liczba zarejestrowanych pojazdów na świecie w roku 2006 wynosiła 947 tysięcy [?]. W 2015 roku na świecie jeździło już 1282 tysiące pojazdów. Wzrost przez te 9 lat był niemalże liniowy. Co roku rejestrowano około 39,4 tysiące nowych samochodów rocznie, co wyznacza stopę wzrostu liczby pojazdów na poziomie 3,7%.



Rysunek 1.1: Liczba pojazdów na świecie

W Polsce wzrost ilości pojazdów w latach 2006 - 2015 był jeszcze większy [?]. 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 oznacza 5 procentowy roczny wzrost. 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 [?]. 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). 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 [?]. Następny raport z 2017 roku może napawać optymizmem, gdyż przedstawione w nim wyliczenia określiły jedynie 0,77% straty całkowitego PKB wspólnoty [?]. 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 brakuje często miejsca na wybudowanie dróg o większej przepustowości. Rozwiązaniem może być wprowadzenie większej ilości sygnalizacji świetlnej. Istotną kwestią jest optymalizacja ustawień sygnalizacji świetlnej. Praca moja jest poświęcona temu problemowi.

Rozdział 2

Cel i zakres pracy

Celem pracy jest stworzenie programu, który zoptymalizuje fazy sygnalizacji świetlnej, co przyczyni się do zwiększenia przepustowości sieci dróg.

Jako środowisko zostanie stworzony symulator ruchu drogowego. Symulacje ruchu będą w pełni zgodne z makroskopowym modelem ruchu. Sam makroskopowy model ruchu zostanie przedstawiony w rozdziale X. Jest to model ciągły. Pożądanym jest dyskretny model ruchu drogowego ze względu na łatwość implementacji komputerowej. Zostanie zatem przedstawiona w sekcji X.Y dyskretyzacja makroskopowego modelu ruchu. W rozdziale X zostanie zdefiniowany model sieci dróg. Początkowy model zaplanowano jako podstawowy z pominięciem większości aspektów. W każdej kolejnej sekcji model będzie stopniowo rozwijany. Sieci dróg zdefiniowane według końcowego modelu będą środowiskiem treningowym dla algorytmów uczenia maszynowego. Rozdział X opisuje uczenie ze wzmocnieniem - algorytm treningowy procesu optymalizacji sygnalizacji świetlnej. Rozdział X przedstawia cztery modele sieci dróg, dla których został stworzony program symulacyjny. Rozdział X opisuje optymalizację sygnalizacji świetlnej dla wspomnianych sieci dróg.

Rozdział 3

Siatka czasowa i przestrzenna

Dyskretny charakter modelu przedstawianego w pracy obliguje do określenia siatki czasowej i przestrzennej. Dla par czasu i miejsc należących do tych dwóch siatek będą określane zmienne stanu.

Siatka czasowa jest zdefiniowana jako skończony ciąg liczb naturalnych:

$$(0, 1, \dots, K). \quad (3.1)$$

Niech będzie ustalona droga e , która jest odcinkiem $[0, L_e]$. Droga zostaje podzielona na $L + 1$ odcinków o równej długości $\Delta x = \frac{L_e}{L+1}$. **Siatka przestrzenna** drogi to ciąg odcinków:

$$(b_l)_{l=0}^L = [l\Delta x, (l+1)\Delta x]$$



Rysunek 3.1: Siatka przestrzenna

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 [?] - makroskopowy, mezoskopowy oraz mikroskopowy. Czasem [?] 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 [?]. Przykładowo pojazdy są zgrupowane w grupę podróżującą z pewnego punktu startowego do celu. Inne modele [?] 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 [?]. 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 [?] przedstawili pomysł przyrównania ruchu ulicznego na zatłoczonych drogach do przepływu wody w rzekach. Z tego powodu nie rozróżniamy w nim indywidualnie pojazdów, ani też nawet grupowo. Rozważamy natomiast gęstość ruchu w danym punkcie na drodze i czasie - czyli ilość pojazdów na danym odcinku drogi. Sposób w jaki poruszają się pojazdy jest wyliczany jedynie na podstawie gęstości ruchu. Jest to najmniej kosztowny obliczeniowo model. Właśnie w modelu makroskopowym zostało stworzone środowisko symulacyjne. Szczegóły modelu są przedstawione w następnym podrozdziale.

4.2 Wstęp

Istotą makroskopowego modelu ruchu jest pojęcie gęstości ruchu. Jest to zmienna stanowa określona dla każdego punktu drogi w czasie. Formalnie gęstość można rozumieć jako czynnik definiujący dynamikę ruchu. Im większa gęstość tym mniejsza prędkość ruchu. W niektórych

artykułach gęstość ruchu [?] jest przedstawiona jako iloraz ilości pojazdów znajdujących się na pewnym odcinku i długości tego odcinka drogi. Nie są to jednak czysto matematyczne formalne definicje. W makroskopowym modelu nie rozróżniamy pojedynczych pojazdów, ani nawet grup, więc taka definicja gęstości ruchu może być odebrana jako nieściśła z ideą modelu.

4.3 Rozwój gęstości ruchu na drodze

Makroskopowy model ruchu jest oparty o równanie różniczkowe (4.2) wraz z warunkiem początkowym (4.1). Model makroskopowy traktuje ruch uliczny na drogach podobnie do przepływu wody w rzece[ref]. Gęstość ruchu można utożsamiać z polem powierzchni przekroju poprzecznego rzeki, co dla ustalonej szerokości rzeki - upraszcza się do wysokości wody w rzece. Istotną uwagą w tym miejscu jest zaznaczenie, iż rzeka zazwyczaj posiada pewien spadek, który zapewnia ruch cieczy ze źródła do ujścia. Ruch makroskopowy zdefiniowany przez równanie (4.2) z kolei odnosi się do rzeki która jest na całym swoim odcinku pozioma. W takim przypadku de facto nie ma zdefiniowanego zwrotu ruchu.

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

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

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 [?] rozwój gęstości ruchu na drodze. Funkcja płynności ruchu f powinna być wklęsła [ref]. W przedstawionym w tej pracy modelu funkcja ma następującą definicję:

$$f(p) = \begin{cases} \lambda p & \text{dla } p \in [0, p^*] \\ \lambda \cdot (2p^* - p) & \text{dla } p \in (p^*, p^{max}] \end{cases} \quad (4.3)$$

$$(4.4)$$

Gdzie λ jest stałym parametrem funkcji trójkątnej oraz $p^* = \frac{1}{2}p^{max}$.

4.4 Dyskretyzacja makroskopowego modelu ruchu

Niech będzie ustalona droga e oraz jej siatka przestrzenna b_l . Celem jest przedstawienie wartości gęstości dla odcinków siatki przestrzennej w chwilach $k = 0, 1, \dots, K$. Gęstość w odcinku b_l i czasie k jest zdefiniowana jako:

$$p_l^k = \int_{b_l} \frac{p(x, k)}{\Delta x} dx. \quad (4.5)$$

Na podstawie (4.2) można wywnioskować, że:

$$\int_{b_l} p(x, k+1) - p(x, k) dx + \int_k^{k+1} f(b_{l+1}, k) - f(b_l, k) dk = 0 \quad (4.6)$$

Upraszczając otrzymujemy:

$$\Delta x(p_l^{k+1} - p_l^k) + \int_k^{k+1} (f(b_{l+1}, k) - f(b_l, k)) dk = 0 = 0 \quad (4.7)$$

Wartości gęstości zmieniają się w tylko w chwilach k . Wtedy wartości $f(b_{l+1}, k)$ i $f(b_l, k)$ są stałe na całym przedziale całkowania $[k, k+1)$. Otrzymujemy równanie:

$$\Delta x(p_l^{k+1} - p_l^k) + (f(b_{l+1}, k) - f(b_l, k)) = 0 \quad (4.8)$$

Rezultatem jest końcowy rekurencyjny wzór na gęstość ruchu:

$$p_l^{k+1} = p_l^k - \frac{1}{\Delta x} (f(b_{l+1}, k) - f(b_l, k)) \quad (4.9)$$

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 stanowych. Początkowo zmienna stanowa jest identyfikowana jako ilość pojazdów na danym odcinku drogi.

5.2.1 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 jednej drogi

Początkowy model przepływu pojazdów zakłada, iż wszystkie pojazdy w chwili $t+1$ są o jeden odcinek dalej w swojej podróży niż w momencie t . Założone jest, iż żadne nowe pojazdy nie pojawiają się w sieci dróg. Ujście pojazdów znajduje się na końcu ostatniego odcinka. Wszystkie pojazdy będące w chwili t na ostatnim odcinku w chwili $t+1$ opuszczają układ. 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. Definiuje ona sposób przepływu pojazdów. \mathbf{A} jest rzadką, kwadratową macierzą o wartościach równych 1 jedynie bezpośrednio 1 wiersz pod główną przekątną macierzy. Takie wartości gwarantują przepływ pojazdów o jeden odcinek w jednym interwale czasowym.

5.3.1 Przykład

Dla przykładu przedstawionego w (5.2.1) zostanie przedstawiony rozwój wektora stanu. Niech zatem

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

Macierzą systemu jest:

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

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

$$\mathbf{x}(1) = \mathbf{A}\mathbf{x}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 4 \\ 3 \end{bmatrix} \quad (5.5)$$

$$\mathbf{x}(2) = \mathbf{A}\mathbf{x}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 4 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 4 \end{bmatrix} \quad (5.6)$$

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

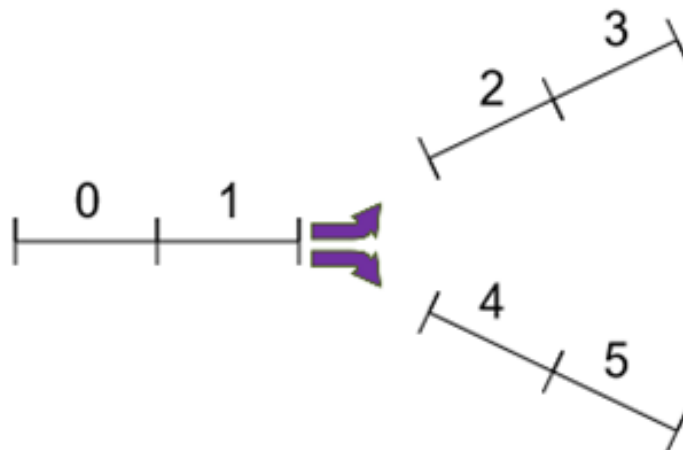
$$\mathbf{x}(4) = \mathbf{A}\mathbf{x}(3) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (5.8)$$

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. Sposób przedstawienia wartości stanów jednak jest bardzo podobny. Każda z dróg e_1, \dots, e_n ma k wydzielonych odcinków oznaczanych jako b_1, \dots, b_{nk} . Dla każdego z odcinków definiowana jest wartość stanowa.

5.4.1 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.

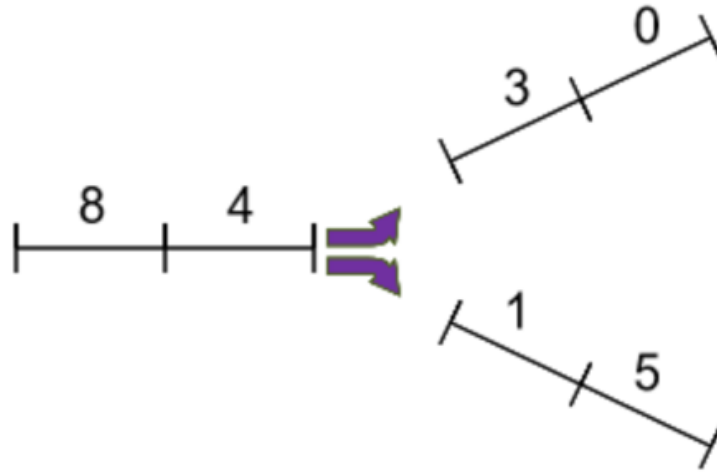


Rysunek 5.2: Numeracja odcinków środowiska

Niech przykładowym wektorem stanu będzie:

$$\mathbf{x}(t) = \begin{bmatrix} 8 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix} \quad (5.9)$$

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 $\mathbf{x}(t)$



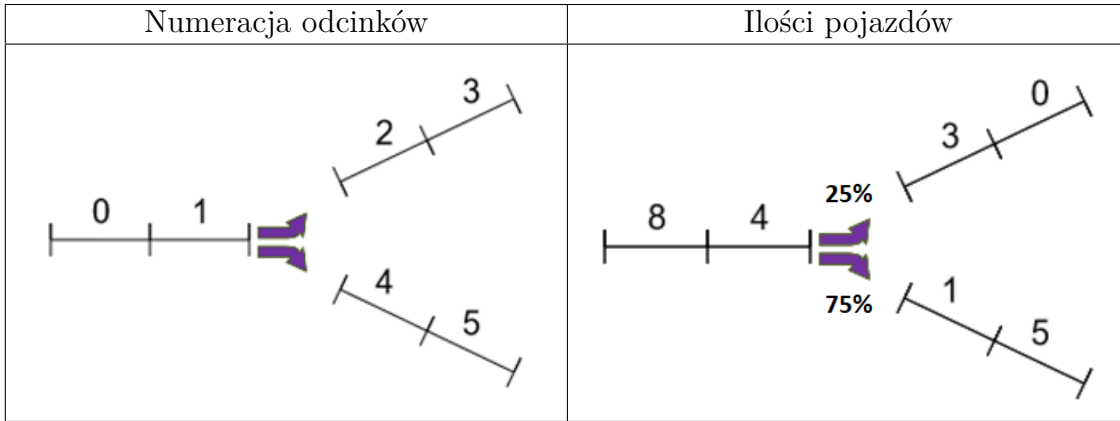
Rysunek 5.3: Sieć dróg z ilościami pojazdów na poszczególnych odcinkach

5.5 Rozwój wektora stanu sieci dróg

Przepływ pojazdów niezmiennie jest oparty o założenie, iż w trakcie trwania jednego interwału czasowego pojazdy pokonują 1 odcinek drogi. Nie pojawiają się nowe pojazdy w trakcie trwania symulacji, a ujścia ruchu znajdują się na końcu odcinków 3 i 5. Macierz systemu \mathbf{A} powinna uwzględnić przepływy pojazdów na skrzyżowaniach. Do tej pory wszystkie pojazdy poruszały się do przodu. W przypadku sieci dróg trzeba uwzględnić przypadek skrzyżowania, gdzie pojazdy mogą obrać różne kierunki ruchu. Zdefiniowana zostanie zatem macierz \mathbf{P} prawdopodobieństwa manewrów. Jest to niezmienna w czasie rzadka macierz. Wartości macierzy \mathbf{P} w kolumnie i oraz wierszu j to prawdopodobieństwo przejazdu pojazdu będącego na odcinku i do odcinka j .

5.5.1 Przykład

Rozważone zostanie środowisko (5.4.1) z dodatkowym założeniem, że 75 procent pojazdów ma zamiar skręcić w prawo. Pozostałe 25 procent wybiera skręt w lewo. Środowisko przedstawiają poniższe obrazki.



Taki sposób przepływu definiuje rzadką macierz prawdopodobieństwa \mathbf{P} o wymiarach 6 na 6.

- $\mathbf{P}[1, 2] = 0.25$ wyznacza manewr skrętu w lewo
- $\mathbf{P}[1, 4] = 0.75$ wyznacza manewr skrętu w prawo
- $\mathbf{P}[0, 1] = 1$ $\mathbf{P}[2, 3] = 1$ $\mathbf{P}[4, 5] = 1$ co wynika z przepływu pojazdów pomiędzy sekcjami na jednej drodze
- Pozostałe wartości macierzy \mathbf{P} to zera.

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

W sytuacji gdy nie ma uwzględnionej sygnalizacji świetlnej oraz pojęcia zatoru macierz systemu jest macierzą prawdopodobieństwa, czyli $A = P$.

Wartości wektora stanu zmieniają się zgodnie równaniem stanu $\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1)$. Poniżej zapisane są obliczenia.

t	Równanie stanu	Podgląd środowiska
0	$\mathbf{x}(0) = \begin{bmatrix} 8 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix}$	
1	$\mathbf{x}(1) = \mathbf{Ax}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 8 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 1 \\ 3 \\ 3 \\ 1 \end{bmatrix}$	
2	$\mathbf{x}(2) = \mathbf{Ax}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 8 \\ 1 \\ 3 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 6 \\ 3 \end{bmatrix}$	
3	$\mathbf{x}(3) = \mathbf{Ax}(2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 2 \\ 1 \\ 6 \\ 3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 0 \\ 2 \end{bmatrix}$	

5.6 Wprowadzenie sygnalizacji świetlnej

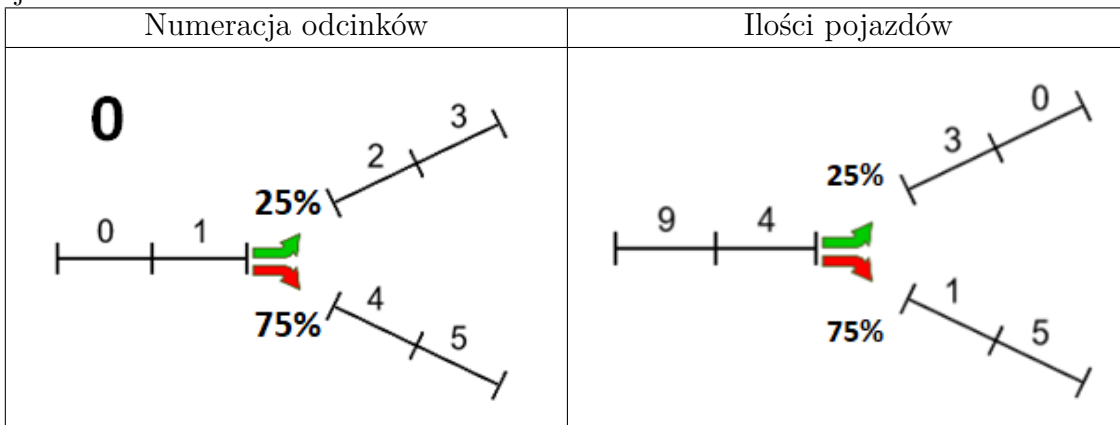
Kolejnym etapem rozwoju modelu jest wprowadzenie sygnalizacji świetlnej. Warto zauważyć, że do tej pory rozważane układy były pozbawione jakiegokolwiek sterowania, czego bezpośrednim skutkiem była niezmiennosc macierzy \mathbf{A} w czasie. Niech wektor $[i, j]$ będzie manewrem polegającym na bezpośrednim przejeździe z odcinka i na odcinek j . Zostanie teraz zdefiniowana macierz sygnalizacji świetlnej \mathbf{S} . Określa ona wykonalność dowolnego manewru.

$$S_{ij} = \begin{cases} 1 & \text{dla manewru zezwolonego przez zielone światło} \\ 1 & \text{dla manewru nie wymagającego sygnalizacji świetlnej} \\ 0 & \text{dla manewru wstrzymanego przez czerwone światło} \\ 0 & \text{dla niemożliwego manewru} \end{cases} \quad \begin{matrix} (5.10) \\ (5.11) \\ (5.12) \\ (5.13) \end{matrix}$$

Sygnalizacja świetlna zawsze posiada pewną fazę, która określa dozwolone (5.10) oraz niedozwolone (5.12) manewry. Fazy świetlne będą oczywiście zmieniane w trakcie symulacji zatem macierz \mathbf{S} też jest zmienna w czasie. Wartości macierzy \mathbf{S} obejmujące przypadki 5.13 oraz 5.11 są z kolei niezmiennie dla ustalonego środowiska.

5.6.1 Przykład macierzy sygnalizacji świetlnej

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



Poszczególne manewry są następujące:

- (5.10) - Manewrem zezwolonym przez zielone światło dla fazy 0 jest $[1, 2]$.
- (5.11) - Prawidłowym manewrem bez sygnalizacji świetlnej są manewry $[0, 1]$, $[2, 3]$, $[4, 5]$.
- (5.12) - Dla fazy świetlnej 0 manewrem zatrzymanym przez czerwone światło jest $[1, 4]$.
- (5.13) - Pozostałe manewry są niemożliwe. Przykładem niech będzie $[0, 2]$, gdyż nie ma możliwości bezpośredniego przejazdu z odcinka 0 do 2. Teoretycznie możliwy mógłby się wydawać manewr $[1, 1]$ jednak jest on uznany za niemożliwy.

Macierz sygnalizacji świetlnej dla tego przykładu jest następująca:

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

5.6.2 Macierz systemu uwzględniająca sygnalizację świetlną

Mając zdefiniowaną zarówno macierz prawdopodobieństwa przejazdów \mathbf{P} jak i macierz sygnalizacji świetlnej \mathbf{S} można ustalić macierz systemu \mathbf{A} uwzględniającą już sygnalizację świetlną. Niech Q będzie zbiorem odcinków, które znajdują się bezpośrednio przed ujściem ruchu. Wtedy macierz systemu zdefiniowana jest jako:

$$A[i, j] = \begin{cases} 0 & \text{dla } S[i, j] = 0 \vee i \in Q \\ P[i, j] & \text{dla } S[i, j] = 1 \\ 1 - \delta(i) & \text{dla } i = j \wedge i \notin Q \end{cases} \quad (5.14)$$

$$(5.15)$$

$$(5.16)$$

Równanie systemu jest następujące:

$$\mathbf{x}(t+1) = \mathbf{A}(t)\mathbf{x}(t) \quad (5.17)$$

Macierz systemu jest teraz zmienna w czasie, zatem w miejsce \mathbf{A} dotychczasowego równania pojawiło się $\mathbf{A}(t)$.

5.6.3 Przykład rozwoju wektora stanowego

Niech dla chwili $t=0$ będzie dany układ z przykładu (5.6.1). W chwili $t=2$ zostanie zmieniona faza świetlna. Od tego momentu obydwie manewry na skrzyżowaniu są dozwolone. W chwili $t=3$ wartości stanowe nie są całkowite, co nie przeszkadza modelowi matematycznemu. Wartości stanowe były identyfikowane do tej pory jako ilość pojazdów na odcinku i dalej jest to dobra interpretacja o ile przyjmujemy, że pojazdy są możliwe do podzielenia na części. Rozwój wektora stanu został przedstawiony w poniższej tabeli.

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	$\mathbf{x}(3) = \mathbf{A}(2)\mathbf{x}(2) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{3}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 9 \\ 3 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2\frac{1}{4} \\ 3 \\ 7\frac{3}{4} \\ 0 \end{bmatrix}$	

5.7 Wprowadzenie źródeł ruchu

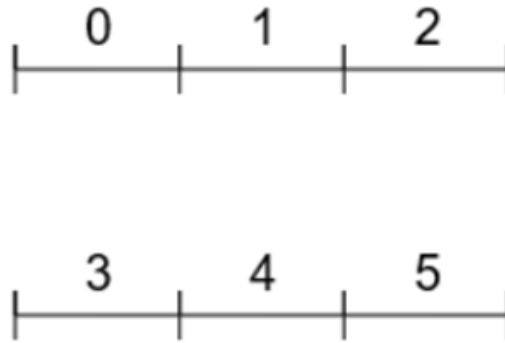
Wszystkie poprzednie przykłady układów ruchu drogowego szybko kończyły się stanem w którym nie było już żadnych pojazdów na drogach. W tej sekcji zostanie przedstawiony sposób napływania nowych pojazdów do układu. Wprowadzony zostanie wektor źródła $\mathbf{u}(t)$

definiujący pojazdy pojawiające się w układzie. Jest on zmienny w czasie, a jego wartości określają ile pojazdów pojawia się w następnej chwili w poszczególnych odcinkach układu. Równanie systemu uwzględniające źródła ruchu to:

$$\mathbf{x}(t) = \mathbf{A}(t-1)\mathbf{x}(t-1) + \mathbf{u}(t-1) \quad (5.18)$$

5.7.1 Przykład

Rozważony zostanie prosty przykład, który 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 5.4: Numeracja odcinków środowiska

Pojazdy wzdłuż dróg kończąc swój bieg po przejechaniu przez odcinki 5 i 2 co przedstawia macierz systemu A :

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

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

5.7.2 Zatory drogowe

Dotychczasowe przedstawienie macierzy systemu \mathbf{A} nie zawiera w sobie jeszcze pojęcia zatoru drogowego. W jednym interwale czasowym może przejechać przez odcinek astronomiczna wręcz liczba pojazdów. Wprowadzona zostanie zatem funkcja zatoru ograniczająca przejazdy w przypadku zbyt dużej liczby pojazdów. Dziedziną funkcji zatoru f jest zbiór wszystkich manewrów $[i, j]$, takich że $i \neq j$. Funkcja zatoru jest jednak zależna także od wartości stanowych wektora \mathbf{x} , a przede wszystkim od ilości pojazdów na odcinkach i oraz j czyli $x[i], x[j]$. Wartości funkcji f należą do przedziału $[0, 1]$. Macierz stanowa \mathbf{A} z uwzględnieniem zatorów dla układu z n odcinkami przedstawia się następująco:

$$\mathbf{A} = \begin{bmatrix} 1 - \delta(0) & f(1, 0) & \dots & f(n, 0) \\ f(0, 1) & 1 - \delta(1) & \dots & f(n, 1) \\ f(0, 2) & f(1, 2) & \dots & f(n, 2) \\ \dots & \dots & \dots & \dots \\ f(0, n) & f(1, n) & \dots & 1 - \delta(n) \end{bmatrix} \quad (5.19)$$

Gdzie δ to suma wszystkich wartości danej kolumny z wyłączeniem tych na głównej przekątnej. W przypadku odcinków zakończonych ujściem, aby wyzerować wartości δ przyjmuje

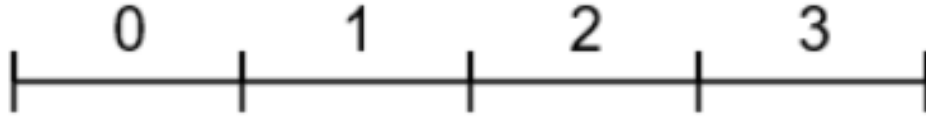
wartość 1. Formalnie δ to:

$$\delta(i) = \begin{cases} \sum_{j \in \{0, \dots, n\}, j \neq i} f(i, j) & \text{dla odcinka } i \text{ nie zakończonego ujściem} \\ 1 & \text{dla odcinka } i \text{ zakończonego ujściem} \end{cases} \quad (5.20)$$

$$(5.21)$$

5.7.3 Przykład zatoru na pojedynczej drodze

Rozważmy poniższe środowisko składające się z drogi posiadającej 4 odcinki.



Rysunek 5.5: Droga z numeracją odcinków

Należy skonstruować pewną funkcję zatoru. Niech jej celem będzie, aby przez jeden interwał czasowy maksymalnie 10 pojazdów przejeżdżało przez odcinek drogi. Następująca funkcja gwarantuje takie zachowanie:

$$f(i, j) = \begin{cases} \mathbf{P}[i, j] & \text{dla } j = i + 1 \wedge x[i] < 10 - \text{przejazd bez zatoru} \\ \frac{10}{x[i]} & \text{dla } j = i + 1 \wedge x[i] \geq 10 - \text{przejazd z zatorem} \\ 0 & \text{dla pozostałych przypadków - niemożliwe manewry} \end{cases} \quad (5.22)$$

$$(5.23)$$

$$(5.24)$$

Macierze stanowe zostały wyliczone na podstawie wzoru 5.19. W tym przypadku jest to:

$$A = \begin{bmatrix} 1 - \delta(0) & f(1, 0) & f(2, 0) & f(3, 0) \\ f(0, 1) & 1 - \delta(1) & f(2, 1) & f(3, 1) \\ f(0, 2) & f(1, 2) & 1 - \delta(2) & f(3, 2) \\ f(0, 3) & f(1, 3) & f(2, 3) & 1 - \delta(3) \end{bmatrix}$$

Funkcja δ dla tego środowiska to:

$$\delta(i) = \begin{cases} \sum_{j \in \{0, 1, 2, 3\}, j \neq i} f(i, j) & \text{dla } i \in 0, 1, 2 \\ 1 & \text{dla } i = 3 \end{cases} \quad (5.25)$$

$$(5.26)$$

Rozważmy następujący stan początkowy środowiska:

t	Równanie stanu	Podgląd środowiska
0	$\mathbf{x}(0) = \begin{bmatrix} 17 \\ 2 \\ 11 \\ 4 \end{bmatrix}$	

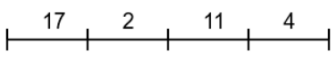
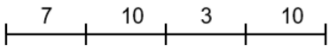
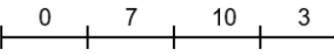
W celu wyliczenia $\mathbf{x}(1)$ należy obliczyć $\mathbf{A}(0)$, wedle wzoru (5.7.3). Wartości zerowej kolumny macierzy $A(0)$ są wyliczane w następującej kolejności:

- $f(0, 1)$ to przypadek manewru z zatorem (??). Zatem $f(0, 1) = \frac{10}{17}$, gdyż $x[0] = 17$
- $f(0, 2) = 0$ $f(0, 3) = 0$ to przypadki niemożliwych manewrów (5.13).
- $A[0, 0] = 1 - \delta(0) = 1 - f(0, 1) - f(0, 2) - f(0, 3) = 1 - \frac{10}{17} - 0 - 0 = \frac{7}{17}$ pozostaje na odcinku 0.

Analogiczne operacje zostają przeprowadzone dla kolejnych kolumn co daje rezultat w postaci:

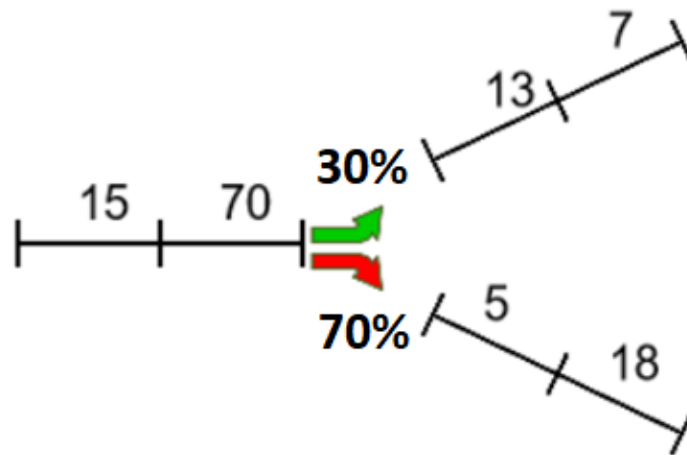
$$A(0) = \begin{bmatrix} \frac{7}{17} & 0 & 0 & 0 \\ \frac{10}{17} & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{11} & 0 \\ 0 & 0 & \frac{10}{11} & 0 \end{bmatrix}$$

Dalszy rozwój wektora stanowego jest przedstawiony w tabeli poniżej.

t	Równanie stanu	Podgląd środowiska
0	$\mathbf{x}(0) = \begin{bmatrix} 17 \\ 2 \\ 11 \\ 4 \end{bmatrix}$	
1	$\mathbf{x}(1) = \mathbf{A}(0)\mathbf{x}(0) = \begin{bmatrix} \frac{7}{17} & 0 & 0 & 0 \\ \frac{10}{17} & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{11} & 0 \\ 0 & 0 & \frac{10}{11} & 0 \end{bmatrix} \begin{bmatrix} 17 \\ 2 \\ 11 \\ 4 \end{bmatrix} = \begin{bmatrix} 7 \\ 10 \\ 3 \\ 10 \end{bmatrix}$	
2	$\mathbf{x}(2) = \mathbf{A}(1)\mathbf{x}(1) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 7 \\ 10 \\ 3 \\ 10 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \\ 10 \\ 3 \end{bmatrix}$	

5.7.4 Przykład zatoru na skrzyżowaniu

Niech funkcją zatoru będzie funkcja przedstawiona w 6.4-6.5 Rozważmy poniższą sytuację na drogach w chwili $t = 0$:



- Zatory są na odcinkach 0,1,2 i 5 (odpowiednio ilości pojazdów to 15,70,13,18).
- Obecna faza świetlna to 0.

Najistotniejszym punktem macierzy sygnalizacji świetlnej jest $S[1,2] = 1$, co odpowiada zielonemu światłu dla lewoskrętu. Cała macierz \mathbf{S} to:

$$\mathbf{S} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (5.27)$$

Macierz systemu to zatem:

$$\mathbf{A} = \begin{bmatrix} \frac{5}{15} & 0 & 0 & 0 & 0 & 0 \\ \frac{10}{15} & \frac{60}{70} & 0 & 0 & 0 & 0 \\ 0 & \frac{10}{70} & \frac{3}{13} & 0 & 0 & 0 \\ 0 & 0 & \frac{10}{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.28)$$

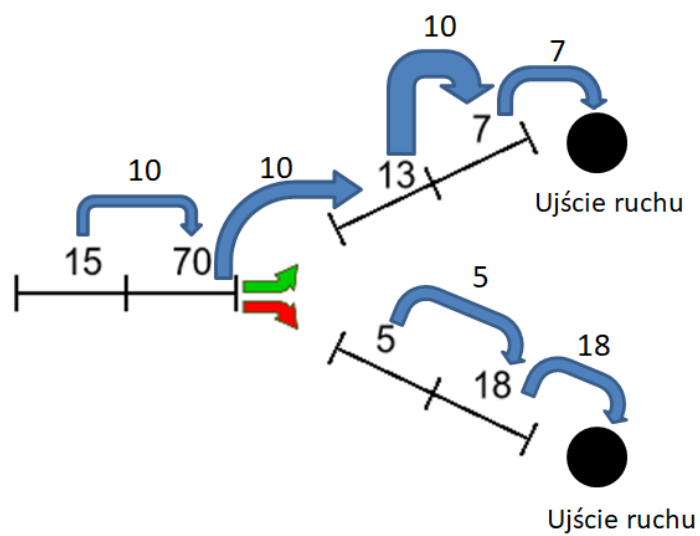
Dla przykładu szczegółowo zostaną policzone wartości zerowej kolumny macierzy \mathbf{A} , która dotyczy pojazdów wyjeżdżających z odcinka 0. Pozostałe wartości są liczone analogicznie.

- Korzystając ze wzoru 6.5 ustalone zostaje $f(0,1) = \frac{10}{15}$. Jest to przypadek zatoru i jedynie 10 pojazdów z 15 przemieszcza się do następnego odcinka.
- Wartości $f(0,2), (0,3), (0,4), (0,5)$ są równe 0, gdyż są to przypadki ?? nieistniejących manewrów.
- $A[0,0] = 1 - \delta(0) = 1 - f(0,1) - f(0,2) - f(0,3) - f(0,4) - f(0,5) = 1 - \frac{10}{15} - 0 - 0 - 0 - 0 = \frac{5}{15}$

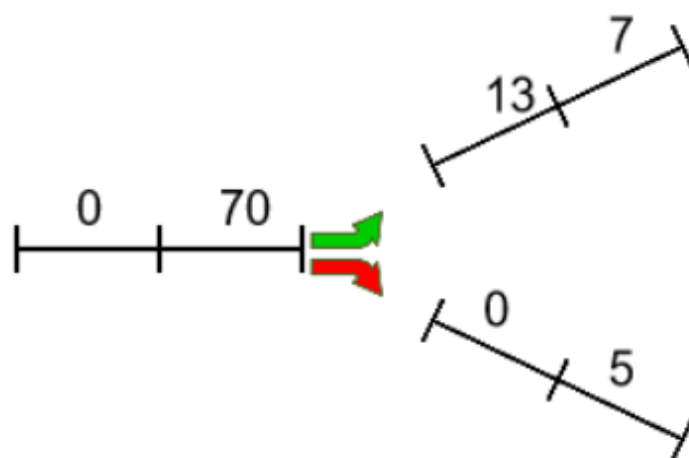
Stan w momencie $t = 1$ wyliczony z równania stanu jest następujący:

$$\mathbf{x}[1] = \mathbf{A}[0]\mathbf{x}[0] = \begin{bmatrix} \frac{5}{15} & 0 & 0 & 0 & 0 & 0 \\ \frac{10}{15} & \frac{60}{70} & 0 & 0 & 0 & 0 \\ 0 & \frac{10}{70} & \frac{3}{13} & 0 & 0 & 0 \\ 0 & 0 & \frac{10}{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 15 \\ 70 \\ 13 \\ 7 \\ 5 \\ 18 \end{bmatrix} = \begin{bmatrix} 12 \\ 70 \\ 13 \\ 10 \\ 0 \\ 5 \end{bmatrix}$$

Poniższy obraz przedstawia sytuację w środowisku w chwili $t-1$ oraz przepływ pojazdów, który miał miejsce między momentem $t-1$ a t .



W rezultacie w chwili t obecny jest następujący stan:



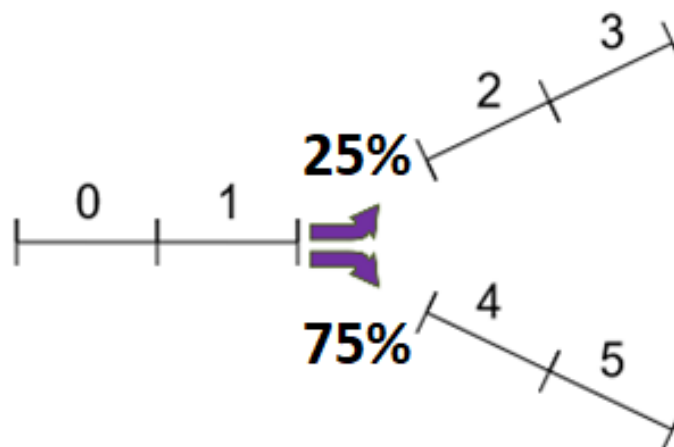
Rozdział 6

środowiska symulacyjne

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. Drugim elementem środowiska symulacyjnego są agenci. Są oni odpowiedzialni za dokonywanie decyzji związanych z sygnalizacją świetlną.

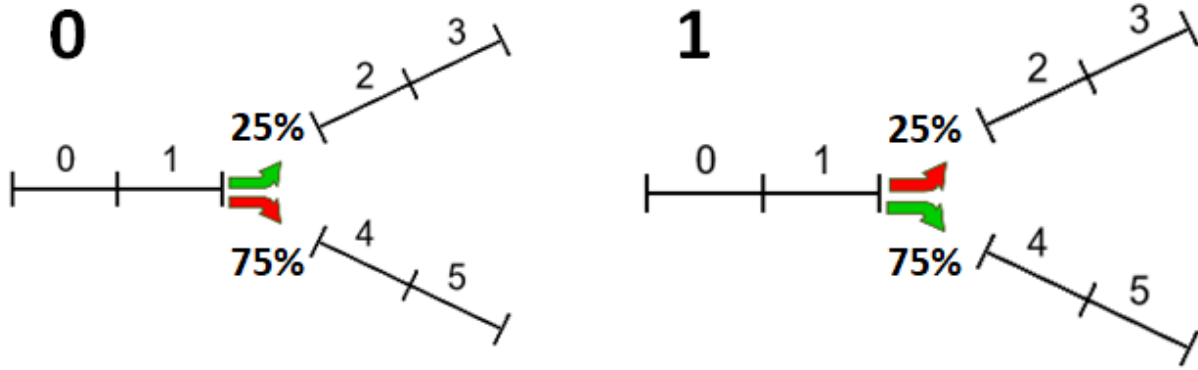
6.1 Środowisko 1(11 na froncie)

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 11

6.1.1 Sygnalizacje świetlne



Rysunek 6.2: środowisko 11 - 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 Zatory drogowe

Wykorzystana w modelu zostanie funkcja zatoru przedstawiona w rozdziale (5.7.3), czyli:

$$f(i, j) = \begin{cases} P[i, j] & \text{dla } j = i + 1 \wedge x[i] < 10 - \text{przejazd bez zatoru} & (6.1) \\ \frac{10}{x[i]} & \text{dla } j = i + 1 \wedge x[i] \geq 10 - \text{przejazd z zatorem} & (6.2) \\ 0 & \text{dla pozostałych przypadków - manewr [i,j]} & (6.3) \end{cases}$$

6.1.4 Uczenie agenta

6.1.5 Pożądane zachowanie

Pożądanym zachowaniem jest utrzymywanie fazy 1. Jest w każdym przypadku lepsza lub równa fazie 0. Spowodowane jest to tym, że więcej pojazdów skręca w prawo.

6.1.6 Sposób uczenia

Zastosowana została metoda uczenia przedstawiona w (7.6). Następujące stałe określające model uczenia to:

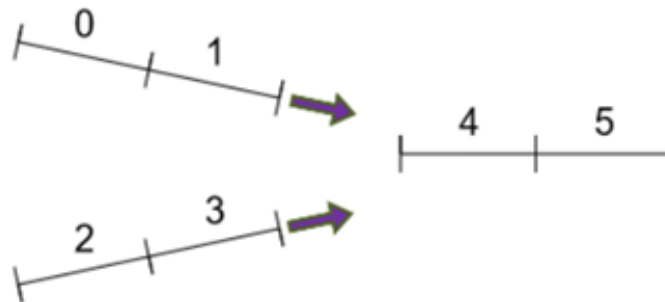
- $\gamma = 0.9$ - stopa dyskontowa
- $\alpha = 0.01$ - stała szybkości uczenia (ang. learning rate)
- Generowanych jest jednorazowo 50 epizodów
- Wygenerowane zbiory danych są dzielone w proporcjach 80 i 20 procent dla odpowiednio treningu i walidacji.
- Warunkiem stopu jest 60 sekund działania algorytmu.
- Sieć neuronowa posiada 1 ukrytą warstwę z 10 neuronami.

6.1.7 Wyniki uczenia

Za każdym razem już po trzeciej (często po drugiej) iteracji algorytmu osiągnane jest pożądane zachowanie agenta. Dla każdego stanu agent wybiera akcję 0.

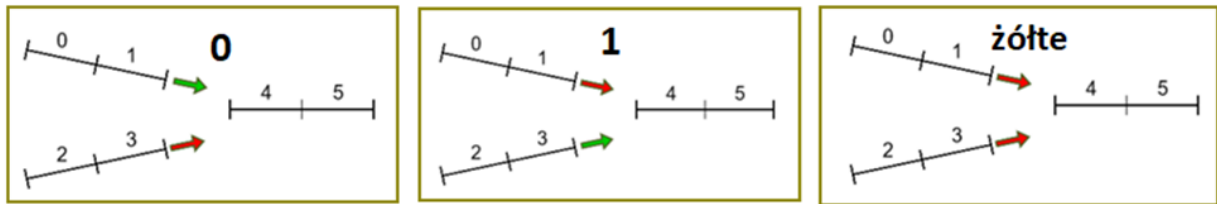
6.2 Środowisko 2(14 na froncie)

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

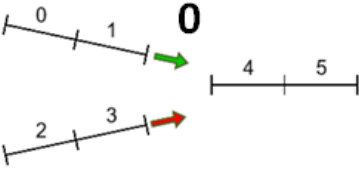
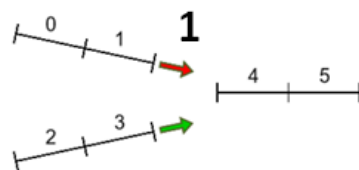
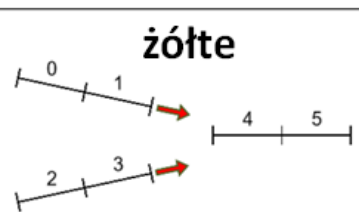


Rysunek 6.4: środowisko 2 - fazy świateł

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 Zatory drogowe

Wykorzystana w modelu zostanie funkcja zatoru przedstawiona w rozdziale (5.7.3), czyli:

$$f(i, j) = \begin{cases} P[i, j] & \text{dla } j = i + 1 \wedge x[i] < 10 - \text{przejazd bez zatoru} \\ \frac{10}{x[i]} & \text{dla } j = i + 1 \wedge x[i] \geq 10 - \text{przejazd z zatorem} \\ 0 & \text{dla pozostałych przypadków - manewr [i,j]} \end{cases}$$

(6.4)

(6.5)

(6.6)

6.2.4 Cel nauki i pożądane zachowanie agenta

Początkowo do każdej z dróg wjeżdżają po 2 pojazdy. Agent ma za zadanie wykonywać akcje, które w trakcie symulacji pozwolą na opuszczenie skrzyżowania przez co najmniej 98 procent pojazdów, które wjechały do sieci dróg. W przypadku spełnienia tego warunku następuje zwiększenie ilości wjeżdżających do układu pojazdów. Ten proces jest powtarzany, aż do momentu gdy przez 10 iteracji uczenia nie uzyskany zostanie warunek 98 procent pojazdów opuszczających skrzyżowanie. Moment stopu spodziewany jest na chwilę, gdy na każdą z dróg będzie wjeżdżało około 5 pojazdów. Nie ma możliwości, by przejeżdżało więcej niż 10 pojazdów, a tyle właśnie pojawia się w układzie. Muszą zatem wtedy pojawiać się zatory.

Pożądane zachowanie agenta w tym przypadku nie jest jednoznaczne do określenia. Z pewnością jeśli na drodze na której jest zielone światło jest conajmniej 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 musi odnaleźć złoty środek i jeśli na drugiej drodze jest dużo pojazdów - może zmienić światło.

6.2.5 Sposób uczenia

Zastosowana została metoda uczenia przedstawiona w (7.6).

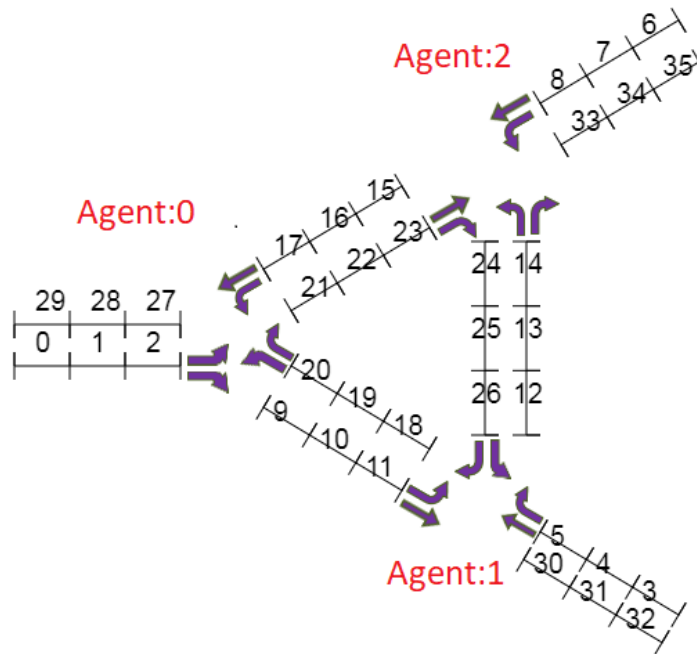
Następujące stałe oraz ustalenia określają szczegóły uczenia:

- $\gamma = 0.9$ - stopa dyskontowa
- 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. 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 $learning_rate = 0,01$.

6.2.6 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.

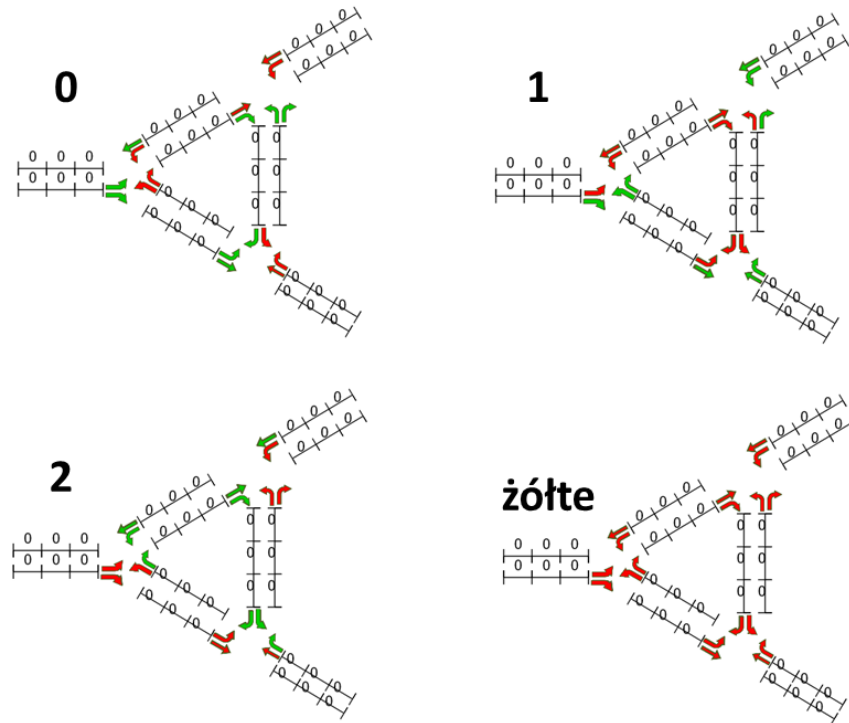
6.3 Środowisko 4



Rysunek 6.5: środowisko 4

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.

Model ruchu: Pojazdy w jednym interwale czasowym pokonują jeden odcinek. Na skrzyżowaniach w przypadku zielonego światła przejeżdża maksymalnie 10 pojazdów w jedną stronę. Fazy świetlne: Każde skrzyżowanie posiada 4 fazy świetlne przedstawione poniżej. Fazy 0,1 i 2 są fazami, które posiadają pewne zielone światła. Agent podejmuje decyzję o zmianie tych faz. Zmiana ta nie jest natychmiastowa i następuje dopiero po 2 interwałach fazy żółtych świateł.



Rysunek 6.6: środowisko 4 - fazy świateł

Uczenie: Każdy agent jako stan przyjmuje 10 elementową tablicę. 9 elementów to ilości pojazdów na odcinkach będących przed skrzyżowaniem przypisanym do agenta.

Rozdział 7

Przegląd metod optymalizacyjnych

7.1 Kategorie uczenia maszynowego

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

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 o ich wysokościach i szerokości korony drzew.

3. Wzmocnione

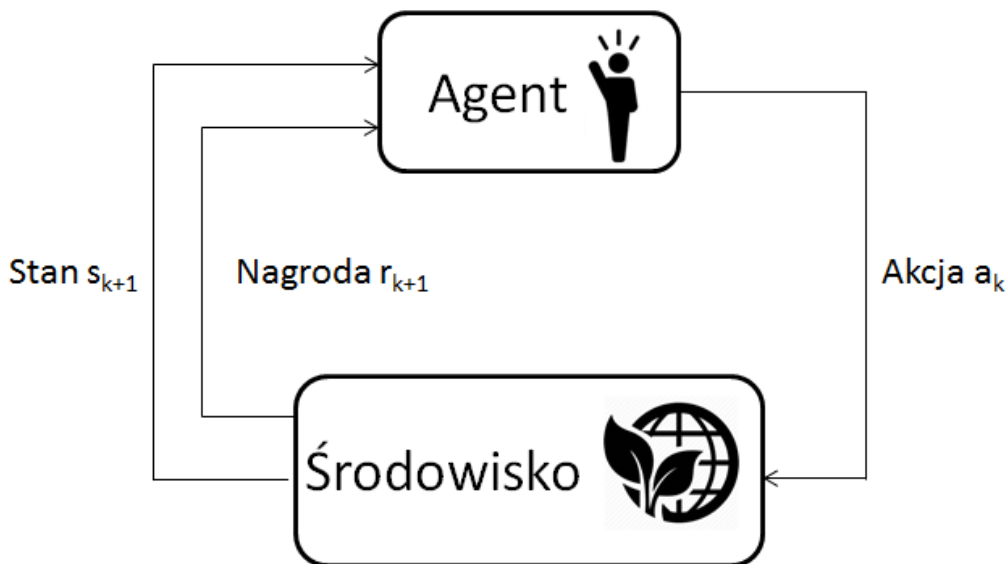
W środowisku uczenia ze wzmocnieniem istnieje agent, który jest odpowiedzialny za podejmowanie pewnych decyzji. Każda decyzja ma wpływ na stan środowiska, które zwraca agentowi nagrodę. Celem uczenia ze wzmocnieniem jest ustalenie strategii maksymalizującej skumulowaną wartość nagród.

Ze wszystkich trzech kategorii uczenie ze wzmocnieniem odpowiada problemowi rozważanym w rozdziale Y. Szczegółowy opis uczenia ze wzmocnieniem zostanie przedstawiony w następnej sekcji.

7.2 Uczenie ze wzmocnieniem

Schemat uczenia ze wzmocnieniem składa się z następujących elementów:

1. **Agent** jest odpowiedzialny za podejmowanie pewnych decyzji. Ma on wiedzę na temat obecnego stanu środowiska i otrzymuje w każdym kroku czasowym sygnał nagrody. Jego decyzje niebezpośrednio wpływają na stan 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. Środowisko jest także nadawcą sygnału nagrody.
3. **Strategia** definiuje sposób doboru akcji przez agenta w danej chwili. Jest to funkcja, która przyjmuje stan środowiska i zwraca akcję, która ma być przeprowadzona.
4. **Sygnał nagrody** definiuje cel problemu uczenia ze wzmocnieniem. W każdym kroku czasowym środowisko wysyła agentowi liczbę rzeczywistą, która jest nazywana nagrodą(reward). 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.
5. **Funkcja wartości** zwraca wartość stanu czyli oczekiwaną sumę nagród jakie agent osiągnie w przyszłości będąc aktualnie w tym stanie.



Rysunek 7.1: Interakcje pomiędzy agentem a środowiskiem.

Algorytmy uczenia ze wzmocnieniem zazwyczaj stosuje się do rozwiązywania problemu procesu decyzyjnego Markowa. Sam **proces decyzyjny Markowa** jest zdefiniowany jako uporządkowana czwórka (S, A, P_a, R_a) , gdzie:

1. S to zbiór stanów
2. A to zbiór akcji. Notacją A_s oznaczane są możliwe akcje dla stanu s .
3. $P_a(s, s') = \text{Pr}(s_{t+1} = s' | s_t = s, a_t = a)$ 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' .

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 wartość:

$$G = \sum_{k=0}^K \gamma^k R_k \quad (7.1)$$

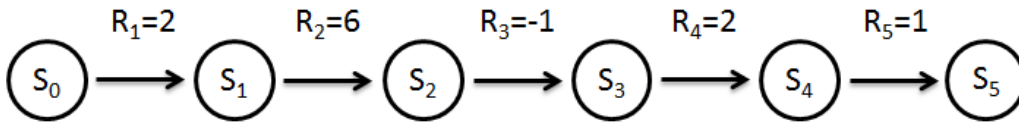
Chociaż strategii $\pi(s)$ nie ma we wzorze 7.1, to strategia wpływa na otrzymywane nagrody R_k w każdej chwili k .

$\gamma \in (0, 1]$ jest czynnikiem dyskontującym. Idea dyskontowania nagród zaczerpnięta jest z rachunku finansowego. Przykładowo wpływ na konto kapitału 1000 złotych po upływie roku czasu jest z pewnością bardziej wartościowy niż za 20 lat. Innymi słowy - pieniądze są 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.

Analogicznie do (7.1) jest ustalona funkcja wartości stanu. Jako wartość stanu s określone jest:

$$G_t = \sum_{k=t}^K \gamma^k R_k = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \dots + \gamma^K R_K \quad (7.2)$$

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 G oraz G_1, G_2, G_3, G_4, G_5 ?



Rysunek 7.2: Ciąg nagród i stanów.

Najłatwiej obliczenia rozpocząć od G_5 , ponieważ

$$G_t = R_t + \gamma G_{t+1} \quad (7.3)$$

$$G_5 = R_5 = 1.$$

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

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

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

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

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

7.3 Programowanie dynamiczne

Termin programowania dynamicznego odnosi się do algorytmów wyliczających optymalne strategie procesu decyzyjnego Markowa w przypadku posiadanej całkowitej wiedzy na temat modelu środowiska [?]. Środowisko nie musi być w pełni deterministyczne tzn. nie za każdym razem akcja przeprowadzana ze stanu s_k musi w efekcie doprowadzić do tego samego stanu s_{k+1} . Jednak w takim przypadku musi być znany rozkład prawdopodobieństwa przydzielania nowego stanu na podstawie poprzedniego i właśnie podjętej przez agenta akcji. Dodatkowo wymagana jest możliwość ustalenia dowolnego stanu w trakcie uczenia.

Początkowa strategia $\pi(s)$ jest dowolna, najczęściej losowa. Przedstawiony algorytm jest podzielony na 2 części. Część predykcji(prediction) oraz kontroli (control).

Proces predykcji ma za zadanie ustalenie wartości stanów na podstawie ustalonej strategii. Jej algorytm jest następujący:

1. Przyjmij daną z góry π jako strategię podejmowania akcji
2. Zainicjuj tablicę wartości stanów $V(s)$. Dla wszystkich możliwych stanów $s \in S$ przyjmij wartość 0.
3. $\Delta = 0$
4. Dla każdego $s \in S$:
 - (a) $v = V(s)$
 - (b) $V(s) = \sum_{s'} Pr(s'|s, \pi(s)) [r + \gamma V(s')]$
 - (c) $\Delta = \max(\Delta, |v - V(s)|)$
5. Jeśli $\Delta < \theta$ to wróć do 3
6. Zwróć $V(s)$ jako tablicę wartości stanów $V_\pi(s)$ dla strategii π .

Parametr $\theta \geq 0$ definiuje w kroku 5. moment stopu.

Wartość $Pr(s', s, \pi(s))$ to prawdopodobieństwo, że akcja $\pi(s)$ podjęta w stanie s doprowadzi do stanu s' . Z kolei r jest właśnie otrzymaną nagrodą.

Algorytm **kontroli** ma za zadanie odnaleźć bardziej optymalną strategię niż dotychczas.

Jako argument przyjmuje on wyliczoną właśnie tablicę wartości stanów $V_\pi(s)$. Wprowadzona zostaje macierz $Q_\pi(s, a)$. Jest ona zdefiniowana następująco:

$$\begin{aligned} Q_\pi(s, a) &= E[R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s, A_t = a] \\ &= \sum_{s'} Pr(s' | s, a) [r + \gamma V_{\pi}(s')] \end{aligned} \quad (7.4)$$

Oczywistym minusem tego algorytmu jest konieczność przeiterowania wszystkich stanów. W przypadku gdy stanów jest bardzo dużo algorytm staje się nieopłacalny. Macierz przedstawia wartość akcji a podjętej w stanie s . Algorytm kontroli jest następujący:

1. Przyjmij wyliczoną przez algorytm predykcji macierz wartości stanów $V_\pi(s)$
2. Zainicjuj tablicę wartości stanów $V(s)$. Dla wszystkich możliwych stanów $s \in S$ przyjmij wartość 0.
3. *policy_stable* = *false*
4. Dla każdego $s \in S$:
 - (a) *old_action* = $\pi(s)$
 - (b) $\pi(s) = \operatorname{argmax}_{a \in A_s} \sum_{s'} Pr(s' | s, a) [r + \gamma V(s')]$
 - (c) Jeśli *old_action* $\neq \pi(s)$ to *policy_stable* = *true*
5. Jeśli *policy_stable* = *true* to zwróć strategię $\pi(s)$.

7.4 Metoda Monte Carlo On-Policy

Metody Monte Carlo są szeroką klasą algorytmów, których wyniki oparte są o losowe próbkowanie. Nie potrzebują one żadnej wiedzy na temat środowiska - akceptowalne są zarówno środowiska deterministyczne jak i stochastyczne. Algorytmy Monte Carlo uczenia ze wzmocnieniem są dzielone na dwie kategorie - On-Policy oraz Off-policy. W pracy zostanie przedstawiona metoda on-policy, która różni się od off-policy jedynie sposobem doboru momentu eksploracji. Metoda on-policy zakłada, iż ustalana jest pewna liczba ϵ bliska zeru. Określa ona prawdopodobieństwo kroku eksploracji, czyli wyboru losowej akcji. Pozostałe akcje są wybierane w sposób zachłanny, czyli podejmowana jest najbardziej wartościowa dostępna akcja. Algorytm przedstawiony jest poniższym pseudokodem.

1. Zainicjuj słowniki:
 - (a) $Q(s, a)$ - Wartość określa opłacalność wyboru akcji a w stanie s
 - (b) $Returns(s, a)$ - Wartość słownika to tablica wartości G wyliczonych na podstawie wzoru (7.2).
 - (c) $\pi(s)$ - Określa jaka akcja ma zostać podjęta dla stanu s . Początkowo wszystkie akcje są wybrane losowo.
2. Zasymuluj pełny epizod wedle strategii π

3. Zoptymalizuj strategię π - dla każdej pary (s,a) - stanu i akcji, które pojawiły się w epizodzie
 - (a) Wylicz G rekurencyjnie wedle wzoru (7.3) t.j. $G_t = R_t + \gamma G_{t+1}$. *
 - (b) Do tablicy Returns(s,a) dodaj wartość G
 - (c) $Q(s, a) = \text{avg}(\text{Returns}(s, a))$
 - (d) $\pi(s) = a^*$. Gdzie a^* to taka akcja, że $Q(s, a^*) \geq Q(s, a)$ dla każdego $a \in A$. Jeśli został jednak wylosowany krok eksploracji, wtedy $\pi(s)$ jest losowo wybraną akcją $a \in A$.

* - Rozsądnym jest iterowanie w 3 kroku w kolejności odwrotnej do chronologicznej, gdyż ułatwia wykorzystanie wzoru (7.3)

Warto zauważyć następujące zalety metody Monte Carlo, których nie zapewniał algorytm programowania dynamicznego:

1. Brak potrzeby wiedzy na temat modelu
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. Dobre skalowanie dla dużej przestrzeni stanu. Więcej czasu treningowego jest poświęcane dla stanów, które są często odwiedzane. Z kolei te, które nie pojawiają się prawie wcale - nie marnują dużo czasu w procesie nauki.

7.5 Mój algorytm który potrzebuje nazwy ale proponuję: Monte Carlo DQN full exploration

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a' \in A'} Q(s', a')] \quad (7.5)$$

- s, a to para stanu i akcji, które pojawiły się w trakcie symulacji
- $Q(s, a)$ to opłacalność podjęcia akcji a w stanie s .
- α to stała szybkości uczenia (ang. learning rate)
- r to nagroda przydzielona podczas symulacji dla akcji a podjętej w stanie s
- $\gamma \in [0, 1]$ to stopa dyskontowa
- s' to stan do którego doprowadziła akcja a podjęta w stanie s podczas symulacji
- A' to przestrzeń możliwych akcji do podjęcia w stanie s'

Algorytm jest następujący:

1. Utwórz sieć neuronową, która na wejście przyjmować będzie stany środowiska. Wyjściowym sygnałem jest opłacalność akcji możliwych do podjęcia w stanie danym na wejściu.
2. Zasymuluj pewną ilość epizodów wedle losowej strategii. Zapisuj dane wygenerowane podczas tych epizodów:
 - (a) Stany środowiska
 - (b) Podjęte akcje
 - (c) Przydzielone nagrody
3. Na podstawie zapisanych danych utwórz następujące zbiory wykorzystane w sieci neuronowej:
 - (a) Zbiór sygnałów wejściowych (x-batch). Pojedynczym elementem tego zbioru jest tablica przedstawiająca stan środowiska.
 - (b) Zbiór oczekiwanych sygnałów wyjściowych (y-batch), czyli opłacalności akcji do podjęcia w stanie zadanym na wejściu. Dla elementu odpowiadającego akcji, która została podjęta w symulacji wylicz wartość wedle równania (7.6). Dla elementów odpowiadających reszcie akcji, które nie zostały podjęte - sieć neuronowa przewiduje nagrodę.
4. Podziel powyższe zbiory na segment treningowy i walidacyjny.
5. Trenuj sieć neuronową do momentu, gdy wyniki dla zbioru walidacyjnego się polepszają.
6. Jeśli spełniony jest warunek stopu - przerwij algorytm. W przeciwnym razie przejdź do 2.

7.6 Mój algorytm który potrzebuje nazwy ale proponuję: Monte Carlo DQN

$$Q(s, a) = (1 - \alpha)Q(s, a) + \alpha[r + \gamma \max_{a' \in A'} Q(s', a')] \quad (7.6)$$

- s, a to para stanu i akcji, które pojawiły się w trakcie symulacji
- $Q(s, a)$ to opłacalność podjęcia akcji a w stanie s .
- r to nagroda przydzielona podczas symulacji dla akcji a podjętej w stanie s
- s' to stan do którego doprowadziła akcja a podjęta w stanie s podczas symulacji
- A' to przestrzeń możliwych akcji do podjęcia w stanie s'
- α to stała szybkości uczenia (ang. learning rate)

- $\gamma \in [0, 1]$ to stopa dyskontowa

Algorytm jest następujący:

1. Utwórz sieć neuronową, która na wejście przyjmować będzie stan środowiska. Wyjściowym sygnałem jest $Q(s,a)$ - opłacalność akcji możliwych do podjęcia w stanie danym na wejściu.
2. Zasymuluj pewną ilość epizodów. Akcje są wybierane w ϵ - greedy sposób. Zapisuj dane wygenerowane podczas tych epizodów:
 - (a) Stany środowiska
 - (b) Podjęte akcje
 - (c) Przydzielone nagrody
3. Na podstawie zapisanych danych utwórz następujące zbiory danych wykorzystane w sieci neuronowej:
 - (a) Zbiór sygnałów wejściowych (x-batch). Pojedynczym elementem tego zbioru jest tablica przedstawiająca stan środowiska.
 - (b) Zbiór oczekiwanych sygnałów wyjściowych (y-batch), czyli opłacalności akcji do podjęcia w stanie zadanym na wejściu. Dla elementu odpowiadającego akcji, która została podjęta w symulacji wylicz wartość wedle równania (7.6). Dla elementów odpowiadających reszcie akcji, które nie zostały podjęte - sieć neuronowa przewidyje nagrodę.
4. Podziel powyższe zbiory na segment treningowy i walidacyjny.
5. Trenuj sieć neuronową do momentu, gdy wyniki dla zbioru walidacyjnego się polepszają.
6. Jeśli spełniony jest warunek stopu - przerwij algorytm. W przeciwnym razie przejdź do 2.