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

Michał Lis

7 kwietnia 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 21
5.7 Wprowadzenie źródeł ruchu 24
5.7.1 Przykład dla pojedynczej drogi 25
5.7.2 Przykład dla złożonej sieci dróg 26
5.8 Wprowadzenie gęstości ruchu 26
5.9 Układy symulacyjne 26

6 Przedstawienie problemu optymalizacyjnego **27**

7 Przegląd metod optymalizacyjnych **29**
7.1 Kategorie uczenia maszynowego 29
7.2 Uczenie ze wzmocnieniem 29
7.3 Programowanie dynamiczne 32

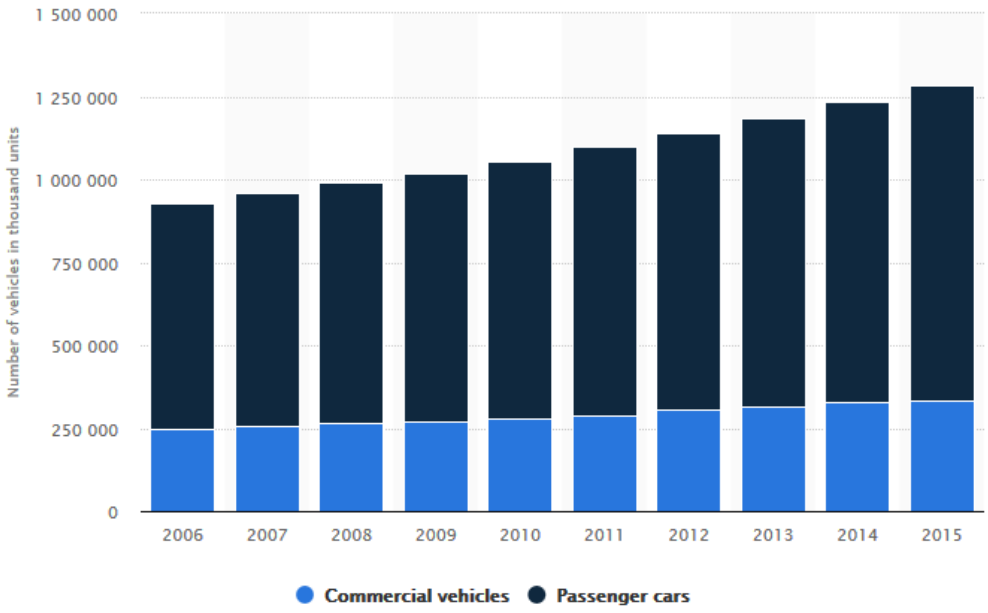
7.4 Metoda Monte Carlo On-Policy 33

8 Optymalizacja sygnalizacji świetlnej 37

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 [1]. W 2015 roku na świecie jeździło już 1282 tysięcy pojazdów. Wzrost przez te 9 lat był niemalże liniowy. Co roku rejestrowano około 39,4 tysięcy 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 [2]. 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 [3]. 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 [4]. Następny raport z 2017 roku może napawać optymizmem, gdyż przedstawione w nim wyliczenia określiły jedynie $0,77\%$ straty całkowitego PKB wspólnoty [5]. 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 świetlnych. 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

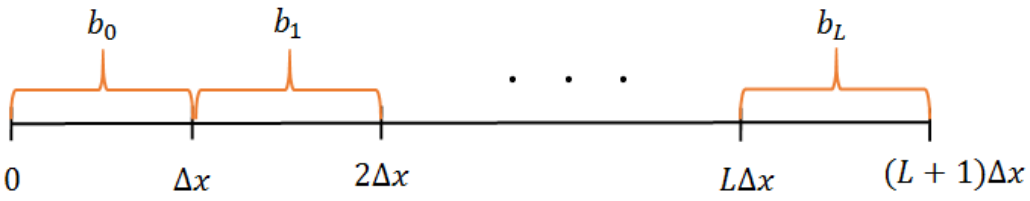
Dyskretny charakter modelu przedstawianego w pracy obliguje do określenia siatki czasowej i przestrzennej. Dla par czasu i miejsc należących do tych dwóch siatek będą określane zmienne stanu.

Siatka czasowa jest zdefiniowana jako skończony ciąg liczb naturalnych:

$$(0, 1, ..., K). \tag{3.1}$$

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 [6] - makroskopowy, mezoskopowy oraz mikroskopowy. Czasem [7] 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 [8]. Przykładowo pojazdy są zgrupowane w grupę podróżującą z pewnego punktu startowego do celu. Inne modele [9] 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 [10]. 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 [11] 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 [12] 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 nieścisła 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:

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

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 $p_0(x)$. Równianie (4.2) określa wedle założeń modelu makroskopowego [11] rozwój gęstości ruchu na drodze. Funkcja płynności ruchu f powinna być wklęsła [ref]. W przedstawionym w tej pracy modelu funkcja ma następującą definicję:

$$f(p) = \begin{cases} \lambda p & \text{dla } \ p \in [0, p^*] \quad (4.3) \\ \lambda \cdot (2p^* - p) & \text{dla } \ p \in (p^*, p^{max}] \quad (4.4) \end{cases}$$

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 $k = 0, 1, ..., K$. 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:

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

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:

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

Rezultatem jest końcowy rekurencyjny wzór na gęstość ruchu:

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

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: ujęcie sygnalizacji świetlnej, brak kolizyjnych manewrów, makroskopowy przepływ ruchu, przepływ ruchu na skrzyżowaniu, struktura sieci dróg. Zestawienie w jednej sekcji wszystkich tych kwestii byłoby bardzo przytłaczające.

5.2 Wektor stanu drogi

Wektor stanu jest strukturą w pełni przedstawiającą aktualny stan drogi. Dla każdego odcinka drogi składuje on wartości zmiennych stanuch. Początkowo zmienna stanowa jest identyfikowana jako ilość pojazdów na danym odcinku drogi.

5.2.1 Przykład

Niech będzie dana droga e z wydzielonymi czterema odcinkami b_1, b_2, b_3, b_4. Przykładowy wektor stanu to

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

Zawiera on w sobie następujące informacje dla chwili t:

- Są 2 pojazdy na odcinku b_1
- Są 4 pojazdy na odcinku b_2
- Są 3 pojazdy na odcinku b_3
- Nie ma żadnego pojazdu na odcinku b_4

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, a pojazdy będące w chwili t w ostatnim odcinku drogi układ. Formalnym wzorem definiującym rozwój wektora stanu jest:

$$\mathbf{x(t+1)} = \mathbf{Ax(t)} \tag{5.2}$$

Gdzie A jest macierzą systemu. Definiuje ona sposób przepływu pojazdów. 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

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

Macierzą systemu jest:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5.4}$$

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

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

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

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

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

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 stanuch jednak jest bardzo podobny. Każda z dróg $e_1, ..., e_n$ ma k wydzielonych odcinków oznaczanych jako $b_1, ..., b_{nk}$. Dla każdego z odcinków definiowana jest wartość stanowa.

5.4.1 Przykład

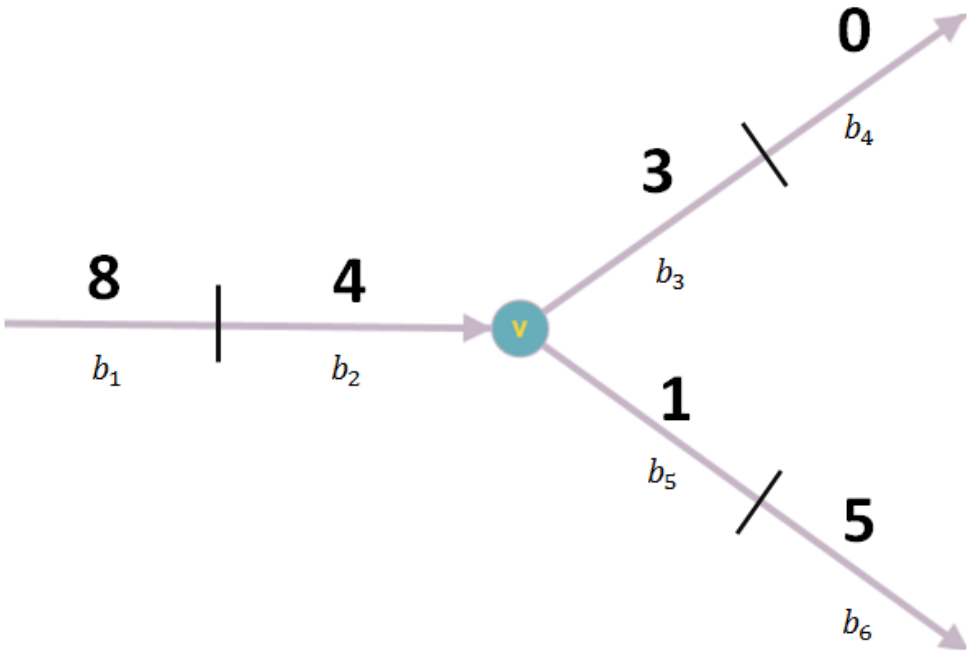
Niech będzie dana sieć składająca się z trzech dróg $E = \{e_1, e_2, e_3\}$. Dla każdej drogi zostaną wydzielone 2 odcinki. Przykładowy wektor stanu

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

Zawiera w sobie następujące informacje dotyczące ilości pojazdów na poszczególnych odcinkach w chwili t:

- $\boldsymbol{b_1}$ - Na pierwszym odcinku drogi e_1 jest 8 pojazdów
- $\boldsymbol{b_2}$ - Na drugim odcinku drogi e_1 są 4 pojazdy
- $\boldsymbol{b_3}$ - Na pierwszym odcinku drogi e_2 są 3 pojazdy
- $\boldsymbol{b_4}$ - Na drugim odcinku drogi e_2 nie ma żadnych pojazdów
- $\boldsymbol{b_5}$ - Na pierwszym odcinku drogi e_3 jest 1 pojazd

- $\boldsymbol{b_6}$ -Na drugim odcinku drogi e_3 jest 5 pojazdów



Rysunek 5.2: Sieć dróg z ilościami pojazdów na poszczególnych odcinkach

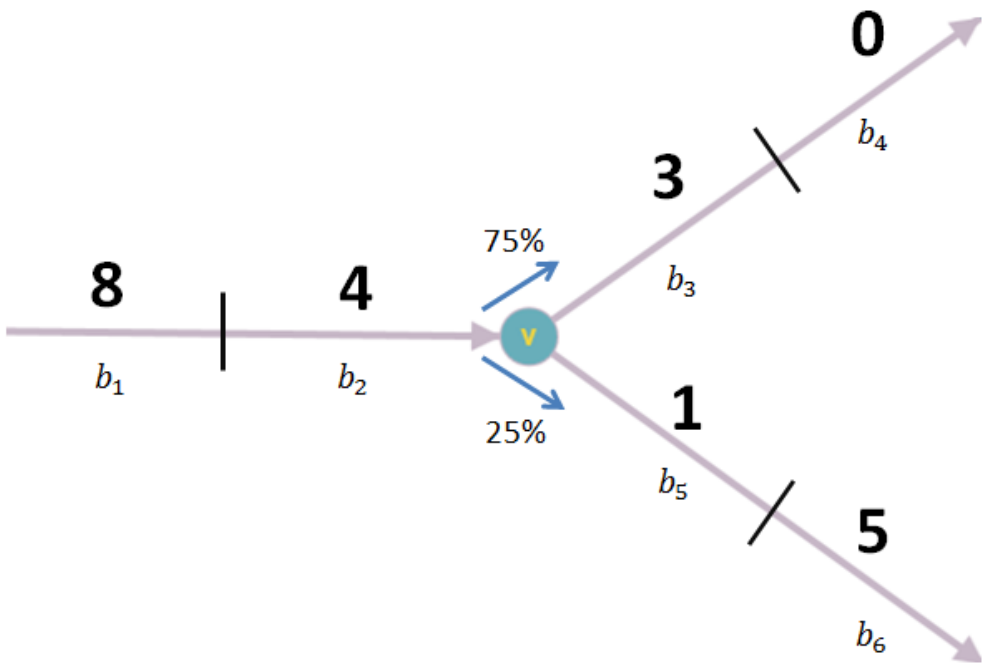
5.5 Rozwój wektora stanu sieci dróg

Przepływ pojazdów niezmiennie jest oparty o założenie, iż w trakcie trwania jednego interwału czasowego pojazdy pokonują 1 odcinek drogi. Równaniem systemu pozostaje $\mathbf{x(t+1)=Ax(t)}$, gdyż do układu niezmiennie nie wpływają nowe pojazdy.

Macierz systemu **A** powinna uwzględnić przepływy pojazdów na skrzyżowaniach. W tym momencie należy przedstawić następującą definicję macierzy **A**:
Wartości macierzy **A** określają jaka część pojazdów z odcinka zadanego przez indeks kolumny przejeżdża do odcinka zadanego przez indeks wiersza.

5.5.1 Przykład

Dla przykładu (5.4.1) przedstawiony zostanie rozwój wektora stanu. Założone zostaje, iż 75% pojazdów będących na odcinku b_2 opuszcza skrzyżowanie na odcinku b_3 a pozostałe 25% przejeżdża do b_5.



Rysunek 5.3:

Wtedy:

$$\mathbf{A} = \begin{array}{c} \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 75\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 25\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{matrix} \tag{5.10}$$

Niech zatem zgodnie z rysunkiem 5.3:

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

Kolejne wartości wektora stanu to:

$$\mathbf{x(1)} = \mathbf{Ax(0)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 75\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 25\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 8 \\ 4 \\ 3 \\ 0 \\ 1 \\ 5 \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x(2)} = \mathbf{Ax(1)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 75\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 25\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 8 \\ 3 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 6 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

$$\mathbf{x(3)} = \mathbf{Ax(2)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 75\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 25\% & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 6 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 6 \\ 0 \\ 2 \end{bmatrix}$$

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 niezmienność macierzy **A** w czasie. W chwili pojawienia się sygnalizacji świetlnej macierz systemu będzie oparta o aktualną fazę sygnalizacji świetlnej. Równanie systemu pozostaje takie samo jak w przypadku braku sygnalizacji świetlnej z jedną małą różnicą:

$$\mathbf{x(t+1)} = \mathbf{A(t)x(t)} \tag{5.11}$$

Macierz systemu jest zmienna w czasie, zatem w miejse **A** pojawiło się $\mathbf{A(t)}$.

Dla rozważanej sieci dróg zostaje przedstawiona **macierz topologii** układu **T**. Jej wartości podobnie jak macierzy **A** odnoszą się do tego jaka część pojazdów z odcinka zadanego przez indeks kolumny przejeżdża do odcinka zadanego przez indeks wiersza. Macierz **A** dotyczy jednak możliwych przejazdów w konkretnej fazie sygnalizacji świetlnej. Natomiast macierz topologii **T** odnosi się do wszystkich możliwych przejazdów - uwzględniając wszystkie fazy sygnalizacji świetlnej i jest stała w czasie.

Macierz sterowania (sygnalizacją świetlną) oznaczana będzie jako $\mathbf{S(t)}$. Ustala ona macierz systemu na podstawie następującej równości:

$$\mathbf{A(t)} = \mathbf{S(t)T} \tag{5.12}$$

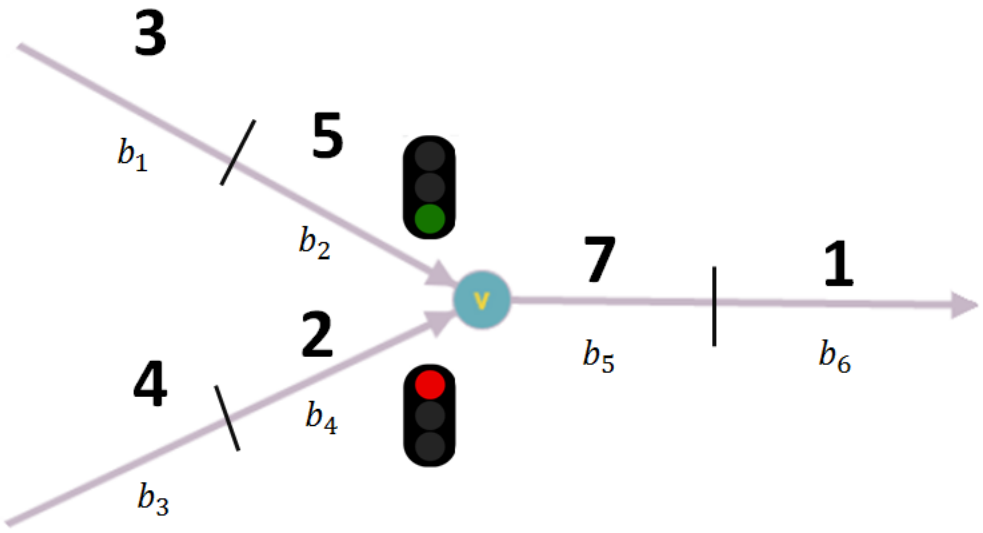
Jeśli $S(t)$ określa przejazdy, które są rzeczywiście możliwe w układzie, to:

$$\mathbf{A(t)} = \mathbf{S(t)} \tag{5.13}$$

Zatem celem wprowadzenia macierzy topologii oprócz przedstawienia wszystkich możliwych przejazdów jest także walidacja przejazdów zadanych przez macierz sterowania $\mathbf{S(t)}$.

5.6.1 Przykład

Rozważony zostanie przykład sieci trzech dróg e_1, e_2, e_3. Każda z nich jest podzielona na dwa odcinki. Drogi e_1 i e_2 zbiegają się na skrzyżowaniu z sygnalizacją świetlną. Założone jest, że sygnalizacja zmienia się w każdym kroku czasowym, jednak w międzyczasie przez jeden interwał czasowy jest żółte światło dla obydwu dróg. Matematycznie ujmując żółte światło jest równoważne czerwonemu - pojazdy czekają na skrzyżowaniu. Początkowy stan sieci przedstawia rysunek:



Rysunek 5.4: Sieć dróg z sygnalizacją świetlną w chwili t=0

Należy zastanowić się nad macierzą topologii układu **T**. Jest ona następująca:

$$\mathbf{T} = \begin{array}{c} \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{matrix} \tag{5.14}$$

- Wartości w kolumnach b_1, b_3, b_5 są równe 1 tylko w wierszach odpowiadającym odcinkom b_2, b_4, b_6, co wynika z tego, że pojazdy będące na odcinkach b_1, b_3, b_5 mają tylko możliwość przejazdu do odpowiednio b_2, b_4, b_6.

- Wartości w kolumnie b_6 są zerowe, gdyż pojazdy z odcinku b_6 opuszczają układ.

- Najbardziej interesujące są kolumny b_2 i b_4. Odpowiadają one odcinkom leżącym bezpośrednio przed skrzyżowaniem. Obydwie kolumny posiadają dwie wartości 1, gdyż pojazdy będące na odcinkach b_2, b_4 mają dwie możliwości przejazdu. Pojazdy będące na odcinku b_2 mogą przejechać przez skrzyżowanie i wjechać na b_5, albo pozostać dalej na b_2 w przypadku czerwonego światła. Analogiczna sytuacja dotyczy pojazdów na odcinku b_4. Zielone i czerwone tło określają przy jakiej sygnalizacji świetlnej opisany przejazd jest możliwy.

Początkowy wektor stanu (zgodny z rysunkiem 5.4) to:

$$\mathbf{x(0)} = \begin{bmatrix} 4 \\ 5 \\ 4 \\ 2 \\ 7 \\ 1 \end{bmatrix}$$

Macierz stanu w momentach $t = 0, 4$ uwzględnia zielone światło dla drogi e_1 oraz czerwone dla e_2.

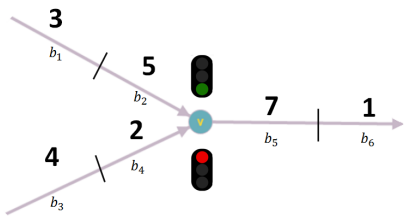
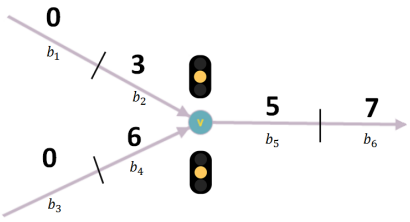
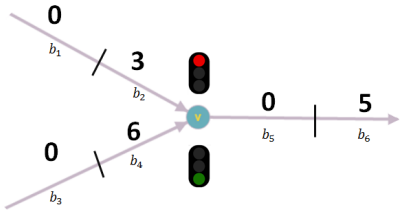
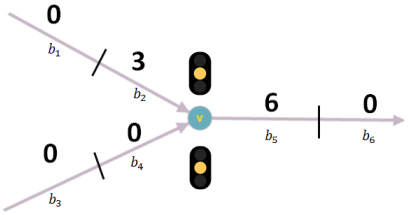
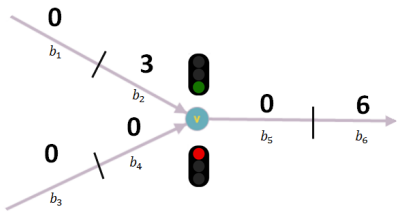
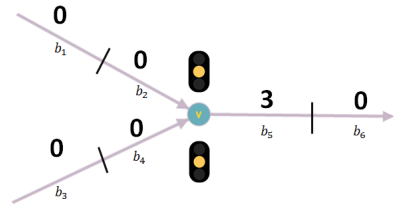
$$\boldsymbol{A_{e_1}} = \begin{array}{c} \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{matrix} \tag{5.15}$$

Macierz stanu w momentach $t = 1, 3, 5$ uwzględnia żółte światło zarówno dla e_1 jak i e_2. Jak zostało już wspomniane, jest ono równoważne czerwonemu światłu.

$$\boldsymbol{A_{yellow}} = \begin{array}{c} \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{matrix} \tag{5.16}$$

Macierz stanu w momencie $t = 2$ uwzględnia zielone światło dla drogi e_2 oraz czerwone dla e_1.

$$\boldsymbol{A_{e_2}} = \begin{array}{c} \begin{matrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array} \begin{matrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \end{matrix} \tag{5.17}$$

t	$\mathbf{x(t)}$	$\mathbf{A(t)}$	Rysunek stanu układu
0	$\begin{bmatrix}3\\5\\4\\2\\7\\1\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\\0&0&0&0&0&0\\0&0&1&1&0&0\\0&1&0&0&0&0\\0&0&0&0&1&0\end{bmatrix}$	
1	$\begin{bmatrix}0\\3\\0\\6\\5\\7\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&0&0\\1&1&0&0&0&0\\0&0&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&0&0\\0&0&0&0&1&0\end{bmatrix}$	
2	$\begin{bmatrix}0\\3\\0\\6\\0\\5\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&0&0\\1&1&0&0&0&0\\0&0&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\end{bmatrix}$	
3	$\begin{bmatrix}0\\3\\0\\0\\6\\0\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&0&0\\1&1&0&0&0&0\\0&0&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&0&0\\0&0&0&0&1&0\end{bmatrix}$	
4	$\begin{bmatrix}0\\3\\0\\0\\0\\6\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&0&0\\1&0&0&0&0&0\\0&0&0&0&0&0\\0&0&1&1&0&0\\0&1&0&0&0&0\\0&0&0&0&1&0\end{bmatrix}$	
5	$\begin{bmatrix}0\\0\\0\\0\\3\\0\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&0&0\\1&1&0&0&0&0\\0&0&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&0&0\\0&0&0&0&1&0\end{bmatrix}$	

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 tym rozdziale zostanie przedstawiony sposób napływania nowych pojazdów do układu. Drogi układu, które nie rozpoczynają się na

skrzyżowaniu będą nazywane drogami zródłowymi. Niech zbiór wszystkich dróg zródłowych to $E_s = e_1, ..., e_k$. Wektorem zródła w chwili t nazywany bedzie wektor:

$$\boldsymbol{u(t)} = \begin{bmatrix} u_1 \\ ... \\ u_k \end{bmatrix}$$

Równanie systemu uwzględniające zródła ruchu to:

$$\mathbf{x(t+1)} = \mathbf{A(t)x(t)} + \mathbf{Bu(t)} \tag{5.18}$$

Gdzie B jest macierzą odpowiedzialną za zrzutowanie pojazdów napływających do dróg zródłowych do odpowiednich odcinków.

5.7.1 Przykład dla pojedynczej drogi

Niech dana będzie droga e podzielona na 4 odcinki b_1, b_2, b_3, b_4. Początkowo niech wektor stanu to

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

Dane są następujące wektory zrodlowe:

$$\mathbf{u(0)} = [7] \quad \mathbf{u(1)} = [3] \quad \mathbf{u(2)} = [5]$$

Są to wektory o wymiarze 1, gdyż w układzie jest tylko 1 droga. Spodziewany następny wektor stanu to:

$$\mathbf{x(1)} = \begin{bmatrix} 7 \\ 2 \\ 4 \\ 3 \end{bmatrix} \tag{5.19}$$

gdyż na pierwszym odcinku pojawia się 7 pojazdów ze zródła $\mathbf{u(0)}$, a pozostałe pojazdy przejeżdżają jeden odcinek drogi. Oczywiście $\mathbf{Ax(0)} = \begin{bmatrix} 0 & 2 & 4 & 3 \end{bmatrix}^T$, zatem koniecznie $\mathbf{Bu(0)} = \begin{bmatrix} 7 & 0 & 0 & 0 \end{bmatrix}^T$ Macierz $\boldsymbol{B}$ zatem musi być dla tego układu następująca:

$$\boldsymbol{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.20}$$

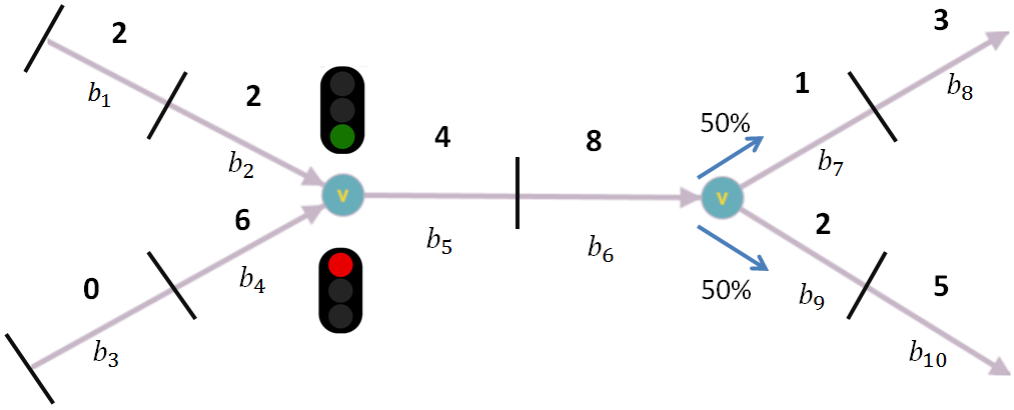
Jest ona stała w czasie. Wartości kolejnych wektorów stanu wyznaczone ze wzoru (5.18) to:

$$\mathbf{x(2)} = \begin{bmatrix} 3 \\ 7 \\ 2 \\ 4 \end{bmatrix}$$

$$\mathbf{x(3)} = \begin{bmatrix} 5 \\ 3 \\ 7 \\ 2 \end{bmatrix}$$

5.7.2 Przykład dla złożonej sieci dróg

Niech dany będzie układ składający się z 5 dróg, z czego każda jest podzielona na dwa odcinki. Strukturę sieci dróg przedstawia poniższy rysunek.



Rysunek 5.5: Złożona sieć dróg

Dalszy rozwój ruchu- Do zrobienia :)

5.8 Wprowadzenie gęstości ruchu

Do zrobienia :)

5.9 Układy symulacyjne

Do zrobienia :)

Rozdział 6

Przedstawienie problemu optymalizacyjnego

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 [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 wiedzy 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:

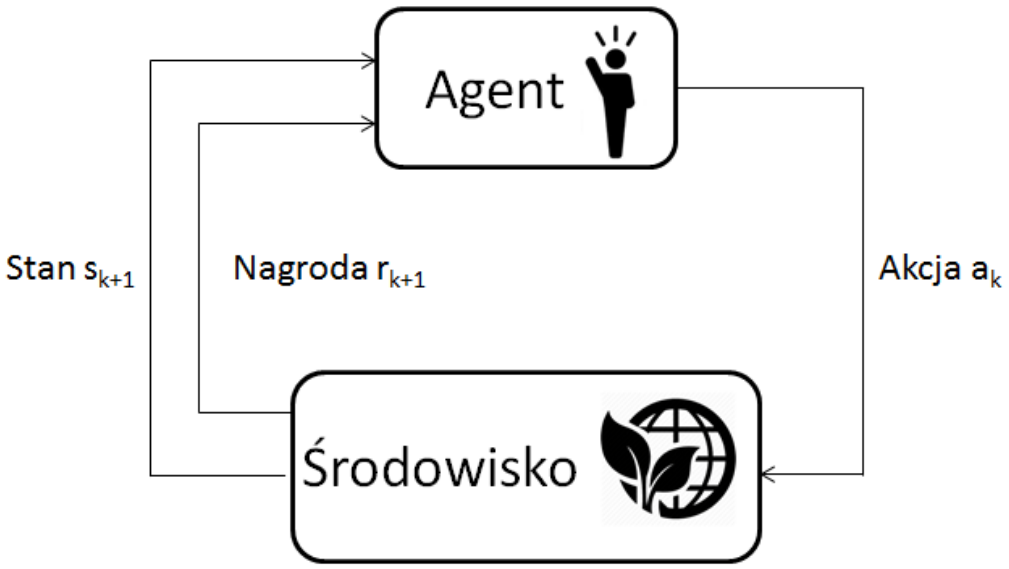
1. **Agent** jest odpowiedzialny za podejmowanie pewnych decyzji. Ma on wiedzę na temat obecnego stanu środowiska i otrzymuje w każdym kroku czasowym sygnał nagrody. Jego decyzje wpływają na stan środowiska.

2. **Środowisko** jest przestrzenią posiadającą dynamiczny stan widoczny dla agenta. Chociaż to 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 akcje, 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. $P_a(s, s') = Pr(s_{t+1} = s' | s_t = s, a_t = a)$ 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'.

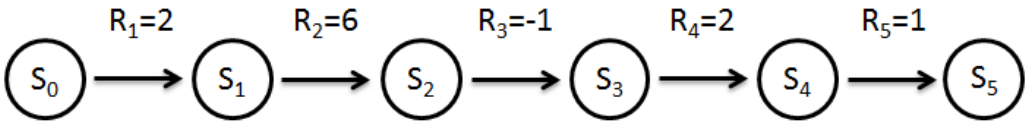
Problemem procesu decyzyjnego Markowa jest odnalezienie optymalnej strategii. Strategia określona jest jako funkcja $\pi(s)$ przyjmująca jako argument stan, a zwracająca podejmowaną akcję. Celem optymalizacji jest odnalezienie strategii maksymalizującej wartość:

$$G = \sum_{k=0}^{K} \gamma^k R_k \tag{7.1}$$

Chociaż strategii $\pi(s)$ nie ma we wzorcze 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 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:

$$G_t = \sum_{k=t}^{K} \gamma^k R_k = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + ... + \gamma^K R_K \tag{7.2}$$

Zostanie przedstawiony teraz przykład obliczeniowy. Agent podejmuje decyzje na których podstawie otrzymuje ciąg nagród $R_0 = 0, R_1 = 2, R_2 = 6, R_3 = -1, R_4 = 2, R_5 = 1$. Czynnik 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

$$G_t = R_t + \gamma G_{t+1} \tag{7.3}$$

.
$G_5 = R_5 = 1.$
$G_4 = R_4 + \gamma G_5 = \ \ 2 + 0,9 \cdot 1 = 2,9$
$G_3 = R_3 + \gamma G_4 = -1 + 0,9 \cdot 2.9 = 1,61$
$G_2 = R_2 + \gamma G_3 = \ \ 6 + 0,9 \cdot 1,61 \approx 7,45$
$G_1 = R_1 + \gamma G_2 = \ \ 2 + 0,9 \cdot 7,45 \approx 8,70$
$G \ \ = R_0 + \gamma G_1 = \ \ 0 + 0,9 \cdot 8,70 \approx 7,83$

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 [14]. 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:

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

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) $\pi(s) = argmax_{a \in A_s} \sum_{s'} Pr(s'|s,a)[r + \gamma V(s')]$
   (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. Są one wyjątkowo użyteczne w problemach o dużej przestrzeni stanów. 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 on-policy jedynie sposobem 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. Monte Carlo podobnie jak algorytm programowania dynamicznego posiada część predykcji i kontroli. Pseudokod cześci predykcji, która jest odpowiedzialna za wyliczenie wartości stanów przy danej strategii π byłby następujący, ale kurwa nie jest bo tego sie nie robi w monte carlo, bo zamiast wartości stanu V(s) sie liczy wartość Q(s,a)!:

1. Przyjmij daną z góry π jako strategię podejmowania akcji
2. Zainicjuj:
   (a) Pusty słownik wartości stanów $V(s)$

   (b) Pusty słownik $Returns(s)$ dla zwracanych wartości stanu podczas symulacji

3. Zasymuluj pełny epizod przy strategii π, który kończy się w chwili T
4. Dla każdego stanu s_t obecnego w przeprowadzonej symulacji począwszy od $t = T$ aż do $t = 0$
   (a) G_t to wartość stanu wyliczona ze wzoru (7.2)
   (b) Dodaj G_t do słownika $Returns(s_t)$
5. $V(s) = avg(Returns(s))$

Jeśli środowisko jest deterministyczne - wystarczy jedna symulacja epizodu.

Część kontroli ma za zadanie polepszyć aktualną strategię π: TODO to jest dla stochastycznego, a ja chce dla deterministycznego kurde od nowa:

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 otrzymanych reward'ów oraz 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. Dla każdego stanu s i akcji a, które pojawiły się w epizodzie
   (a) Wylicz G rekurencyjnie wedle wzoru (7.3)*
   (b) Do tablicy Returns(s,a) dodaj wartość G
   (c) $Q(s, a) = avg(Returns(s, a))$
   (d) $\pi(s) = argmax_{a \in A} Q(s, a)$. Chyba, że wylosowany został krok ekspolarcji, wtedy $\pi(s)$ jest losowo wybraną akcją $a \in A$.

*(Warto rozważyć iterowanie 3. w kolejności odwrotnej do chronologicznej, gdyż ułatwia to wykorzystanie wzoru (7.3))

aaaaaaaaaaaaaaaaaaaa

1. Zainicjuj:
   (a) Pusty słownik $Q(s, a)$ - dla wartości par stan - akcja.
   (b) Pusty słownik $Returns(s, a)$ dla wartości stanu i akcji zwracanych podczas symulacji.
   (c) Dowolny słownik $\pi(a|s)$ - dla
2. W pętli

(a) Przeprowadz symulacje epizodu uzywajac strategii π

(b) Dla kazdej pary s,a pojawiającej sie w epizodzie:

i. Wylicz G ze wzoru 7.2

ii. Dodaj G do wartości $Returns(s, a)$

iii. Wylicz $Q(s, a) = avg(Returns(s, a))$

(c) Dla kazdego s w epizodzie

i. $a^* = argmax_a(Q(s, a))$

ii. Dla wszystkich $a \in A(s)$:

A. $\pi(a|s) = 1 - \epsilon + \epsilon/|A(s)|$

B. $\epsilon/|A(s)|$

Rozdział 8

Optymalizacja sygnalizacji świetlnej

Bibliografia

[1] 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).

[2] polskawliczbach.pl, "Samochody osobowe w polsce w latach 2003-2016 (źródło: Gus)," http://www.polskawliczbach.pl/#transport-i-komunikacja, 2019 (accessed March 3, 2019).

[3] 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).

[4] "Plan działania na rzecz wdrażania inteligentnych systemów transportowych w europie," *Komisja Wspólnot Europejskich, KOM*, vol. 886, 2008.

[5] "Study on urban mobility – assessing and improving the accessibility of urban areas," *Komisja Wspólnot Europejskich, KOM*, vol. 886, 2017.

[6] S. Boubaker, F. Rehimi, and A. Kalboussi, "Comparative analysis of microscopic models of road traffic data," in *Logistics (LOGISTIQUA), 2011 4th International Conference on.* IEEE, 2011, pp. 474–478.

[7] P. Kumar, R. Merzouki, B. Conrard, V. Coelen, and B. O. Bouamama, "Multilevel modeling of the traffic dynamic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1066–1082, 2014.

[8] W. Burghout, H. N. Koutsopoulos, and I. Andreasson, "A discrete-event mesoscopic traffic simulation model for hybrid traffic simulation," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE.* IEEE, 2006, pp. 1102–1107.

[9] M. Ben-Akiva, M. Bierlaire, D. Burton, H. N. Koutsopoulos, and R. Mishalani, "Network state estimation and prediction for real-time traffic management," *Networks and spatial economics*, vol. 1, no. 3-4, pp. 293–318, 2001.

[10] V. A. Vu and G. 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.* IEEE Press, 2017, pp. 127–135.

[11] M. J. Lighthill and G. B. Whitham, "On kinematic waves ii. a theory of traffic flow on long crowded roads," *Proc. R. Soc. Lond. A*, vol. 229, no. 1178, pp. 317–345, 1955.

[12] D. Helbing, A. Hennecke, V. Shvetsov, and M. Treiber, "Master: macroscopic traffic simulation based on a gas-kinetic, non-local traffic model," *Transportation Research Part B: Methodological*, vol. 35, no. 2, pp. 183–211, 2001.

[13] Z. Guan, L. Bian, T. Shang, and J. Liu, "When machine learning meets security issues: A survey," in *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. IEEE, 2018, pp. 158–165.

[14] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE Control Systems Magazine*, vol. 12, no. 2, pp. 19–22, 1992.