

Symulacja ruchu drogowego na przykładzie ronda Grunwaldzkiego w Krakowie

Projekt zespołu 05 na przedmiot
Symulacja Systemów Dyskretnych

Kamień milowy 2 - Model Formalny

Łukasz Łabuz
Dawid Małecki
Mateusz Mazur

19 grudnia 2023

Spis treści

1	Wstęp	1
1.1	Cel projektu	1
1.2	Materiały źródłowe	1
2	Model formalny	2
2.1	Model TSF	2
2.2	Automat komórkowy	4
	Bibliografia	9

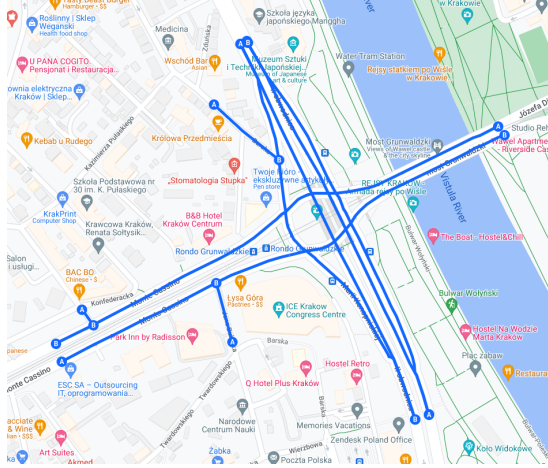
1 Wstęp

1.1 Cel projektu

Celem projektu jest stworzenie modelu symulacyjnego ruchu drogowego na rondzie Grunwaldzkim w Krakowie. Obszar symulacji przedstawia rysunek 1.

1.2 Materiały źródłowe

Materiały [1], [3] stanowią podstawę, na bazie której stworzony został omawiany model formalny.



Rysunek 1: Obszar symulacji. Źródło: Google My Maps

2 Model formalny

2.1 Model TSF

Współcześnie najskuteczniejsze modele symulacji ruchu drogowego opierają się na teorii automatów komórkowych i modelu Nagela-Schreckenberga ([2], [1]). Model ten służy on do symulacji ruchu pojazdów na prostym odcinku drogi. Jego istotnym rozszerzeniem tego modelu jest model TSF ([1]). Umożliwia on przeprowadzanie symulacji ruchu na rzeczywistych sieciach drogowych i rzeczywistych danych. Uwzględnia m.in. takie elementy jak:

- skrzyżowania,
- rozróżnialność kierowców,
- rozróżnialność typów dróg,
- światła drogowe,
- wielopasmowość dróg.

Nasz model bazować będzie na modelu TSF.

2.1.1 Sieć drogowa w modelu TSF

Sieć drogowa w modelu TSF reprezentowana jest jako graf skierowany $G = (V, E)$, gdzie V to zbiór wierzchołków, a $E \subseteq V \times V$ to zbiór krawędzi. Krawędzie reprezentują odcinki dróg, posiadają określone atrybuty:

- d - długość,
- $lanes$ - liczba pasów ruchu,
- v_{avg} - średnią maksymalną prędkość jazdy,
- v_{std} odchylenie standardowe wartości v_{avg} ,
- $type$ - typ drogi.

W obrębie danej krawędzi pasy ruchu są numerowane liczbami naturalnymi. W obrębie pojedynczego pasa, komórki również są numerowane. Komórkę jednoznacznie wyznacza więc trójka: *edge*, *lane*, *cell*. Wszystkie komórki na tej samej krawędzi mają jednakowość długość i wszystkie pasy ruchu na tej krawędzi są podzielone na tyle samo komórek o takiej samej długości.

2.1.2 Sygnalizacja świetlna

W modelu TSF sygnalizacja świetlna jest zlokalizowana w wierzchołkach grafu. Każdą sygnalizację charakteryzują następujące parametry:

- *id* - unikalny identyfikator,
- *position* - lokalizacja (krawędź, na końcu której znajduje się sygnalizacja),
- *t_{green}* - czas trwania fazy zielonej,
- *t_{red}* - czas trwania fazy czerwonej,
- *\$t_{change}* - czas trwania zmiany fazy,
- *state* - aktualny stan (*GREEN/RED*).

Wartość *t_{change}* jest nieujemna i zmniejsza się o 1 w każdym kroku symulacji aż do osiągnięcia wartości 0. Wtedy następuje zmiana stanu sygnalizacji na przeciwny i *t_{change}* przyjmuje wartość *t_{green}* lub *t_{red}* w zależności od stanu, do którego zmieniła sygnalizacja.

Zbiór wszystkich sygnalizacji w modelu oznaczamy jako *SIGNALS*.

2.1.3 Pojazdy

W modelu TSF pojazdy mają następujące atrybuty:

- *id* - unikalny identyfikator,
- *edge* - krawędź, na której znajduje się pojazd,
- *distance* - odległość od początku krawędzi,
- *id_{lane}* - pas ruchu, na którym znajduje się pojazd,
- *id_{cell}* - komórka, w której znajduje się pojazd,
- *profile* - parametry ruchu pojazdu (wartość z przedziału $[0, 1]$),
- *velocity* - aktualna prędkość pojazdu,
- *path* - ścieżka, którą pokonuje pojazd,
- *start* - czas rozpoczęcia symulacji pojazdu.

Zbiór wszystkich pojazdów w modelu oznaczamy jako *CARS*.

2.1.4 Piesi

Dodani przez nas do modelu piesi mają następujące atrybuty:

- *id* - unikalny identyfikator,
- *edge* - krawędź, na której znajduje się pieszy,
- *distance* - odległość od początku krawędzi,
- *id_{cell}* - komórka, w której znajduje się pieszy,

- r_1 - prawdopodobieństwo, że pieszy przestrzega prawa,
- r_2 - prawdopodobieństwo, że pieszy zatrzyma się na czerwonym świetle,
- t_{walk} - czas, przez który pieszy uzna za bezpieczny, aby przejść na drugą stronę ulicy,
- $velocity$ - aktualna prędkość pieszego,
- $path$ - ścieżka, którą pokonuje pieszy,
- $start$ - czas rozpoczęcia symulacji pieszego.

Zbiór wszystkich pieszych w modelu oznaczamy jako *PEDESTRIANS*.

2.2 Automat komórkowy

Automat komórkowy w naszym modelu to krotka:

$$CA = \langle T, C, N, S, S_0, F \rangle$$

gdzie:

- T - przedział czasu, w którym odbywa się ewolucja automatu,
- C - zbiór komórek,
- $N : C \rightarrow 2^C$ - funkcja, która każdej komórce przyporządkowuje zbiór jej otoczenie,
- S - zbiór możliwych stanów komórek,
- $S_0 : C \rightarrow S$ - funkcja, która każdej komórce przyporządkowuje jej stan początkowy,
- $F : T \times C \rightarrow S$ - reguła przejścia - funkcja, która każdej komórce przyporządkowuje jej stan po ewolucji.

2.2.1 Zbiór komórek

W naszym modelu komórki są reprezentowane przez pojazdy, pieszych oraz sygnalizacje świetlne. Dla dalszego ułatwienia, zastosujemy oznaczenie $CELLS = CARS \cup PEDESTRIANS$. Wtedy:

$$C = CELLS \cup SIGNALS$$

2.2.2 Funkcja otoczenia

O ruchu pojazdów oraz pieszych decyduje przede wszystkim ich otoczenie, ale również mogą to być sygnały globalne, pochodzące np. z nawigacji.

Ponieważ stan sygnalizacji w chwili $t + 1$ zależy nie tylko od jej stanu w chwili t , ale również od stanu innych sygnalizacji w chwili t , ponieważ sygnalizacja może być ustawiane globalnie, w celu optymalizacji ruchu. Zatem:

$$\forall_{c \in C} N(c) = C \cup SIGNALS$$

2.2.3 Zbiór stanów

Komórki znajdujące się na krawędziach grafu mogą być zajmowane przez pojazdy, pieszych, lub być puste. Jeżeli w komórce znajduje się pojazd lub pieszy, to jest on jej stanem. Jeżeli komórka jest pusta, to jej stanem jest *null*. W przypadku sygnalizacji, stanem komórki jest fasa sygnalizacji (*GREEN/RED*). Zatem:

$$S = \{null\} \cup CARS \cup PEDESTRIANS \cup \{GREEN, RED\}$$

2.2.4 Funkcja stanu początkowego

Stan początkowy komórki jest zależny od tego, czy jest to sygnalizacja, czy komórka na krawędzi. W przypadku sygnalizacji, stan początkowy jest dowolnym elementem ze zbioru $\{GREEN, RED\}$. Konfiguracja początkowa komórek ze zbioru CELLS to rozmieszczenie pojazdów i pieszych, którzy rozpoczynają ruch wraz z początkiem symulacji na odpowiednich komórkach CELLS.

2.2.5 Funkcja przejścia

Funkcja przejścia jest najbardziej złożonym elementem automatu komórkowego.

Dla komórek ze zbioru SIGNALS regułę przejścia definiuje algorytm przedstawiony na rysunku 2.

Algorithm 1 Algorytm przejścia dla sygnalizacji świetlnej s w kroku t

Require: $s \in SIGNALS(G), t \in T$

Ensure: $state(s, t + 1) \in \{GREEN, RED\}$

```

if  $t_{change}(s) > 0$  then
     $t_{change}(s) := t_{change}(s) - 1$ 
     $state(s, t + 1) := state(s, t)$ 
    return  $state(s, t + 1)$ 
else
    if  $state(s, t) = RED$  then
         $t_{change}(s) := t_{green}(s)$ 
         $state(s, t + 1) := GREEN$ 
    else
         $t_{change}(s) := t_{red}(s)$ 
         $state(s, t + 1) := RED$ 
    end if
    return  $state(s, t + 1)$ 
end if

```

Rysunek 2: Algorytm opisujący funkcję przejścia dla komórek ze zbioru *SIGNALS*. Źródło: [1]

Dla komórek ze zbioru CARS regułę przejścia definiuje algorytm przedstawiony na rysunku 3. W tej funkcji wykorzystywane są następujące parametry pomocnicze:

- $turnPenalty \in [0, 1]$ - zachowanie pojazdu przed skrętem,
- $crossroadPenalty \in [0, 1]$ - zachowanie pojazdu w trakcie przejazdu przez skrzyżowanie (bez skręcania),
- $prob \in [0, 1]$ - losowa redukcja prędkości pojazdu.

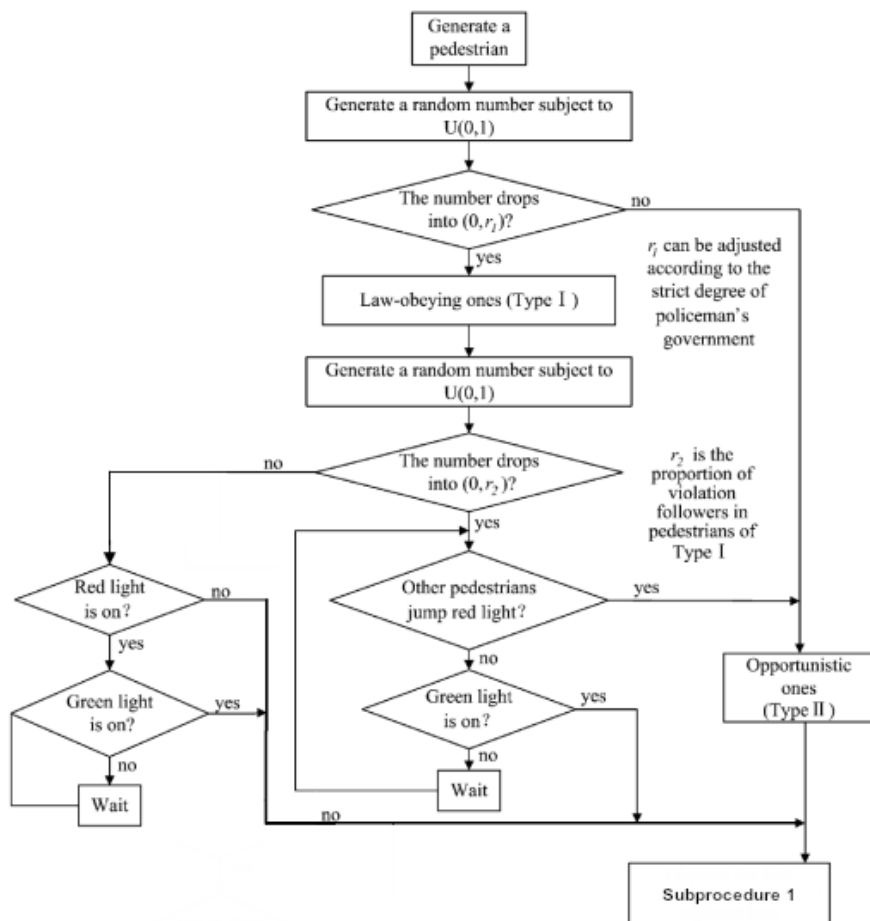
Algorithm 2 Algorytm ruchu pojazdu car w kroku t

Require: $G = (V, E)$, $t \in T$, $car \in CARS(t)$, $turnPenalty$, $crossroadPenalty$, $prob$
increaseVelocity(car, t);
if *stopOnSignal(car, t)* **then**
 reduceVelocityOnSignal(car, t)
else
 if *turnOnCrossroad(car, t)* **then**
 reduceVelocity(car, t, turnParameter);
 else
 if *crossroad(car, t)* **then**
 reduceVelocity(car, t, crossroadParameter);
 end if
 end if
end if
if *shouldChangeLane(car, t)* **then**
 changeLane(car, t);
end if
 safeReduceVelocity(car, t)
 with probability $prob$: *reduceVelocity(car, t);*
 makeMove(car, t);

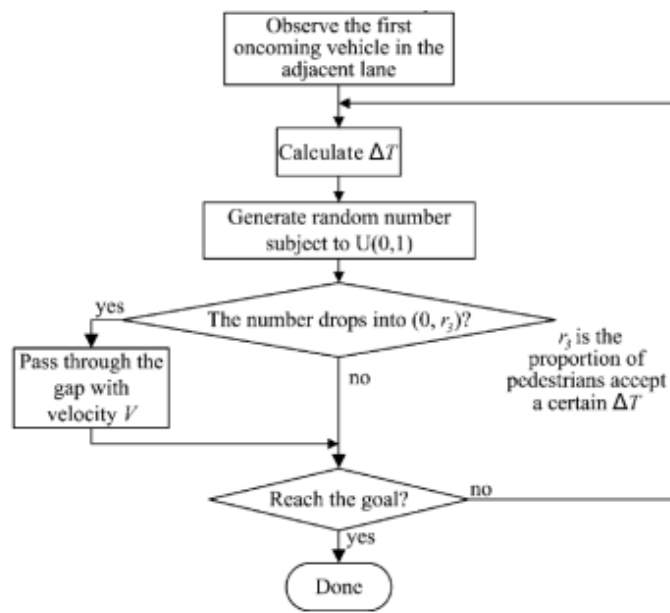
Rysunek 3: Algorytm opisujący funkcję przejścia dla komórek ze zbioru $CARS$.
Źródło: [1]

Dla komórek ze zbioru PEDESTRIANS regułę przejścia definiuje głównie macierz prawdopodobieństw P oraz, w przypadku przechodzenia przez przejście dla pieszych, fragment algorytmu przedstawionego na rysunkach 4, 5.

P jest macierzą o wymiarach 3×3 , która tworzona jest na bazie najbliższego otoczenia pieszego. Pieszy może zmienić pozycję na jedną z 8 sąsiednich komórek. W zależności od tego, czy pieszy chce skręcić, wyprzedzić innego pieszego, czy dalej kontynuować ruch prosto, prawdopodobieństwo zajęcia danej komórki jest różne.



Rysunek 4: Algorytm opisujący część funkcji przejścia dla komórek ze zbioru *PEDESTRIANS*. Źródło: [3]



Rysunek 5: Algorytm opisujący część funkcji przejścia dla komórek ze zbioru *PEDESTRIANS* - podproces 1. Źródło: [3]

Bibliografia

- [1] Gora, P. 2010. *Adaptacyjne planowanie ruchu drogowego*. Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki.
- [2] Nagel, K. and Schreckenberg, M. 1992. A cellular automaton model for freeway traffic. *Journal de physique I*. 2, 12 (1992), 2221–2229.
- [3] Rasouli, A. 2021. Pedestrian simulation: A review. *arXiv preprint arXiv:2102.03289*. (2021).