

Symulacja II obwodnicy Krakowa

Miłosz Wrzesień, Krzysztof Sawina, Paweł Pytlowski

Czerwiec 2020

1 Wprowadzenie do tematu

1.1 Opis problemu

W dzisiejszych czasach w centrach dużych miast, zwłaszcza w tzw. godzinach szczytu, duży problem stanowią korki uliczne i zbytnie zagęszczenie ruchu na drogach.

Powód tej materii jest wiele i można je podzielić na trudne lub niemożliwe do zmiany oraz takie, które mogą być eliminowane przez właściwe planowanie.

Do pierwszej grupy należą: wąskie ulice, nieodpowiednia ilość pasów ruchu czy położenie najczęstszych obiektów będących głównymi celami podróży.

Natomiast niektóre z czynników, które można poprawić to: umiejętności kierowców (unikanie wypadków i upłygnienie ruchu), czy efektywna organizacja ruchu, w tym działanie sygnalizacji świetlnej z odpowiednim stałym lub zmiennym czasem oczekiwania.

1.2 Wstępny opis projektu

Projekt obejmuje zagadnienie ruchu drogowego na II obwodnicy Krakowa, na której kierowcy codziennie zmagają się z wymienionymi trudnościami. Obwodnica ta ma ponad 12 kilometrów długości i składają się na nie skrzyżowania z ulicami [1]:

- ul. M. Konopnickiej,
- Aleja Z. Krasińskiego,
- Aleja A. Mickiewicza,
- Aleja J. Słowackiego,
- Aleja 29 Listopada,
- ul. W. Stwosza,
- ul. A. Lubomirskiego,
- Rondo Mogilskie,
- Aleja Powstania Warszawskiego,
- Rondo Grzegórzeckie,
- ul. Kotlarska,
- ul. G. Herlinga-Grudzińskiego,
- ul. S. Klimeckiego,
- al. Powstańców Wielkopolskich,

- Aleja Powstańców Śląskich,
- ul. H. Kamieńskiego,
- Rondo Matecznego.

Celem projektu jest stworzenie symulacji ruchu pojazdów na II obwodnicy Krakowa oraz dojazdów do jej skrzyżowań oraz prostego interfejsu, który klarownie w czasie rzeczywistym przedstawiałby statystyki poruszających się pojazdów, a ponadto dałby użytkownikowi możliwość bezpośredniego wpływu na wyniki symulacji przez manipulację zmiennymi wejściowymi.

Przeglądając potencjalne możliwości rozwiązania problemu, czyli gotowe symulacje poświęcone ruchowi drogowemu, znaleźliśmy między innymi rozbudowaną stronę z generycznymi typami skrzyżowań i sieci dróg, traffic-simulation.de, czy też projekt symulacji skrzyżowania ul. Czarnowiejskiej z Aleją Adama Mickiewicza, którym zainspirowaliśmy się w naszej implementacji [2, 3].

2 Przegląd literatury oraz dostępnych narzędzi

2.1 Literatura

W niniejszym rozdziale zostanie dokonany przegląd literatury tematu oraz przedstawiony zostanie opis wykorzystanych narzędzi.

2.1.1 Model Nagela-Schreckenberga

W projekcie korzystamy z modelu Nagela-Schreckenberga. Jest on wykorzystywany przede wszystkim do symulacji ruchu na drogach ekspresowych i autostradach, jednak można go używać do symulacji ruchu miejskiego. Model został opracowany na początku lat 90. XX wieku przez niemieckich fizyków Kai Nagela i Michaela Schreckenberga. Bazuje on na prostym wykorzystaniu automatu komórkowego. Modelowanie dynamiki przepływu ruchu za pomocą automatu pozwala nam przeprowadzać duże symulacje ruchu sieciowego z przy stosunkowo niskiej złożonością obliczeniową. Projektując mechanikę modelu w aspektach takich jak wytyczanie trasy pojazdów jest oczywiste, że należy je rozpatrywać w całej sieci. Nawet lokalne zmiany w regulacjach pierwszeństwa na pojedynczych skrzyżowaniach muszą zostać zbadane na dużą skalę, aby uniknąć niepożądanych efektów globalnych z powodu wzajemnych korelacji. Testowanie strategii kontroli w rzeczywistości jest zazwyczaj niewykonalne lub przynajmniej niezwykle wymagające jeśli chodzi o czas i zużycie. Dlatego pożąданie jest użycie uogólnionych rozwiązań sprowadzających się do środowiska laboratoryjnego.

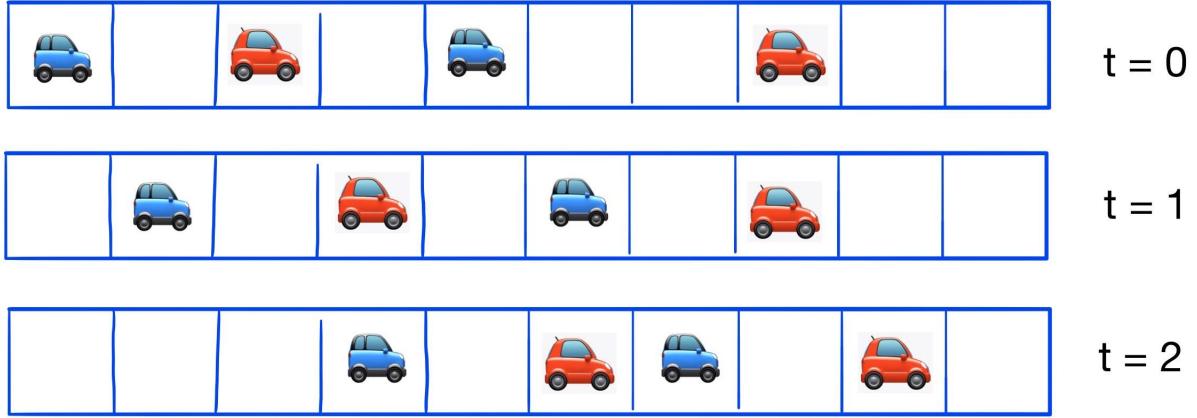
Projektowanie narzędzia do symulacji ruchu sieciowego wymaga ogólnego kompromisu: Ogranicza się lub rezygnuje z niektórych danych dotyczących poszczególnych pojazdów, takich jak czas podróży lub liczba postojów, aby ułatwić ich gromadzenie. Jest to szczególnie ważne gdy chcemy symulować ruch w czasie rzeczywistym [4, 5].

2.1.2 Model automatu komórkowego

Model 2 Nagela – Schreckenberga został pierwotnie zdefiniowany na drodze jednopasmowej. Droga jest podzielona na komórki, które mogą być puste lub zajmowane przez jeden pojazd. W wymienionym modelu przyjęto rozmiar komórki równy $d = 7.5\text{m}$. Każdy pojazd ma nieujemną prędkość opisywaną poprzez liczbę komórek pokonanych w określonej chwili czasu. Model Na-Sch opisują następujące reguły [12, 13]:

- Przyspieszenie: $v(t+1) \rightarrow \min(v(t) + 1, v_{max})$, gdzie $v(t)$ to prędkość aktualna,
- Hamowanie: $v(t+1) \rightarrow \min(v(t), g(t)-1)$, gdzie $g(t)$ jest liczbą pustych komórek pomiędzy,

- Element losowy (losowe hamowanie): prawdopodobieństwo p , że zajdzie $v(t+1) \rightarrow \max(v(t)-1)$, jeżeli $v(t) \geq 1$,
- Ruch (zmiana położenie w czasie): $x(t+1) = x(t) + v(t)$.



Rys. 1. Ruch w modelu Nagela Schreckenberga na pasie ruchu w kolejnych chwilach czasowych

2.1.3 Automaty komórkowe

AK weszły do bibliotek fizyków na początku lat 80-tych. Jednym z głównych popularyzatorów tej idei był Stephen Wolfram, znany jako twórca pakietu Mathematica. Próbował on również sklasyfikować AK [4].

Deterministyczny automat komórkowy jest pojęciem matematycznym definiowanym przez:

- sieć komórek i D-wymiarowej przestrzeni,
- zbiór S_i stanów pojedynczej komórki — najczęściej ten sam dla wszystkich komórek i n-elementowy,
- regułę F określającą stan komórki w chwili $t+1$ w zależności od stanu w chwili t tej komórki i komórek ją otaczających: $s_i(t+1) = F(s_j(t))$, gdzie $j \in O(i)$ i $O(i)$ jest otoczeniem i-tej komórki,

Najbardziej zbadaną klasą automatów są jednowymiarowe automaty deterministyczne ($D = 1$), o dwóch stanach komórki ($n = 2$) i otoczeniu składającym się z najbliższych sąsiadów ($r = 1$) - zgodnie z najprostszą notacją te automaty oznaczamy jako $(2, 1)$. Argumentem funkcji F jest stan $2r+1 = 3$ komórek, każda z nich może być w $n = 2$ stanach — funkcja F jest więc określona dla $2^3 = 8$ różnych konfiguracji. Dla każdej z nich funkcja F może przyjmować $k = 2$ wartości. Można ją więc opisać na $2^8 = 256$ sposobów. Za Wolframem te 256 jednowymiarowych AK nazywamy elementarnymi.

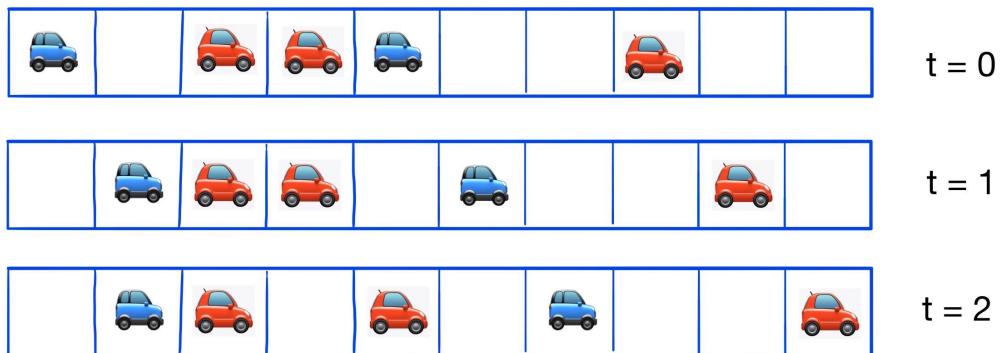
Najprostszy automat komórkowy, który symuluje ruch pojazdów na jednokierunkowej i jednopasmowej drodze należy do rodziny elementarnych automatów zdefiniowanych przez Wolframa i odpowiada regule numer 184. Droga, po której poruszają się cząstki (samochody) przedstawiona jest przez łańcuch komórek. Każda z nich może przyjmować jeden z dwóch stanów: pusta lub zajęta. Wszystkie obiekty poruszają się w jedną stronę zgodnie z następującymi zasadami:

- każdym kroku cząstka może zostać przesunięta o jedno miejsce lub pozostać na poprzedniej pozycji,
- przesunięcie następuje wtedy, gdy komórka docelowa jest pusta,
- wszystkie pola są uaktualniane jednocześnie,

Zakładając, że ruch odbywa się w prawą stronę, można napisać:

$$S_i(t+1) = S_{i-1}(t)[1 - S_i(t)] + S_i(t) \cdot S_{i+1}(t)$$

gdzie: $S_i(t)$ — stan i-tej komórki w chwili t : wartość 0 odpowiada pustej komórce, 1 — zajętej.



Rys. 2. Ilustracja przepływu opisanego elementarnym automatem Wolframa

2.2 Narzędzia

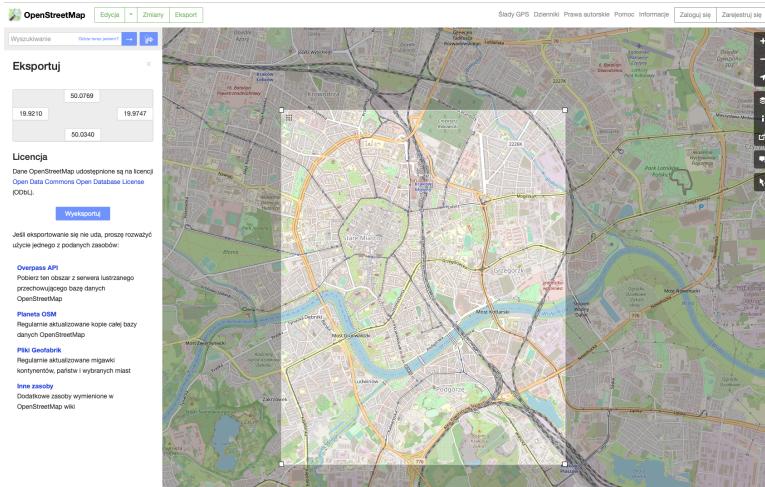
Do wykonania projektu wykorzystane zostały języki: Python, HTML, CSS, JavaScript oraz R. Python został użyty do wygenerowania z pliku .KML komórek niezbędnych do zwizualizowania symulacji oraz do znalezienia sąsiadów poszczególnych komórek. HTML i CSS wykorzystane zostały do wyświetlenia symulacji na stronie. W JavaScript przygotowany został kod do symulowania i wizualizowania ruchu samochodów a w R kod przedstawiający na wykresach uzyskane dane z symulacji.

2.2.1 OpenStreetMap

OpenStreetMap [6] jest to narzędzie tworzone i aktualizowane przez społeczność dostarczające dane dotyczące dróg, ścieżek oraz miejsc takich jak dworce, restauracje. Korzystając z danych z tego serwisu, jesteśmy w stanie dokładnie odwzorować wygląd ulic, jeśli chodzi o rozmieszczenie i kierunki pasów na jezdni, ograniczenia prędkości, zjazdy oraz wjazdy na drogę główną. Zaangażowanie społeczności posiadającej wiedzę lokalną, początkujących kartografów i specjalistów od tworzenia map daje pewność, że dane z tej strony są aktualne i dokładne. Niestety nie ma możliwości pobrania obszaru w odpowiednim formacie, lecz z pomocą przychodzą nam inne narzędzia.

2.2.2 MyGeodata Cloud

MyGeodata Cloud [7] to darmowe narzędzie online obsługujące m.in. formaty GIS i CAD oraz układy współrzędnych. Narzędzie to pozwala na pobrać obszar z OpenStreetMap.



Rys. 3. Zrzut ekranu strony OpenStreetMap

Rys. 4. Zrzut ekranu strony MyGeodataCloud

```

<nd ref="-104201"/>
<tag k="color" v="#ff0000"/>
<tag k="highway" v="residential"/>
<tag k="maxspeed:type" v="Pt:urban"/>
<tag k="name" v="Rogatka"/>
<tag k="surface" v="asphalt"/>
</way>
<way id="-102249">
<nd ref="-105235"/>
<nd ref="-105239"/>
<tag k="color" v="#ff0000"/>
<tag k="cycleway" v="no"/>
<tag k="foot" v="yes"/>
<tag k="highway" v="secondary"/>
<tag k="lanes" v="3"/>
<tag k="lit" v="yes"/>
<tag k="maxspeed" v="50"/>
<tag k="maxspeed:conditional" v="60 @ (23:00-05:00)"/>
<tag k="name" v="Czarnowiejska"/>
<tag k="sidewalk" v="separate"/>
<tag k="surface" v="asphalt"/>
</way>
<way id="-102250">
<nd ref="-105240"/>
<nd ref="-105241"/>
<tag k="color" v="#ff0000"/>
<tag k="cycleway" v="no"/>
<tag k="dual_carriageway" v="yes"/>
<tag k="foot" v="no"/>

```

Rys. 5. Zrzut ekranu pliku .OSM

2.2.3 Osmium

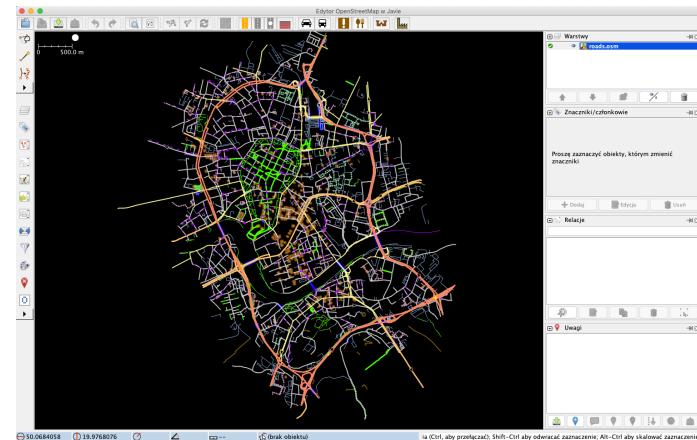
Osmium [8] to narzędzie do pracy z danymi OSM. Pozwala na szybką konwersję formatu plików OSM oraz na wyodrębnianie obszarów geograficznych z pliku.

2.2.4 Mapbox

Mapbox [9] oferuje użytkownikowi możliwość tworzenia i korzystania z niestandardowych map. Obsługuje serwisy takie jak Facebook, Snapchat czy Foursquare. Map box ma wkład w tworzenie aplikacji i bibliotek open source m.in. bibliotekę Leaflet JavaScript, MBTiles i bibliotekę Mapbox GL-JS JavaScript.

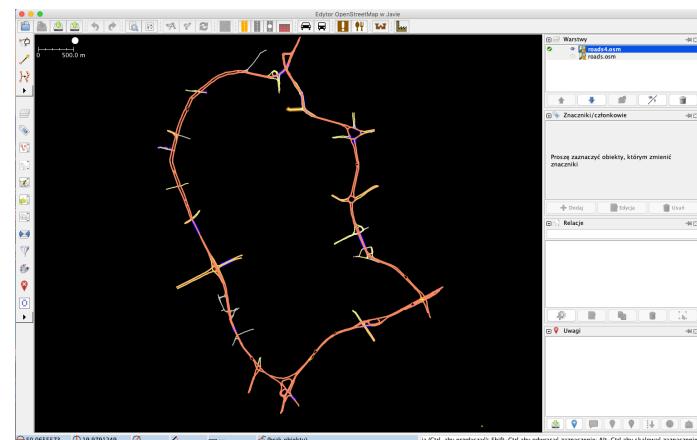
2.2.5 JOSM

JOSM (Java OpenStreetMap Editor) jest narzędziem do edycji danych z OpenStreetMap stworzonych w javie. Jego twórcą jest Immanuel Scholz, lecz obecnie jego utrzymaniem zajmuje się Dirk Stöcker. Jest bardziej zaawansowany i posiada więcej funkcji niż przeciętny edytor iD. Oprogramowanie JOSM zapewnia m.in. import plików GPX, monitorowanie poprawności danych oraz obsługę warstw i relacji. Dostępne są również różnego rodzaju wtyczki (repozytorium zawiera ponad 100) służące między innymi do obsługi technologii 3D.



Rys. 6. Zrzut ekranu programu JOSM

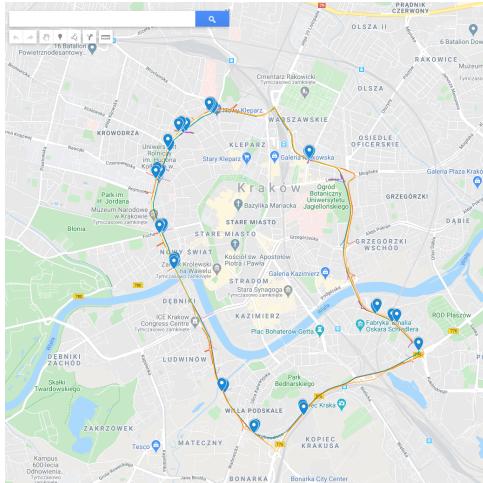
Mapa ta jednak nie jest optymalna dla symulacji dlatego też używamy narzędzia Osmium, żeby ją dostosować.



Rys. 7. Zrzut ekranu programu JOSM po użyciu Osmium

2.2.6 Google Mymaps

Google MyMaps [10] służy do tworzenia własnych map poprzez import danych (np. z arkuszy kalkulacyjnych) z możliwością nanoszenia punktów oraz rysowania kształtów. Aplikacja umożliwia wyszukiwanie i zapisywanie na mapie miejsc, a także personalizację graficzną obejmującą dodawanie zdjęć i filmów w danych miejscach jak i edycję kolorów i ikon. Zapewnia ona również łatwość udostępniania map utworzonych przez użytkownika.



Rys. 8. Zrzut ekranu strony Mymaps

```
<Placemark>
  <name>B003_00</name>
  <styleUrl>#line-0F9D58-1200-nodesc</styleUrl>
  <LineString>
    <tessellate>1</tessellate>
    <coordinates>
      19.9607911,50.042128,0
      19.9599395,50.0417507,0
      19.9594255,50.0415638,0
      19.9588981,50.0414096,0
      19.9583826,50.0413043,0
      19.9574871,50.0411432,0
      19.9565904,50.0409476,0
      19.9558512,50.0407356,0
      19.9551522,50.0404823,0
      19.9545237,50.0402018,0
      19.9540383,50.0399564,0
      19.9535891,50.0396924,0
      19.953164,50.039413,0
      19.9483939,50.0358316,0
    </coordinates>
  </LineString>
</Placemark>
```

Rys. 9. Zrzut ekranu pliku .KML

2.2.7 Leaflet

Leaflet [11] to biblioteka graficzna typu open-source do JavaScript, dzięki której do swoich projektów z łatwością można implementować elementy rzeczywistych map z całego świata. Dodatkowo pozwala tworzyć niestandardowe znaczniki powiązane z konkretnymi punktami geograficznymi, czy też renderować obiekty o kształcie prostych brył geometrycznych oraz grupować je w warstwy i umieszczać je na konkretnym miejscu na mapie. LeafletMarkerCluster ogranicza wyświetlanie się punktów na mapie usuwając te które nie są w kadrze.

3 Model zjawiska

W rozdziale tym opisano niezbędne informacje dotyczące przygotowanego modelu.

3.1 Wejścia do modelu

Model przewiduje bazowanie na zmiennych, które mają kluczowy wpływ na efektywność przepływu pojazdów. Wejścia do symulacji ustawiane są przez użytkownika i stanowią one:

- liczbę samochodów na całej mapie
- czas trwania cyklu sygnalizacji świetlnej
- liczba samochodów wjeżdżających na obwodnicę z dróg podzielonych na minutę

3.2 Wyjścia z modelu

Wyniki symulacji w czasie rzeczywistym obrazowane są w postaci następujących danych:

- aktualna liczba samochodów na obwodnicy

- liczba samochodów, które wjechały z dróg podporządkowanych
- liczba samochodów, które w trakcie symulacji zjechały z obwodnicy
- aktualna średnia prędkość wszystkich pojazdów znajdujących się na symulowanej mapie
- średnia z maksymalnych możliwych prędkości wszystkich pojazdów znajdujących się na symulowanej mapie
- przepustowość pojazdów na minutę na każdym skrzyżowaniu

3.3 Założenia algorytmu

Symulacja ma opierać się na poniższych założeniach:

- każdy pojazd ma generowane losowo przyspieszenie i maksymalną prędkość, aby odtworzyć różnice możliwości technicznych rzeczywistych pojazdów oraz umiejętności i stylów prowadzenia samochodów przez kierowców
- każdy pojazd zajmuje dokładnie jedno pole, które reprezentuje powierzchnię o długości 3 metrów, implementacja pojazdów dłuższych (autobusów czy ciężarówek) zbytnio skomplikowałaby logikę poruszania się, jednocześnie wprowadzając jedynie marginalne odchyły w wynikach symulacji
- każdy pojazd w danej iteracji symulacji porusza się do przodu z zadaną aktualną prędkością lub przyspieszeniem/opóźnieniem dostosowanym do sytuacji przed nim, zgodnie z założeniami modelu Nagela-Schreckenberga
- każdy pojazd zbliżający się do możliwego zjazdu o obwodnicy losowo decyduje o zjeździe lub pozostaniu na obwodnicy
- sygnalizacja świetlna na każdym skrzyżowaniu jest uproszczona do 4 sygnalizatorów prostych S-1, po jednym dla każdego kierunku jazdy na skrzyżowaniu; pominięte są tzw. strzałki warunkowe S-2 i sygnalizacja kierunkowa S-3
- pominięte są przejścia dla pieszych, ponieważ większość z nich jest zintegrowana z sygnalizacją świetlną na skrzyżowaniach

4 Implementacja symulacji zjawiska

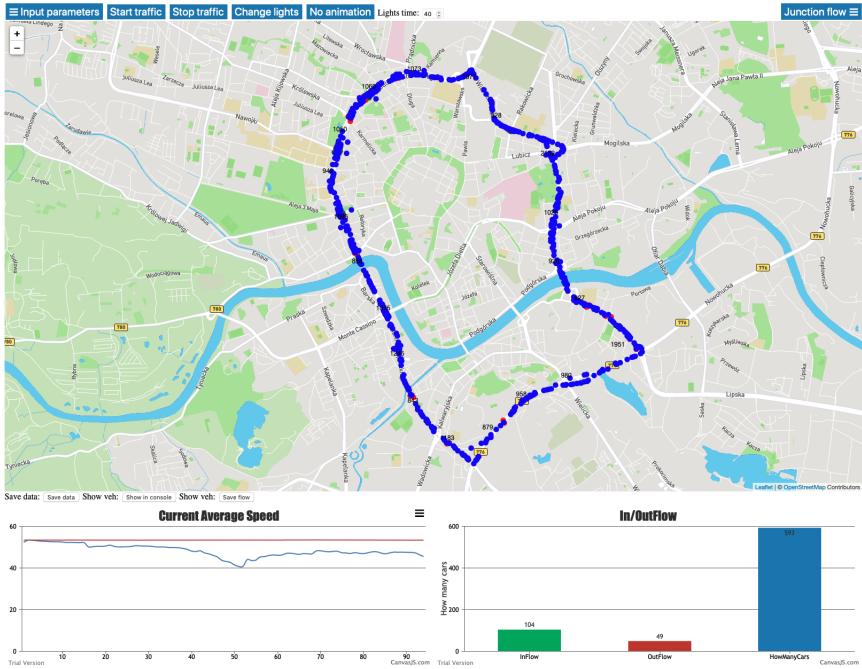
Symulacja została napisana w JavaScript, ponieważ jest on dobrze znany całemu zespołowi. Mogliśmy również łatwo zastosować bibliotekę Leaflet, dedykowaną temu językowi, która stanowi podstawę prezentacji wyników modelu.

Komórki potrzebne do symulacji zjawiska zostały wygenerowane na podstawie pobranych danych. Dane nie były dokładne, więc zostały poprawione w Google MyMaps. Następnie z wyeksportowanego pliku .KML w Pythonie wygenerowano potrzebne komórki.

4.1 Opis części wizualnej

Głównym elementem ekranu jest mapa generowana z biblioteki Leaflet, oraz wyrysowane na niej interesujące nas punkty. Każdy z nich, w postaci okręgu, ma średnicę długości 3 metrów i składa się na całą sieć wielopasmowych dróg i skrzyżowań. Poniżej znajdują się wykresy prezentujące dane wyjściowe symulacji, obliczane w czasie rzeczywistym, opisane w punkcie 3.3.

Ruch pojazdów animowany jest poprzez renderowanie punktów w odpowiednim kolorze. Niebieski punkt oznacza pojazd, natomiast biały to puste pole. Dla znacznej optymalizacji symulacji, występowanie pojazdów renderowane jest jedynie na obszarze aktualnie wyświetlanym, więc dla zwiększenia płynności animacji zalecana jest obserwacja na jak największym przybliżeniu.



Rys. 10. Zrzut ekranu z działającej aplikacji.

4.2 Opis back-endu

Pojazdy pogrupowane są w warstwach według aktualnej prędkości poruszania, ich ruch obliczany jest cyklicznie co odpowiedni czas delayu dla danej prędkości, dzięki zastosowaniu javascriptowej funkcji setTimeout. Dla każdego pojazdu algorytm sprawdza 8 punktów do przodu (24 metry) w poszukiwaniu wolniej jadących pojazdów, czerwonych świateł czy też możliwości zjazdy z obwodnicy i decyduje o potencjalnym zmianie pasa lub zmianie prędkości.

4.3 Struktury danych

Elementy symulacji mają swoje dane zorganizowane w formie słowników.

4.3.1 Każdy punkt na mapie powiązany jest z:

- swoim indeksem w słowniku 'neigh'
- lt - latitude, szerokość geograficzna punktu
- ln - longitude, długość geograficzna punktu
- s - jeśli punkt jest początkiem drogi
- e - jeśli punkt jest zakończeniem drogi
- nl - indeks punktu po lewej lub pusty string
- nf - indeks następującego punktu lub pusty string
- nr - indeks punktu po prawej lub pusty string
- tl - identyfikator świateł powiązanych z punktem lub pusty string
- v - identyfikator pojazdu w punkcie lub pusty string

4.3.2 Każdy pojazd reprezentowany jest przez:

- po - indeks punktu, na którym się znajduje
- s - aktualna prędkość
- d - obecność/brak obecności na obwodnicy
- ms - maksymalna prędkość (40, 50 lub 60)
- cl - liczba cykli symulacji od ostatniej zmiany pasa
- l - liczba cykli symulacji od wjazdu na obwodnicę

4.3.3 Każdy element sygnalizacji świetlnej ma:

- lt - latitude, szerokość geograficzna punktu
- ln - longitude, długość geograficzna punktu
- id - identyfikator sygnalizatora
- st - aktualny stan (zielone/czerwone)

4.4 Szczegółowy opis funkcjonalności

Oprócz tego do dyspozycji użytkownika jest panel do konfiguracji symulacji, na który składają się następujące opcje:

- Input parameters - możliwość ustawienia liczby samochodów na każdej drodze wjazdowej (Input), znajdujących się już na obwodnicy (Cars), oraz dostosowania inputu indywidualnie dla każdej ulicy podzędnej osobno
- Add points - rysuje wszystkie punkty, po których poruszają się pojazdy
- Start traffic - uruchomienie symulacji
- Change lights - natychmiastowa zmiana światła
- Lights time - użytkownik może ustalić czas zielonego światła na głównych ulicach (w sekundach). Na cykl sygnalizacji składa się również 3-krotnie krótszy czas zielonego światła dla dróg podzędnych
- No animation - możliwość wyłączenia animacji i przeprowadzania samej symulacji w tle, dla odciążenia sprzętu lub przy braku płynności
- Junction flow - prezentacja przepustowości każdego ze skrzyżowań (liczba pojazdów, które pokonały dane skrzyżowanie w bieżącej minucie symulacji)

5 Obserwacje, badania i wyniki symulacji

W rozdziale tym przedstawione zostaną wyniki symulacji oraz informacja o przeprowadzonych badaniach.

5.1 Badania

Zakładając, że obwodnica 2 miasta Kraków dwa ma około 12 kilometrów samochód wjeżdżający na nią nie będzie chciał przejechać więcej niż około 6 kilometrów. Dlatego zakładamy, że każdy samochód obecny lub wjeżdżający na obwodnicę będzie pokonywał założony dystans i po jego pokonaniu będzie poszukiwał zjazdu z obwodnicy. Zakładając również, że ruch może odbywać się, gdy liczba samochodów na obwodnicy pozwala na ruch, czyli około połowa długości jezdni jest dostępna (przy liczbie średnio 2 pasów na obwodnicy), to maksymalna liczba samochodów na obwodnicy nie powinna przekroczyć 4000. Ze względu na pewne ograniczenia (wydajność zaproponowanego rozwiązania) przyjęto, że wizualizacja ruchu na mapie może się odbywać (w czasie rzeczywistym do 1500 samochodów), po przekroczeniu tej liczby samochodów dostępna będzie jedynie symulacja (bez wizualizacji).

Parametry wejściowe:

- liczba aut na poszczególnych 26 wjazdach na obwodnicę może zostać ustawiona taka sama na wszystkich wjazdach: 2,5,10,15 na minutę, co daje łącznie 52, 130, 260, 390.
- liczba aut w losowo wybranych miejscach na obwodnicy 750, 1500, 2000, 3000, ze względu na losowość (możliwość powtórzenia pozycji komórki na obwodnicy oraz ograniczenia: nie wstawiane są auta na pasach wyjazdowych) należy przyjąć, że liczba aut na obwodnicy będzie około 70% z zadeklarowanej wartości podczas uruchamiania wymulacji (sprawdzone na podstawie wielu testów) - tabela 1.
- czas świateł – czas świateł zielonych na obwodnicy jest trzykrotnie większy niż czas świateł dróg wjeżdżających na obwodnicę, założono 40 sekund na obwodnicy – oprogramowanie umożliwia zmianę długości świateł, ale badanie przeprowadzono przy powyższych parametrach.

Numer symulacji	Liczba aut wjeżdżających na obwodnicę w 1 minucie	Liczba losowych aut w momencie rozpoczęcia symulacji
1	52	750
2	130	750
3	260	750
4	390	750
5	52	1500
6	130	1500
7	260	1500
8	390	1500
9	52	2000
10	130	2000
11	260	2000
12	390	2000
13	52	3000
14	130	3000
15	260	3000
16	390	3000
17	52	4000

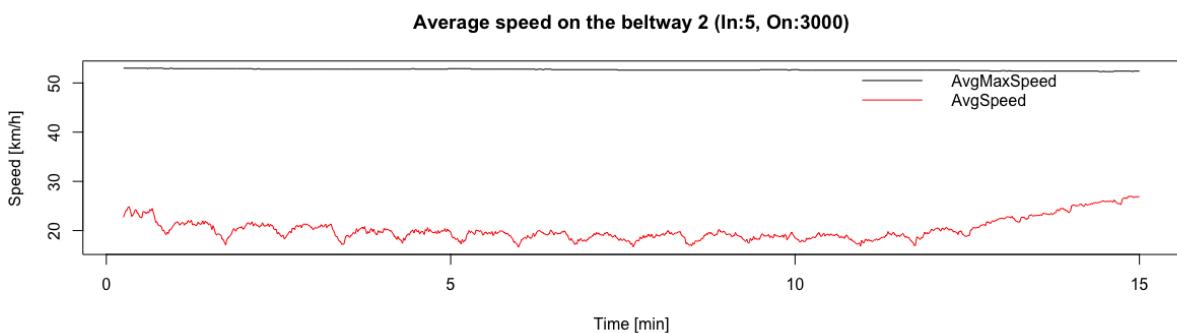
Tablica 1: Liczba samochodów wjeżdżających na obwodnicę oraz liczba samochodów rozpoczynających daną symulację

Ze względu na dużą liczbę możliwych parametrów wejściowych symulację ograniczono do 15 minut – ponieważ bez regulacji długości świateł, liczba symulacji wynosi 16, a każda zmiana długości świateł, to kolejne 15 symulacji. Wyniki przeprowadzonych symulacji zamieszczono w plikach.

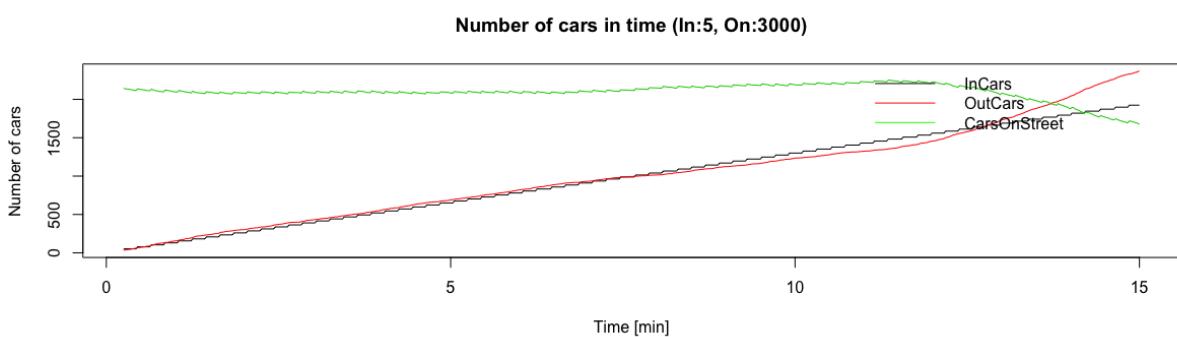
5.2 Analiza wybranych wyników symulacji

W przedstawionych symulacjach najważniejszym elementem badań było sprawdzenia średniej prędkości ruchu wszystkich samochodów na obwodnicy 2 Krakowa (z uwzględnieniem samochodów wjeżdżających) do liczby samochodów będących na obwodnicy.

Na zaprezentowanych wykresach przedstawiono maksymalną prędkość jaką mogłyby uzyskać wszystkie samochody w warunkach idealnych, bieżącą (średnią wszystkich) prędkość samochodów na obwodnicy oraz liczbę samochodów wjeżdżających, zjeżdżających oraz przebywających na obwodnicy.



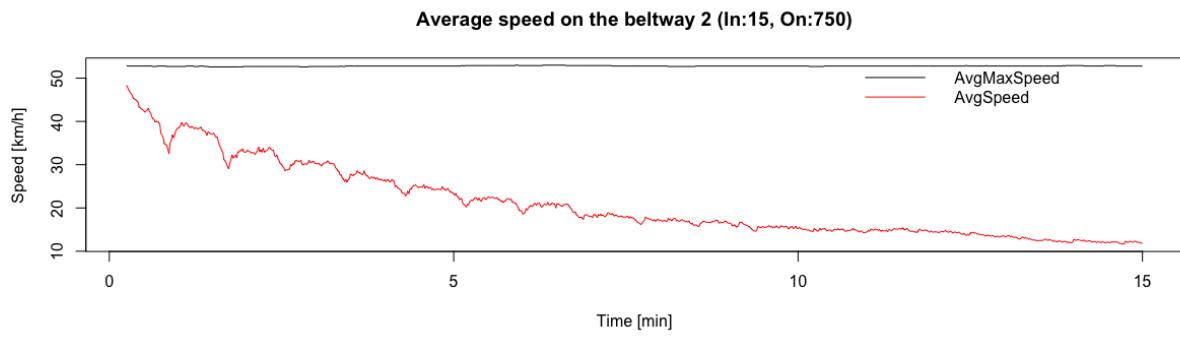
Rys. 11. Średnia prędkość samochodów 5/3000.



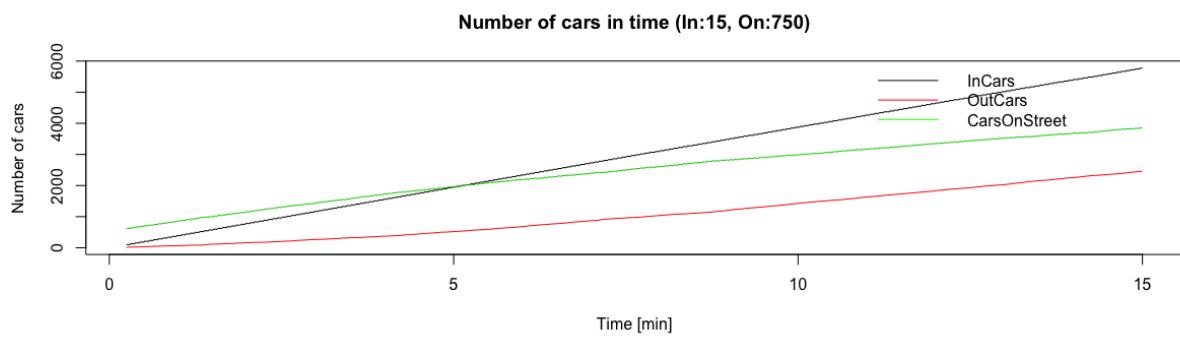
Rys. 12. Liczba samochodów na obwodnicy 5/3000.

Na rysunkach 11 i 12 przedstawiono średnią prędkość samochodów oraz liczbę samochodów na obwodnicy przy tych samych parametrach startowych. Na przedstawionych wykresach można zaobserwować, że w przypadku, gdy liczba samochodów na obwodnicy się nie zmienia (liczba wjeżdżających i zjeżdżających samochodów jest podobna 130 na minutę), to prędkość również niewiele się zmienia w zależności od świateł, które zmieniają się co 40 sekund.

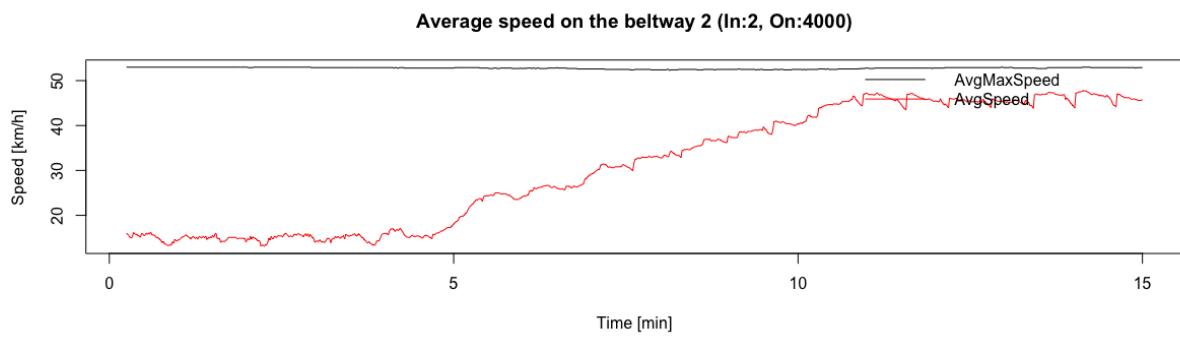
Na rysunkach 13 i 14 przedstawiono przypadek, gdy liczba samochodów wjeżdżających na obwodnicę była trzykrotnie większa niż liczba samochodów zjeżdżających, rozpoczęto od 500 samochodów na obwodnicy i co minutę wjeżdżało 390 samochodów na (po 15 na każdym wjeździe). Po 15 minutach było ponad 4000 aut na obwodnicy, co spowodowało wyraźny spadek prędkości. Można również zauważać, że gdy prędkość średnia wszystkich samochodów na obwodnicy spada poniżej 15 km/h przestają być widoczne jej oscylacje związane ze światłami na skrzyżowaniach.



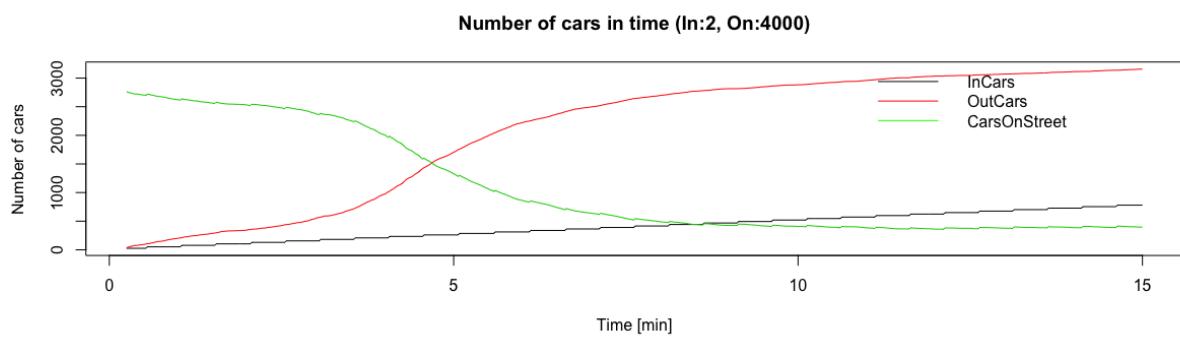
Rys. 13. Średnia prędkość samochodów 15/750.



Rys. 14. Liczba samochodów na obwodnicy 15/750.



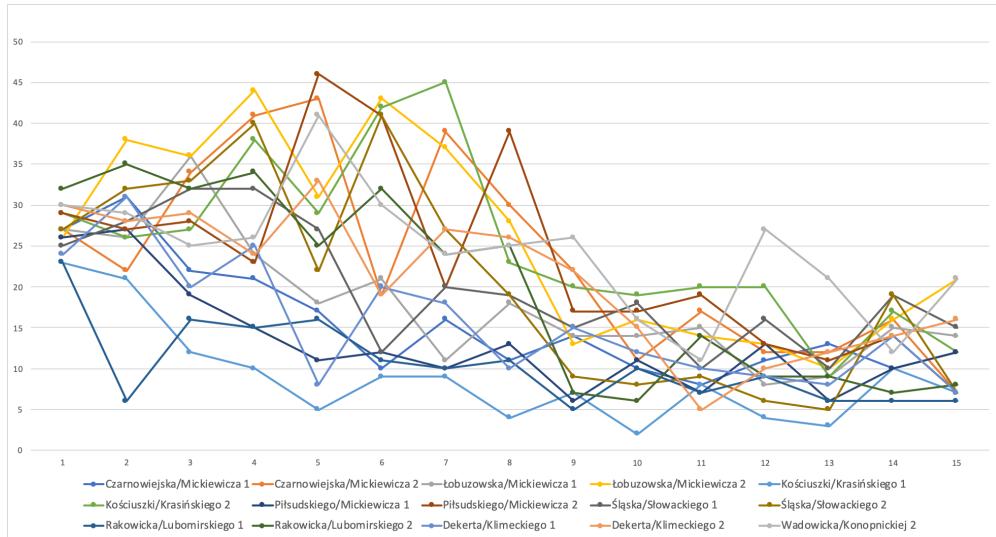
Rys. 15. Średnia prędkość samochodów 2/4000.



Rys. 16. Liczba samochodów na obwodnicy 2/4000.

Na rysunkach 15 i 16 przedstawiono przypadek, gdy liczba samochodów wjeżdżających na obwodnicę była trzykrotnie mniejsza niż liczba samochodów zjeżdżających, rozpoczęto od około 3000 samochodów na obwodnicy i co minutę wjeżdżało 52 samochodów na minutę (po 2 na każdym wjeździe). Po 15 minutach zostało jedynie ponad 500 aut na obwodnicy, co spowodowało wyraźny wzrost średniej prędkości z około 15 km/h do prawie 45 km/h.

Dodatkowo na rysunku 17 zamieszczono również wykres przedstawiający liczbę samochodów przez dane skrzyżowanie w danej minucie (część skrzyżowań posiadających prawoskrętny nie jest zależna od długości świateł, tylko od możliwości wjazdu na obwodnicę).



Rys. 17. Przepływ na skrzyżowaniach.

Załączone pliki na githubie zawierają szczegółowe dane dotyczące każdej z symulacji, nazwy plików pozwalają na rozszyfrowanie konkretnej symulacji; pierwsze dwa znaki informują o liczbie aut wjeżdżających na każdym z przygotowanych zjazdów (realna liczba wjeżdżających aut na wszystkich wjazdach jest 26 razy większa), kolejne cztery miejsca (w nazwie pliku) informują o liczbie losowanych samochodów podczas startu symulacji, kolejne dwa znaki, to informacja o długości świateł (w sekundach) na poszczególnych skrzyżowaniach obwodnicy.

5.3 Analiza szczegółowa przeprowadzonych badań

W tablicy 2 zamieszczono szczegółowe wyniki dotyczące danych związanych z rozpoczęciem symulacji oraz informację na temat liczby aut pozostających na obwodnicy, gdy symulacja została zakończona.

Numer symulacji	Liczba aut na starcie symulacji	Średnia prędkość na starcie symulacji [km/h]	Liczba aut na końcu symulacji
1	500	44	360
2	580	45	1080
3	630	43	2200
4	800	39	3950
5	1050	38	410
6	1060	38	950
7	1130	35	2550
8	1180	32	4100
9	1480	34	430
10	1490	34	1310
11	2050	29	3500
12	1550	26	4500
13	2100	21	610
14	2050	23	1650
15	2040	20	2550
16	2150	19,5	5300
17	2650	14	450

Tablica 2: Liczba samochodów oraz średnia prędkość wszystkich samochodów w danej symulacji

Dane zebrane podczas kilkunastu symulacji wyraźnie wskazują, że opracowany system wydaje się działać poprawnie, gdyż wraz ze wzrostem liczby aut na obwodnicy 2 miasta Kraków średnia prędkość aut będących na niej spada.

W tablicy 3 zamieszczone dodatkowo informację dotyczącą średniej prędkości aut na obwodnicy po zakończeniu symulacji.

Dane zamieszczone w tablicy 3 potwierdzają skuteczność działania opracowanego modelu i symulacji. Wraz ze wzrostem liczby aut średnia prędkość wszystkich aut spada, natomiast przy zmniejszającej liczbie aut prędkość wzrasta.

W tablicy 4 zamieszczone szczegółowe wyniki dotyczące przeprowadzonych symulacji.

Przedstawione dane w tablicy 4 wyraźnie wskazują, że średnia prędkość samochodów na obwodnicy 2 miasta Kraków jest zależna od liczby samochodów poruszających się po niej. Z przeprowadzonych symulacji wynika, że przy liczbie samochodów większej niż 5000 ruch na obwodnicy 2 miasta Kraków ustaje, ponieważ średnia prędkość wszystkich samochodów spada poniżej 7 km/h.

W tablicy 5 przedstawiono dane dotyczące średniego przepływu samochodów. Ponieważ przepływ samochodów był badany na wielu skrzyżowaniach, postanowiono zaprezentować uśrednione wyniki, prezentacja wszystkich badanych skrzyżowań mogłaby nic nie wnosić do zrealizowanego projektu, a jedynie mogłaby zwiększyć objętość opisu, a co za tym idzie zaciemnić (ilościowo) otrzymane wyniki.

Dane zamieszczone w tablicy 5 należy porównać z danymi w tablicy 1, gdzie zamieszczone informację na temat liczby aut w danej symulacji. Wyraźnie widać, że średnia liczba samochodów przejeżdżających przez poszczególne skrzyżowania w ciągu jednej minuty jest skorelowana z liczbą samochodów będących na obwodnicy. Wydaje się, że minimalna liczba samochodów przez dane skrzyżowania nie jest powiązana z liczbą aut na obwodnicy, może się przecież zdarzyć, że na danym skrzyżowaniu nie pojawi się ani jeden samochód w ciągu minuty (może być korek kilka światel wcześniej).

Numer symulacji	Liczba aut na starcie symulacji	Średnia prędkość na starcie symulacji [km/h]	Liczba aut na końcu symulacji	Średnia prędkość na końcu symulacji [km/h]
1	500	45	360	48
2	580	45	1080	37,5
3	630	43	2200	21
4	800	39	3950	12
5	1050	38	410	47
6	1060	38	950	36,5
7	1130	35	2550	18
8	1180	34	4100	11,5
9	1480	34	430	48
10	1490	34	1310	31
11	2050	29	3500	17
12	1550	26	4500	10
13	2100	21	610	41
14	2050	23	1650	27
15	2040	20	2550	14,2
16	2150	19,5	5300	7
17	2650	14	450	46

Tablica 3: Liczba samochodów oraz średnia prędkość na końcu symulacji

Numer symulacji	Minimalna średnia prędkość wszystkich samochodów [km/h]	Maksymalna średnia prędkość wszystkich samochodów [km/h]	Średnia prędkość wszystkich samochodów [km/h]
1	31,4	49,5	42,71
2	28,4	48,9	37,6
3	20,4	47,9	28,26
4	11,6	49	21,54
5	29,1	47,8	39,61
6	27,4	45,6	37,6
7	17,8	43,6	24,33
8	11,1	43,1	18,71
9	24,3	49,3	34,83
10	27,4	40,6	27,68
11	14,9	38,6	20,39
12	10,1	36	15,98
13	17,9	40,8	24,33
14	16,6	29	20,21
15	10,2	28	14,82
16	7,12	24,4	11,86
17	13,1	47,5	30,7

Tablica 4: Średnie prędkości w poszczególnych symulacjach

Numer symulacji	Minimalna liczba samochodów przejeżdżających przez skrzyżowanie	Maksymalna liczba samochodów przejeżdżających przez skrzyżowanie	Średnia liczba samochodów przejeżdżających przez skrzyżowanie
1	2	46	18,95
2	0	70	22,21
3	11	88	46,38
4	18	180	60,58
5	2	77	26,19
6	4	69	26,21
7	11	111	51,67
8	11	147	63,07
9	1	86	31,14
10	3	99	36,6
11	9	105	56,63
12	15	182	68,8
13	1	92	36,93
14	2	92	51,15
15	9	181	66,4
16	17	189	74,21
17	0	86	16,8

Tablica 5: Przepustowość skrzyżowań w poszczególnych symulacjach

6 Wnioski

Z zamieszczonych w dokumentacji wyników badań można wywnioskować, że przygotowane oprogramowanie w oparciu o automat komórkowy działa poprawnie, gdyż wydaje się być zgodny z rzeczywistym zachowaniem samochodów na drodze (im więcej aut tym mniejsza średnia predkość i odwrotnie).

Można z pewnością założyć, że przygotowane oprogramowanie do symulacji z wykorzystaniem automatu komórkowego będzie obarczona pewnymi błędami, gdyż przygotowany model zawiera pewne ograniczenia związane z dostępem do darmowych danych wejściowych. Serwisy OSM, GoogleMaps, MyGeoData.cloud udostępniające dane dotyczące koordynat oraz opisu jezdni w wersji darmowej nie pozwalają na dostęp do pełnych/pewnych danych. W realizowanym projekcie postanowiono wprowadzić pewne ograniczenia dotyczące niektórych rond, skrzyżowań, wiaduktów oraz dróg wielopoziomowych. Ograniczenie to wprowadzono ze względu na pewne ograniczenie czasowe (realizacji projektu) oraz na obecną/panującą sytuację epidemii, która częściowo uniemożliwiła nam zasięgnięcia wiedzy od ekspertów. Analiza (części darmowych) danych pozwala wyciągnąć wnioski, że dane opisujące drogi (np. w mieście Kraków) są obarczone wieloma błędami, np. typy/rodzaj ulic zdarzają się być błędnie oznaczone, liczba pasów danej drogi nie uwzględnia ograniczeń typu buspasy, itp. Dane umieszczone na OpenStreetmap są gromadzone przez tzw. społeczność, często jakość przekazanych danych (na podstawie przeprowadzonych analiz wybranych punktów, skrzyżowań czy dokładności koordynat dł/sz geograficznej) pozostawia wiele do życzenia, przykładowo ulica Reymonta została oznaczona jako ulica osiedlowa, a nie jako ulica główna posiadającej w pewnych fragmentach wiele pasów w różnych kierunkach.

7 Literatura

- [1] https://pl.wikipedia.org/wiki/Obwodnice_Krakowa
- [2] http://www.zis.agh.edu.pl/app/MSc/Rafal_Socha/
- [3] http://www.zis.agh.edu.pl/app/MSc/Kamil_Nowak/
- [4] <http://www.zis.agh.edu.pl/ak/ca.pdf>
- [5] <http://home.agh.edu.pl/~malarz/dyd/ak/>
- [6] <https://www.openstreetmap.org/>
- [7] <https://mygeodata.cloud>
- [8] <https://osmcode.org/osmium-tool/manual.html>
- [9] <https://www.mapbox.com>
- [10] <https://www.google.pl/maps/d/u/0/>
- [11] <https://leafletjs.com>
- [12] J. Esser, M. Schreckenberg, Microscopic Simulation of Urban Traffic Based on Cellular Automata. International Journal of Modern Physics, vol. 8, No. 5, 1997, 10251036.
- [13] J. Wąs, R. Bieliński, B. Gajewski, P. Orzechowski; Problematyka modelowania ruchu miejskiego z wykorzystaniem automatów komórkowych, Automatyka, AGH, 2009