

Symulacja ruchu drogowego na IV obwodnicy Krakowa

Szymon Gałuszka, Michał Worsowicz, Maciej Nalepa

10 czerwca 2020

Część I

Omówienie projektu

1 Definicja problemu

Nasz cel to symulacja ruchu drogowego na IV obwodnicy Krakowa. Konieczna jest definicja tras oraz sposobu poruszania się po nich. Obwodnica tworzy zamkniętą pętlę (uwzględniając odcinek w budowie) i dzieli się na ruch zgodny i przeciwny do wskazówek zegara.

Ruch drogowy powinien uwzględniać pojazdy osobowe, ciężarowe i transport publiczny. Ciężkie pojazdy obowiązują inne ograniczenia prędkości oraz zajmują więcej przestrzeni na jezdni. Napływ ruchu powinien odbywać się przez wjazdy na obwodnicę, które wpuszczają samochody z określoną częstotliwością, która może zależeć od pory dnia.

Symulacja może obejmować wydarzenia losowe, takie jak:

- blokada pasa ruchu,
- zamknięcie zjazdu,
- nagłe hamowanie.

2 Obszar symulacji

Symulowany przez nas obszar to IV obwodnica Krakowa, znana także jako obwodnica autostradowa Krakowa, ponieważ większość jej odcinka stanowi autostrada A4.

Na obwodnicy nie ma sygnalizacji świetlnej, skrzyżowania znajdują się najczęściej pod wiaduktem i najpierw należy zjechać z drogi szybkiego ruchu. Trasa ma zróżnicowane ograniczenia prędkości oraz różną ilość pasów ruchu.

Fragment ten na zachodnich obrzeżach miasta jest dwupasmowy, a na południowych - trójpasmowy. W najbliższym czasie przewidywana jest budowa odcinka północnego, który uwzględniliśmy w naszej symulacji, odtwarzając i dodając ten fragment do projektu. Poniżej przedstawiony jest wykaz węzłów drogowych, które obejmuje symulacja (zgodnie z kierunkiem ruchu wskazówek zegara):

- Kraków Nowa Huta,
- Kraków Przewóz,
- Kraków Bieżanów,
- Kraków Wieliczka,
- Kraków Łagiewniki,
- Kraków Południe,
- Kraków Skawina,
- Kraków Tyniec,
- Kraków Bielany,
- Kraków Balice (lotnisko),
- Balice I,
- Modlniczka,
- Modlnica,
- Kraków Północ – dla uproszczenia przyjęto taką nazwę planowanego węzła (wariant I)

3 Algorytm

Zastosowany przez nas algorytm ruchu drogowego to zmodyfikowany model Nagela-Schreckenberga, będący teoretycznym modelem mikroskopowym o charakterze dyskretnym, w którym obie jednokierunkowe jezdnie są podzielone na komórki, które zawierają informację na temat ilości pasów oraz ilości pojazdów w danej chwili na jezdni. Dzięki takiemu uproszczeniu, nie jest konieczna symulacja zmiany pasu ruchu, ponieważ nie jest on określony wprost.

Każda z komórek może być interpretowana na jeden ze sposobów:

- komórka pusta – fragment pustej drogi
- komórka używana – fragment zawierający pojazd na nieokreślonym pasie
- komórka pełna – fragment drogi z pojazdem na wszystkich pasach ruchu

Każdy samochód n posiada swoją prędkość $v(n)$ o wartości liczby naturalnej, nie większej od prędkości maksymalnej v_{max} – symbolizuje ona maksymalną prędkość osiąganą przez samochód.

Czas t w modelu przyjmuje wartość dyskretną o stałym kroku. Nasz algorytm można podzielić na cztery następujące po sobie czynności:

1. Losowość:

Czynność ta symbolizuje wszelkie przypadki losowe z jakimi kierowca może spotkać się na drodze. Dla każdego samochodu n prędkość może zostać zmniejszona, zwiększona lub dopasowana do ograniczenia na drodze (hamowanie lub przyspieszenie o wartość 1 w kierunku przepisowej prędkości).

2. Ruch:

Jeżeli odległość $d(n)$ samochodu n od samochodu znajdującego się przed nim jest mniejsza od prędkości $v(n)$ tego samochodu to prędkość samochodu ulega zmniejszeniu. Jako że odległość mierzona jest w liczbie komórek, a prędkość w liczbie komórek na jednostkę czasu, to prędkość ostateczna może wynosić maksymalnie $d(n)$ na jednostkę czasu, a minimalnie zero.

3. Populacja:

Punkty startowe z pewnym prawdopodobieństwem generują nowy pojazd, jeśli jest dla niego miejsce.

4. Zapis danych:

Ostatnim krokiem jest zapisanie aktualnych stanów do monitorów zdefiniowanych przez użytkownika.

Część II

Realizacja

4 Struktury danych

Obsługiwany plik mapy SimpleMap jest w formacie json i przybiera następujący format:

```
{
    "nazwa_wjazdu;nazwa_zjazdu": {
        "lanes": liczba_pasow,
        "maxspeed": ograniczenie_predkosci,
        "geometry": [
            [lat, lon],
            ...
        ]
    },
    ...
}
```

Rysunek 1: Przykład mapy w formacie SimpleMap.

Dla kompatybilności z pierwszą realizacją mapy za pomocą formatu OpenStreetMap-json powstał moduł `OpenMap`. Użycie tego formatu wymaga konwersji z formatu `.osm` do `.json`, może do tego posłużyć program JOSM.

Automat komórkowy – `Cellular` przechowuje listę następujących obiektów: pojazdów (agentów), komórek, punktów startowych i końcowych. Umożliwia budowanie mapy na podstawie pliku `json` kompatybilnego z modulem opracowania własnego `SimpleMap`.

Komórka – `Cell` przechowuje wskaźnik do następnej komórki, ilość dostępnych pasów, dopuszczalną prędkość oraz aktualne wykorzystanie.

Logi – `Stat` zapisuje dane w postaci ramki danych `pandas`. W każdej iteracji dopisywane są wiersze z monitorowanymi danymi.

Konfiguracja – `CONFIG` udostępniony przez moduł `utils` pozwala na podłączenie klasy definiującej statycznie następujące parametry:

- `Timestep` – krok czasu t ,
- `Radius` – rozmiar komórki,
- `Spawn rate` – domyślne prawdopodobieństwo pojawienia się nowego pojazdu w danym kroku,
- `Agent driveoff` – prawdopodobieństwo opuszczenia obwodnicy,
- `Agent slow` – domyślne prawdopodobieństwo hamowania,
- `Agent fast` – domyślne prawdopodobieństwo przyspieszenia,
- `Agent limit` – prawdopodobieństwo dążenia do ograniczenia prędkości,
- `Agent Vmax` – domyślna maksymalna prędkość samochodu

5 Implementacja

6 Usprawnienie

Najsłabsze punkty implementacji:

- Zapis jednego stanu mapy to ramka danych o rozmiarze ilości komórek na mapie. Oznacza to, że pojedyncza sekunda symulacji generuje nawet 30000 wpisów do ramki. W celu uniknięcia problemu z zapełnianiem pamięci stworzono specjalną klasę `LastCellStat` przechowującą określoną ilość stanów wstecz. Lepszym rozwiązaniem może być zapis logów bezpośrednio do pliku.
- Konfiguracja prawdopodobieństwa opuszczenia skrzyżowania jest globalna, powinna być zależna od węzła.

Część III

Podsumowanie

7 Wyniki

Wszystko jest gites i działa!

8 Od autorów

Nasz framework umożliwia w obecnym stanie analizę interesującego parametru: średniego czasu jaki samochód spędza na obwodnicy. Postanowiliśmy zostawić to jako propozycję do samodzielnego eksperymentu z tą symulacją.

9 Dalsza praca

Aby uczynić projekt bardziej uniwersalnym, należy dodać obsługę innych zaawansowanych map. Można to zrobić poprzez rozbudowanie obecnego modułu `SimpleMap` lub zaimplementować obsługę znanych już formatów, np. `OpenStreetMap`, `Google MyMaps`.

Projekt nie posiada innych pojazdów, niż domyślny osobowy. Zatem kolejnym usprawnieniem jest wprowadzenie pojazdów, które:

- obowiązują inne ograniczenia prędkości
- zajmują więcej, niż jedną komórkę

Obsługa całej platformy może zostać zawarta w wygodnej dla użytkownika oprawie graficznej, np. `kivy`. Na chwilę obecną najbardziej interaktywny interfejs uzyskujemy poprzez `jupyter notebook`, który ma duże ograniczenia wydajności.

Literatura

- [1] Wiki, *Kraków Obwodnica IV*, wikipedia.org
- [2] OpenStreet, *mapa*, openstreetmap.org
- [3] K. Nagel, M. Schreckenberg, *Two lane traffic simulations using cellular automata*, arxiv.org