

MAŁECKI Krzysztof<sup>1</sup>  
SZMAJDZIŃSKI Maciej<sup>2</sup>

## Symulator do mikroskopowej analizy ruchu drogowego

*Symulacja komputerowa, mikrosymulacja,  
modele ruchu drogowego,  
automaty komórkowe*

### Streszczenie

*W niniejszym artykule przedstawiono opracowany symulator do badania efektywności ruchu drogowego dla wyznaczonego obszaru. Warstwę logiczną narzędzia stanowi teoria automatów komórkowych, ze szczególnym uwzględnieniem modeli ruchu drogowego. W pierwszej części artykułu autorzy wprowadzają pojęcie automatu komórkowego i prezentują wybrane modele ruchu drogowego, by w dalszej części przedstawić założenia i części składowe opracowanego oprogramowania oraz możliwości jego wykorzystania. Całość wsparta jest licznymi rysunkami ukazującymi poszczególne ekrany aplikacji i wnioskami końcowymi.*

### MICROSCOPIC ROAD TRAFFIC ANALYSIS SIMULATOR

### Abstract

*This article presents the developed simulator for measuring traffic flow efficiency for a designated area. Logical layer of the simulator is based on the cellular automata theory with particular emphasis on road traffic models. In the first part of the article authors introduce the concept of cellular automata and show some of the road traffic models to later present the principles and components of the developed software and its possible usage. Everything is supported by numerous illustrations showing the various application screens and the final conclusions.*

### 1. WSTĘP

Utrudnienia komunikacyjne spowodowane zbyt małą przepustowością obecnych dróg są stałym problemem wielu polskich aglomeracji miejskich. Możliwość płynnej i sprawnej komunikacji na obszarze miejskim jest dla mieszkańców szczególnie ważnym aspektem. Niestety możliwości przebudowy miejskich węzłów drogowych są mocno ograniczone a w niektórych przypadkach wręcz niemożliwe. Pozostaje zatem wykorzystać możliwości wysterowania sygnalizacji świetlnej (tam, gdzie one istnieją) w celu poprawienia przepustowości określonych miejsc. Jednakże wykonywanie takich działań, dobrze by było poprzedzić odpowiednią symulacją komputerową potwierdzającą słuszność idei oraz poprawność wprowadzanych parametrów. W tym celu został opracowany, opisywany w niniejszym artykule, mikro-symulator ruchu drogowego. Do sterowania poruszającymi się pojazdami wykorzystano modele matematyczne, które w precyzyjny, bezbłędny sposób pozwalają na opisanie wybranego wycinka rzeczywistości i zachodzących w nim zjawisk.

Interesującym przykładem wspomnianych modeli matematycznych są automaty komórkowe, których idea i cechy pozwalają na efektywne wykorzystanie do rozwiązywania problemów symulacji poruszających się obiektów. Teoria automatów komórkowych znajduje zastosowanie w wielu dziedzinach nauki m.in. w biologii, biofizyce czy fizyce [4][5], w tym w dziedzinach transportu (hydrodynamika, gaz sieciowy, zatory drogowe, materiały granulowane, itp.). Opisywane narzędzie do symulacji ruchu bazuje na modelach ruchu opisywanych w [6][7]. Zadaniem programu jest: wyznaczenie ruchu realizowanego w warunkach zbliżonych do rzeczywistych w stosunku do ruchu odbywającego się po modernizacji obszaru, dobór odpowiednich ustawień sygnalizacji świetlnej w celu zwiększenia przepustowości węzła oraz popularyzacja mikro-symulatorów ruchu drogowego o dostępie darmowym.

### 2. WYBRANE ZAGADNIENIA TEORETYCZNE

W niniejszym rozdziale przedstawione zostaną podstawowe pojęcia i ich definicje, co umożliwi lepsze rozumienie opisywanych aspektów.

#### 2.1 Definicja automatu komórkowego

Najprostszy automat komórkowy, to model matematyczny składający się z trzech elementów [4][5]:

- dyskretnej  $n$ -wymiarowej przestrzeni (siatki) podzielonej na jednakowe komórki,

<sup>1</sup> Zachodniopomorski Uniwersytet Technologiczny w Szczecinie; Wydział Informatyki; 71-210 Szczecin; ul. Żołnierska 52. Tel: +48 91 449 55 14; E-mail: kmalecki@wi.zut.edu.pl

<sup>2</sup> Zachodniopomorski Uniwersytet Technologiczny w Szczecinie; Wydział Informatyki; 71-210 Szczecin; ul. Żołnierska 52. E-mail: mszmajdzinski@wi.zut.edu.pl

- skończonego  $k$ -elementowego zbioru stanów dla pojedynczej komórki,
- reguły  $F$  określającej stan komórki w chwili  $t+1$ , w zależności od jej stanu i stanu komórek ją otaczających w chwili  $t$ .

$N$ -wymiarowa siatka identycznych komórek podlega ewolucji w dyskretnych chwilach czasu. Każda z komórek siatki może przyjąć jeden z  $k$  stanów określonych przez regułę:

$$S_i(t+1) = F(s_j(t)), \text{ j należy do } O(i) \quad (1)$$

gdzie:  $O(i)$  – otoczenie  $i$ -tej komórki,  $S_i$  – stan komórki,  $F$  – reguła (funkcja) przejścia.

## 2.2 Siatka automatu komórkowego

Siatka automatu to  $n$ -wymiarowa, dyskretna przestrzeń zbudowana z identycznych komórek o takich samych parametrach, tj. o identycznej liczbie sąsiadów, takim samym kształcie czy przestrzeni stanów. Nie jest dopuszczalne istnienie dwóch komórek posiadających różne elementy. Na jej strukturę ma wpływ wymiar przestrzeni automatu, kształt komórek i związana w tym liczba sąsiadów.

## 2.3 Reguła przejścia

Automat komórkowy ewoluuje poprzez zmianę stanów komórek na siatce. To, w jaki sposób odbywa się zmiana pomiędzy ich stanami określa tzw. reguła przejścia. W określonych, dyskretnych chwilach czasu automat dla każdej lub wybranych komórek zbiera informację o stanach jej sąsiadów jak i własnym, po czym za pomocą reguły decyduje o jej nowym stanie [3][8].

## 2.4 Warunki brzegowe

Próbując implementować nieskończoną z definicji przestrzeń automatu, problemem stanie się ograniczona pamięć maszyny, na której będziemy wykonywać automat. Z tego powodu siatka przyjmuje kształt skończonej tablicy o odpowiednim wymiarze. Ograniczony rozmiar siatki zmusza do określania dodatkowych warunków brzegowych odnośnie zachowania się komórek na jej granicach. Wyróżnia się trzy typy warunków brzegowych [4][5]:

- periodyczne – po przejściu przez krawędź część pojawia się po drugiej stronie,
- pochłaniające – po przekroczeniu krawędzi przestaje istnieć,
- odbijające – część odbija się od krawędzi siatki.

Periodyczne warunki brzegowe najlepiej sprawdzają się przy reprezentacji przestrzeni sferycznej. Pochłaniające warunki brzegowe implementowane są, gdy podczas ewolucji automatu mają być tworzone nowe obiekty, których ogólna liczba musi być ograniczona. Do symulacji zachowań w zamkniętej przestrzeni stosuje się odbijające warunki brzegowe.

## 2.5 Sąsiedztwo komórek

Sąsiedztwo komórek określa, które z komórek znajdujących się w najbliższym otoczeniu brane są pod uwagę podczas wykonywania funkcji przejścia automatu. Najczęściej występujące sąsiedztwa to [4][5]:

- sąsiedztwo Moore'a – składa się ze wszystkich komórek otaczających komórkę, dla której określamy sąsiedztwo,
- sąsiedztwo von Neumana – składa się z komórek graniczących krawędziami z komórką, dla której określamy sąsiedztwo.

## 2.6 Automaty komórkowe w modelowaniu ruchu drogowego

Idea wykorzystania automatów komórkowych w symulacji ruchu drogowego spotkała się z dużym zainteresowaniem świata naukowego, o czym może świadczyć spora liczba powstałych modeli, np. publikowanych w [1][2][6][7]. Automaty komórkowe z są proste w budowie i działaniu. Użycie zestawu prostych reguł nałożonych na dany automat prowadzi do skomplikowanych zachowań. Z tego powodu modele bazujące na teorii automatów komórkowych wiernie odtwarzają świat rzeczywisty i zachodzące w nim reakcje, jak np. tworzenie się zatorów na drogach przy wymuszeniu chwilowego zatrzymania pojazdu [12]. Dodatkowo założenia techniczne modeli automatów komórkowych sprawiają, że złożoność modelu nie rośnie wraz ze wzrostem wielkości problemu. Mając więc działający symulator dla małych węzłów, będzie on działał prawidłowo po załadowaniu do niego map z zamodelowanymi całymi dzielnicami [13].

W tabeli 1 przedstawione zostały wybrane i jednocześnie najważniejsze z opracowanych modeli – od prostego modelu reprezentowanego przez ciąg komórek w jednowymiarowej tablicy, do najnowszych, w których symulacja może odbywać się na drogach dwukierunkowych, skrzyżowaniach, z udziałem sygnalizacji świetlnej a także przy wykorzystaniu pętli indukcyjnych [6].

Tab. 1. Modele bazujące na automatach komórkowych w logistyce miejskiej

Model	Opis modelu
Model podstawowy – model Choparda-Luthego-Queloza	Elementarny automat komórkowy zdefiniowany przez Wolframa o regule numer 184 przedstawia ruch pojazdów na drodze jednokierunkowej i jednopasmowej. Droga ta jest wyrażona przy pomocy jednowymiarowej tablicy, w której każda z komórek może być zajęta przez pojazd lub wolna, co odpowiada dwóm stanom 1 lub 0 odpowiednio. Wszystkie samochody poruszają się w jedną stronę (np. w prawo) maksymalnie o jedną pozycję pod warunkiem, że komórka docelowa jest wolna.
Model Nagela-Schreckenberga	Model opisujący ruch samochodów na drodze jednopasmowej i jednokierunkowej. Pas ruchu podzielony jest na odcinki (komórki) o długości 7,5 metra (odpowiada to długości przeciętnego samochodu), a każdy pojazd porusza się z prędkością nie większą od prędkości maksymalnej, ale nie mniejszą od zera. Prędkość pojazdu odnoszona jest do liczby komórek przemierzanych przez ten pojazd w jednostce czasu (np. 5 odpowiada prędkości 135 km/h). Model posiada pochlaniające warunki brzegowe. Funkcja przejścia podzielona jest na 4 etapy: przyspieszenie, hamowanie, zdarzenie losowe i przesunięcie.
Model Chowdhuryego-Schadschneidera	Model skrzyżowania oparty na siatce $N \times N$ z jednokierunkowymi ulicami równoległymi do osi X lub Y. Ruch pojazdów odbywa się tylko w kierunku: w prawo lub do góry. Na każdym skrzyżowaniu (węzle sieci) znajduje się sygnalizacja świetlna, która zmienia się, co określoną liczbę jednostek czasu $t$ i umożliwia ruch tylko w jednym kierunku. Odcinki pomiędzy węzłami sieci podzielone są na $D$ komórek. Model posiada okresowe warunki brzegowe. Funkcja przejścia jak w modelu NaSch.
Model Ring-Road – ruch okrężny	Ruch pojazdów odbywa się na drodze dwupasmowej i jednokierunkowej reprezentowanej przez tablicę o dwóch wierszach i $n$ kolumnach. Funkcja przejścia jak w modelu NaSch z tym, że pojazdy mogą zmieniać pas ruchu. Model jest systemem zamkniętym, przedstawia rondo, na które nie wjeżdża i z którego nie wyjeżdża żaden pojazd. W symulacji występują samochody osobowe i ciężarowe. Model zaprezentowany przez Martina Treibera.
Model On Ramp – pas rozbiegowy	Model przedstawia drogę dwupasmową, jednokierunkową wraz z pasem rozbiegowym. Drogę reprezentuje tablica dwuwymiarowa z trzema wierszami. Pierwsze dwa wiersze to droga główna ( $k_1$ kolumn), a trzeci to droga dojazdowa ( $k_2$ , przy czym $k_2 < k_1$ ). Brzegowe $m$ komórek trzeciego wiersza przedstawiają pas rozbiegowy, z którego kierowca może się włączyć do ruchu na drogę główną. Komórki są opisane tak jak w modelu NaSch. Funkcja przejścia składa się z 5 etapów: zmiana pasa ruchu, przyspieszenie, hamowanie, zdarzenie losowe i przesunięcie. W symulacji występują samochody osobowe i ciężarowe. Model zaprezentowany przez Martina Treibera.
Model Laneclosing – zamknięcia drogi z symulacją ograniczenia prędkości	Model opisuje ruch pojazdów na drodze dwupasmowej, jednokierunkowej. Fragment tej drogi jest zamknięty, np. z powodu robót drogowych. Siatka automatu komórkowego ma postać tablicy dwuwymiarowej o dwóch wierszach i $n$ kolumnach. Komórka oprócz stanu zajętej przez pojazd lub wolnej może przyjąć stan „wyłączenia z ruchu”. Funkcja przejścia składa się z 5 etapów: zmiana pasa ruchu, przyspieszenie, hamowanie, zdarzenie losowe i przesunięcie. W symulacji występują samochody osobowe i ciężarowe. Dodatkowo wprowadzono ograniczenie prędkości dla pojazdów ciężarowych (w początkowej konfiguracji do 80 km/h). Model zaprezentowany przez Martina Treibera.
Model Uphill Grade – stromy podjazd	Model przedstawia ruch na drodze dwupasmowej, i jednokierunkowej w postaci tablicy o dwóch wierszach i $n$ kolumnach. Symulowane są utrudnienia w ruchu spowodowane przez wolno jadące pojazdy poruszające się po stromych podjazdach. W modelu wybrane $n-k$ komórki reprezentują fragment drogi, na której samochody zwalniają. Komórki mogą przyjmować dwa stany: wolna lub zajęta. Funkcja przejścia przebiega następująco: zmiana pasa ruchu, przyspieszenie, hamowanie, zdarzenie losowe i przesunięcie. Dodatkowo wprowadzono ograniczenie prędkości Model zaprezentowany przez Martina Treibera.
Model Traffic Lights – sygnalizacja świetlna	Model symulujący ograniczenia w ruchu, jakim jest sygnalizacja świetlna. Sieć komórek o dwóch wierszach i $n$ kolumnach przedstawia dwupasmową drogę jednokierunkową. Każda komórka ma dwa stany – zajęta przez pojazd lub wolna. Co więcej ustalone jest miejsce, które odpowiada linii warunkowego zatrzymania się przed sygnalizacją. Światła zmieniają się w określonej jednostce czasu. Funkcja przejścia: zmiana pasa ruchu, przyspieszenie, hamowanie, zdarzenie losowe i przesunięcie. Dodatkowo wprowadzono ograniczenie prędkości. Model zaprezentowany przez Martina Treibera.
Model Lane Changes – zmiany pasa ruchu	Model identyczny z modelem ruchu okrężnego, tyle, że na drodze dodatkowo umieszczone zostały przeszkody na obydwu pasach ruchu, co zmusza pojazdy do zmiany pasa. Model zaprezentowany przez Martina Treibera.
Rozwinięcie modelu skrzyżowania	Rozwinięcie modeli skrzyżowania [1] i [11]; modyfikacja drugiego etapu funkcji przejścia, poprzez uwzględnienie detektora ruchu (np. pętli indukcyjnej), co powoduje eliminację możliwości zablokowania skrzyżowania. Model zaproponowany przez Krzysztofa Małeckiego w [6].

źródło: opracowanie własne na podstawie analizy literatury.

### 3. SYMULATOR AUTOMAT

#### 3.1 Założenia i możliwości opracowanej aplikacji

Celem opracowanego programu jest symulacja i wizualizacja ruchu drogowego, przy wykorzystaniu wyżej przedstawionych modeli. Program ma budowę modułową, co umożliwia równoległe prowadzenie różnych wątków implementacyjnych. Aby uczynić opracowane narzędzie bardziej elastycznym, wyodrębniono moduł tworzenia mapy (obszar, dla którego będzie prowadzona symulacja ruchu), którego wynikiem końcowym jest odpowiednio zdefiniowany plik .xml. Takowy, zewnętrzny plik mapy jest następnie wczytywany do głównego programu w celu przeprowadzenia stosownych badań eksperymentalnych.

Przed przystąpieniem do implementacji zdefiniowano efekty końcowe, czyli możliwości symulatora:

- umożliwia wczytywanie map jedno- lub wielopoziomowych z zewnętrznych plików .xml,
- pozwala na sterowanie przebiegiem symulacji,
- umożliwia regulację natężenia ruchu,
- dopuszcza wybór modelu odzwierciedlającego poruszanie się pojazdów,
- umożliwia sterowanie sygnalizacją świetlną wraz z powiązanymi z nią pętlami indukcyjnymi,
- realizuje symulację znaków drogowych określających miejsca, gdzie należy ustąpić pierwszeństwa, co pozwala wyodrębnić drogi uprzywilejowane i podporządkowane,
- umożliwia wyświetlenie statystyk badanego drogowego węzła komunikacyjnego, tj. procentowy udział pojazdów poruszających się w stosunku do stojących pojazdów oraz ich średnią prędkość.

#### 3.2 Technologia wykonania

Implementacja programu została wykonana w technologii JAVA, która pozwala na tworzenie aplikacji dających się uruchomić na dowolnej platformie. Warunkiem koniecznym jest posiadanie zainstalowanej maszyny wirtualnej JAVA.

JAVA jest prostym, w pełni obiektowym i czytelnym językiem, w którym bardzo dobrze widać hierarchię użycia klas. JAVA jako język kompilowany do postaci byte code'u powoduje nieznacznie wolniejsze wykonywanie się programów, jednak przy obecnych prędkościach procesorów jest to niemal niezauważalne. Dodatkowo JAVA jest językiem ciągle rozbudowywanym i mocno rozpowszechnionym. Posiada wiele bibliotek ze zoptymalizowanymi klasami, podobieństwo w składni do C# czyni go prostym w nauce a dobre narzędzia takie jak NetBeans czy Eclipse ułatwiają tworzenie aplikacji [10].

#### 3.3 Opis pliku mapy

Jak już wyżej zostało wspomniane mapa, na której realizowana jest symulacja przechowywana jest w osobnym pliku .xml (rys. 1), pomiędzy odpowiednimi znacznikami. Dzieli się ona na dwie sekcje:

```
<?xml version="1.0" encoding="UTF-8"?>
<map>
  <mapinfo>
    <mapsize x>50 </mapsize x>
    <mapsize y>40 </mapsize y>
    <numof finds>0 </numof finds>
    <numof lights>0 </numof lights>
    <numof levels>2 </numof levels>
  </mapinfo>
  <cell>
    <type>R </type>
    <pos>23,0,1 </pos>
    <paths>10,23:1:1 </paths>
    <lights>N </lights>
    <signs>N </signs>
    <speedlimit>2 </speedlimit>
    <ind>N </ind>
    <start>Y </start>
    <end>N </end>
  </cell>
```

Rys.1. Fragment pliku .xml opisującego mapę.

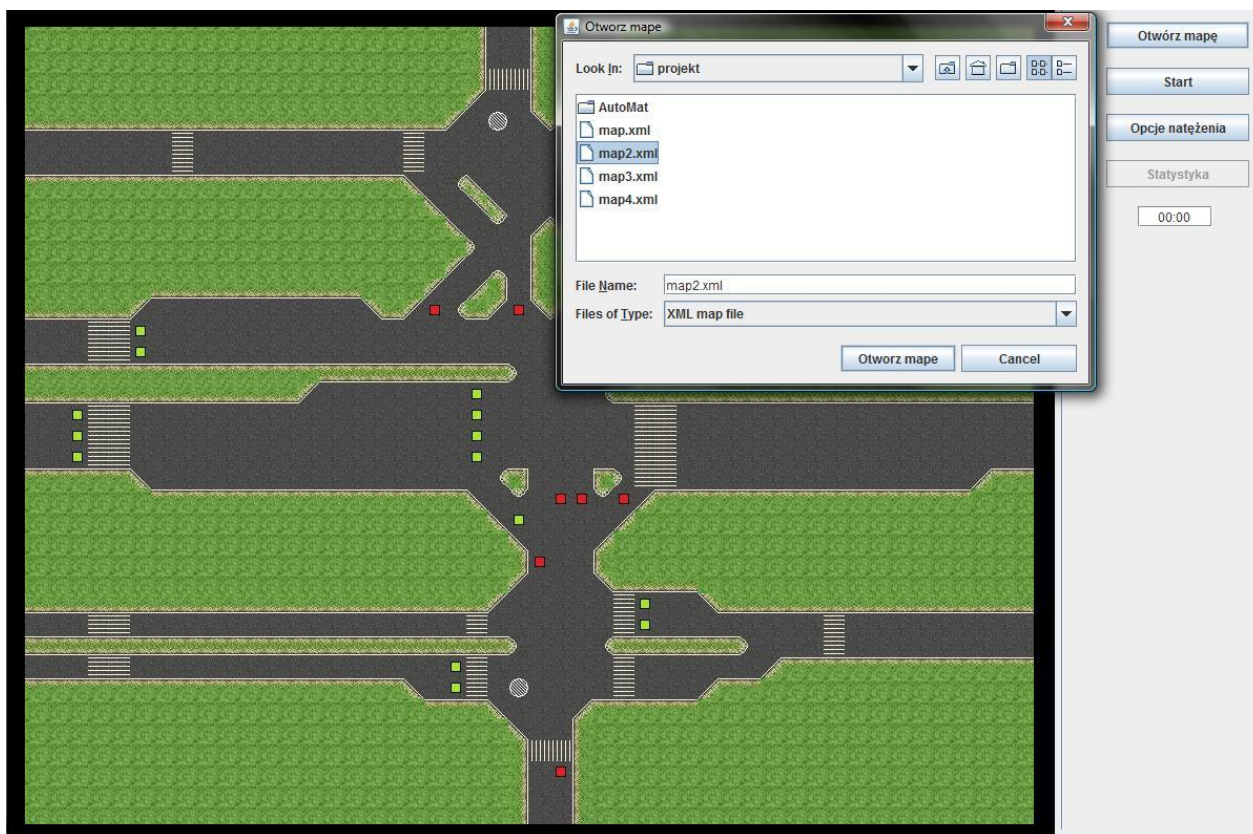
- określającą podstawowe parametry mapy, takie jak: liczba wierszy i kolumn, liczba poziomów, liczba pojedynczych sygnalizatorów świetlnych oraz liczba pętli indukcyjnych dla każdego ze świateł. Sekcja ta może wystąpić tylko jeden raz dla całej mapy;
- opisującą pojedynczą komórkę mapy. Występuje tyle razy, ile komórek, na których znajduje się droga, znajduje się na mapie. Zawiera w sobie określenie typu komórki, jej pozycję na mapie, informacje o ścieżkach przechodzących przez tę komórkę, czasach sygnalizatora (jeśli go zawiera), znaki z polami do sprawdzenia pod kątem zajętości przed zezwoleniem na ruch, ograniczenie prędkości, koordynaty pętli indukcyjnych (jeśli istnieją) dla przechowywanego sygnalizatora, określenie czy jest polem startowym oraz czy jest polem końcowym ścieżki.

### 3.4 Program główny

Program wykonuje się w głównej klasie odpowiadającej zarazem za interfejs użytkownika i logikę aplikacji. Klasa ta poprzez obiekt typu Timer uruchamia cyklicznie główną pętlę programu oraz obsługuje wywołania metod rysujących wybrane elementy. Po uruchomieniu symulatora na ekranie pojawi się główne okno, którego znaczną część zajmuje obszar roboczy. Na nim prezentowany jest węzeł i symulowany jest ruch drogowy. Po prawej stronie znajduje się panel sterowania symulatorem, który pozwala na wczytanie mapy, sterowanie symulacją oraz wyświetlaniem statystyk (rys. 2).

Ładowanie mapy odbywa się poprzez wybór odpowiedniego pliku z okna, które otworzy się po wciśnięciu przycisku oznaczonego „Otwórz mapę”. Należy wówczas wybrać jedną z dostępnych map (mapy mogą być generowane poprzez edytor map lub tworzone ręcznie poprzez założenie i edycję pliku .xml) i zatwierdzić wybór. Załadowana mapa pojawi się w obszarze roboczym symulatora. Mapa, będąca siatką automatu komórkowego, to kolekcja odpowiednich typów komórek, na których, według zasad w nich zawartych odbywa się ewolucja automatu. Przechowuje ona też obiekty świateł drogowych, punktów początkowych ścieżek i kolekcję samochodów. Ścieżki poruszania się pojazdów zawarte są w tablicy wewnątrz komórek o typie odpowiadającym aktywnej drodze. W takiej komórce każda ścieżka przez nią przebiegająca ma swój unikalny numer oraz koordynaty kolejnej komórki ścieżki. Ruch pojazdu odbywa się, więc na zasadzie przechodzenia odpowiedniej listy jednokierunkowej komórek siatki automatu komórkowego.

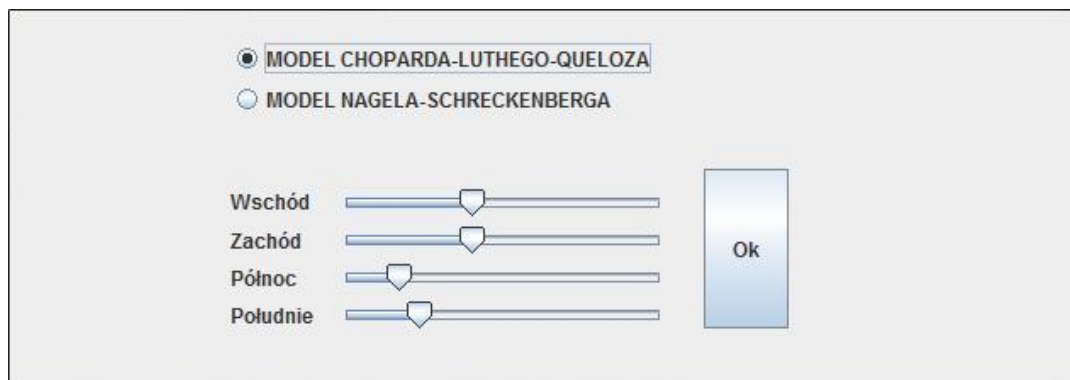
Każda komórka siatki automatu będąca komórką aktywnej drogi może zawierać w sobie sygnalizator świetlny. Jego definicja zawarta jest w strukturze komórki w pliku .xml w postaci ciągu znaków zawierającym numer sygnalizatora i czasy trwania każdego ze świateł. Wszystkie sygnalizatory rozpoczynają pracę od początku swojego cyklu a odpowiednie ustalenie czasów świecenia pozwala na konfigurację i synchronizację sygnalizacji świetlnej. Dodatkowo każdy sygnalizator może mieć zero lub więcej pętli indukcyjnych pozwalających na kontrolę zapalania się światła zielonego. Pętle te działają na zasadzie sprawdzania, czy w komórce, na której się znajdują zatrzymał się samochód. Jeśli tak, światło sygnalizatora jest zmieniane, aby zablokować możliwość wjazdu na skrzyżowanie. Pętle mogą się na siebie nakładać tj. wiele sygnalizatorów może być zależnych od pętli znajdujących się w tej samej komórce.



Rys.2. Główne okno programu wraz z wczytaną mapą oraz oknem wyboru mapy.



Po załadowaniu mapy otrzymujemy możliwość zmiany ustawień symulacji. Zmiana parametrów jest możliwa w każdym dowolnym momencie trwania symulacji. Za pomocą suwaków (rys. 3) można ustalać prawdopodobieństwo, z jakim będą pojawiać się pojazdy na każdym z brzegów mapy. Dodatkowo użytkownik ma też do wyboru możliwość zmiany modelu ruchu, według którego poruszają się pojazdy. Zmiany zostają wprowadzone bez konieczności zatrzymywania symulacji.

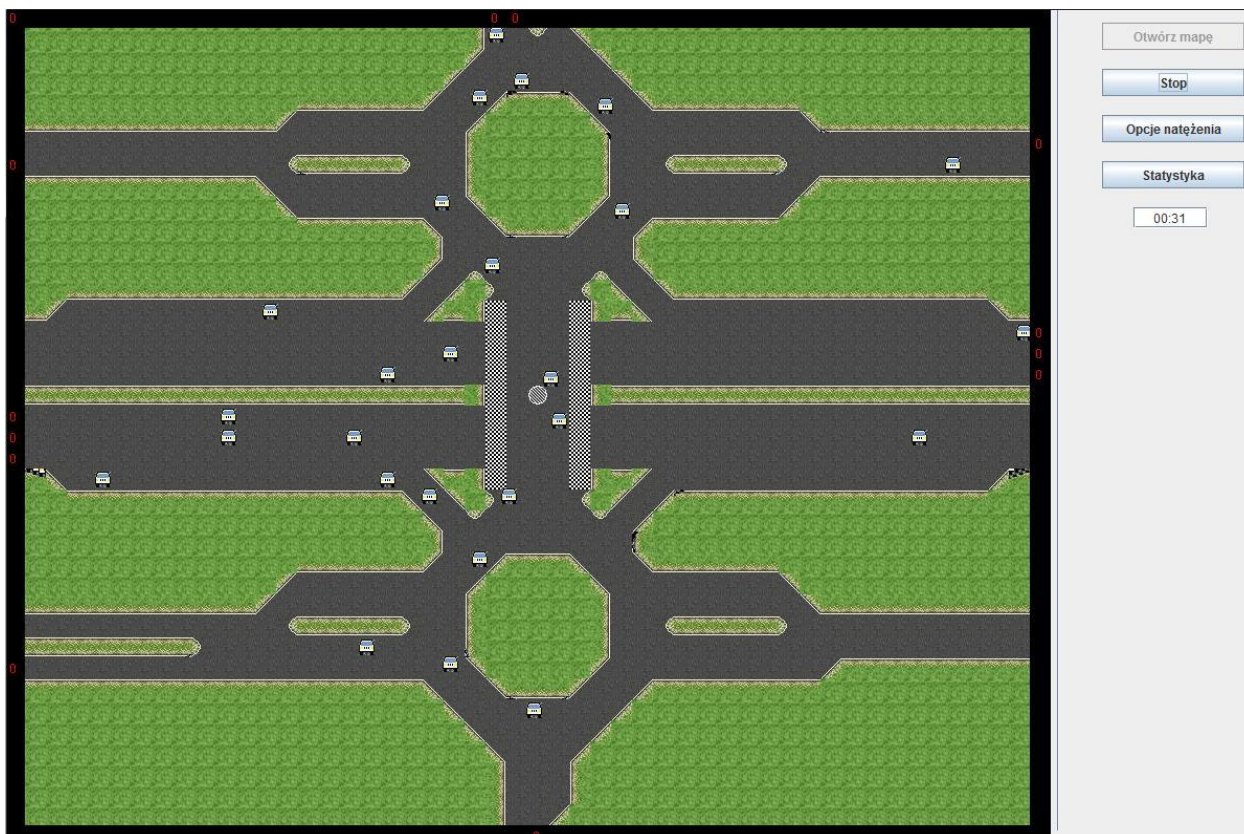


Rys.3. Okno zmiany parametrów symulacji.

Start symulacji skutkuje rozpoczęciem wykonywania się głównej pętli programu i zawartych w niej metod zgodnie z przyjętym modelem ruchu oraz wartościami kontrolującymi natężenia. Przekłada się to bezpośrednio na rozpoczęcie generowania pojazdów i ich ruch, po wyznaczonych na mapie ścieżkach. W trakcie symulacji użytkownik ma możliwość sprawdzania statystyk dla badanego obszaru. Program posiada też możliwość pomiaru kolejek pojazdów oczekujących na wjazd do symulowanego obszaru. Dzięki temu można symulować drogi dojazdowe o dowolnej długości w przypadku, gdy mapa nie zawiera wystarczająco dużo miejsca aby odwzorować odpowiednio długą ścieżkę dojazdową do węzła. W taki sposób zakolejkowane pojazdy, po otrzymaniu zezwolenia na ruch są tworzone i umieszczane na węźle tak, aby odwzorować płynny wjazd na skrzyżowanie kolumny pojazdów. Po uruchomieniu symulacji znika też możliwość załadowania nowej mapy do symulatora.

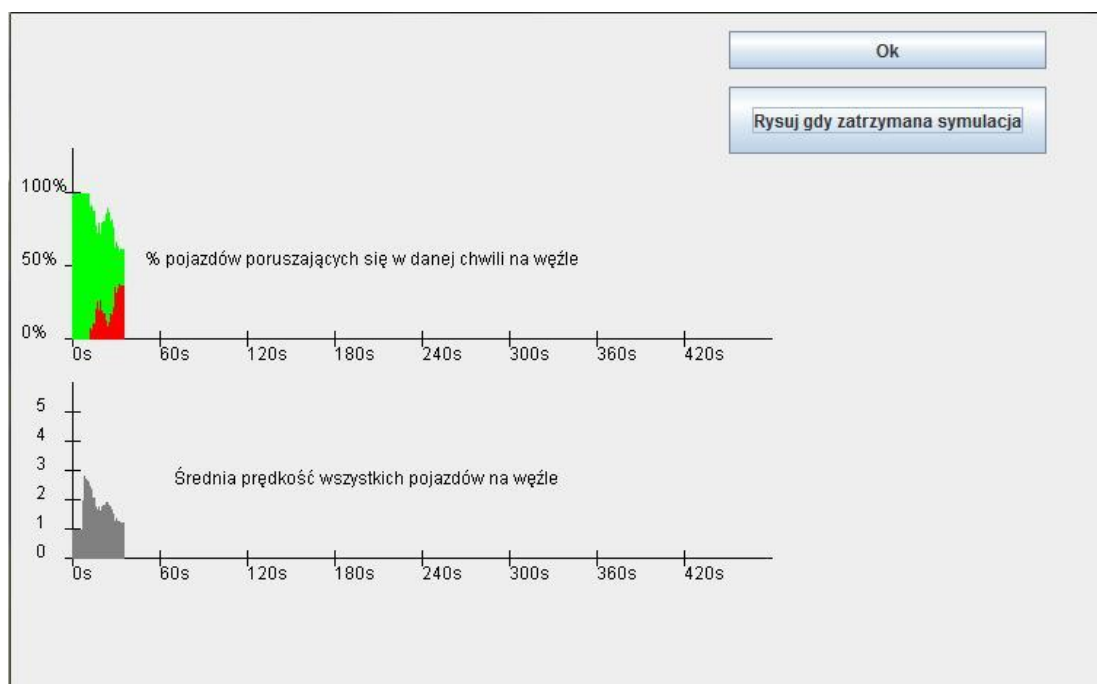


Rys.4. Symulacja ruchu na węźle jednopoziomowym.



Rys.5. Symulacja ruchu na węźle dwupoziomowym.

Aby lepiej zobrazować przepustowość ruchu w granicach badanego węzła komunikacyjnego, stworzony został obiekt zbierający dane z samochodów i prezentujący je na dwóch wykresach: jednym określającym procent samochodów stojących i procent samochodów poruszających się oraz drugim, określającym średnią prędkość pojazdu. Dane zbierane są co każdą iterację głównej pętli z wszystkich pojazdów znajdujących się na obszarze w danym momencie i w zależności od modelu prezentowane na wykresach o odpowiedniej skali (rys. 6).



Rys.6. Przykładowe statystyki dostępne w programie AutoMat.

Program AutoMat jest aktualnie rozbudowywany, projektowane są nowe funkcjonalności, które będą przedmiotem badań nad zachowaniami się kierowców w ruchu drogowym.

#### 4. WNIOSKI

Dostępne rozwiązania bazujące na teorii automatów komórkowych pozwalają na realizację tylko i wyłącznie prostych i opracowanych już wcześniej modeli ruchu drogowego (takich jak: model ruchu po drodze jednokierunkowej, skrzyżowanie, ruch okrężny itp.). W rzeczywistości jednak możemy spotkać dodatkowe elementy, które nie są uwzględnione w istniejących rozwiązaniach jak np. wielopoziomowe skrzyżowania czy pętle indukcyjne sterujące sygnalizacją.

Opracowana aplikacja rozwiązuje te potrzeby, uwzględniając jednocześnie aspekty zawarte w istniejących rozwiązaniach. Do zalet omawianego systemu należy symulacja ruchu na wielopoziomowych mapach, co pozwala na symulowanie nowoczesnych rozjazdów i skomplikowanych węzłów drogowych głównych arterii miejskich. Sygnalizację świetlną można wzbogacić o dodatkowe elementy sterujące a sposób wykonania pozwala na szybką rozbudowę o dodatkowe elementy.

#### 5. BIBLIOGRAFIA

- [1] Bartodziej M.: *Modelowanie ruchu ulicznego za pomocą automatów komórkowych*, Politechnika Wroclawska, 2007.
- [2] Chopard B., Luthi P., Quelo P-A.: *Cellular automata model of car traffic in a two-dimensional street network*, Journal of Physics, 1996.
- [3] Eck D.J.: *Introduction to One-dimensional cellular automata*, Department of Mathematics and Computer Science, Hobart and William Smith Colleges, <http://math.hws.edu/xJava/CA/CA.html>.
- [4] Kułakowski K.: *Automaty komórkowe*, Akademia-Górnictwo-Hutnicza, 2000.
- [5] Malarz K.: *Automaty komórkowe*, Akademia-Górnictwo-Hutnicza, 2008.
- [6] Małecki K., Rokita M. Wątróbski J.: *Wykorzystanie automatów komórkowych w modelowaniu ruchu drogowego*, Polska, Pomiary Automatyka Kontrola, Nr 7, 2010.
- [7] Nagel K., Schreckenberg M.: *A cellular automaton model for freeway traffic*, Journal of Physics, 1992.
- [8] Wolfram S.: *A new kind of science*, Wolfram Media, 2002.
- [9] <http://www.traffic-simulation.de/>.
- [10] <http://javastart.pl/wprowadzenie/dlaczego-java/>.
- [11] Chowdhury D., Schadschneider A.: *Self-organization of traffic jams in cities: effects of stochastic dynamics and signal periods*, Physical Review E 59, pp. 1311-1314, 1999
- [12] <http://sjsu.rudyrucker.com/~han.jiang/paper/>
- [13] Dascalu M., Goschin S.: *Using cellular automata in traffic modeling*, University of Bucharest, 2002.