

# AKADEMIA GÓRNICZO-HUTNICZA

Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki



KATEDRA INFORMATYKI

## Programowanie systemów agentowych i wielorobotowych

Kraksim – wielopasowość i predyk-  
cja ruchu

Dokumentacja do projektu

**Autorzy:**

Maciej Zalewski ([mzal@student.agh.edu.pl](mailto:mzal@student.agh.edu.pl))  
Łukasz Dziewoński ([dziewon@student.agh.edu.pl](mailto:dziewon@student.agh.edu.pl))

**Wersje:**

Wersja	Data	Opis
1.0	22.06.2008	Wersja wstępna dokumentacji
1.1	13.09.2008	Wersja uwzględniająca oba semestry, z uwzględnieniem uwag do poprzedniej wersji (bez testów predykcji)
1.2	15.09.2008	Wersja 1.1 rozszerzona o testy predykcji
1.3	01.10.2008	Wersja 1.2 uzupełniona o testy krawędziowe
1.4	02.10.2008	Ostateczne poprawki, wnioski

1. System Kraksim i wprowadzone zmiany.....	3
2. Rozbudowa modelu o wielopasowość.....	4
2.1. Zmiany w architekturze systemu.....	5
2.2. Zmiany w modelu.....	6
2.3. Równoważenie obciążenia.....	7
3. Rozbudowa systemu o predykcję ruchu.....	8
3.1. Etapy działania mechanizmu predykcji.....	9
3.2. Sposób realizacji rozwiązania.....	10
3.2.1. Podstawowe pojęcia.....	10
3.2.2. Sposób przechowywania zagregowanych danych.....	13
3.2.3. Algorytm agregacji.....	14
3.2.4. Algorytm wyszukiwania wzorców.....	14
3.2.5. Zastosowanie wzorców do bieżącego stanu modelu.....	15
3.3. Konfiguracja mechanizmu predykcji.....	16
3.3.1. Przykładowa konfiguracja.....	16
3.3.2. Konfiguracja algorytmu predykcji.....	16
3.3.3. Konfiguracja sposobu dyskretyzacji.....	17
3.3.4. Domyślne ustawienia sposobu dyskretyzacji.....	18
4. Zmiany w architekturze systemu Kraksim.....	19
4.1. Zmiany architektoniczne systemu Kraksim związane z wielopasowością.....	19
4.2. Zmiany architektoniczne systemu Kraksim związane z predykcją.....	23
4.3. Zmiany w interfejsie użytkownika i mechanizmie uruchamiania.....	28
4.4. Opis całościowy nowej architektury.....	29
5. Testy.....	30
5.1. Testy wielopasowości.....	30
5.2. Testy mechanizmu predykcji.....	34
5.2.1. Sposób testowania.....	34
5.2.2. Rezultaty testów – model D.....	36
5.2.3. Model Krakowa.....	47
5.2.4. Wnioski.....	58
6. Podsumowanie.....	59
6.1. Propozycje dalszych zmian.....	59
7. Załączniki.....	60
7.1. Załącznik 1: prediction.xml.....	61
7.2. Załącznik 2: config.properties.....	62
7.3. Załącznik 3: model-D.xml.....	63
7.4. Załącznik 4: traffic-D-med.xml.....	68
7.5. Załącznik 5: model-krakow.xml.....	69
7.6. Załącznik 6: traffic-krakow_med.xml.....	99

# 1. System Kraksim i wprowadzone zmiany

System Kraksim jest systemem agentowym służącym do symulacji ruchu miejskiego. Symulacja ruchu odbywa się przy wykorzystaniu automatów komórkowych. Sterowanie ruchem poszczególnych pojazdów odbywa się poprzez niezależne agenty, analizujące posiadane dane na temat stanu modelu. Sam model ma postać grafu, którego wierzchołki stanowią skrzyżowania i bramy (punkty wejścia do modelu), zaś krawędziami są ulice łączące poszczególne skrzyżowania i bramy. Model ruchu na skrzyżowaniach jest nieskomplikowany – każdemu ze skrzyżowań przypisane są fazy, które kolejno się zmieniają. Dla każdej fazy zdefiniowane są te ulice wchodzące do skrzyżowania, z których można przejechać przez skrzyżowanie bez groźby kolizji z innymi pojazdami. W tym sensie skrzyżowania w systemie Kraksim są bezkolizyjne. Za zmianę faz skrzyżowań również odpowiedzialne są autonomiczne agenty. W systemie Kraksim zaimplementowane zostały zarówno statyczne jak i dynamiczne mechanizmy zmiany fazy światła na skrzyżowaniach – pierwsze z nich zakładają zmiany fazy w określonych odstępach czasowych, niezależnie od natężeń ruchu, drugie natomiast analizują natężenia na ulicach dochodzących do skrzyżowania i dostosowują długość trwania poszczególnych faz do aktualnego zapotrzebowania.

Wersja systemu Kraksim, jaką otrzymaliśmy na początku roku posiadała model, który był zdecydowanie uproszczony w stosunku do jakichkolwiek rzeczywistych sieci drogowych – uniemożliwiał tworzenie sieci drogowych, w których ulice składałyby się z wielu pasów. Właśnie wielopasowość była elementem, który został przez nas dodany do systemu Kraksim. Okazało się to zadaniem dość złożonym, gdyż architektura wewnętrzna systemu Kraksim nie wspierała tego typu możliwości. Zmiany wprowadzone w architekturze zostały bardziej szczegółowo opisane w kolejnym rozdziale. Dzięki wprowadzeniu wielopasowości model sieci wykorzystany przez system Kraksim stał się o wiele bardziej złożony, a równocześnie o wiele bardziej bliski rzeczywistości.

Celem drugiej zmiany było zrealizowanie mechanizmu służącego do przewidywania korków i ostrzegania przed nimi. Wprowadzenie takiej zmiany umożliwiłoby symulację mechanizmów służących do zadań optymalizacji ruchu drogowego, zarazem dając możliwość implementacji różnych algorytmów i porównywania ich skuteczności dla różnych modeli i natężeń ruchu. Zaprojektowane rozwiązanie architektoniczne praktycznie nie ingeruje w system Kraksim, jest natomiast łatwe w rozbudowie i umożliwia prosta podmianę sposobu realizacji predykcji. Wybrana realizacja agreguje dane na temat charakterystyki ruchu w kolejnych okresach czasu, a następnie na ich podstawie przewiduje dalsze zachowania modelu i kierunki przepływu ruchu.

## 2. Rozbudowa modelu o wielopasowość

Podstawowym założeniem wprowadzanej zmiany było umożliwienie modelowania ruchu po ulicach wielopasowych. Dotychczas zrealizowanej wersji możliwe było jedynie wprowadzenie kilku pasów służących do skrętu, jednak z założeń architektonicznych wynikało, iż każdy musiał umożliwiać skręt w inną ulicę. Było to dość silne ograniczenie modelu.

Poza podstawową wersją, otrzymaliśmy również wersję, w której wprowadzono ograniczoną wielopasowość poprzez przedłużenie pasów bocznych na całą długość ulicy, przez co samochody skręcające w wybranym kierunku nie zajmowały pasa głównego, przeznaczonego do jazdy prosto. Zmiana ta jednak nie umożliwiała dwóm pojazdom zmierzającym do tego samego celu przemieszczania się równolegle po jednej ulicy.

W istniejących wersjach systemu Kraksim, wybór trasy odbywał się w zasadzie wyłącznie na poziomie ulic i skrzyżowań. Ze względu na ograniczenie, iż na jednej ulicy może znajdować się tylko jeden pas dla każdej z ulic docelowych, zmiany pasa ruchu na odbywały się natychmiast, gdy tylko nastąpiła taka możliwość i nie wiązały się w żaden sposób z mechanizmami równoważenia obciążenia na trasie.

Wprowadzona zmiana usunęła powyższe ograniczenie, wiązało się to jednak z szerszymi zmianami w systemie Kraksim. Były nimi:

- zbudowanie zależności kolejności pomiędzy pasami w obrębie jednej ulicy,
- rozbudowa mechanizmu akcji dostępnych dla pasa o możliwość wielu pasów prowadzących do jednego celu,
- rozbudowa algorytmu umieszczającego pojazdy na pasach tak, aby obciążenie było zrównoważone,
- rozbudowa elementu prezentującego model miasta o możliwość wyświetlania ulic wielopasowych,
- rozbudowa parsera konfiguracji o elementy wielopasowe.

## 2.1. Zmiany w architekturze systemu

Podstawową zmianą, jaka dokonała się w architekturze systemu Kraksim było odejście od założenia o jedyności akcji prowadzącej do wskazanej ulicy w obrębie całej ulicy. Koniecznym okazało się umożliwienie dodania właściwych, różnych akcji dla poszczególnych pasów (ponieważ akcja zawiera także informację o pasie, z którego może zostać wykonana), a także rozbudowa mechanizmu pobierania dostępnych akcji prowadzących do wskazanej ulicy – w tym momencie zamiast pojedynczego pasa, otrzymywana jest lista pasów. Siłą rzeczy, pociągnęło to za sobą konieczność wyboru najlepszej z pośród dostępnych listy akcji. W tym celu została stworzona osobna klasa, która wspomaga wybór. Dodatkową zaletą takiego rozwiązania jest fakt, że w razie zmiany koncepcji dotyczącej sposobu optymalizacji zrównoważenia ruchu bądź strategii obieranej przez pojazdy, zmiany dotkną wyłącznie nowo utworzoną klasę odpowiedzialną za wybór akcji.

Drugą ważną zmianą było wykorzystanie również tej klasy w celu doboru najlepszego pasa przy umieszczaniu pojazdów na ulicy. Konieczność ta wiązała się głównie z faktem, że w przypadku węzłów końcowych (gateway) nie istnieją żadne akcje, więc wybór w oparciu o akcje byłby nieskuteczny.

Ostatnią z istotnych zmian było wprowadzenie dodatkowego założenia – zgodnego z przepisami ruchu drogowego w Polsce – że pierwszeństwo ma pojazd poruszający się po najbardziej prawym pasie. W związku z tym w przypadku gdy więcej niż jeden pas jednej ulicy umożliwiał akcję prowadzącą do wskazanej ulicy, hierarchia kolejności pomiędzy nimi daje pierwszeństwo pasom poczynając od najbardziej prawego.

## 2.2. Zmiany w modelu

Zmiany w modelu dotyczą dwóch elementów:

- formatu pliku opisującego model,
- sposobu prezentacji ulic.

Jeśli chodzi o pierwszy z powyższych elementów, zmiana dodaje opcjonalną właściwość ***numberOfLanes*** do węzła ***main*** w pliku xml. Fragment przykładowego pliku konfiguracyjnego znajduje się poniżej.

```
<RoadNet>
  <nodes>
    <gateway id="Nroad" x="10" y="10" />
    <intersection id="X1" x="10" y="160" />
  </nodes>

  <roads>
    <road
      id="NroadX1"
      street="ulica Pionowa"
      from="Nroad"
      to="X1">

      <uplink>
        <left length="40" />
        <main length="50" numberOfLanes="3"/>
      </uplink>

      <downlink>
        <main length="50"/>
      </downlink>

    </road>
  </roads>
</RoadNet>
```

Jak widać, właściwość ta jest opcjonalna, dzięki czemu nowa konfiguracja zachowuje wsteczną kompatybilność z modelami bez wielopasowości.

Jeśli chodzi o sposób reprezentacji ulic, konieczne było oczywiście dodanie do algorytmu rysującego poszczególne ulice pętli do rysowania wszystkich głównych pasów zamiast wywołania rysującego pojedynczy główny pas.

## 2.3. Równoważenie obciążenia

Wprowadzenie wielu pasów do modelu nie miałoby sensu, gdyby nie następowało proste równoważenie obciążenia pomiędzy pasami. Zostało ono zrealizowane w wyodrębnionej klasie. Opiera się na zliczaniu ilości samochodów znajdujących się na poszczególnych pasach oraz odległości najbliższego samochodu od wjazdu ze skrzyżowania.

Pierwszeństwo wprowadzania samochodów na pasy jest następujące:

- pas z najmniejszą ilością samochodów
- wśród pasów z równą ilością samochodów – pas, na którym jest największa odległość do kolejnego samochodu
- jeśli powyższe kryteria nie pomogą – pas najbardziej po prawej.

Jest to rozwiązanie zgodne z logiką oraz przepisami. Możliwe jest jednak łatwe zmodyfikowanie sposobu wybierania najlepszego pasa poprzez modyfikację jednej tylko klasy.

### 3. Rozbudowa systemu o predykcję ruchu

Podstawową kwestią w każdym systemie symulującym ruch miejski jest moment wyboru trasy, po jakiej poruszać się będą pojazdy. Najprostszym z pomysłów jest taki, aby każdy pojazd wjeżdżający w sieć ulic ustalał trasę, jaką zamierza przejechać. Tak też zrealizowane to było w systemie Kraksim. Ponadto, w celu zwiększenia realizmu tego modelu, z określonym prawdopodobieństwem kierowca mógł ponownie wybrać trasę. W obu przypadkach wykorzystywany był algorytm Dijkstry, wagami krawędzi grafu były średnie czasy przejazdu przez daną krawędź. Czasy te, zbierane na bieżąco przez moduł statystyczny, dawały możliwość dostosowania wybieranej drogi do aktualnego stanu zatłoczenia poszczególnych krawędzi grafu połączeń.

Zrealizowany mechanizm predykcji opiera się na wyszukiwaniu wzorców wśród sekwencji następujących po sobie stanów systemu. Metody oparte na wzorcach cechują się przede wszystkim tym, że w początkowej swej fazie wymagają zebrania odpowiedniej ilości danych, w których można rozpocząć wyszukiwanie wzorców, więc zaczynają działać faktycznie z pewnym opóźnieniem w stosunku do całego systemu. Następnie dopiero rozpoczyna się korzystanie ze znalezionych wzorców i stosowanie ich do aktualnej sytuacji.

W dalszej części rozdziału opisana jest bardziej szczegółowo architektura i realizacja mechanizmu predykcji, który został wprowadzony do systemu Kraksim.



### 3.1. Etapy działania mechanizmu predykcji

Jak już zostało wspomniane, mechanizm predykcji działa w kilku kolejnych etapach:

- zbieranie danych na temat zachowania modelu (jego stanów)
- wyszukiwanie wzorców predykcyjnych w zebranych danych
- zastosowanie wzorców do bieżącego stanu modelu

Faktycznie, pierwszy z etapów rozciągnięty jest na cały czas działania systemu, pozostałe natomiast włączają się stopniowo.

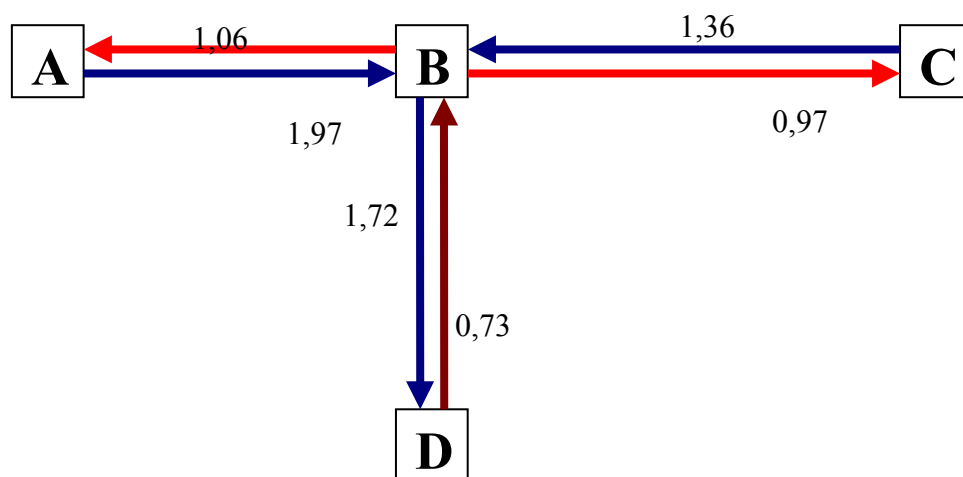
## 3.2. Sposób realizacji rozwiązania

Aby uniknąć konieczności dołączania do aplikacji symulacyjnej dodatkowej bazy danych oraz mechanizmów związanych z komunikacją z nią – co znacząco opóźniłoby prowadzenie samej symulacji – zdecydowaliśmy się na stworzenie mechanizmu agregowania zachodzących relacji. Mechanizm ten oczywiście także spowalnia nieco działanie aplikacji, jest jednak rozwiązaniem zdecydowanie lepszym.

Pomysł opiera się na stworzeniu wielowymiarowej macierzy rzadkiej przechowującej liczniki wystąpień poszczególnych zdarzeń. W ten sposób cała historia symulacji nie musi być przechowywana – zapisywane są jedynie ilości poszczególnych sekwencji stanów. Aby wyjaśnić, czym jest wzorzec, który jest wyszukiwany, konieczne jest wprowadzenie paru pojęć.

### 3.2.1. Podstawowe pojęcia

**Graf stanu modelu** jest to graf ważony skierowany, którego wierzchołkami są wjazdy i skrzyżowania, a krawędziami połączenia między nimi. Wagą krawędzi jest aktualny średni czas przejazdu krawędzią.



Rysunek 1: przykładowy graf stanu modelu

**Stan świata** jest to reprezentacja jednego grafu stanu modelu w postaci uporządkowanej sekwencji wartości wag poszczególnych jego krawędzi.

Powyżej przedstawiono przykładowy graf stanu modelu. Zakładając, że uporządkowanie odbywałoby się w kolejności: AB, BA, BC, CB, BD, DB stan świata wyglądałby następująco:

1,97	1,06	0,97	1,36	0,73	1,72
------	------	------	------	------	------

Tabela 1: Przykładowy stan świata

**Historia świata** jest to szereg ułożonych kolejno stanów świata.

Jednym z parametrów algorytmu predykcji jest ilość kroków czasowych, na jakie ma wpływ aktualny stan świata. Należy przy tym pamiętać, że te kroki czasowe nie są tożsame z krokami symulacji – mechanizm dynamicznego routingu definiuje ilość kroków symulacji przypadających na jeden tak rozumiany krok czasowy.

Przykładowa historia świata wyglądałaby więc następująco:

	AB	BA	BC	CB	BD	DB
$t_{i+2}$	1,97	1,77	1,48	1,63	0,73	1,79
$t_{i+1}$	1,97	1,51	1,17	1,36	0,73	1,75
$t_i$	1,97	1,16	0,97	1,36	0,73	1,72

Tabela 2: przykładowa historia świata

**Wysokość historii (h)** jest to liczba stanów świata, na które ma wpływ aktualny stan świata. W przedstawionej powyżej przykładowej historii parametr h jest równy 2 – zakładamy, że aktualny stan świata w czasie  $t_i$  ma wpływ na stan świata w punktach czasowych od  $t_{i+1}$  do  $t_{i+h}$ .

**Sąsiedztwo** wskazanej krawędzi jest to zbiór wszystkich krawędzi oddalonych od wskazanej krawędzi nie dalej niż o określoną ilość skrzyżowań (czyli takie, do których można się dostać ze wskazanej krawędzi przejeżdżając nie więcej niż określoną ilość skrzyżowań)

**Szerokość sąsiedztwa** - oznaczane jako r – jest to parametr algorytmu określający, na ile sąsiednich krawędzi ma wpływ stan ruchu we wskazanej krawędzi. Na przykładowym grafie (Rysunek 1) sąsiadami krawędzi DB (kolor brązowy) o szerokości sąsiedztwa  $r = 1$  są oznaczone na czerwono krawędzie BA oraz BC. Zbyt wąskie sąsiedztwo może skutkować niewielkim wpływem mechanizmu predycyjnego, zbyt duża jego wartość natomiast może doprowadzić do wyraźnego spowolnienia działania aplikacji, jak również do powstawania błędnych wzorców.

**Następstwem** wskazanej wartości w danej krawędzi jest zbiór sąsiedztw w kolejnych krokach czasowych. W Tabeli 2 następstwo krawędzi DB (kolor brązowy) zaznaczone jest na kolor czerwony (dla parametrów  $r = 1$ ,  $h = 2$ )

**Wysokość następstwa** – oznaczane jako h – jest to parametr algorytmu określająca, jaka liczba kolejnych sąsiedztw jest liczona jako następstwo danego stanu krawędzi. Zbyt niska wartość tego parametru może skutkować bardzo krótkotrwałym wpływem mechanizmu predykcji, a nawet prowadzić do koncentracji korków. Zbyt duża jego wartość z kolei wyraźnie wydłuża wyszukiwanie wzorców oraz ich stosowanie, może także skutkować odnajdowaniem błędnych wzorców.

**Stan krawędzi** jest to pierwotnie czas przejazdu daną krawędzią – tym jest stan krawędzi przechowywany przez system Kraksim i służący do wyliczania za pomocą algorytmu Dijkstry optymalnej ścieżki przejazdu przez model. Jednakże w celu sensownej agregacji danych, konieczna jest dyskretyzacja wartości liczbowych do kilku kategorii. Dyskretyzacja odbywa się w następujący sposób:

- Dla każdej krawędzi obliczany jest minimalny czas przejazdu (długość krawędzi podzielony przez maksymalną prędkość)
- Z pliku konfiguracyjnego pobierana jest lista możliwych stanów definiowana poprzez górną oraz dolną granicę wartości. Przykładowo konfiguracja:

```
<level
  description="DENSE"
  lowerBound="1.8"
  upperBound="2.5"
  influence="2"
  prevDescription="SMOOTH"
  nextDescription="STUCK">
</level>
```

opisuje poziom którego dolną granicę stanowi 1.8 wartości minimalnej, górną zaś 2.5 wartości minimalnej. Jeśli więc średni czas przejazdu jakimś odcinkiem wynosi 5 jednostek, stan jej jest uznawany za gęsty (DENSE) dla średnich czasów przejazdu od 9 do 12.5 jednostki (od  $1.8 \cdot 5$  do  $2.5 \cdot 5$ )

- Każda z wartości liczbowych zamieniana jest na jeden z tak zdefiniowanych stanów

**Przyczyna** to para (nazwa krawędzi, stan krawędzi), dla której tworzony jest sekwencja. Przykładowo, jeśli w przykładzie pokazanym w Tabeli 2 przyjąć dla krawędzi DB średni czas przejazdu jako 1, a dyskretyzację taką, jak przytoczono powyżej, to przyczyną w tym przykładzie byłaby między innymi oznaczona kolorem brązowym para (DB, SMOOTH)

**Następnik wzorca** to para (nazwa krawędzi, ilość kroków od przyczyny). Przykładowo w Tabeli 2 zaznaczono kolorem czerwonym 4 następni:

- (BA, 1)
- (BA, 2)
- (BC, 1)
- (BC, 2)

**Sekwencja** jest to para (przyczyna, następnik). Z każdą sekwencją związany jest Agregat stanów (licznik wystąpień poszczególnych stanów dla danej sekwencji). Przykładowo sekwencja ( (DB, SMOOTH), (BA, 1) ) służy do przechowania informacji o tym, ile razy po stanie SMOOTH w krawędzi DB, w krawędzi BA po 1 kroku czasowym wystąpiły poszczególne stany. Przykład takiego licznika znajduje się poniżej:

	EMPTY	SMOOTH	DENSE	STUCK
Wystąpienie	1	2	16	1
Procent	5%	10%	80%	5%

### Tabela 3: Przykładowy agregat stanów dla sekwencji

W powyższym przypadku, rezultatem stanu SMOOTH w krawędzi DB był w kolejnym kroku czasowym w krawędzi BA w 80% wystąpił stan DENSE.

**Wzorzec** jest parą (sekwencja, stan wynikowy krawędzi), w powyższym przypadku wzorcem mogłaby być para ( ( **DB, SMOOTH**), (**BA, 1**) ), DENSE ). Aby jednak uznać taką parę za wzorzec, musi ona spełniać dwa kryteria:

- Procent przypadków potwierdzających wzorzec musi być wyższy od zadanej wartości progowej (aby odrzucić nieprawdopodobne zdarzenia)
- Ilość przypadków potwierdzających wzorzec musi być wyższa od zadanej ilości minimalnej (aby usunąć zakłócenia powstające na początku pracy mechanizmu predykcji)

Nietrudno wyobrazić sobie sytuację, gdy agregat stanów będzie zawierał tylko jedno pole, z licznoscią 1 – wówczas samo kryterium procentowe nie wystarcza.

**Ufność** wzorca to procent przypadków potwierdzających wzorzec. Aby wzorzec został uznany za prawdziwy, jego ufność musi przekroczyć wartość minimalną, zadaną jako parametr algorytmu. Oczywiście nadanie temu parametrowi zbyt niskiej wartości może skutkować akceptowaniem pewnej ilości fałszywych wzorców, zaś zbyt wysoka jego wartość przyczyni się do zniwelowania działania mechanizmu predykcji

**Minimalna ilość zdarzeń potwierdzających** jest to współczynnik algorytmu odpowiadający za to, z jakim opóźnieniem algorytm rozpoczyna działanie. Może się bowiem zdarzyć, że w początkowych momentach działania algorytmu, przy niewielkiej ilości zagregowanych obserwacji (zanim algorytm „zdąży” nauczyć się wzorców) – przykładowo dla pojedynczego wystąpienia – ufność osiągnie poziom 100%, wzorzec zaś będzie fałszywy. Stąd wprowadzony został parametr określający, jaka minimalna ilość obserwacji musi potwierdzać dany wzorzec, by został on uznany i zastosowany. Oczywiście zbyt wysoka wartość tego współczynnika wyraźnie opóźni działanie algorytmu.

#### 3.2.2. Sposób przechowywania zagregowanych danych

Powyższe definicje powinny dać pewien obraz przechowywanych danych. Ze względu na to, że przechowywane dane stanowią macierz rzadką, zdecydowaliśmy się wykorzystać do ich przechowywania tablice haszujące zaimplementowane w języku Java. Jako klucz posłużył łańcuch znaków składający się kolejno z następujących elementów:

- Identyfikator krawędzi „przyczynowej”
- Stan krawędzi „przyczynowej”
- Identyfikator krawędzi „skutkowej”
- Ilość kroków czasowych pomiędzy „przyczyną” a „skutkiem”

Widać więc wyraźnie, że jest to **Sekwencja** (zgodnie z definicją w poprzednim podrozdziale). Dla każdego takiego klucza wartością jest klasa przechowująca agregat stanów.

Zaletami takiego rozwiązania są:

- Istnienie (dobrej) implementacji w języku Java

- Stosunkowo niewielki rozmiar przechowywanych danych (brak konieczności przechowywania danych „pustych”)
- Intuicyjny sposób wyszukiwania (łatwy w ewentualnym debugowaniu), oparty o łańcuchy znakowe, które są czytelne dla człowieka

### 3.2.3. Algorytm agregacji

Znając sposób przechowywania, łatwo domyślić się, w jaki sposób działa mechanizm agregacji – otrzymując co określony krok czasowy kolejny stan świata, dopisuje go do swojej historii. Na tym etapie stany poszczególnych krawędzi są dyskretyzowane.

Historia nie jest przechowywana ciągle – jedynie ostatnie  $h+1$  wpisów, które są kolejno zastępowane nowymi. Dzięki temu wzrost czasu działania algorytmu nie zwiększa zużycia zasobów przez niego zajmowanych (jedynie rozmiar macierzy agregatów może się zwiększać – ale to dzieje się głównie w początkowej fazie działania algorytmu, szybko osiągając stan nasycenia).

Od momentu, w którym wpisów w historii jest przynajmniej  $h+1$ , przeprowadzana jest analiza – dla najstarszego stanu w historii, dla każdej przyczyny (para (krawędź, stan) ), a następnie dla każdego z elementów jej następstwa budowana jest sekwencja, z której powstaje łańcuch znakowy – klucz tablicy haszującej. Jeżeli pod wskazanym kluczem znajduje się już agregat stanów, liczniki dla odpowiednich wartości są inkrementowane, jeśli nie, tworzony jest nowy agregat.

Po dokonaniu agregacji uruchamiany jest mechanizm wyszukiwania wzorców.

### 3.2.4. Algorytm wyszukiwania wzorców

Aby agregowane dane mogły być użyte, konieczne jest wyszukanie wśród nich wzorców określających następstwa stanów.

W tym celu pobierany jest aktualny stan świata. Dla każdej z krawędzi (przyczyn) i przeszukiwane jest jej następstwo. Dla każdego z elementów następstwa, dla którego istnieje agregat stanów, znajdowany jest element najczęściej występujący.

Określane jest następnie prawdopodobieństwo jego wystąpienia w obrębie agregatu (odpowiada współczynnikowi ufności w algorytmach wyszukiwania asocjacji używanych w procesach Data Mining), poprzez obliczenie ilości jego wystąpień do łącznej ilości wystąpień wszystkich elementów. Rezultat ten jest porównywany z minimalnym progiem ufności, powyżej którego wzorec zaczyna być traktowany jako prawdziwy. Ponadto sprawdzane jest, czy ilość przypadków potwierdzających przekracza wartość minimalną.

**Przykład:**

<i>Stan</i>	EMPTY	SMOOTH	DENSE	STUCK
<i>Licz- ność</i>	3	5	52	0

Tabela 4: Przykład znajdowania wzorców

W powyższym przykładzie najczęściej występującym następstwem jest stan DENSE. Jego ufność to  $52/(3+5+52)$ , czyli ok. 87%. Jeśli za minimalny poziom ufności przyjęłoby np. 80%, wzorec ten zostałby zaakceptowany, jeśli poziom ten byłby ustawiony na 90%, wynik ten nie zostałby uznany za wzorec.

Dodatkowo zaimplementowany został mechanizm starzenia się wzorców. Polega on na tym, że po każdym kroku czasowym wartości we wszystkich istniejących agregatach mnożone są przez współczynnik starzenia – liczbę z przedziału [0, 1]. Dzięki temu jeżeli ruch zmienia się w trakcie symulacji, stare wzorce po pewnym czasie zanikną, dając miejsce nowym. Oczywiście, nie stanie się to natychmiast, szybkość wygasania będzie zależna od współczynnika starzenia.

### 3.2.5. Zastosowanie wzorców do bieżącego stanu modelu

Mechanizm predykcyjny integruje się z mechanizmem dynamicznego routingu zaimplementowanym w systemie Kraksim. Mechanizm dynamicznego routingu bazuje na aktualnych wartościach wag krawędzi grafu stanu modelu. Mechanizm predykcji zmienia te wagi w oparciu o znalezione wzorce. Każdemu z poziomów natężenia ruchu przyporządkowany jest współczynnik wpływu. Współczynnik ten określa mnożnik, jaki powinien być zastosowany w odniesieniu do zmienianego połączenia jeśli rezultatem wzorca jest wskazany poziom ruchu.

Przykładowo, jeśli współczynnik wpływu dla natężenia ruchu DENSE wynosi 2, oznacza to, że waga krawędzi grafu może zostać zwiększona nawet dwukrotnie. Dodatkowo, tu także uwzględniana jest ufność wzorca – wpływ zmniejsza się liniowo od 100% współczynnika wpływu dla wartości ufności równej 100% do 0% dla współczynnika ufności znajdującego się na granicy dopuszczalności.

Wzór na nową wagę krawędzi jest następujący:

$$V_{\text{new}} = V_{\text{old}} \cdot (1 + I_{\text{max}} \cdot (T - T_{\text{min}}) / (1 - T_{\text{min}})) ,$$

gdzie:

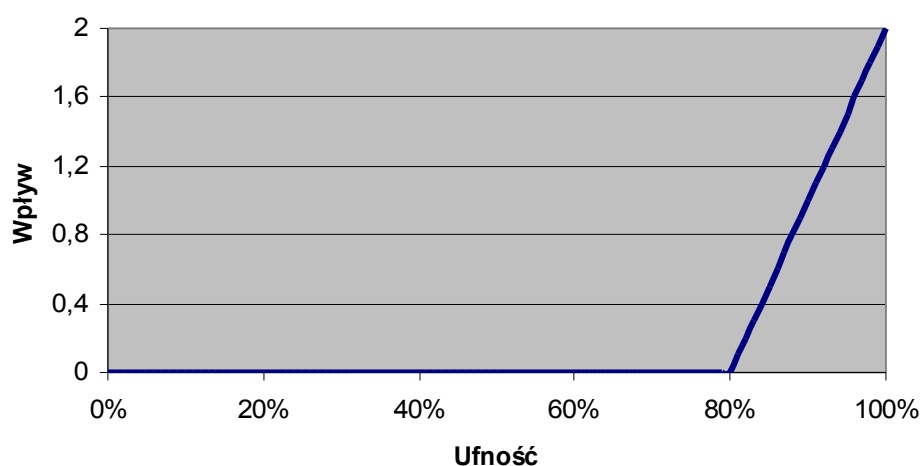
$V_{\text{new}}$  – nowa wartość wagi

$V_{\text{old}}$  – pierwotna wartość wagi

$I_{\text{max}}$  – maksymalny wpływ dla danego stanu krawędzi

$T$  – poziom ufności wzorca

$T_{\text{min}}$  – minimalny poziom ufności (od którego wzorec uznawany jest za prawdziwy)



Rysunek 2: Wykres współczynnika wpływu

Przykład powinien wyjaśnić ewentualne wątpliwości:

Zajmujemy się krawędzią, której aktualna waga to 10 ( $V_{old}$ ). Maksymalny wpływ ( $I_{max}$ ) dla stanu, z którym mamy do czynienia to 2, a minimalny poziom ufności ( $T_{min}$ ) wynosi 80%. Możemy więc łatwo policzyć, że dla kolejnych wartości ufności otrzymamy następujące wartości wagi:

$T = 80\%$	$V_{new} = 10$ (bez zmian)
$T = 82\%$	$V_{new} = 12$
$T = 85\%$	$V_{new} = 15$
$T = 90\%$	$V_{new} = 20$
$T = 98\%$	$V_{new} = 38$

Należy zauważyć, że wpływ może być także ujemny – wówczas waga krawędzi zmniejsza się – dzięki czemu więcej pojazdów wybiera taką trasę.

W rzeczywistości podany powyżej wzór jest pewnym uproszczeniem – w rzeczywistości w odpowiedniej tablicy sumowane są wpływy od różnych krawędzi. Pogrubiony fragment jest wpływem od pojedynczej krawędzi. Może się więc zdarzyć, że dwa lub więcej wzorców zniweluje wzajemnie swój wpływ, lub też go wzmocni.

Zmiana wag krawędzi odbywa się dopiero po przeanalizowaniu wszystkich wzorców i obliczeniu wpływu od każdego z nich.

### 3.3. Konfiguracja mechanizmu predykcji

Wprowadzenie mechanizmu predykcji przyniosło za sobą niemałą ilość zmian w konfiguracji systemu Kraksim. Przede wszystkim zmianę wprowadzono w głównym pliku konfiguracyjnym systemu Kraksim (config.properties). Zmiana polega na dodaniu dwóch parametrów – obu opcjonalnych – do tego pliku. Są nimi:

- enablePrediction – flaga określająca, czy mechanizm predykcji jest włączony czy nie. Dopuszczalne wartości to „true” i „false”
- predictionFile – pole określające położenie pliku xml z konfiguracją dla predykcji

Drugi z parametrów Jest ścieżką do pliku zawierającego pełną konfigurację mechanizmu predykcji. Konfiguracja ta składa się z dwóch elementów:

- konfiguracji algorytmu predykcji
- konfiguracji sposobu dyskretyzacji

Poniżej znajduje się przykładowy plik konfiguracyjny, pod nim zaś szczegółowy opis poszczególnych jego elementów.

#### 3.3.1. Przykładowa konfiguracja

Przykładowy plik konfiguracyjny znajduje się na końcu niniejszego dokumentu jako załącznik w rozdziale 7.1

#### 3.3.2. Konfiguracja algorytmu predykcji

Parametrami konfiguracyjnymi samego mechanizmu predykcji są:



- `cutOutProbability` – minimalny poziom ufności (por. definicja w rozdziale 3.2.1). Sensowne wartości należą do przedziału  $[0, 1]$ , preferowane  $[0.5, 0.95]$
- `cutOutMinimumNumber` – minimalna ilość wzorców potwierdzających wzorzec (por. definicja w rozdziale 3.2.1). Sensowne wartości to liczby całkowite, w granicach od 2 do 10, zależnie od rozmiaru i kształtu modelu.
- `neighborhoodSize` – szerokość sąsiedztwa (por. definicja w rozdziale 3.2.1 – oznaczone jako  $r$ ). Sensowne wartości to liczby całkowite, w zasadzie od 1 do 3.
- `influencedTimesteps` – wysokość następstwa (por. definicja w rozdziale 3.2.1). Sensowne wartości to liczby całkowite, w zasadzie od 1 do 4.
- `ageingRate` – współczynnik starzenia wzorców (por. rozdział Error: Reference source not found 3.2.4). Sensowne wartości są w granicach  $[0, 1]$ , preferowane  $[0.9, 1]$

### 3.3.3. Konfiguracja sposobu dyskretyzacji

To, na czym polega dyskretyzacja, opisane zostało w rozdziale 3.2.1, przy okazji definiowania stanu krawędzi. Definicja dyskretyzacji podawana w pliku konfiguracyjnym powinna zawierać wybraną ilość stanów. Stany te powinny spełniać następujące kryteria:

- powinny posiadać różne nazwy
- powinny być rozłączne
- powinny pokrywać wszystkie możliwe stany krawędzi, w praktyce wystarczającym jest w zupełności, by zawierały w sobie cały przedział  $[1, 100]$
- stany zawierają przedział lewostronnie otwarty, prawostronnie domknięty, a więc wskazane jest, aby pierwszy z przedziałów rozpoczynał się od wartości mniejszej niż 1 (najlepiej od 0)

Wartość, jaka będzie dyskretyzowana przez stany to stosunek aktualnego średniego czasu przejazdu daną krawędzią do minimalnego czasu przejazdu. Przykład:

- minimalny czas przejazdu krawędzią: 10
- aktualny średni czas przejazdu: 17
- wartość podlegająca dyskretyzacji: 1.7

Każdy ze stanów posiada następujące pola:

- `description` – opis poziomu natężenia ruchu. Pole tekstowe, unikalne, obowiązkowe
- `lowerBound` – dolna granica stanu (wyjaśnienie znajduje się powyżej w tym rozdziale). Wskazane jest, aby poziom najniższy zaczynał się od 0.
- `upperBound` – górna dolna stanu (wyjaśnienie znajduje się powyżej w tym rozdziale). Wskazane jest, aby poziom najwyższy przyjmował wartość o wiele wyższą niż najwyższa spodziewana (np. 100)
- `influence` – maksymalny wpływ wzorca, którego rezultatem jest dany stan (por. rozdział 3.2.5)
- `prevDescription` – opis poziomu natężenia ruchu bezpośrednio poprzedzającego dany poziom (niepotrzebne dla najniższego poziomu)
- `nextDescription` – opis poziomu natężenia ruchu bezpośrednio następującego dany poziom (niepotrzebne dla najwyższego poziomu)

### 3.3.4. Domyślne ustawienia sposobu dyskretyzacji

W przypadku, gdy w pliku konfiguracyjnym nie wystąpi definicja dyskretyzacji, używana jest dyskretyzacja domyślna. Jej właściwości są następujące:

Nazwa	Minimum	Maksimum	Wpływ	Poprzedni	Następny
Empty	0	1.2	-0.5	-----	Occupied
Occupied	1.2	1.6	0.1	Empty	Stuck
Stuck	1.6	100	1.5	Occupied	-----

## 4. Zmiany w architekturze systemu Kraksim

Wprowadzone w systemie Kraksim zmiany siłą rzeczy musiały zmodyfikować architekturę. Podjęte zostały starania, aby zmiany te jak najmniej zaburzały istniejącą strukturę. Zmiany zostały wprowadzone w taki sposób, by stanowiły jak najbardziej odrębny moduł dołączony w minimalnej ilości miejsc do systemu Kraksim. W rezultacie wszelkie dalsze zmiany w tak dodanym elemencie odbywają się wyłącznie na nim, bez ingerencji w pozostałe części systemu (modularność).

### 4.1. Zmiany architektoniczne systemu Kraksim związane z wielopasowością

Podstawową zmianą, jaka dokonała się w architekturze systemu Kraksim było odejście od założenia o jedyności akcji prowadzącej do wskazanej ulicy w obrębie całej ulicy. Koniecznym okazało się umożliwienie dodania właściwych, różnych akcji dla poszczególnych pasów (ponieważ akcja zawiera także informację o pasie, z którego może zostać wykonana), a także rozbudowa mechanizmu pobierania dostępnych akcji prowadzących do wskazanej ulicy – w tym momencie zamiast pojedynczego pasa, otrzymywana jest lista pasów. Siłą rzeczy, pociągnęło to za sobą konieczność wyboru najlepszej z pośród dostępnych listy akcji. W tym celu została stworzona osobna klasa, która wspomaga wybór. Dodatkową zaletą takiego rozwiązania jest fakt, że w razie zmiany koncepcji dotyczącej sposobu optymalizacji zrównoważenia ruchu bądź strategii obieranej przez pojazdy, zmiany dotkną wyłącznie nowo utworzoną klasę odpowiedzialną za wybór akcji.

Drugą ważną zmianą było wykorzystanie również tej klasy w celu doboru najlepszego pasa przy umieszczaniu pojazdów na ulicy. Konieczność ta wiązała się głównie z faktem, że w przypadku węzłów końcowych (gateway) nie istnieją żadne akcje, więc wybór w oparciu o akcje byłby nieskuteczny.

Ostatnią z istotnych zmian było wprowadzenie dodatkowego założenia – zgodnego z przepisami ruchu drogowego w Polsce – że pierwszeństwo ma pojazd poruszający się po najbardziej prawym pasie. W związku z tym w przypadku gdy więcej niż jeden pas jednej ulicy umożliwiał akcję prowadzącą do wskazanej ulicy, hierarchia kolejności pomiędzy nimi daje pierwszeństwo pasom poczynając od najbardziej prawego.

Konieczne ingerencje wewnątrz istniejących klas związane były głównie ze zmianą rozmiaru tablicy pasów (Lane) w klasie reprezentującej ulicę (Link). Dodatkowo zmodyfikowany został także parser pliku konfiguracyjnego modelu oraz narzędzie do wizualizacji modelu.

Poniżej znajduje się lista wprowadzonych zmian z uwzględnieniem poszczególnych klas:

#### **Stworzone klasy:**

- MultiLaneRoutingHelper (pl.edu.agh.cs.kraksim.real) – klasa zawierająca mechanizmy routingu na poziomie pasów.

#### **Zmodyfikowane klasy:**

- RoadNetXmlHandler (pl.edu.agh.cs.kraksim.parser)

- Pola:
  - numberOfLanes – dodano – pole przechowujące ilość głównych pasów
- Metody:
  - createLane – zmodyfikowano – dodano odczyt dodatkowej właściwości
  - endElement – zmodyfikowano – dodano parametr numberOfLanes do wywołania metody createLink, dodano generowanie listy akcji nadrzędnych dla ulic wielopasowych
  - getPriorityLanesForLane – dodano – tworzy tablicę pasów mających pierwszeństwo nad wskazanym pasem – konieczne do walidacji akcji i zachowania spójności modelu
- City (pl.edu.agh.cs.kraksim.core)
  - Metody
    - createLink – zmodyfikowano – dodano argument (numberOfLanes), zmieniono wywołanie konstruktora klasy Link
- Link (pl.edu.agh.cs.kraksim.core)
  - Pola:
    - mainLaneNum – usunięto
    - numberOfMainLanes – dodano
    - leftMainLaneNumer – dodano
    - rightMainLaneNumer – dodano
  - Metody:
    - Link (konstruktor) – zmodyfikowano – dodano parametr i jego obsługę
    - initializeMainLane – zmodyfikowano – aby obsługiwał wiele pasów
    - initializeLeftLanes – zmodyfikowano – aby uwzględniały zmienioną numerację pasów
    - leftLaneCount – zmodyfikowano – aby uwzględniały nową numerację
    - rightLaneCount – zmodyfikowano – aby uwzględniały nową numerację
    - mainLaneCount – dodano
    - getMainLane – zmodyfikowano – dodano parametr – numer pasa
    - getLeftLane – zmodyfikowano – aby uwzględniały nową numerację
    - getRightLane – zmodyfikowano – aby uwzględniały nową numerację
    - getLength – zmodyfikowano – aby uwzględniały nową numerację
    - getMainLanes – dodano – zwraca listę pasów głównych, w kolejności od prawego
    - findAction – oznaczono jako deprecated (do usunięcia, nie jest używana)
    - findActions – utworzono – znajduje listę akcji prowadzących do wskazanej ulicy
- Lane (pl.edu.agh.cs.kraksim.core)
  - Pola
    - relativeNumber – dodano

- Metody:
  - Lane (konstruktor) – zmodyfikowano – dodano parametr: relativeNumber i jego obsługę
  - getRelativeNumber – zmodyfikowano – uproszczono
  - addAction – zmodyfikowano – usunięto sprawdzanie warunku unikalności akcji w obrębie ulicy
  - checkPriorityRules – zmodyfikowano – dodano sprawdzanie dla wszystkich akcji (w związku z możliwością otrzymania listy akcji zamiast pojedynczej akcji)
- CityMapVisualisator (pl.edu.agh.cs.kraksim.visual)
  - Metody:
    - drawRoad – zmodyfikowano – uwzględniono wiele pasów głównych
- VisualisatorComponent (pl.edu.agh.cs.kraksim.visual)
  - Metody:
    - drawVehicles – zmodyfikowano – uwzględniono wiele pasów głównych
- LinkRealExt (pl.edu.agh.cs.kraksim.real)
  - Metody:
    - enterCar – zmodyfikowano – dodano fragment kodu służący do wyboru właściwego pasa, na który należy wstawić samochód oraz akcji, jaką należy mu ustawić
- LaneRealExt (pl.edu.agh.cs.kraksim.real)
  - Pola:
    - enteringCar – usunięto
    - enteringCars – dodano – aby więcej niż jeden pojazd mógł równocześnie wjechać na skrzyżowanie
  - Metody:
    - hasCarPlace – zmieniono tak, by uwzględniało pojazdy wjeżdżające na pas w danej turze.
    - driveCar – zmodyfikowano – aby uwzględniało zmianę pola enteringCar z pojedynczego na listę
    - finalizeTurnSimulation - zmodyfikowano – aby uwzględniało zmianę pola enteringCar z pojedynczego na listę
    - getAllCarsNumber – dodano – zwraca liczbę pojazdów na danym pasie
    - getFirstCarPosition – dodano – zwraca ilość wolnych miejsc na początku pasa
- GatewayRealExt (pl.edu.agh.cs.kraksim.real)
  - Pola:
    - acceptedCar – usunięto
    - acceptedCars – dodano – aby więcej niż jeden pojazd mógł równocześnie opuścić miasto
  - Metody:
    - GatewayRealExt – konstruktor – zmieniono inicjalizację
    - acceptCar – zmodyfikowano – aby uwzględniało zmianę pola acceptedCar z pojedynczego na listę
    - finalizeTurnSimulation – zmodyfikowano – aby uwzględniało zmianę pola acceptedCar z pojedynczego na listę



## 4.2. Zmiany architektoniczne systemu Kraksim związane z predykcją

Zmiany wprowadzone w systemie w związku z wprowadzeniem mechanizmu predykcji miały z założenia wykorzystać istniejącą architekturę systemu i występujące w nim składniki, modyfikując je jedynie w najmniejszym możliwym stopniu. Dzięki temu wyłączenie lub podmiana mechanizmu predykcji nie wpływa w zupełności na funkcjonowanie systemu Kraksim, ponadto zmiany wprowadzone da się bez problemów usunąć lub przenieść do innej wersji systemu.

Prawie wszystkie z pośród dodanych klas i interfejsów umieszczono w pakiecie *pl.edu.agh.cs.kraksim.routing.prediction*. Wyjątek nowi parser konfiguracji, który znajduje się w pakiecie *pl.edu.agh.cs.kraksim.parser*.

Podstawę wprowadzonej zmiany stanowią dwa interfejsy:

- *ITrafficPredictionSetup*
- *ITrafficPredictor*

Pierwszy z nich odpowiada za przechowywanie i przekazywanie pomiędzy elementami systemu konfiguracji mechanizmu predykcji. Tworzony jest przez wspomniany powyżej parser (*pl.edu.agh.cs.kraksim.parser.PredictionConfigurationXml*), który przetwarza plik konfiguracyjny mechanizmu predykcji, do którego ścieżka podawana jest w konfiguracji systemu Kraksim.

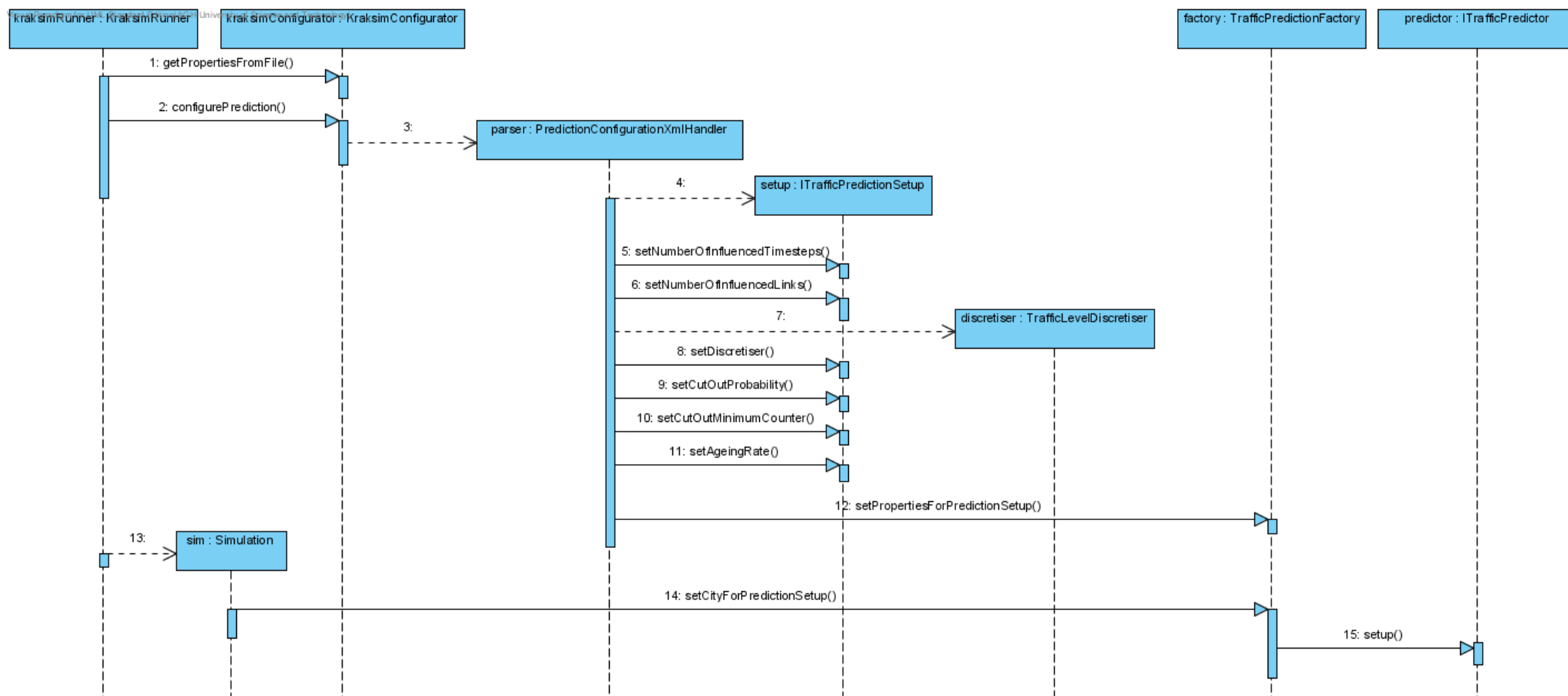
Ustawienia zawarte w tej klasie są następnie przekazywane do klasy implementującej interfejs *ITrafficPredictor*. Jest to w zasadzie interfejs, który udostępnia pełną funkcjonalność mechanizmu predykcji. Metody, które należą do tego interfejsu to:

- *setup (ITrafficPredictionSetup)* – służąca ustawieniu właściwych wartości parametrów mechanizmu predykcji zgodnie z konfiguracją odczytaną z pliku
- *appendWorldState (WorldState)* – służy do zbierania i agregacji danych na temat historii świata i wystąpień poszczególnych sekwencji w jej obrębie. Metoda wywoływana jest z klasy *TimeTable*, wykorzystywanej w mechanizmie routingu dynamicznego
- *adjustCurrentWeightsOfLink (double[])* – służy do wyszukania i zastosowania wzorców predykcyjnych do grafu połączeń, zapisanego w postaci jednowymiarowej macierzy liczb (kolejne wagi krawędzi)

Poniższe diagramy powinny dać pełną jasność co do sposobu działania mechanizmu predykcji.







Rysunek 4: Diagram sekwencji uruchomienia – inicjalizacja predykcji

Jak widać, proces tworzenia instancji klasy odpowiedzialnej za mechanizm predykcji jest dość złożony, wynika to jednak z faktu, że w zrealizowanej implementacji wybrane zostało rozwiązanie z singletonem, korzystające jednak ze wzorca projektowego Fabryka. Na powyższym diagramie zostały również wyszczególnione elementy sekwencji uruchomienia nie wynikające z mechanizmu predykcji – jak choćby odczyt całości konfiguracji z pliku (komunikat oznaczony numerem 1 na rysunku 4), jak również utworzenie symulacji (komunikat 13). Komunikaty od 5 do 11 ustawiają poszczególne parametry algorytmu, między innymi sposób dyskretyzacji, który dokonywany jest za pośrednictwem specjalnej klasy (*TrafficLevelDiscretiser*).

Fabryka służy tu również jeszcze jednemu zadaniu – poza zwróceniem odpowiedniej instancji *ITrafficPredictor*, służy ona także odpowiedniemu jej skonfigurowaniu – poza parametrami mechanizmu predykcji wymaga on bowiem do swego działania wskazania na model statyczny symulowanej sieci drogowej, który jest przekazywany dopiero po utworzeniu symulacji do fabryki. Dopiero po otrzymaniu tych dwóch elementów (konfiguracji predykcji oraz wskazania na model statyczny) mechanizm predykcji może rozpocząć poprawne funkcjonowanie.

Poniżej znajduje się lista zmian z uwzględnieniem poszczególnych klas:

#### Utworzone klasy:

- pakiet *pl.edu.agh.cs.kraksim.parser*
  - *PredictionConfigurationXmlHandler*
- pakiet *pl.edu.agh.cs.kraksim.routing.prediction*
  - interfejs *ITrafficPredictor*
  - interfejs *ITrafficPredictionSetup*
  - klasa *DefaultTrafficPredictor*
  - klasa *DefaultTrafficPredictionSetup*
  - klasa *TrafficLevelDiscretiser*
  - klasa *TrafficLevel*
  - klasa *TrafficPredictionFactory*
  - klasa *WorldState*
  - klasa *TrafficPredictionContainerPath*
  - klasa *TrafficStatisticsForResult*
  - klasa (wyjątek) *TrafficPredictionException*

#### Zmodyfikowane klasy:

- pakiet *pl.edu.agh.cs.kraksim.routing*
  - klasa *TimeTable*
    - dodano wymagane importy
    - dodano pole *predictor* przechowujące wskazanie na instancję klasy realizującej interfejs *ITrafficPredictor*
    - zmodyfikowana konstruktor o uzupełniania pola *predictor*
    - zmodyfikowano metodę *refreshAll* o wywołanie metod agregującej dane do predykcji oraz stosującej mechanizm predykcji
- pakiet *pl.edu.agh.cs.kraksim.main*
  - klasa *Simulation*
    - dodano wymagane importy

- zmodyfikowano metodę ***doStep*** o uzupełnienie konfiguracji mechanizmu predykcji o wskazanie na model statyczny
  - zmodyfikowano metodę ***doRun*** o uzupełnienie konfiguracji mechanizmu predykcji o wskazanie na model statyczny
- klasa ***SampleModuleConfiguration***
  - zmodyfikowano metodę ***setUpModules***, zastępując wykorzystanie statyczne routera dynamicznym
- pakiet ***pl.edu.agh.cs.kraksim.routing***
  - klasa ***TimeBasedRouter***
    - przepisana praktycznie w całości – ze względu na błędy w działaniu i marnotrawstwo zasobów. Klasa była ewidentnie przekopioną dość bezmyślnie klasą routera statycznego, ze wszystkimi jego właściwościami – w tym także przechowywaniem mapy możliwych połączeń pomiędzy każdą parą węzłów w grafie – mapa ta nie była oczywiście wykorzystywana, sprawiałoby to, iż routing przestawałby być dynamiczny. Po modyfikacji klasa nie wykonuje zbędnych operacji i nie przechowuje archiwalnych tras.

### 4.3. Zmiany w interfejsie użytkownika i mechanizmie uruchamiania

Zmiana w sposobie uruchamiania systemu Kraksim związana była z tym, iż poprzednia wersja systemu nie umożliwiała wielokrotnego uruchomienia symulacji. Za rozpoczęcie wizualnej części symulacji odpowiadała klasa **OptionsPanel**, w głównym pakiecie systemu (*pl.edu.agh.cs.kraksim*). Ze względu na brak tej elementarnej funkcjonalności oraz na fakt, że poza uruchomieniem aplikacji zajmowała się ona także przygotowaniem danych otrzymanych z pliku konfiguracyjnego, rozdzieliliśmy tę klasę na 2 nowe klasy: **KraksimRunner**, odpowiedzialną jedynie za uruchomienie aplikacji, oraz **KraksimConfigurator**, służącą do przygotowania danych konfiguracyjnych w sposób umożliwiający uruchomienie symulacji. Część wizualizacyjna została przeniesiona do klasy **MainVisualisationPanel** (znajdującej się w pakiecie main wewnątrz pakietu głównego) – dodane zostało także menu. Sama zawartość wymienionych klas nie zmieniła się znacznie w stosunku do klasy **OptionsPanel**, zmiana umożliwiła jedynie ponowny wybór ustawień symulacji (co wcześniej było możliwe jedynie na początku, przed pierwszym uruchomieniem symulacji) i ponowne jej wystartowanie.

Poniżej znajduje się lista z uwzględnieniem poszczególnych klas:

#### Utworzono klasy:

- W pakiecie *pl.edu.agh.cs.kraksim*
  - klasę **KraksimRunner**
  - klasę **KraksimConfigurator**
- W pakiecie *pl.edu.agh.cs.kraksim.main*
  - klasę **MainVisualisationPanel**
  - klasę **InputPanel** (dawniej klasa wewnętrzna klasy **OptionsPanel**)
  - klasę **SetUpPanel** (wyodrębniona funkcjonalność klasy **OptionsPanel**)

#### Usunięto klasę:

- W pakiecie *pl.edu.agh.cs.kraksim.main*
  - klasę **OptionsPanel**

## 4.4. Opis całościowy nowej architektury

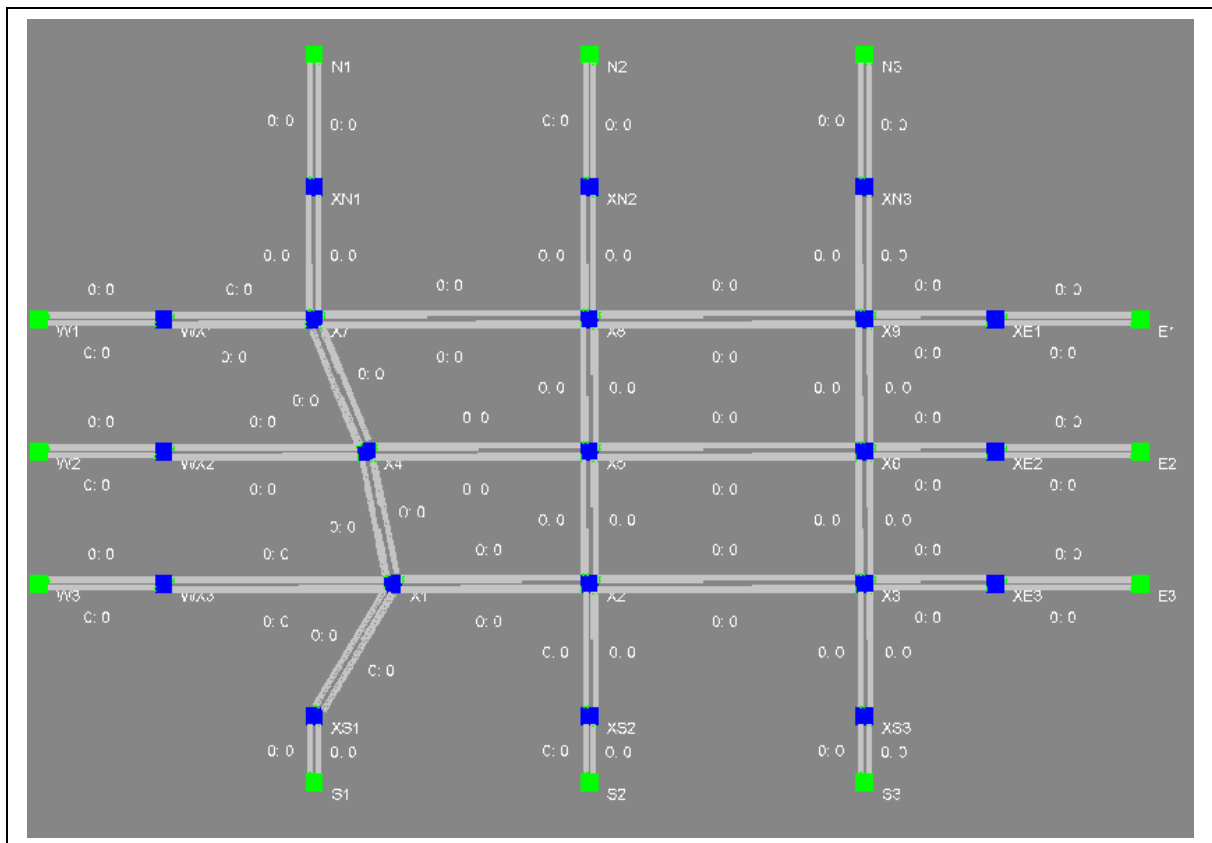
Jak było już wspomniane, w trakcie wprowadzania zmian usiłowaliśmy stosować je tak, by ingerencja w istniejącą architekturę była minimalna. Z powodzeniem można więc powiedzieć, że całość architektury systemu Kraksim nie uległa zmianie. Całość logiki ukrytej za poruszaniem się pojazdów pozostała bez zmian, nowe elementy zostały jedynie umieszczone jako elementy pośredniczące – można porównać to do wstawienia swego rodzaju filtrów – miało to miejsce zarówno w przypadku mechanizmu wielopasowości (gdzie „filtr” został dodany na poziomie parsingu pliku konfiguracyjnego oraz jako element procedury umieszczającej pojazd na nowej ulicy), jak również w przypadku mechanizmu predykcji (gdzie „filtr” został dodany jako element routera dynamicznego).

## 5. Testy

Wykonywane przez nas testy prowadzone były w dwóch etapach – pierwszym był etap testowania wielopasowości, drugim natomiast testy mechanizmu predykcji (na modelach z wykorzystaniem wielopasowości). Ich rezultaty zostaną zaprezentowane osobno.

### 5.1. Testy wielopasowości

Testy zostały przeprowadzone w oparciu o mapę 9xgrid, która najlepiej oddaje charakterystykę ulic w mieście. W testach była wykonywana symulacja, która była przerywana w 3000 turze (turę tą wybraliśmy ze względu na to, iż dla podanej mapy były widoczne w niej już korki, natomiast mapa nie była całkowicie zablokowana).



Rysunek 5: model 9xGrid

W czasie testów zmienialiśmy:

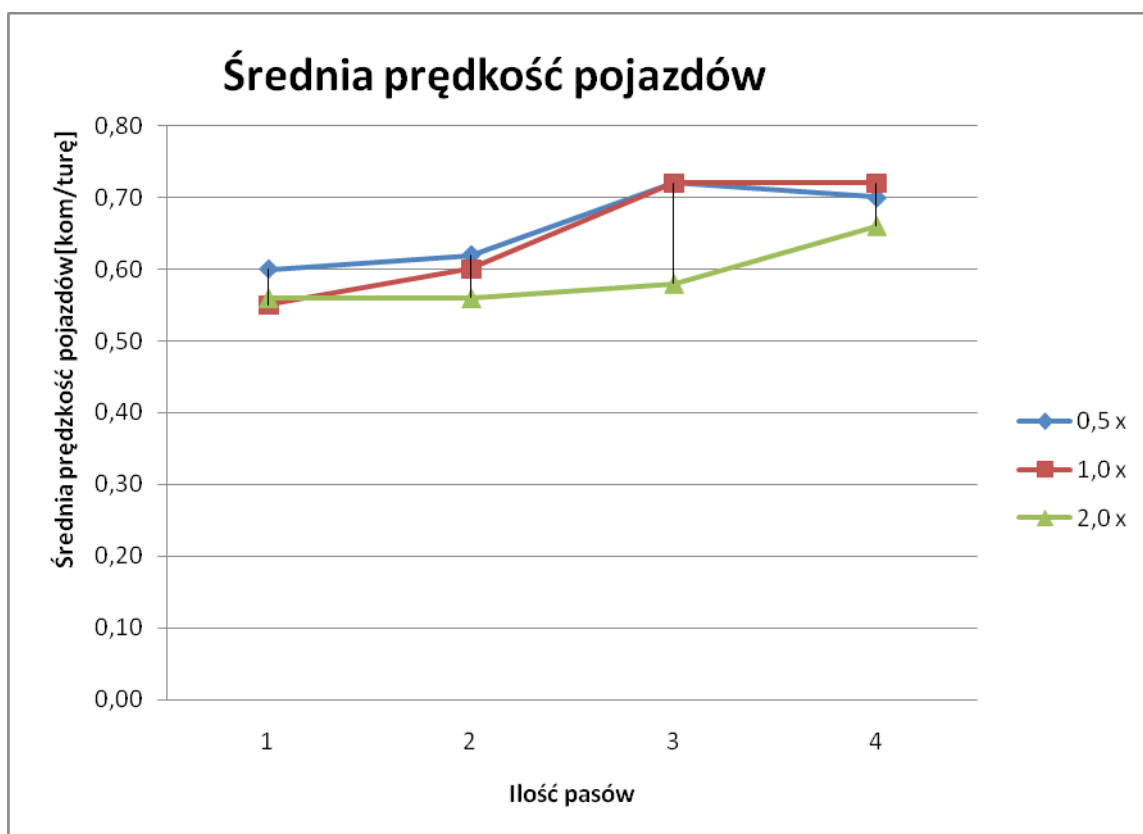
- Ilość pasów dla wszystkich dróg (1, 2, 3 lub 4 pasy)
- Natężenie ruchu (podstawowe, 2 razy większe i 2 razy mniejsze) ruch był generowany różnie z różnych gatewayów (podobnie jak dla rzeczywistych dróg)

Obserwacją poddana była:

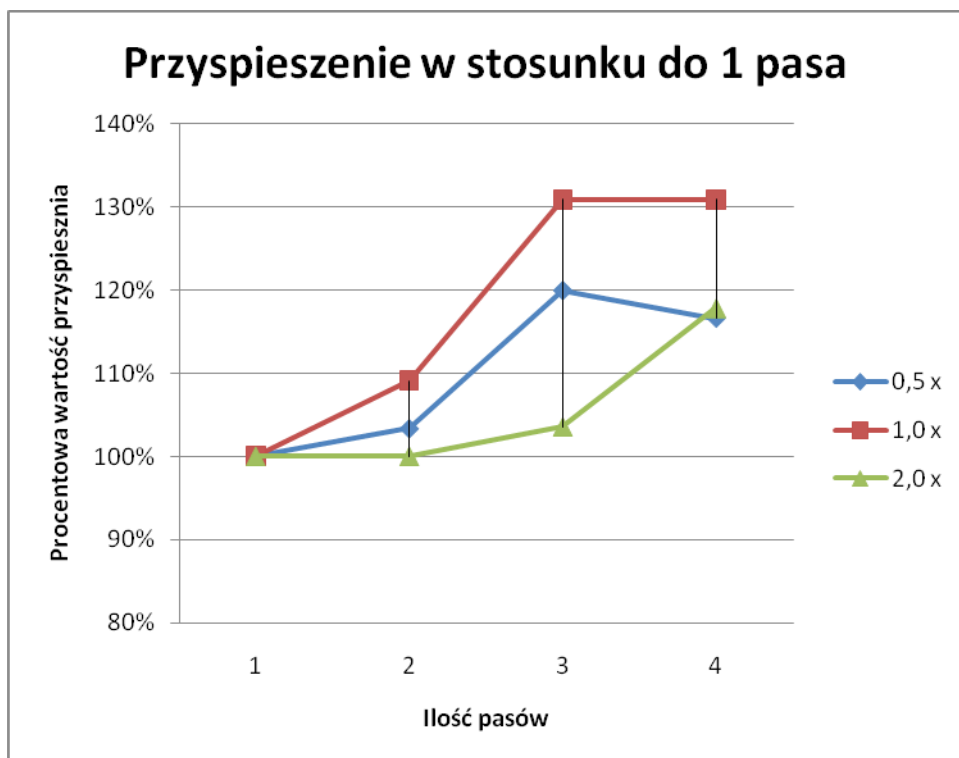
- Ilość samochodów znajdująca się na mapie
- Ilość samochodów, która przejechała całą trasę

- Średnia prędkość wszystkich samochodów

Na poniższym wykresie znajduje się wykres średniej prędkości pojazdów od ilości pasów. Każda linia przedstawia inne natężenie ruchu (niebieska najmniejsze, zielona największe)



Na podstawie przedstawionych powyżej danych został sporządzony wykres procentowego przyspieszenia w stosunku do drogi 1 pasowej.

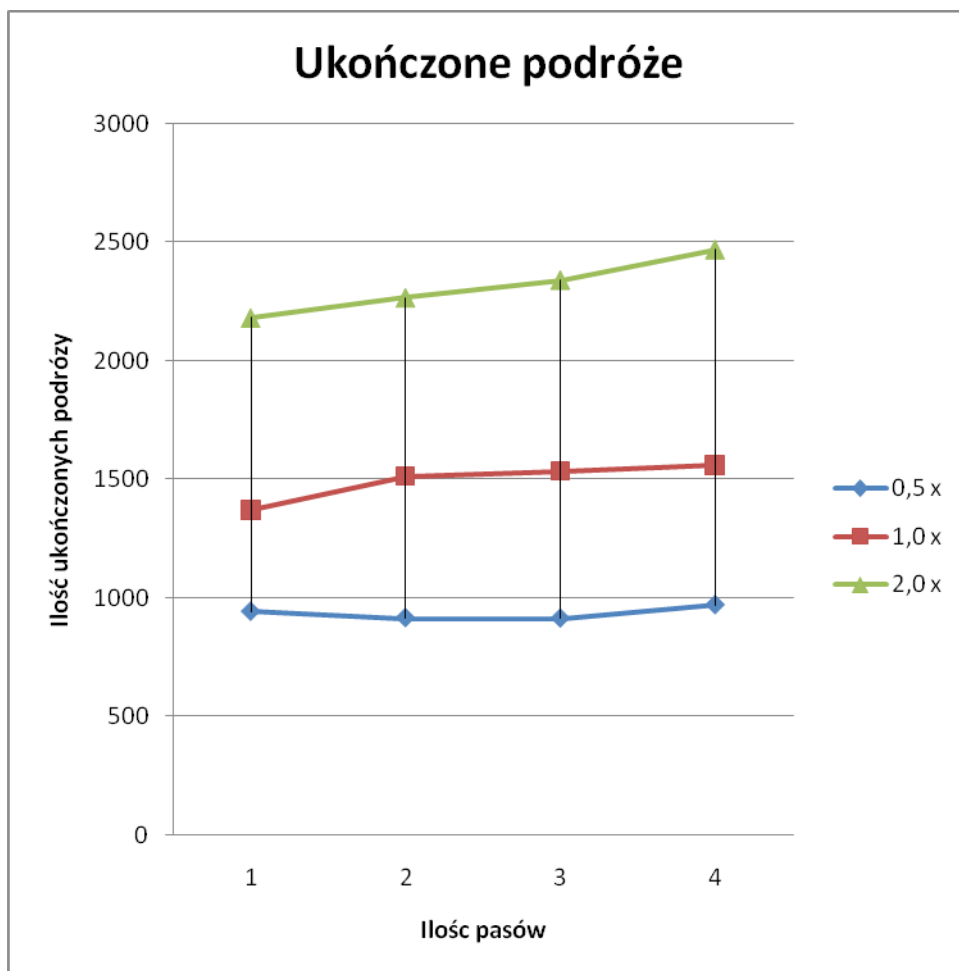


Na podstawie powyższych wykresów można zaobserwować przyspieszenie szybkości samochodów w zależności od ilości pasów. Przyspieszenie to nie jest jednak liniowe. Warto zauważyć, że dla 4 krotnie większej liczby pasów (liczba zwiększona 400%) osiągany zysk to maksymalnie 140%. Jest to spowodowane tym, iż na takiej mapie nadal są „wąskie gardła” w postaci pojedynczych pasów skrzyżowania w lewo oraz sytuacji, w których z dwóch dróg 4 pasowych samochody wjeżdżają w pojedynczą drogę 4 pasową.

Ciekawą obserwacją jest też fakt, iż największe przyspieszenia osiąga się dla średniego natężenia ruchu (choć wartości bezwzględne szybkości największe są dla ruchu najmniejszego). Jest to spowodowane faktem, iż z jednej strony dla małego ruchu robią się małe korki dla 1 pasa, w związku z tym przyspieszenie względne nie jest tak korzystne. Z drugiej strony dla dużego natężenia ruchu, o średniej prędkości decydują drogi, na których korków nie ma co powoduje ich małe względne przyspieszenie.

Poniższy wykres przedstawia ilość pojazdów, które ukończyły podróż.





Jak widać ilość samochodów, które ukończyły podróż wzrasta z liczbą pasów jaką mają te samochody dostępne. Wzrost ten jest jednak nieznaczny w stosunku do zmiany liczby pasów. Największy wzrost jest dla największego ruchu.

Obserwacje te można wytłumaczyć faktem, iż za liczbę samochodów, które ukończyły podróż głównie odpowiadają drogi niezakorkowane. Na takie drogi nie ma wpływu ilość pasów. Wyniki są, więc zgodne z oczekiwaniami.

Poniżej przedstawiamy tabelę ze zbiorczymi danymi pomiarowymi:

traffic	lane nr	Car Count	Travel Count	Avg V	Przyrost Avg v
0,5 x	1	637	943	0,60	100%
	2	664	916	0,62	103%
	3	666	914	0,72	120%
	4	609	971	0,70	117%
1,0 x	1	1800	1372	0,55	100%
	2	1658	1514	0,60	109%
	3	1637	1535	0,72	131%
	4	1612	1560	0,72	131%
2,0 x	1	4170	2183	0,56	100%
	2	4085	2268	0,56	100%

	3	4010	2343	0,58	104%
	4	3882	2471	0,66	118%

## 5.2. Testy mechanizmu predykcji

Wyniki liczbowe testów, w postaci tabel, znajdują się w dołączonym arkuszu programu Excel. Poniżej zaprezentowane zostały jedynie wybrane wyniki.

### 5.2.1. Sposób testowania

Testy mechanizmu predykcji zostały przeprowadzone dla następujących parametrów ulegających zmianie:

- Dla różnych modeli:
  - Wzbogacony model-D (dedykowany do predykcji)
  - Model miasta Krakowa
- Dla różnych natężeń ruchu:
  - Niskie (0.5 \* standard)
  - Średnie (1 \* standard)
  - Wysokie (2 \* standard)

Standard określony jest osobno dla każdego modelu. Jego opis znajduje się w podrozdziałach odpowiednio 5.2.2 dla modelu-D oraz 5.2.3 dla modelu Krakowa, same pliki z konfiguracją można znaleźć w załącznikach w rozdziałach odpowiednio 7.4 dla modelu-D i 7.6 dla modelu Krakowa)

- Dla różnych dyskretyzacji (opis poniżej)
  - Domyślne
  - O dużym wpływie (duża)
  - O małym wpływie (mała)
- Dla różnych trybów mechanizmu predykcji (opis poniżej)
  - Słabej
  - Umiarkowanej
  - Silnej

Ponadto dla każdego modelu i natężenia ruchu przeprowadzone zostały testy z wyłączonym mechanizmem predykcji, a także z wyłączonym mechanizmem dynamicznego routingu

### Typy dyskretyzacji

Dyskretyzacja domyślna:

Nazwa	Minimum	Maksimum	Wpływ	Poprzedni	Następny
Empty	0	1.2	-0.5	-----	Occupied
Occupied	1.2	1.6	0.1	Empty	Stuck
Stuck	1.6	100	1.5	Occupied	-----

Dyskretyzacja ta charakteryzuje się tym, iż jest najmniejszą z pośród testowanych dyskretyzacji – kategoryzuje rezultaty jedynie na 3 grupy. W związku z tym maleje jej dokładność, z drugiej jednak strony takie rozwiązanie jest nieco bardziej uniwersalne – dla każdego modelu najlepiej byłoby dobrać własną, specyficzną dyskretyzację w oparciu o znajomość charakterystyki ruchu. Dyskretyzacja domyślna zaś powinna osiągać przyzwoite wyniki dla wszystkich modeli. Wpływ tej dyskretyzacji jest umiarkowany – ani szczególnie duży, ani mały.

Dyskretyzacja o dużym wpływie (opisywana w skrócie jako duża)

Nazwa	Minimum	Maksimum	Wpływ	Poprzedni	Następny
Empty	0	1.1	-0.8	-----	Smooth
Smooth	1.1	1.2	0.5	Empty	Dense
Dense	1.2	1.5	5	Smooth	Stuck
Stuck	1.5	100	10	Dense	-----

Dyskretyzacja o 4 szczeblach (a więc dokładniejsza od domyślnej). Ponadto wpływ, jaki posiadają poszczególne ze stanów jest większy, a – co za tym idzie – znalezione wzorce w większym stopniu wpływają na działanie aplikacji i zachowanie pojazdów. Wskazana głównie w sytuacji, gdy charakterystyka ruchu jest klarowna i nie zmienia się często w czasie.

Dyskretyzacja o małym wpływie (w skrócie: mała)

Nazwa	Minimum	Maksimum	Wpływ	Poprzedni	Następny
Empty	0	1.1	-0.3	-----	Smooth
Smooth	1.1	1.3	0	Empty	Dense
Dense	1.3	1.6	1	Smooth	Stuck
Stuck	1.6	100	2	Dense	-----

Dyskretyzacja również o 4 szczeblach, których wpływ jest jednak mniejszy niż w przypadku dyskretyzacji domyślnej czy dużej. W rezultacie znajdowane wzorce nie oddziałują tak silnie na wagi krawędzi, co jest dobrym rozwiązaniem w przypadku, gdy wzorce są mniej pewne – na przykład gdy ruch ulega częstym zmianom.

## Typy predykcji

Predykcja słaba ma następujące parametry:

Ufność minimalna	0.5
Minimalna liczba przypadków potwierdzających	2
Szerokość sąsiedztwa (r)	1
Wysokość następstwa (h)	2
Współczynnik starzenia	0.99

Predykcja tak ustawiona przyjmuje jako wzorce sekwencje o niskim stopniu ufnosci – może się więc łatwo zdarzyć, że zaakceptuje jako wzorzec coś, co nim nie jest. Jednakże dzięki takiej konfiguracji predykcja może działać dla modeli, w których ruch często ulega zmianom, na przykład gdy dwa następstwa występują z podobną częstotliwością

Predykcja umiarkowana ma następujące parametry:

Ufność minimalna	0.8
Minimalna liczba przypadków potwierdzających	5
Szerokość sąsiedztwa (r)	1

Wysokość następstwa (h)	2
Współczynnik starzenia	0.99

Predykcja pośrednia – która akceptuje wzorce posiadające dość mocne poparcie w rezultatach, a zarazem rozpoczynająca działanie stosunkowo szybko i dość szybko reagująca na zmiany natężenia ruchu.

Predykcja silna ma następujące parametry:

Ufność minimalna	0.9
Minimalna liczba przypadków potwierdzających	8
Szerokość sąsiedztwa (r)	1
Wysokość następstwa (h)	2
Współczynnik starzenia	0.99

Predykcja stawiająca bardzo silne warunki sekwencjom – dzięki temu wzorce przez nią znalezione z bardzo dużą pewnością nadają się do zastosowania. Siłą rzeczy oporniej reaguje na zmianę dynamiki ruchu, dla silnie zmiennego ruchu może wcale się nie uaktywnić.

Mierzone były następujące właściwości:

- Ilość zakończonych podróży lub tura ukończenia symulacji (dla małych symulacji)
- Przebieg średniej szybkości pojazdów
- Ilość pojazdów w modelu

Konfiguracja, w jakiej uruchamiane były testy znajduje się jako załącznik w rozdziale 7.2

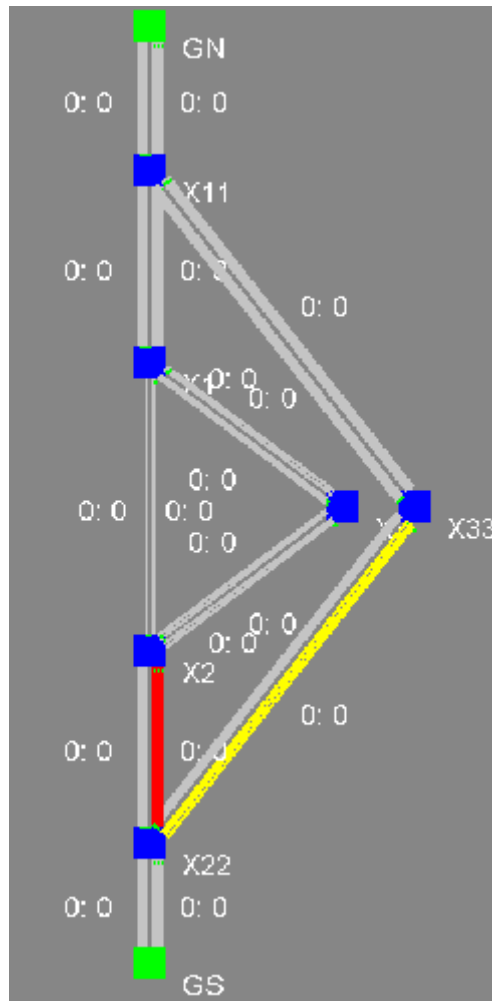
Wersy zakomentowane stanowiły alternatywne konfiguracje (gł. konfiguracje predykcji – odpowiadają one opisowi powyżej).

### 5.2.2. Rezultaty testów – model D

W pierwszej kolejności przeprowadzone zostały testy dla modelu D – jest to model, na którym działanie mechanizmu predykcji powinno być szczególnie wyraźne. Model składa się z głównej, trzypasmowej magistrali łączącej północny i południowy wjazd do modelu. W środkowym odcinku magistrala zwęża się do pojedynczego pasa, co skutkuje powstaniem korka na odcinku przez zwężeniem. Istnieje możliwość ominięcia zwężenia – początki zwężenia połączone są dodatkową, dwupasmową ulicą (dłuższą, oczywiście, od odcinka jednopasmowego). Ponadto istnieje także szeroka, trójpasmowa „obwodnica”, zaczynająca się zaraz za wjazdem do modelu. Plik konfiguracyjny modelu znajduje się w załączniku w rozdziale 7.3. Sam model jest bardzo niewielki – odległość między wjazdami do modelu, w których generowany jest ruch, wynosi niecałe 400 jednostek

Mechanizm routingu statycznego kieruje wszystkie pojazdy przez magistralę – jako trasę najkrótszą. Routing dynamiczny przekierowuje część (stosunkowo niewielką) pojazdów na trasy objazdowe, jednak są one o tyle dłuższe, że ilość posyłanych nimi pojazdów nie odciąża wystarczająco głównej magistrali.

Dla kolejnych natężeń ruchu do modelu zostaje wprowadzone odpowiednio 800, 1500 i 3000 pojazdów, z rozkładem równomiernym, po połowie z każdego z wjazdów (północnego i południowego). Generacja pojazdów trwa przez 500 tur. Standardowy plik konfiguracji natężenia ruchu znajduje się w załączniku w rozdziale 7.4:



Rysunek 6: Model-D

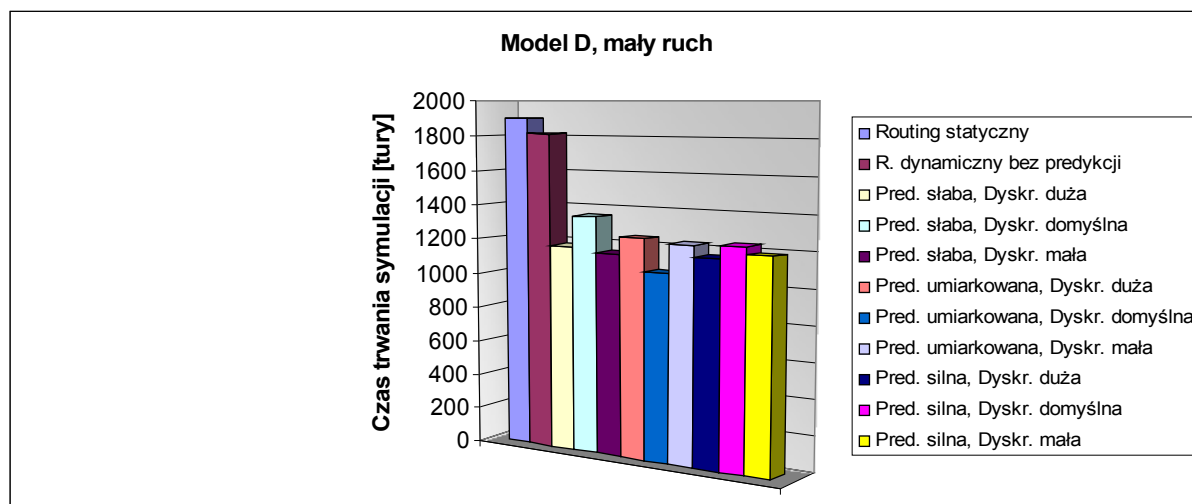
Na etapie testów opracowane zostały wykresy prezentujące wagę krawędzi w mechanizmie routingu dynamicznego. Wagi te zmieniają się samoczynnie na skutek wzrostu natężenia ruchu (odpowiadają czasom przejazdu daną krawędzią w algorytmie Dijkstry), lecz są także modyfikowane przez mechanizm predykcji – ich zmiany mogą wskazać, jaki wpływ miały poszczególne predykcje na wagę wskazanej krawędzi. Dla każdej z krawędzi sporządzone zostały wykresy dla predykcji, która wypadła najlepiej i najgorzej w danym modelu dla danego natężenia ruchu.

W modelu-D jako przykładowe, reprezentatywne krawędzie wybrane zostały krawędzie zaznaczone na powyższym rysunku. Będą one w dalszej części dokumentu prezentowane jako przykłady krawędzi o odpowiednio niskim (żółta) i wysokim (czerwona) natężeniu ruchu.

Poniżej zebrane zostały wyniki dla poszczególnych natężeń ruchu

#### 5.2.2.1. Niskie natężenie ruchu

Symulacja kończy się szybko, poniżej znajduje się zestawienie czasów trwania symulacji dla poszczególnych konfiguracji.



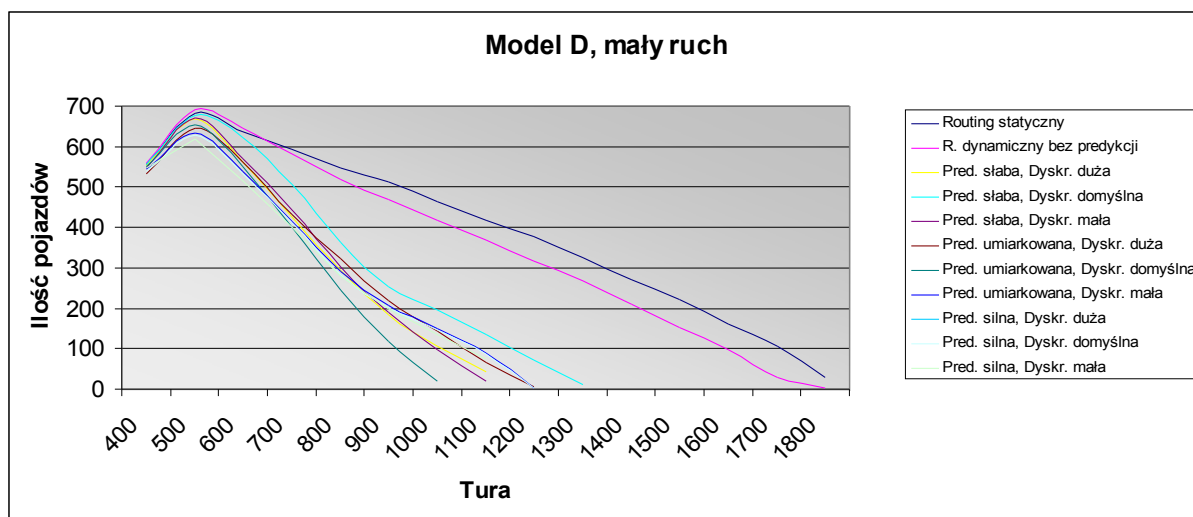
Rysunek 7: Model D, niskie natężenie ruchu – wykres czasu trwania symulacji

Konfiguracja	Czas trwania [tury]
Routing statyczny	1897
R. dynamiczny bez predykcji	1814
Pred. słaba, Dyskr. duża	1176
Pred. słaba, Dyskr. domyślna	1353
Pred. słaba, Dyskr. mała	1151
Pred. umiarkowana, Dyskr. duża	1247
Pred. umiarkowana, Dyskr. domyślna	1061
Pred. umiarkowana, Dyskr. mała	1225
Pred. silna, Dyskr. duża	1166
Pred. silna, Dyskr. domyślna	1236
Pred. silna, Dyskr. mała	1194

Konfiguracja	Średni czas trwania
Predykcja słaba	1226,666667
Predykcja umiarkowana	1177,666667
Predykcja silna	1198,666667
Dyskretyzacja duża	1196,333333
Dyskretyzacja domyślna	1216,666667
Dyskretyzacja mała	1190

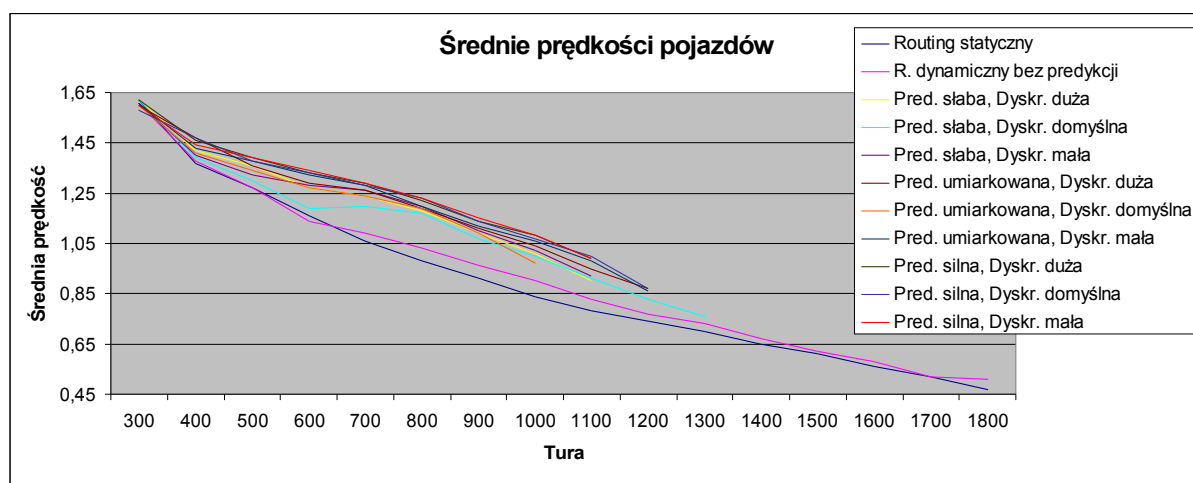
Jak widać, najlepsze wyniki osiągnane są w tym przypadku dla predykcji umiarkowanej.

Poniżej znajduje się wykres przebiegu ilości pojazdów w modelu w zależności od czasu trwania symulacji.



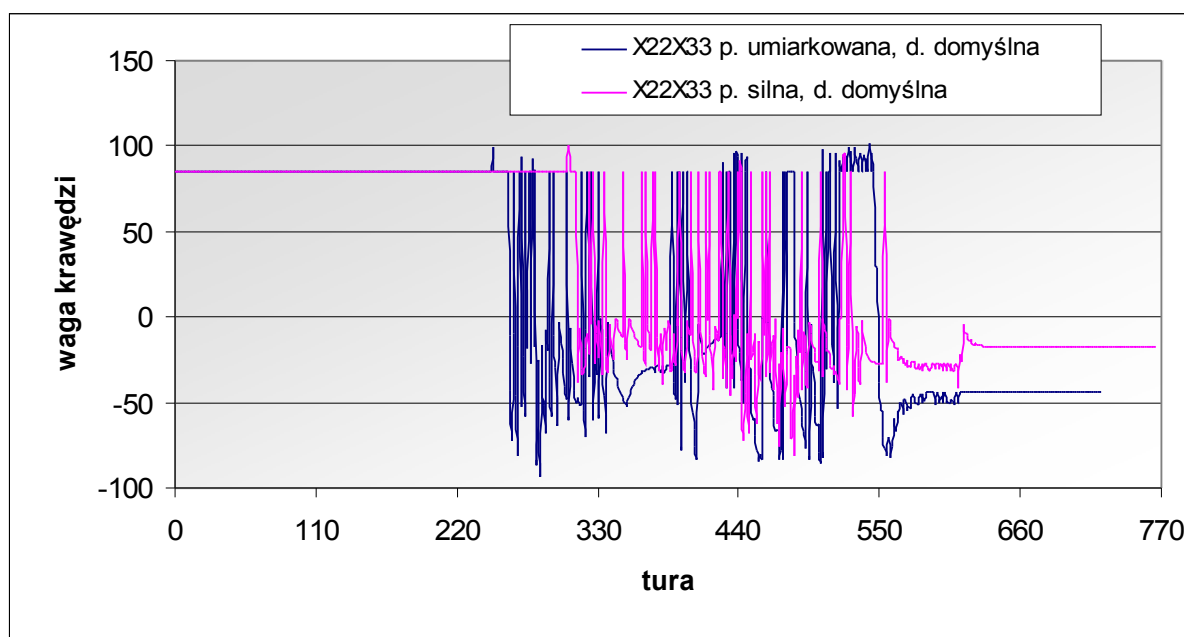
Rysunek 8: Model-D, niskie natężenie ruchu – wykres ilości pojazdów

Dodatkowo, dla każdej konfiguracji, prowadzona była także obserwacja zmiany średnich prędkości pojazdów w modelu. Ze względu jednak na fakt, że prędkości jednostkowe pojazdów mogą zmieniać się jedynie w zakresie  $[0,2]$ , na otrzymanych wykresach niewiele można dostrzec. Wykres dla tego modelu i niskiego natężenia ruchu jest stosunkowo najlepszy, dlatego on został zaprezentowany poniżej (pozostałe znajdują się w arkuszu dołączonym do dokumentacji)



Rysunek 9: Model D, niskie natężenie ruchu – wykres szybkości średniej

Powyższy wykres zmiany średniej szybkości pojazdów w czasie nie zawiera właściwie żadnych kluczowych informacji. Dlatego sięgnęliśmy do wyników (odwrotnie) proporcjonalnych do średnich prędkości średnich czasów przejazdu poszczególnymi krawędziami, zmodyfikowanych dodatkowo poprzez mechanizm predykcji. Rezultaty otrzymane dla najlepszej i najgorszej z konfiguracji predykcji są bardzo ciekawe:

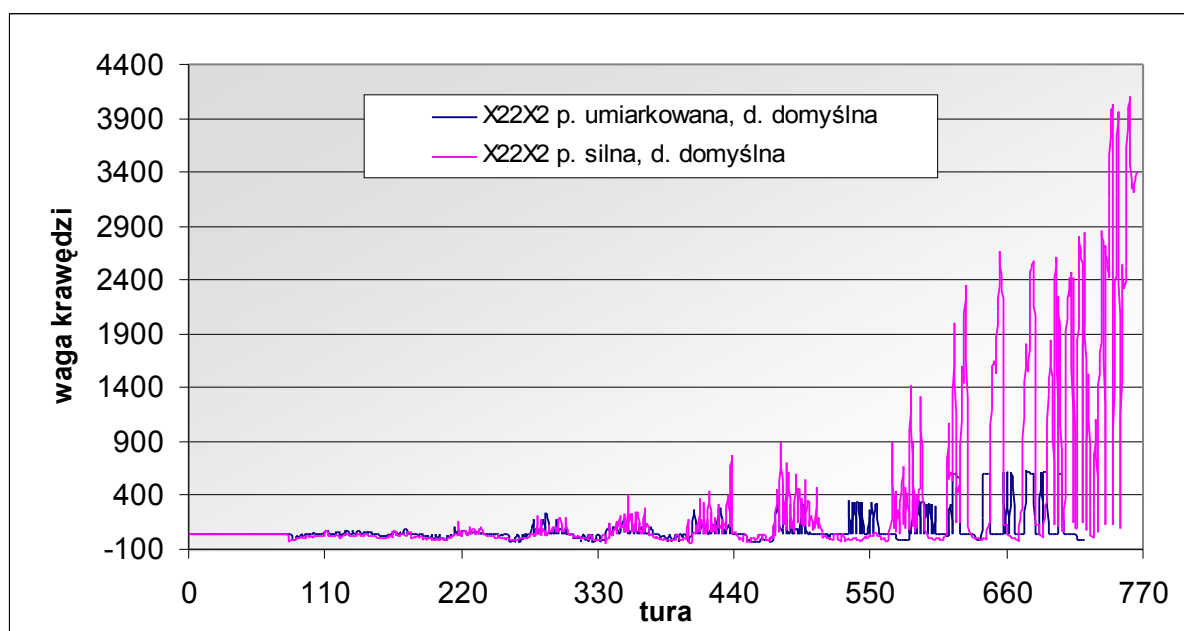


Rysunek 10: Model-D, mały ruch – waga mniej obciążonej krawędzi w czasie

Późny zapłon algorytmu wynika ze sposobu zapisywania modelu – trasy, którymi nie przejechał jeszcze żaden pojazd nie są brane pod uwagę przy obliczaniu wzorców. Wyraźnie jednak widać, iż od momentu uruchomienia się mechanizmu waga krawędzi zaczyna być systematycznie równomiernie obniżana, w podobnym stopniu przez oba algorytmy. Algorytm lepszy przewagę swą zawdzięcza głównie faktowi, iż włączył się nieco szybciej i od razu w pełnym dostępnym zakresie zaczął „zachęcać” do wyboru trasy objazdowej.

Jak natomiast wyglądała sytuacja na krawędzi przed zwężeniem?



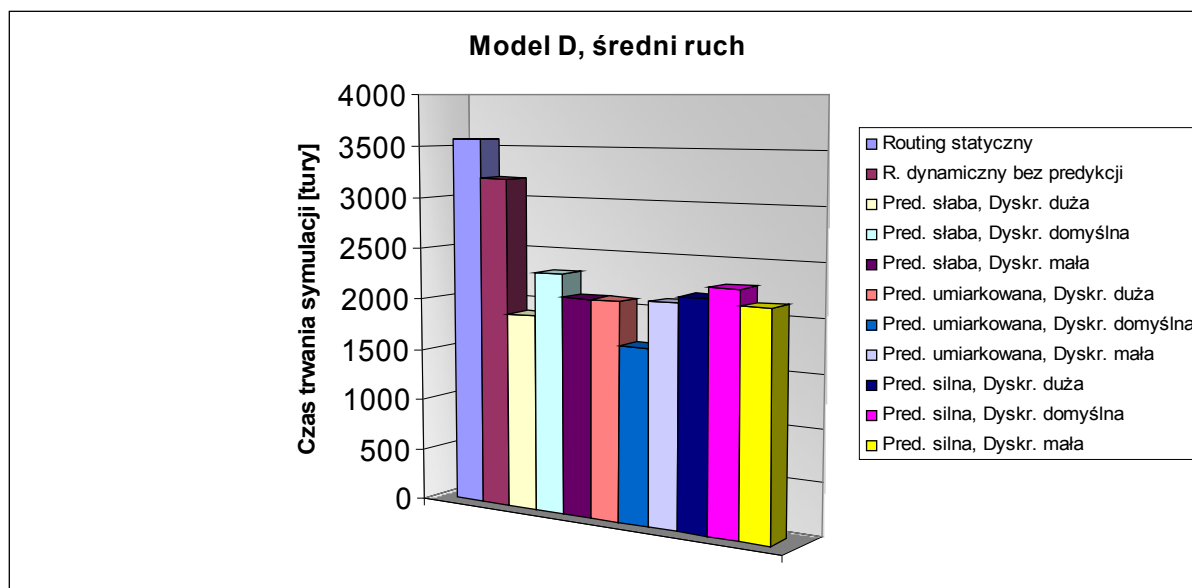


Rysunek 11: Model-D, mały ruch – waga mocniej obciążonej krawędzi w czasie

Jak widać, oba mechanizmy zachowały się dość podobnie (duża skala wykresu nie pozwala dobrze tego ocenić – w „wyniesieniach” pojawiających się na wykresie waga krawędzi osiągała wartość ponad 350 (wartość dla pustej drogi to 85). Efekt na końcu jest rezultatem tego, iż przez pewien czas, już po zakończeniu generacji pojazdów (gdy droga dojazdowa do węzła X22 opróżnia się), na krawędzi X22X2 nadal panuje korek, którego nie ma możliwości rozproszyć – pojazdy, które wjechały na jakąś ulicę, nie mogą zmienić swej trasy aż do przejechania najbliższego skrzyżowania. Pojawia się więc niemożliwy do rozproszenia korek, który skutkuje stałym wysokim natężeniem ruchu na tej krawędzi. W rezultacie pojawiają się bardzo pewne wzorce, które cały czas wzmacniają się. Niewątpliwym jest, że gdyby na końcu symulacji wygenerować jeszcze jedną grupę pojazdów, praktycznie żaden z nich nie pojechałby najkrótszą drogą, wszystkie wybrałyby obwodnicę (na tym właśnie polega uczenie się mechanizmu predykcji)

#### 5.2.2.2. Średnie natężenie ruchu

Dla średniego natężenia ruchu różnica pomiędzy rezultatami bez wykorzystania mechanizmu predykcji a z jego wykorzystaniem są jeszcze bardziej wyraźne



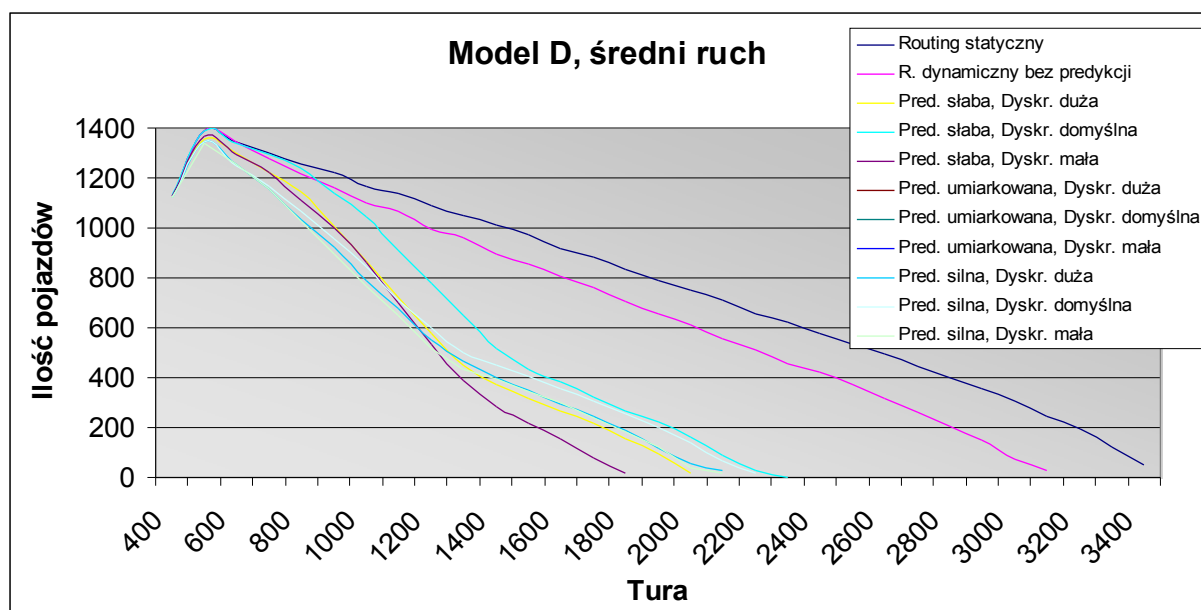
Rysunek 12: Model D, średnie natężenie ruchu – wykres czasów trwania symulacji

Konfiguracja	Czas trwania
Routing statyczny	3563
R. dynamiczny bez predykcji	3196
Pred. słaba, Dyskr. duża	1888
Pred. słaba, Dyskr. domyślna	2306
Pred. słaba, Dyskr. mała	2090
Pred. umiarkowana, Dyskr. duża	2090
Pred. umiarkowana, Dyskr. domyślna	1677
Pred. umiarkowana, Dyskr. mała	2116
Pred. silna, Dyskr. duża	2181
Pred. silna, Dyskr. domyślna	2282
Pred. silna, Dyskr. mała	2138

Konfiguracja	Średni czas trwania
Predykcja słaba	2094,666667
Predykcja umiarkowana	1961
Predykcja silna	2200,333333
Dyskretyzacja duża	2053
Dyskretyzacja domyślna	2088,333333
Dyskretyzacja mała	2114,666667

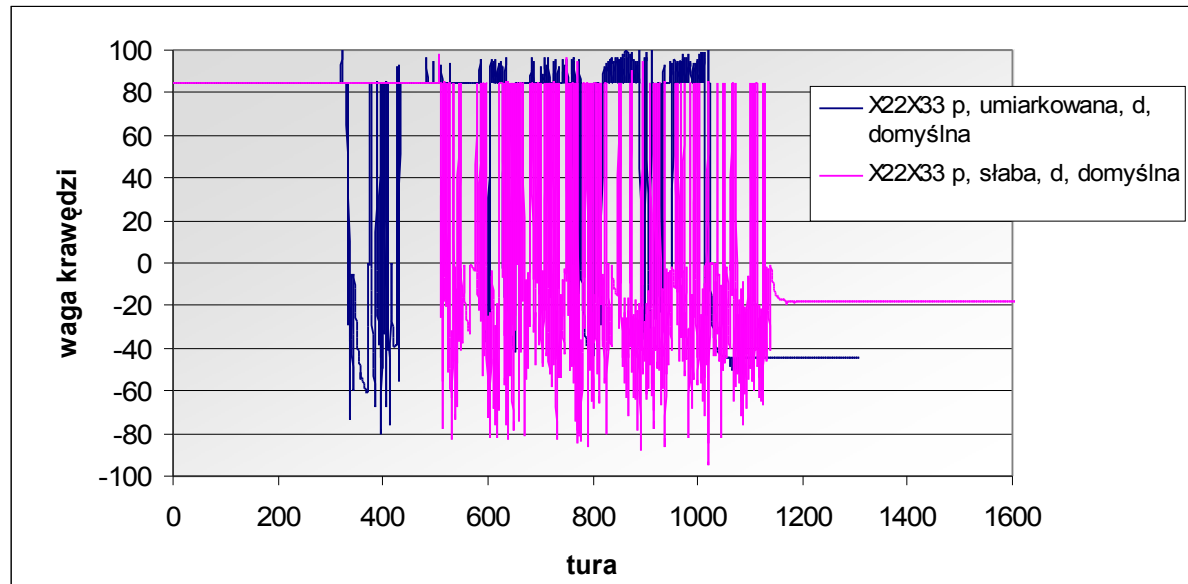
Ponownie, najlepszy wynik udało się uzyskać dla symulacji z predykcją umiarkowaną, trudno jednak dostrzec wyraźne tendencje występujące w pozostałych konfiguracjach. Dyskretyzacja domyślna, wygrywająca wyraźnie w połączeniu z umiarkowanymi ustawieniami predykcji, okazuje się być najbliższa dla konfiguracji bardziej skrajnych.

Wykres ilości pojazdów w czasie prezentuje się następująco:



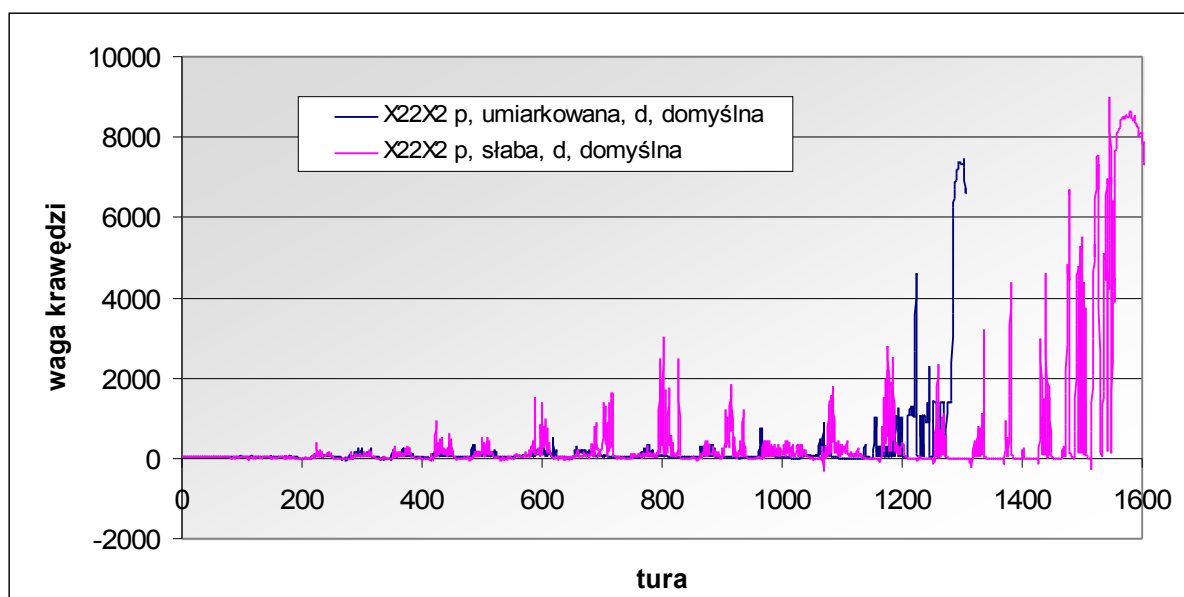
Rysunek 13: Model D, średnie natężenie ruchu – wykres ilości pojazdów w modelu

Jeśli chodzi o wagi krawędzi, zachowują się one bardzo podobnie jak w przypadku małego natężenia ruchu. Dla krawędzi mniej obciążonej można powiedzieć, że wykres jest nieomal identyczny



Rysunek 14: Model-D, średni ruch – waga mniej obciążonej krawędzi w czasie

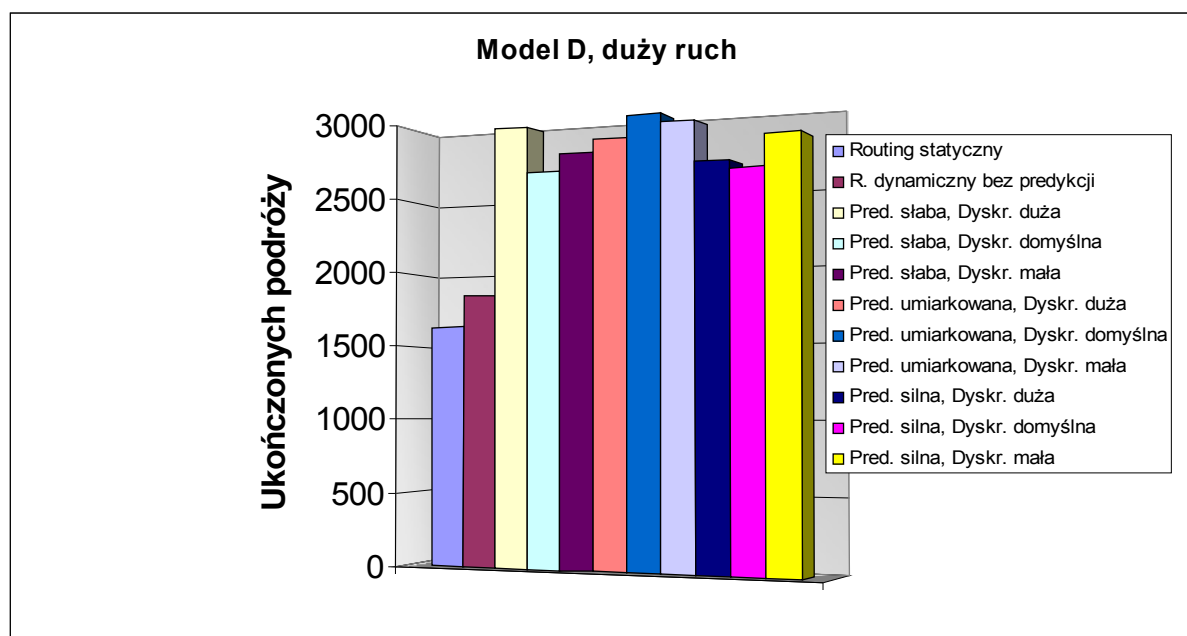
W przypadku ścieżki bardziej obciążonej, następuje dokładnie ten sam efekt jak dla ruchu słabego, wzmocniony dodatkowo faktem, iż ilość oczekujących pojazdów jest większa. Stąd wykres osiąga bardzo wysokie skale:



Rysunek 15: Model-D, średni ruch – waga mocniej obciążonej krawędzi w czasie

### 5.2.2.3. Duże natężenie ruchu

Dla dużego natężenia ruchu, różnica pomiędzy wynikami bez i z predykcją jest niemal dwukrotna. Prezentowane poniżej wyniki prezentują ilość pojazdów, które ukończyły podróż po 4000 tur symulacji.

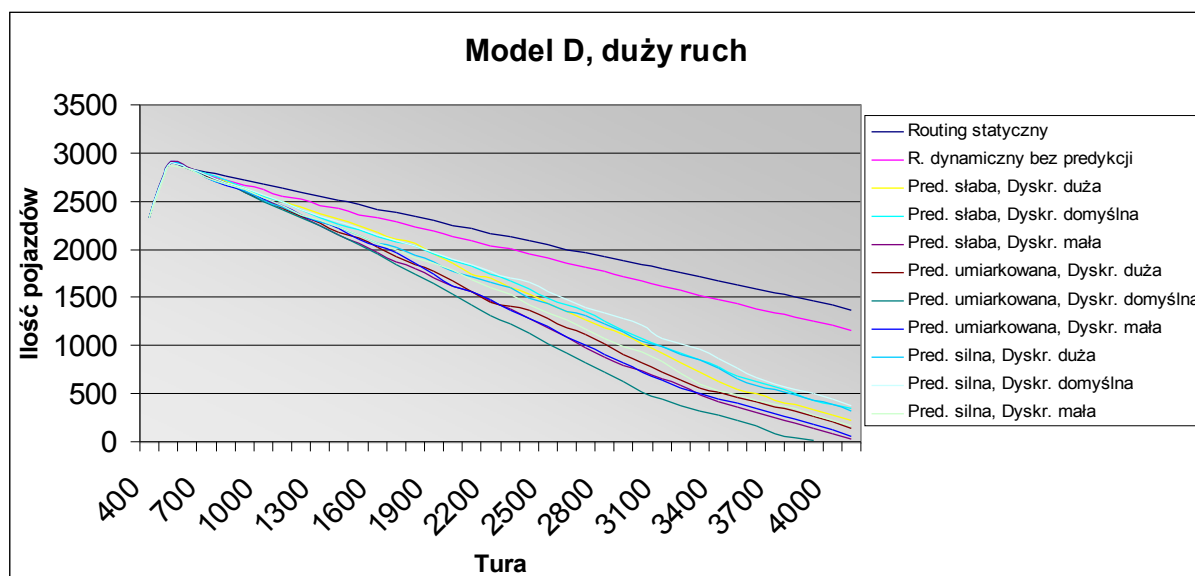


Rysunek 16: Model D, duże natężenie ruchu – ilość zakończonych podróży

Konfiguracja	Ukończonych podróży
Routing statyczny	1628
R. dynamiczny bez predykcji	1839
Pred. słaba, Dyskr. duża	2967
Pred. słaba, Dyskr. domyślna	2658
Pred. słaba, Dyskr. mała	2773
Pred. umiarkowana, Dyskr. duża	2855
Pred. umiarkowana, Dyskr. domyślna	3000
Pred. umiarkowana, Dyskr. mała	2942
Pred. silna, Dyskr. duża	2673
Pred. silna, Dyskr. domyślna	2617
Pred. silna, Dyskr. mała	2829

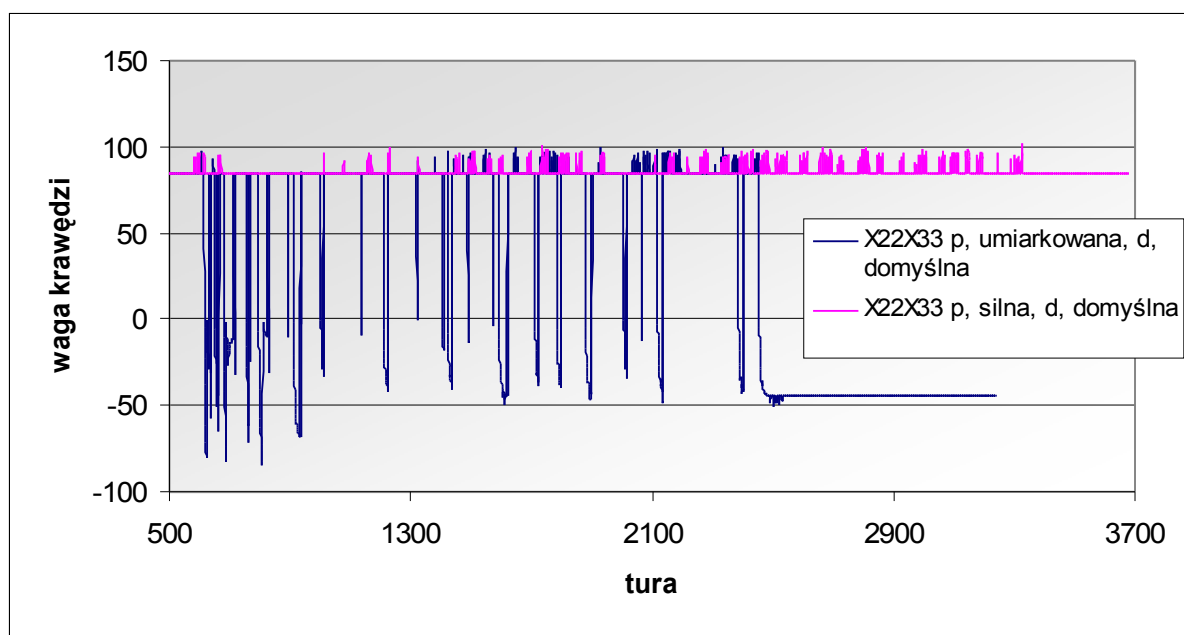
Konfiguracja	Średnia liczba ukończonych podróży
Predykcja słaba	2799,333
Predykcja umiarkowana	2932,333
Predykcja silna	2706,333
Dyskretyzacja duża	2831,667
Dyskretyzacja domyślna	2758,333
Dyskretyzacja mała	2848

Oczywiście w tym przypadku najlepszym wynikiem jest wynik jak najwyższy - po-  
nownie osiągnięty dla konfiguracji z predykcją umiarkowaną i dyskretyzacją domyślną. Nale-  
ży przy tym zaznaczyć, że wynik ten został osiągnięty 200 tur wcześniej – co świadczy o tym,  
że ta konfiguracja jest wyraźnie najlepsza dla tego modelu ruchu. Pozostałe znacząco odbie-  
gają od niej – zarówno te z identycznymi ustawieniami predykcji jak i te z taką samą dyskre-  
tyzacją.



Rysunek 17: Model D, duże natężenie ruchu – ilość pojazdów

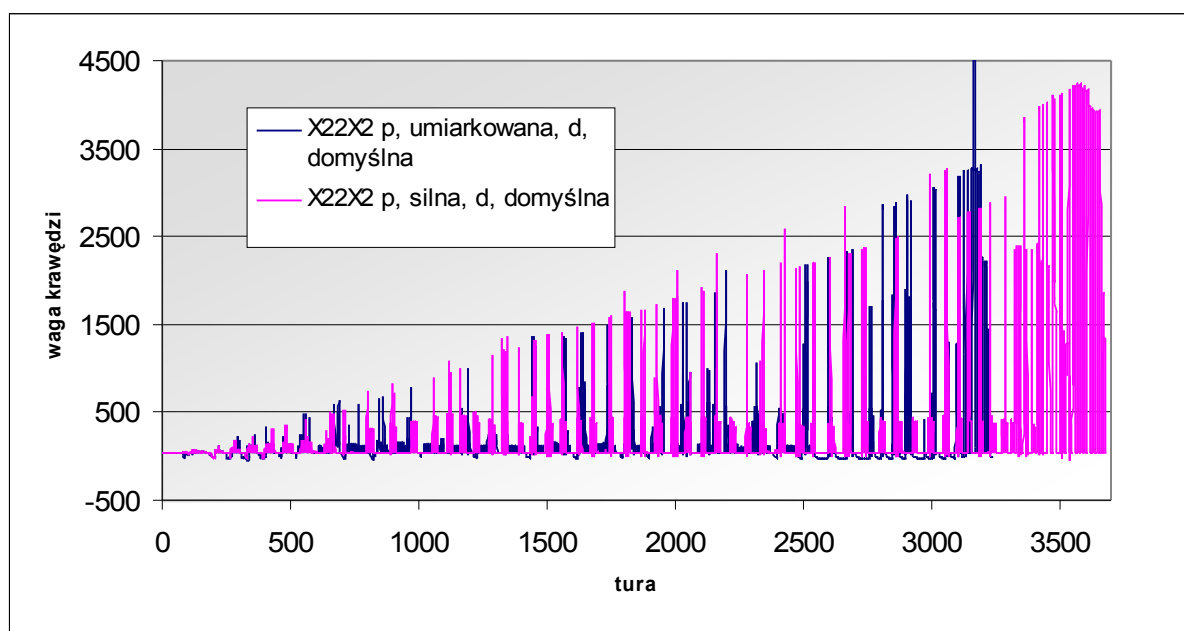
Wykres wag wybranych krawędzi przynosi w zasadzie jedną niespodziankę:



Rysunek 18: Model-D, duży ruch – waga mniej obciążonej krawędzi w czasie

Jak widać, wybrana konfiguracja w ogóle nie odciążała, a wręcz dociążała krawędź objazdową! Zapewne wynikało to z silnych ustawień predykcji, które okazały się niemożliwe do spełnienia, np. ze względu na minimalną ilość wzorców, które muszą daną sekwencję potwierdzić. Pamiętać należy, że krawędzie nieużywane nie są liczone w mechanizmach predykcji, aby uniknąć sytuacji, w której bardzo kluczowa krawędź, będąca daleko od wjazdów do modelu (a więc taka, do której pojazdy dotrą późno) przez tego typu wzorzec zaczęłaby ściągać dodatkowe pojazdy.

Jeśli chodzi o drugi wykres, jest już czymś właściwie znanym z poprzednich konfiguracji natężenia ruchu:

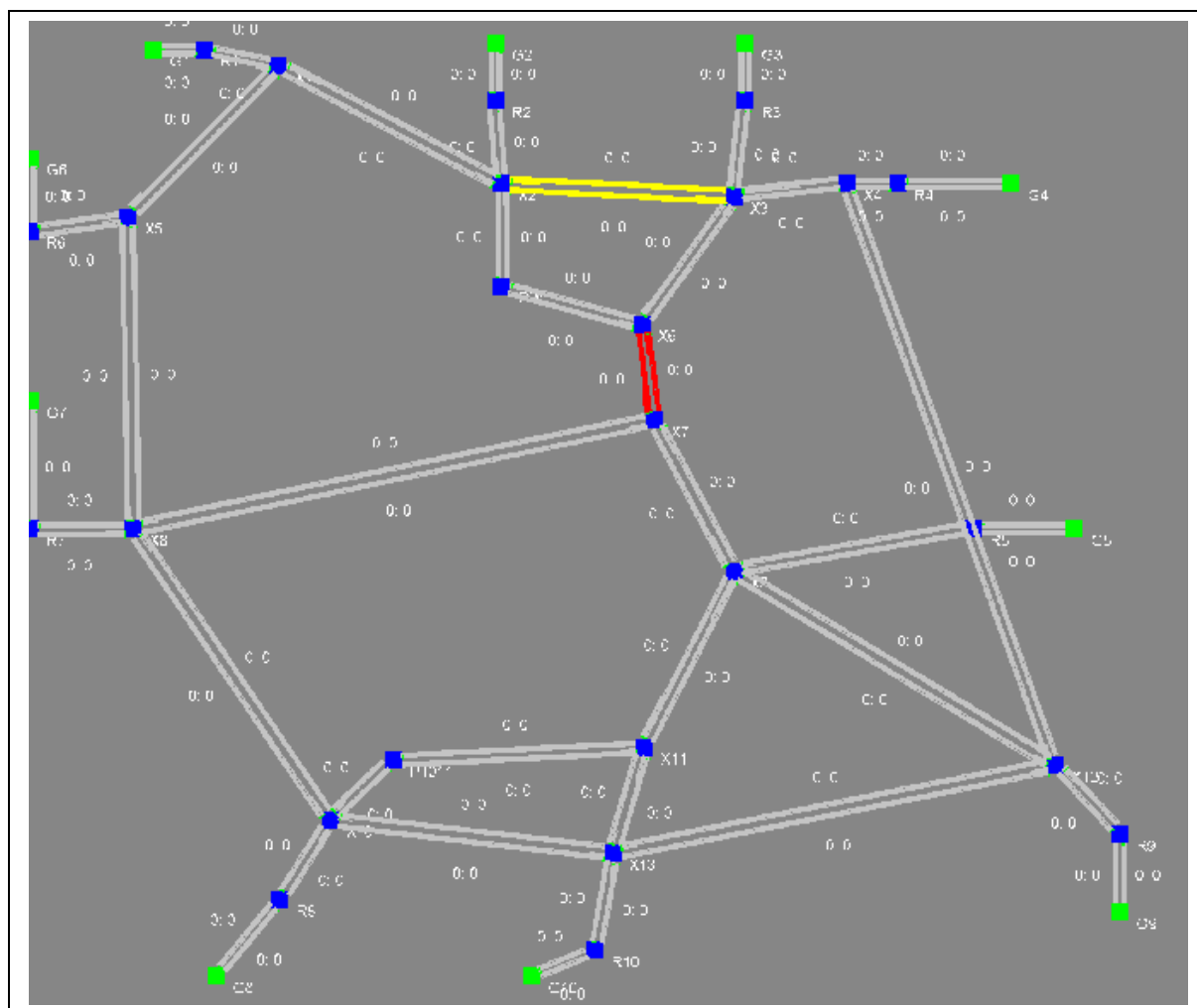


Rysunek 19: Model-D, duży ruch – waga mocniej obciążonej krawędzi w czasie

Tutaj także, podobnie jak poprzednio, następuje swoisty rezonans, stopniowe szybkie zwiększanie wagi krawędzi dociążonej przy pomocy bardzo pewnych (bo długo budowanych) wzorców. Faktycznie, taka sytuacja, we wdrażanym systemie, byłaby bardzo pożądana.

### 5.2.3. Model Krakowa

Model ten opiera się na rzeczywistej mapie ulic w centrum Krakowa. Z tego względu nie występują tu wymuszone, oczywiste objazdy tak, jak w bardziej teoretycznym modelu D. Często dwa węzły łączy tylko jedna droga, lub ewentualna droga alternatywna nie jest zbyt konkurencyjna. Konfiguracja modelu w postaci pliku XML znajduje się w załączniku w rozdziale 7.5. Model jest bardzo rozległy – jego szerokość sięga 900 jednostek, a wysokość 800, najdłuższe połączenia mają długość ok. 1000, średnia długość pojedynczego odcinka to ok. 350. W modelu występuje 10 wjazdów i 41 ulic (trypasmowych), część z nich posiada dodatkowo pas do skrętu.



Rysunek 20: model Krakowa

Wyszczególnione na powyższym rysunku krawędzie były w tym modelu obserwowane jako szczególnie ciekawe ze względu na występujące na nich natężenie ruchu – krawędź oznaczona kolorem czerwonym była miejscem występowania bardzo silnych korków, natomiast krawędź oznaczona kolorem żółtym była stosunkowo mało obciążona w stosunku do krawędzi z nią sąsiadujących.

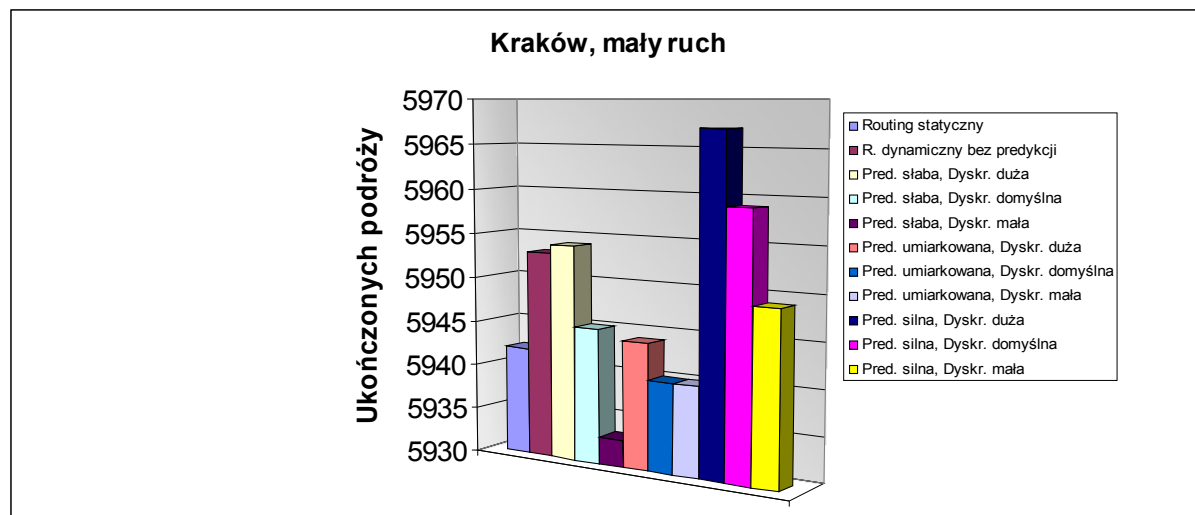
Modele ruchu, które zostały tu użyte, bazują na przybliżeniu ruchu w centrum miasta w godzinach szczytu – gdy z zewnętrznych obszarów miasta duża ilość pojazdów przemieszcza się do centrum, właściwie po wszystkich możliwych drogach. Stąd nie znajdzie się tu zbyt wyraźnych różnic w otrzymanych rezultatach. Pokazują one jednak, że umiejętne dobór współczynników algorytmów predykcji do modelu jest w stanie choćby nieznacznie usprawnić ruch na drogach. Oczywiście, dobór takich współczynników do rzeczywistego modelu jest procesem wymagającym bardzo wielu eksperymentów, w zasadzie wymaga on sprawdzenia praktycznie wszystkich możliwych układów, co dla symulacji dużych modeli (takich jak model miasta Krakowa) jest praktycznie niemożliwe. Na podstawie osiągniętych rezultatów można jednak wnioskować, jakie cechy powinny posiadać dobrze dobrane współczynniki, aby nastąpiła poprawa.

Konfiguracja standardowa (średni) ruchu Znajduje się w załączniku w rozdziale 7.6. Jest to ruch generowany pomiędzy każdą parą węzłów, dla każdej pary węzłów generowa-



nych jest ok. 800 pojazdów, a więc w całości generowanych jest ponad 30 000 pojazdów. Ze względu na to, że generacja odbywa się dla wszystkich par wjazdów, obciążenie dróg jest z grubsza równomierne.

#### 5.2.3.1. Niskie natężenie ruchu



Rysunek 21: Kraków, niskie natężenie ruchu – ilość ukończonych podróży

Konfiguracja	Ukończonych podróży
Routing statyczny	5942
R. dynamiczny bez predykcji	5953
Pred. słaba, Dyskr. duża	5954
Pred. słaba, Dyskr. domyślna	5945
Pred. słaba, Dyskr. mała	5933
Pred. umiarkowana, Dyskr. duża	5944
Pred. umiarkowana, Dyskr. domyślna	5940
Pred. umiarkowana, Dyskr. mała	5940
Pred. silna, Dyskr. duża	5967
Pred. silna, Dyskr. domyślna	5959
Pred. silna, Dyskr. mała	5949

Konfiguracja	Średnia liczba ukończonych podróży
Predykcja słaba	5944
Predykcja umiarkowana	5941,333333
Predykcja silna	5958,333333
Dyskretyzacja duża	5955
Dyskretyzacja domyślna	5948
Dyskretyzacja mała	5940,666667

Jak widać wyraźnie, na rzeczywistym modelu najlepszy indywidualnie wynik zbiega się z wynikiem najlepszym dla średniej z eksperymentów. Świadczy to dość wyraźnie o tym, że dla tego modelu preferowanym rozwiązaniem jest konfiguracja „agresywna” – o dużym współczynniku ufności (co eliminuje błędne wzorce) i równocześnie dużym wpływem (co wzmacnia efekt działania znalezionych wzorców).

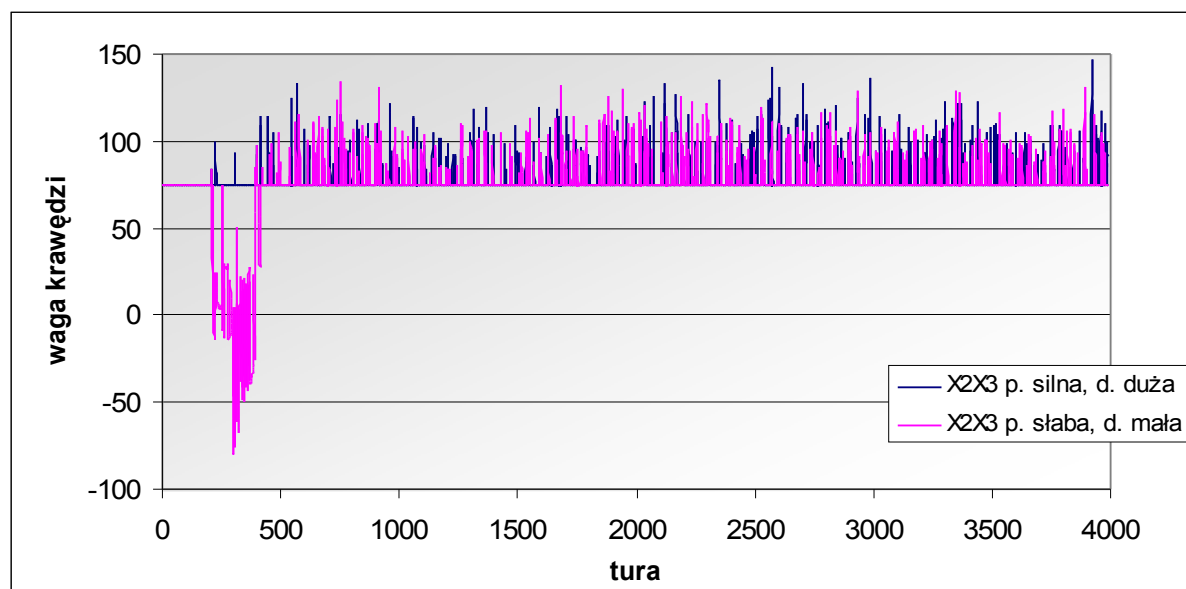
Oczywiście, różnice w uzyskanych wynikach nie są duże, wynika to jednak – jak zostało wspomniane na wstępie – z ogólnego natężenia ruchu w modelu – nie ma w nim całkowicie nieużywanych ścieżek dla routingu statycznego (tak jak w przypadku modelu D), gdyż ruch generowany jest pomiędzy wieloma punktami modelu.

Zwraca uwagę fakt, iż tym razem wyniki dla routingu statycznego nie odbiegają daleko od pozostałych wyników – świadczy to o tym, że dla takiego natężenia ruchu wybór najkrótszej ścieżki, niezależnie od obciążenia dróg, jeszcze się sprawdza.

Wykres ilości pojazdów od czasu zostanie pominięty, gdyż nie wnosi nic nowego do uzyskanych rezultatów – różnice dla tak małego ruchu pomiędzy poszczególnymi konfiguracjami są na tyle nieznaczne, że na wykresie nie znajdzie się zbyt wiele rozbieżności. Wykres jest oczywiście umieszczony w dołączonym arkuszu programu Excel.

Jeśli chodzi o wykresy wag krawędzi grafu, zostały one zrealizowane dla dwóch krawędzi (por. Rysunek 20: model Krakowa i opis poniżej tego rysunku), dla których otrzymano najciekawsze wyniki. Dla niewielkiego natężenia ruchu rezultaty są raczej mało interesujące.

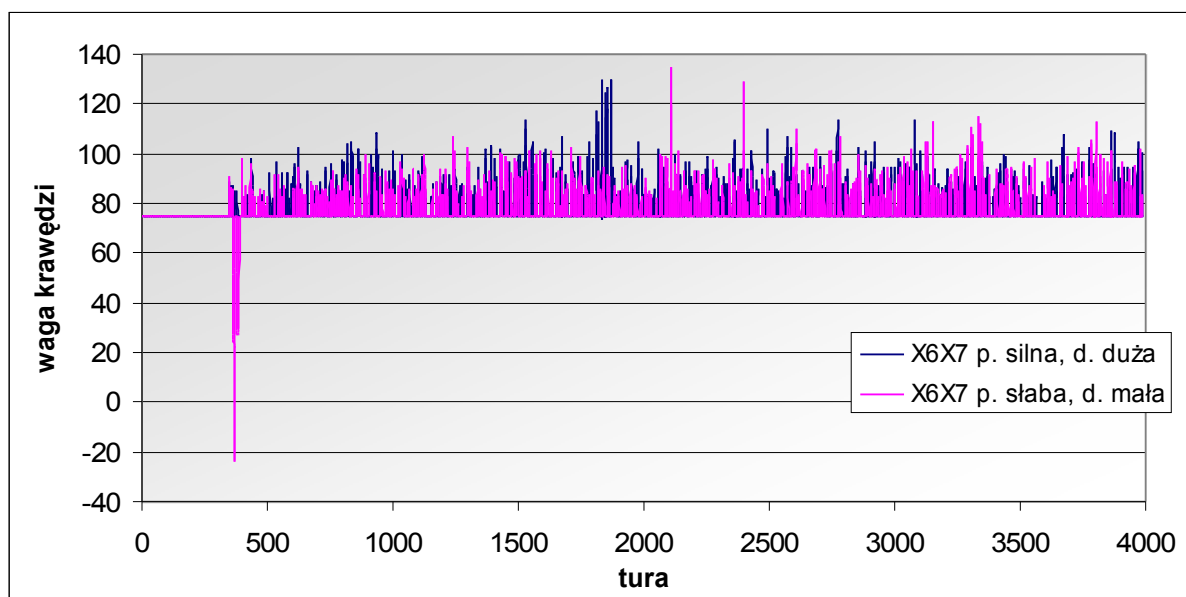
Wykres dla krawędzi mniej obciążonej (oznaczonej kolorem żółtym na rysunku modelu Krakowa):



Rysunek 22: Kraków, niewielki ruch - waga mniej obciążonej krawędzi w czasie

Jak widać, poza pojedynczym wyskokiem w początkowej fazie symulacji (wynikającej zapewne ze zbyt szybkiego przyjęcia sekwencji za wzorzec – błędnie) oba mechanizmy oscylowały wokół podobnego stanu. Wynika to z faktu, iż przy niewielkim natężeniu ruchu

nie istnieje zbyt wyraźna potrzeba przekierowywania pojazdów na lepsze trasy. Wykres dla krawędzi bardziej obciążonej pokazuje jeszcze mniejszą różnicę pomiędzy obydwoimi mechanizmami.

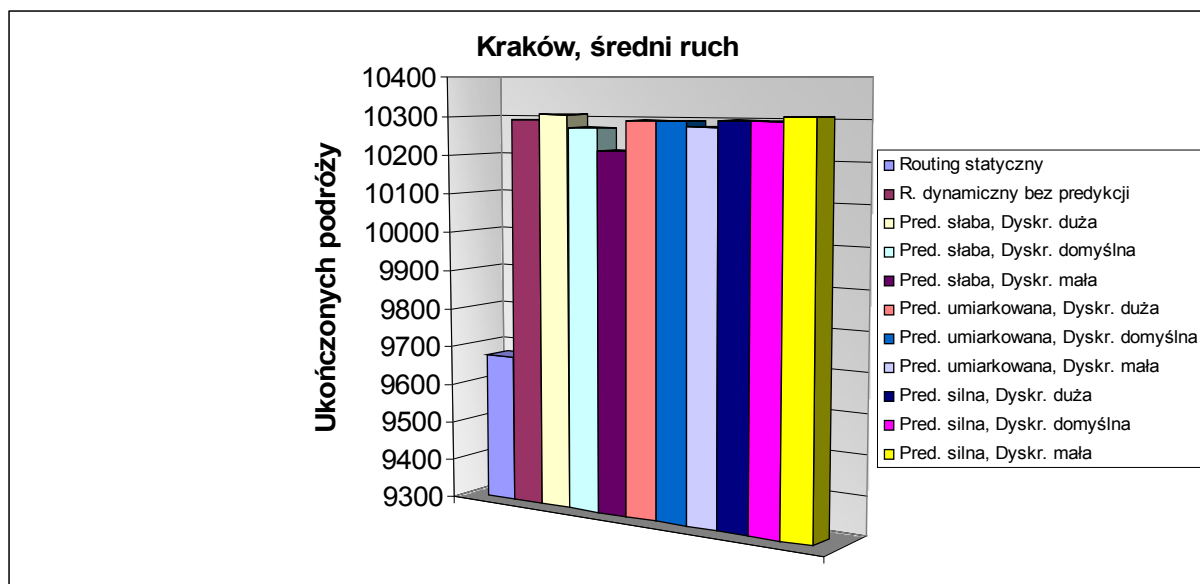


Rysunek 23: Kraków, niewielki ruch - waga mocniej obciążonej krawędzi w czasie

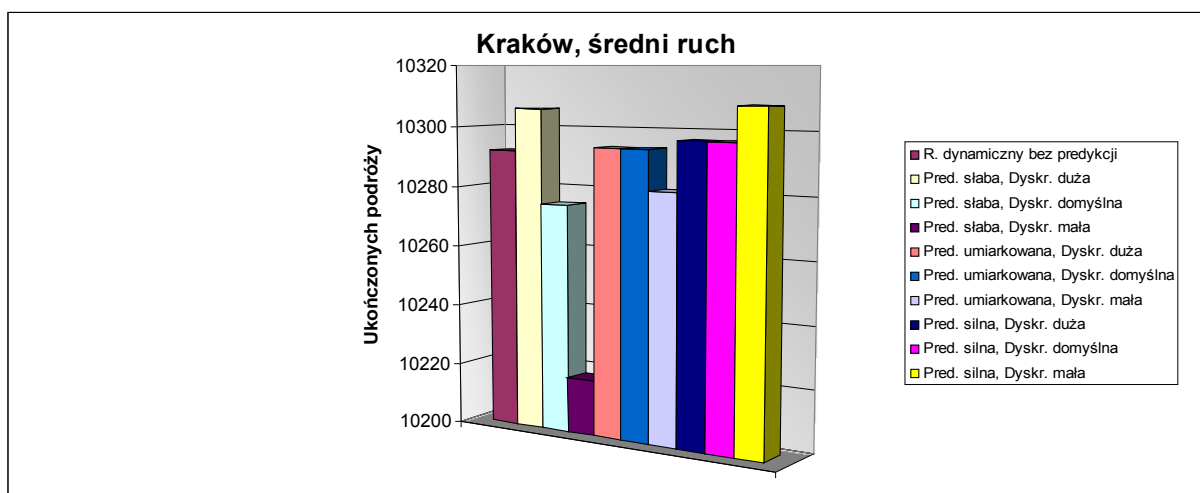
Jak widać, różnice tutaj są jeszcze „węższe” (bardziej krótkotrwałe), ponadto początkowy pik konfiguracji słabszej jest o połowę mniejszy, co wynika także z szybkiego wzrostu natężenia ruchu na tej krawędzi.

#### 5.2.3.2. Średnie natężenie ruchu

Dwukrotne zwiększenie natężenia ruchu dało wyraźniejsze efekty. Zaprezentowane poniżej zostały dwa wykresy ilości podróży ukończonych w czasie 4000 tur – jeden zawierający wynik dla routingu statycznego oraz drugi, który go nie zawiera – co umożliwia dostrzeżenie wyraźniej różnic pomiędzy poszczególnymi konfiguracjami predykcji.



Rysunek 24: Kraków, średnie natężenie ruchu – ilość ukończonych podróży (z routingiem statycznym)



Rysunek 25: Kraków, średnie natężenie ruchu – ilość ukończonych podróży (bez routingu statycznego)

Konfiguracja	Ukończonych podróży
Routing statyczny	9677
R. dynamiczny bez predykcji	10292
Pred. słaba, Dyskr. duża	10306
Pred. słaba, Dyskr. domyślna	10275
Pred. słaba, Dyskr. mała	10218
Pred. umiarkowana, Dyskr. duża	10294
Pred. umiarkowana, Dyskr. domyślna	10294
Pred. umiarkowana, Dyskr. mała	10281
Pred. silna, Dyskr. duża	10297

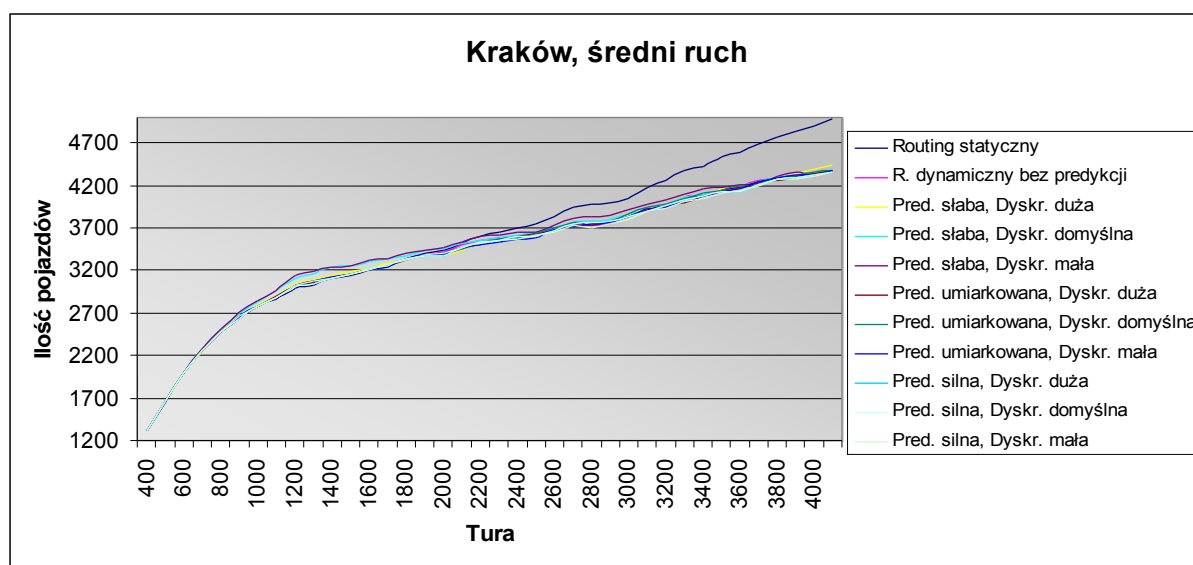
Pred. silna, Dyskr. domyślna	10297
Pred. silna, Dyskr. mała	10308

Konfiguracja	Średnia liczba ukończonych podróży
Predykcja słaba	10266,33333
Predykcja umiarkowana	10289,66667
Predykcja silna	10300,66667
Dyskretyzacja duża	10299
Dyskretyzacja domyślna	10288,66667
Dyskretyzacja mała	10269

Dla tego natężenia ruchu dwa wyniki okazały się lepsze od zwycięzcy dla niewielkiego natężenia ruchu. Należy jednak zwrócić uwagę, że w rankingu średnich nadal przewagę uzyskuje tamto rozwiązanie, a zarówno jego wynik jak i średni są wyższe niż wynik dla routingu dynamicznego bez predykcji. Takie rezultaty pozwalają wyciągnąć dwa wnioski:

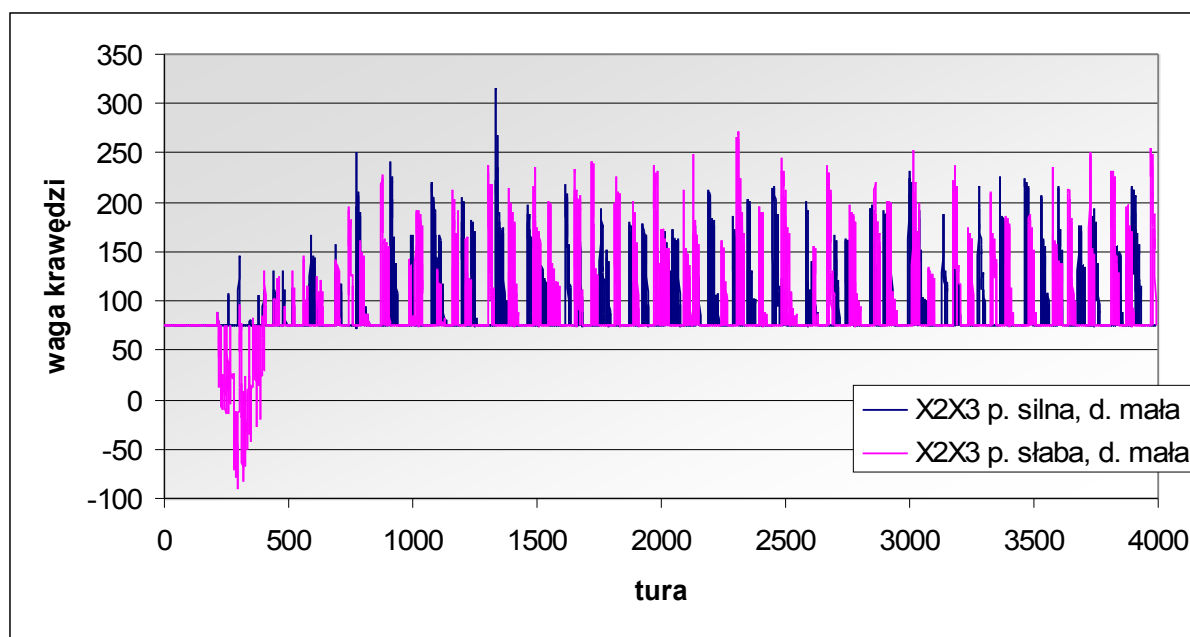
- Dla specyficznych przypadków (takich jak określone natężenia ruchu) rozwiązania średnio słabsze mogą okazać się bardziej skuteczne – a więc jeśli dla średniego natężenia ruchu lepsze byłoby rozwiązanie z dyskretyzacją „małą”, może warto jest wprowadzić możliwość dynamicznej zmiany konfiguracji mechanizmu predykcji (lub chociaż samej dyskretyzacji), by wykorzystać ją np. w okresie, o którym wiadomo, że taką charakterystyką ruchu się cechuje (np. czas po godzinach szczytu)
- Poprawny dobór parametrów predykcji musi być taki, aby rezultat ich działania był zawsze przynajmniej zbliżony (najlepiej: lepszy) od wyników bez predykcji. Dzięki temu nie zostanie utracony zysk osiągany w czasie, gdy wyniki osiągane dla takiej konfiguracji są najlepsze. Tak więc przed ewentualnym wdrożeniem tego typu rozwiązania konieczne jest przetestowanie go dla wszystkich natężeń ruchu.

Dla średniego natężenia ruchu, na wykresie ilości pojazdów wyraźnie widać „ucieczkę” krzywej odpowiadającej za wyniki osiągane z routingiem statycznym.



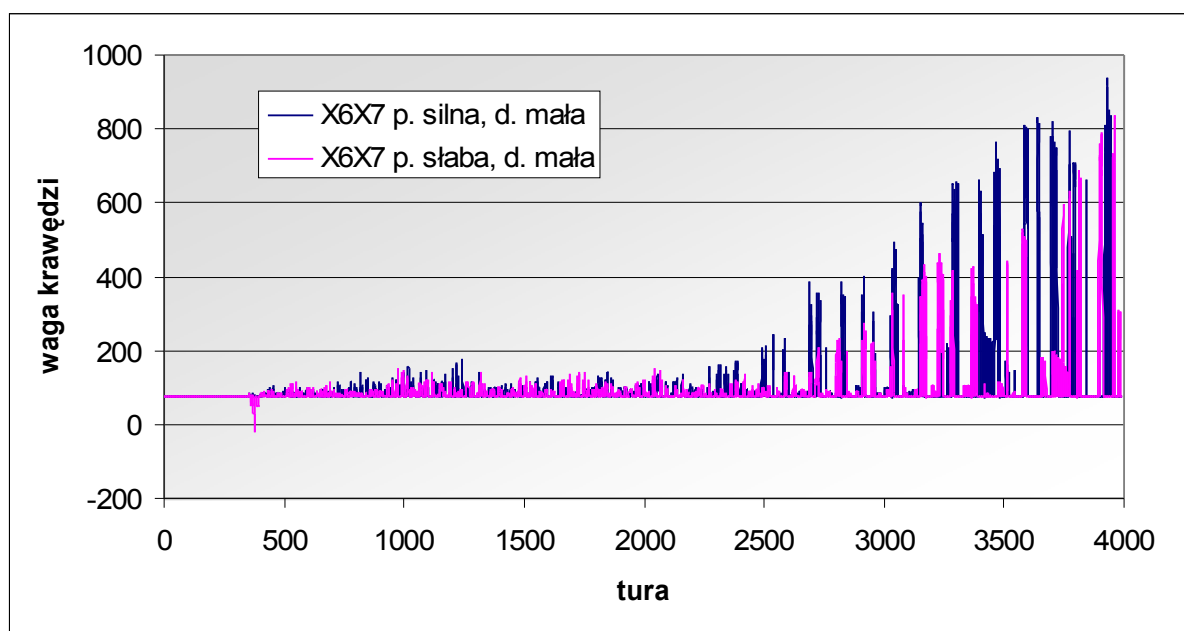
Rysunek 26: Kraków, średnie natężenie ruchu – ilość pojazdów

Najciekawiej prezentują się dla tego natężenia ruchu wyniki na poszczególnych krawędziach. Dla krawędzi mniej obciążonej różnica jest nieznaczna:



Rysunek 27: Kraków, średni ruch - waga mniej obciążonej krawędzi w czasie

Daje się jednak dostrzec, że zwłaszcza w dalszej części symulacji predykcja gorsza (kolor różowy) w większym stopniu dociąża tę krawędź. Natomiast wykres dla krawędzi bardziej obciążonej jest doskonałym przykładem zachowania mechanizmu predykcji dla krawędzi dociążanej.

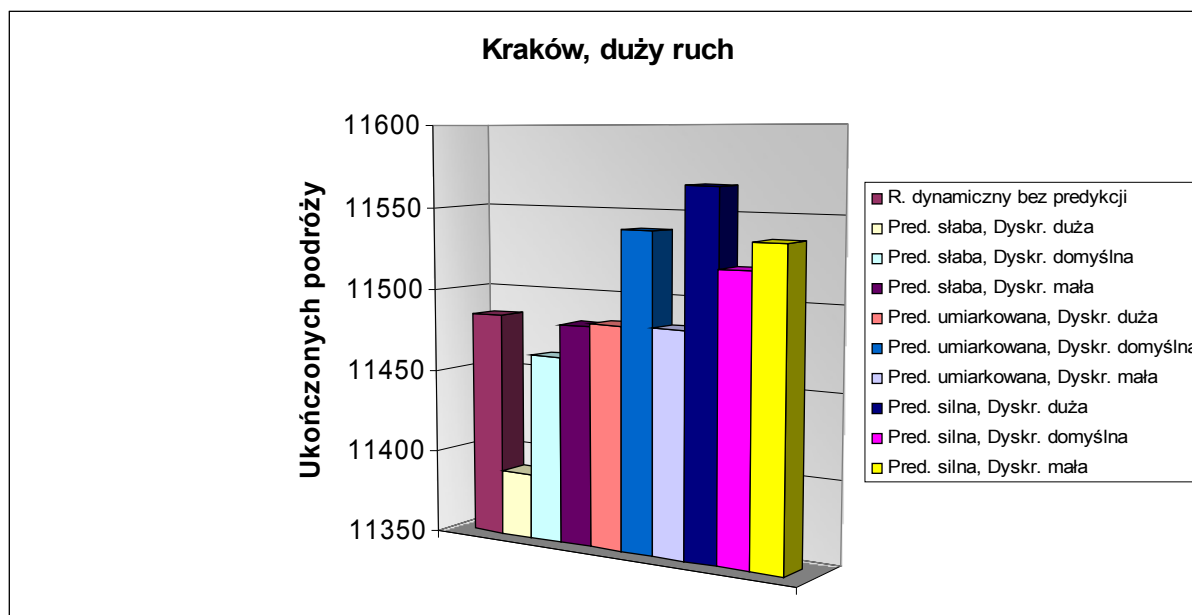


Rysunek 28: Kraków, średni ruch - waga mocniej obciążonej krawędzi w czasie

Jak widać, predykcja, która okazała się lepsza przede wszystkim znacznie bardziej dociąża krawędzie, na których tworzą się korki. Szczególnie dobry efekt otrzymujemy tu dla ruchu, który jest na granicy przepustowości modelu (tak jak właśnie średnie natężenie ruchu zaproponowane dla modelu Krakowa) – nie wszystkie krawędzie są zapchane (tak, jak w przypadku dużego ruchu, więc zwiększanie wagi krawędzi obciążonych skutkuje rzeczywiście przekierowaniem części pojazdów na lepsze, mniej zatłoczone trasy.

#### 5.2.3.3. Duże natężenie ruchu

Dla dużego natężenia ruchu, ponownie rezultaty wskazują na to, iż dla tego modelu najlepszym z rozwiązań jest konfiguracja „agresywna” – o dużym wpływie (duża dyskretyzacja) wzorców o dużej ufności (silna predykcja).



Rysunek 29: Kraków, duże natężenie ruchu – ilość ukończonych podróży (bez routingu statycznego)

Wynik routingu statycznego został usunięty z tego wykresu, gdyż wyraźnie odstawał od pozostałych wyników i zmniejszał czytelność wykresu (zbyt duża skala)

Konfiguracja	Ukończonych podróży
Routing statyczny	10703
R. dynamiczny bez predykcji	11485
Pred. słaba, Dyskr. duża	11389
Pred. słaba, Dyskr. domyślna	11462
Pred. słaba, Dyskr. mała	11482
Pred. umiarkowana, Dyskr. duża	11484
Pred. umiarkowana, Dyskr. domyślna	11540
Pred. umiarkowana, Dyskr. mała	11484
Pred. silna, Dyskr. duża	11566
Pred. silna, Dyskr. domyślna	11520
Pred. silna, Dyskr. mała	11536

Konfiguracja	Średnia liczba ukończonych podróży
Predykcja słaba	11444,33333
Predykcja umiarkowana	11502,66667
Predykcja silna	11540,66667
Dyskretyzacja duża	11479,66667
Dyskretyzacja domyślna	11507,33333
Dyskretyzacja mała	11500,66667

Zwraca uwagę to, która z konfiguracji wypadła najslabiej Dwukrotnie była nią konfiguracja przeciwna do zwycięskiej – dla konfiguracji predykcji dopuszczającej wzorce o niskiej ufności (predykcja słaba) oraz dyskretyzacji o niewielkim wpływie („mała”). Dla duże-

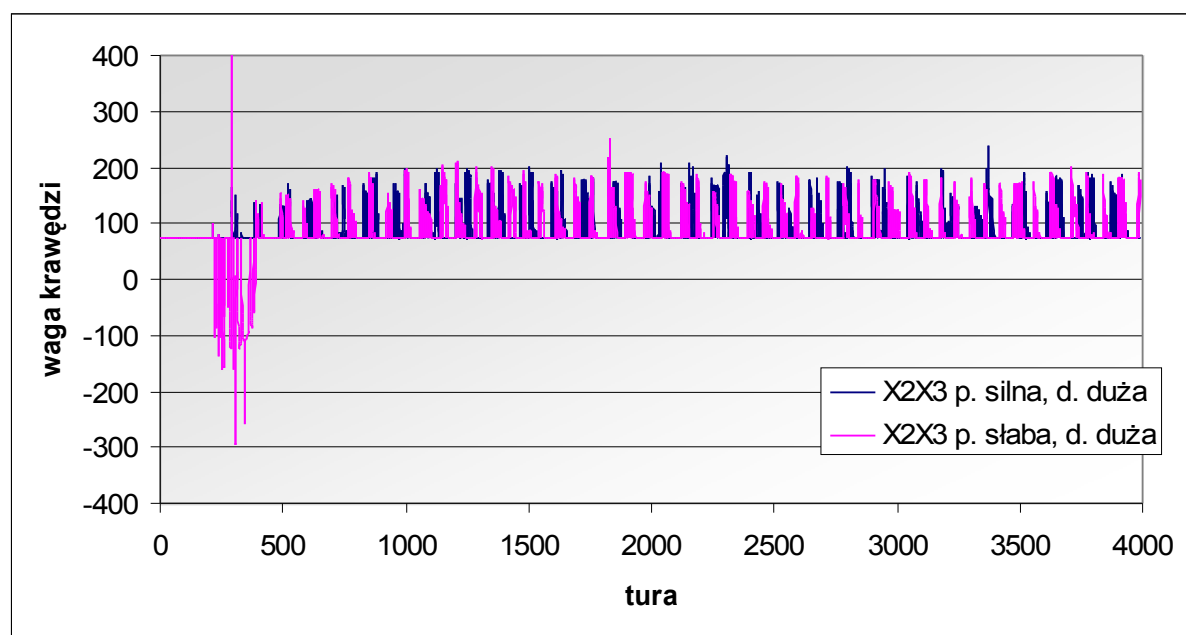


go natężenie ruchu jej wynik był co prawda bliski wynikowi uzyskanemu dla routingu dynamicznego bez predykcji, jednak dla dwóch poprzednich natężeń ruchu wyniki uzyskane były bez porównania gorsze.

Ciekawym jest także fakt, iż dla dużego natężenia ruchu najlepszy średni wynik uzyskała dyskretyzacja domyślna – ponownie dzięki swej dobrej „współpracy” z predykcją umiarkowaną (dla tej predykcji dyskretyzacja domyślna okazała się wyraźnie lepsza od rozwiązań bardziej skrajnych). Świadczy to o fakcie, że jest to konfiguracja obiecująca i należy zwrócić na nią uwagę przy poszukiwaniu optymalnego rozwiązania.

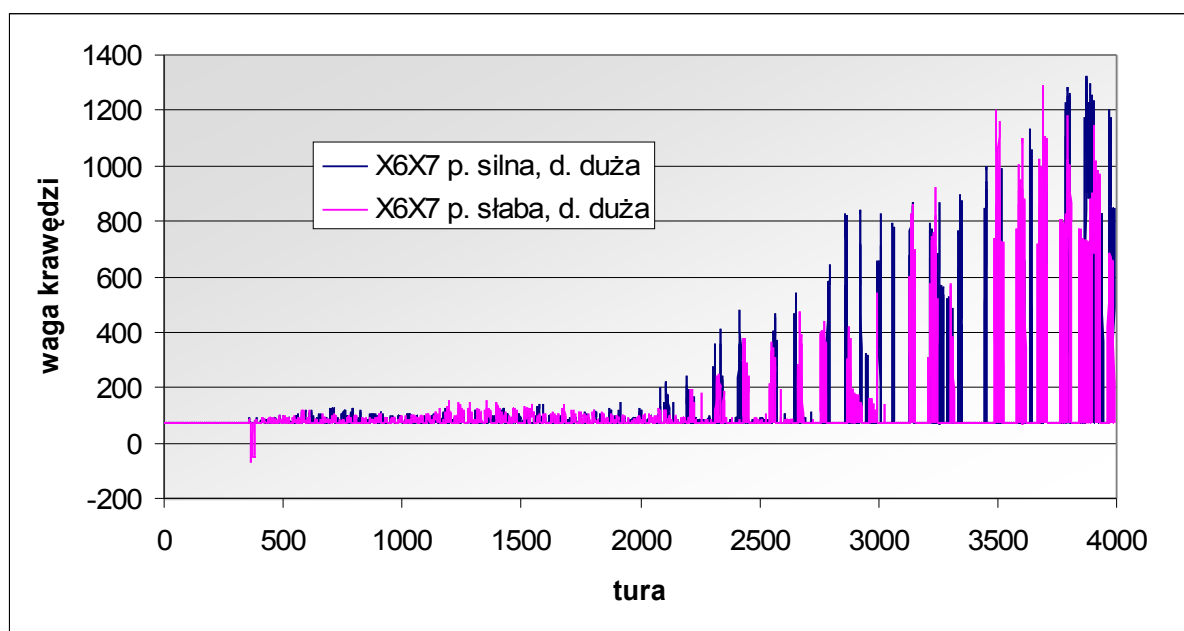
Jeśli chodzi o indywidualne czasy przejazdów na poszczególnych krawędziach, otrzymane rezultaty – spodziewanie – nie powaliły. Wynika to z faktu, że przy takim natężeniu ruchu model jest przesycony, i jakiegokolwiek zmiany niewiele w tym względzie mogą zmienić – przy takiej ilości pojazdów po prostu nie ma możliwości na szybsze przebycie miasta.

Wykres dla mniej obciążonej krawędzi pokazuje, że obie konfiguracje – zarówno najlepsza jak i najgorsza – właściwie zachowują „obojętność” na tę krawędź – natężenie ruchu na niej jest wysokie, nie ma potrzeby jednak dodatkowo go podnosić, gdyż trasa ta nie jest aż tak uczęszczana.



Rysunek 30: Kraków, duży ruch - waga mniej obciążonej krawędzi w czasie

Warto jednak zauważyć charakterystyczną sprawę na temat silnych dyskretyzacji – w połączeniu ze słabą predykcją (taką, która przyjmuje mało wiarygodne wzorce) – tak jak to jest w przypadku serii danych w kolorze różowym – jest bardzo niestabilna, może na tej samej krawędzi w bardzo krótkich odstępach czasu powodować przeciwne rezultaty (tak jak na wykresie powyżej, około 300. tury następują po sobie dwa piki – do góry i do dołu. Takie sytuacje, zdarzające się, jak widać, często na początku symulacji (zwł. dla słabej predykcji) mogą bardzo negatywnie wpływać na wynik całej symulacji.



Rysunek 31: Kraków, duży ruch - waga mocniej obciążonej krawędzi w czasie

Jak zostało wspomniane, dla dużego ruchu, na zapchanej krawędzi następuje dramatyczny wzrost wagi. Niestety, przy takim natężeniu ruchu wzrost taki można zaobserwować na bardzo wielu, zwłaszcza kluczowych, krawędziach – a co za tym idzie zwiększanie ich wag niewiele jest tu w stanie zmienić

#### 5.2.4. Wnioski

Wykonane testy pokazały, że:

- Zaimplementowany mechanizm predykcji wywiera wyraźny wpływ na przebieg symulacji,
- Uniwersalna konfiguracja jest trudna, o ile nie niemożliwa, do znalezienia, jednak ewentualne poszukiwania powinny iść najprawdopodobniej w kierunku predykcji o wysokim współczynniku ufności i o dużym wpływie.
- Konfiguracja opisana wyżej niekoniecznie sprawdzi się we wszystkich przypadkach (dla modelu teoretycznego uzyskane dla niej wyniki były dość przeciętne)
- Dobór niewłaściwych parametrów może prowadzić do pogorszenia zamiast do poprawy rezultatów. Trudno także jednoznacznie wskazać ustawienie domyślne, które byłoby odpowiednie dla każdego typu modelu.

## 6. Podsumowanie

Wprowadzone przez nas zmiany znacznie rozbudowały możliwości i złożoność systemu Kraksim. Wielopasowość była właściwie koniecznością, by model w jakikolwiek sposób odzwierciedlał istniejące sieci dróg, jej realizacja wydaje się poprawnie odzwierciedlać modelowaną rzeczywistość.

Mechanizm predykcji w wersji, jaka została wybrana, jest mechanizmem bardziej złożonym niż wskazywałaby na to jego konstrukcja – dokładna analiza wyników otrzymywanych z jego udziałem pokazuje, że może być to bardzo przydatne rozwiązanie, zarówno w systemie Kraksim, jak i w ewentualnie wdrażanym systemie. Architektura zrealizowanej zmiany umożliwia łatwą rozbudowę lub podmianę zaimplementowanej przez nas realizacji. Jej podstawowymi zaletami jest stosunkowa lekkość (można ją stosować wykorzystując wyłącznie pamięć ram, bez zapisu do jakiegóż bazy danych) i względnie niewielki narzut obliczeniowy.

### 6.1. Propozycje dalszych zmian

1. Rozbudowa modelu o możliwość ustalenia maksymalnej prędkości w całym modelu (na chwilę obecną jest to wprowadzona na sztywno wartość 2) – bardzo dobre na projekt wprowadzający
2. Rozbudowa modelu o możliwość ustalenia maksymalnej prędkości dla poszczególnych krawędzi – bardziej skomplikowana, wymaga także rozbudowy konfiguracji
3. Zmiana mechanizmu wprowadzania na pasy tak, aby pojazdy jeździły „na zamek” – zastąpienie klasy `pl.edu.agh.cs.kraksim.real.MultiLaneRoutingHelper` inną implementacją – prosty projekt wprowadzający
4. Rozbudowa routingu tak, aby mógł następować także przed skrzyżowaniem – skomplikowany projekt, wymaga dobrej znajomości mechanizmów systemu Kraksim
5. Zmiana generatora, utworzenie innych rodzajów pojazdów (poruszających się z większą/mniejszą prędkością, wybierających zawsze najkrótsze/zawsze najmniej zatłoczone trasy) – skomplikowane zadanie, wymagające biegłości zarówno w mechanizmach generacji, jak i – przede wszystkim – w routingu
6. Rozbudowa modelu ruchu o wyprzedzanie – bardzo trudne zadanie, może okazać się niemożliwe bez całkowitej zmiany modelu ruchu w systemie.
7. Stworzenie sensownego generatora plików konfiguracyjnych: definicji modelu oraz ruchu – zadanie bardzo czasochłonne, ale niezbędne. Wymaga niewielkiej znajomości systemu, lecz dokładnego przemyślenia i zaprojektowania interfejsu, aby był rzeczywiście wygodny.

## 7. Załączniki

Poniżej znajduje się lista załączników. Ich zawartość dołączona jest na końcu pliku, we wskazanej kolejności.

1. Przykładowy plik ***prediction.xml***
2. Plik ***config.properties*** dla testów mechanizmu predykcji
3. Plik ***model-D.xml*** użyty w testach mechanizmu predykcji
4. Plik ***traffic-D-med.xml*** określający wartość domyślną natężenia ruchu w testach predykcji dla modelu-D
5. Plik ***model-krakow.xml*** użyty w testach mechanizmu predykcji
6. Plik ***traffic-krakow\_med.xml*** określający wartość domyślną natężenia ruchu w testach predykcji dla modelu Krakowa

## 7.1. Załącznik 1: prediction.xml

```
<prediction
  cutOutProbability="0.8"
  cutOutMinimumNumber="4"
  neighborhoodSize="1"
  influencedTimesteps="2"
  ageingRate="0.98">
<trafficLevels>
  <level
    description="EMPTY"
    lowerBound="0"
    upperBound="1.1"
    influence="-0.3"
    nextDescription="SMOOTH">
  </level>
  <level
    description="SMOOTH"
    lowerBound="1.1"
    upperBound="1.2"
    influence="0"
    prevDescription="EMPTY"
    nextDescription="DENSE">
  </level>
  <level
    description="DENSE"
    lowerBound="1.2"
    upperBound="1.4"
    influence="0.5"
    prevDescription="SMOOTH"
    nextDescription="STUCK">
  </level>
  <level
    description="STUCK"
    lowerBound="1.4"
    upperBound="100"
    influence="2"
    prevDescription="DENSE">
  </level>
</trafficLevels></prediction>
```

## 7.2. Załącznik 2: config.properties

```
workDir = I:/Disc/Studia/S7/SVN/KRAKSIM/src/resources/tests/

# -- Krakow model -- for vms
#cityMapFile = trafficConfigs/model-krakow.xml
#travelSchemeFile = trafficConfigs/traffic-krakow-lo.xml
#travelSchemeFile = trafficConfigs/traffic-krakow-med.xml
#travelSchemeFile = trafficConfigs/traffic-krakow-hi.xml

# -- D model
#cityMapFile = trafficConfigs/model-D.xml
#travelSchemeFile = trafficConfigs/traffic-D-lo.xml
#travelSchemeFile = trafficConfigs/traffic-D-med.xml
#travelSchemeFile = trafficConfigs/traffic-D-hi.xml

statOutFile = I:/Disc/Studia/S7/SVN/KRAKSIM/src/resources/tests/debug.s

visualization = true
algorithm = sotl:zone=18

dynamicRouting=true
#dynamicRouting = false

yellowTransition = 3

#enablePrediction = false
enablePrediction = true

#predictionFile = predConfigs/prediction.weak_big.config.xml
#predictionFile = predConfigs/prediction.weak_default.config.xml
#predictionFile = predConfigs/prediction.weak_small.config.xml

#predictionFile = predConfigs/prediction.medium_big.config.xml
#predictionFile = predConfigs/prediction.medium_default.config.xml
#predictionFile = predConfigs/prediction.medium_small.config.xml

#predictionFile = predConfigs/prediction.strong_big.config.xml
#predictionFile = predConfigs/prediction.strong_default.config.xml
#predictionFile = predConfigs/prediction.strong_small.config.xml
```

## 7.3. Załącznik 3: model-D.xml

```
<?xml version="1.0"?>
<RoadNet>

  <nodes>
    <gateway id="GN" x="60" y="10" />
    <gateway id="GS" x="60" y="400" />

    <intersection id="X1" x="60" y="150" />
    <intersection id="X2" x="60" y="270" />
    <intersection id="X3" x="140" y="210" />

    <intersection id="X11" x="60" y="70" />
    <intersection id="X22" x="60" y="350" /><intersection id="X33"
      x="170" y="210">
    </intersection>
  </nodes>

  <roads>
    <road id="GN_X11" street="ulica Pionowa" from="GN" to="X11">
      <uplink>
        <main length="60" numberOfLanes="3" />
      </uplink>
      <downlink>
        <main length="60" numberOfLanes="3" />
      </downlink>
    </road>

    <road id="X11_X1" street="ulica Pionowa" from="X11" to="X1">
      <uplink>
        <main length="80" numberOfLanes="3" />
      </uplink>
      <downlink>
        <main length="80" numberOfLanes="3" />
      </downlink>
    </road>

    <road id="X1_X2" street="ulica Pionowa" from="X1" to="X2">
      <uplink>
        <main length="120" numberOfLanes="1" />
      </uplink>
      <downlink>
        <main length="120" numberOfLanes="1" />
      </downlink>
    </road>

    <road id="X2_X22" street="ulica Pionowa" from="X2" to="X22">
      <uplink>
        <main length="80" numberOfLanes="3" />
      </uplink>
      <downlink>
        <main length="80" numberOfLanes="3" />
      </downlink>
    </road>

    <road id="X22_GS" street="ulica Pionowa" from="X22" to="GS">
```

```

        <uplink>
            <main length="50" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="50" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="X1_X3" street="ulica Prawoskosna" from="X1" to="X3">
        <uplink>
            <main length="100" numberOfLanes="2" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="2" />
        </downlink>
    </road>

    <road id="X2_X3" street="ulica Lewoskosna" from="X2" to="X3">
        <uplink>
            <main length="100" numberOfLanes="2" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="2" />
        </downlink>
    </road>

    <road id="X11_X33" street="Obwodnica" from="X11" to="X33">
        <uplink>
            <main length="170" numberOfLanes="3" ></main>
        </uplink>
        <downlink>
            <main length="170" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="X33_X22" street="Obwodnica" from="X33" to="X22">
        <uplink>
            <main length="170" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="170" numberOfLanes="3" />
        </downlink>
    </road>
</roads>

<intersectionDescriptions>
    <intersection id="X1">

        <armActions arm="X11" dir="NS">
            <action lane="0" exit="X2"></action>
            <action lane="0" exit="X3"></action>
        </armActions>

        <armActions arm="X2" dir="NS">
            <action lane="0" exit="X11"></action>
            <action lane="0" exit="X3">
                <rule entrance="X11" lane="0" />
            </action>
        </armActions>

        <armActions arm="X3" dir="NS">

```



```

        <action lane="0" exit="X11">
            <rule entrance="X2" lane="0" />
        </action>
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X11" lane="0" state="green" />

            <inlane arm="X2" lane="0" state="red" />

            <inlane arm="X3" lane="0" state="red" />
        </phase>
        <phase num="2">
            <inlane arm="X11" lane="0" state="red" />

            <inlane arm="X2" lane="0" state="green" />

            <inlane arm="X3" lane="0" state="red" />
        </phase>
        <phase num="3">
            <inlane arm="X11" lane="0" state="red" />

            <inlane arm="X2" lane="0" state="red" />

            <inlane arm="X3" lane="0" state="green" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X2">

    <armActions arm="X22" dir="NS">
        <action lane="0" exit="X1"></action>
        <action lane="0" exit="X3"></action>
    </armActions>

    <armActions arm="X1" dir="NS">
        <action lane="0" exit="X22"></action>
        <action lane="0" exit="X3">
            <rule entrance="X22" lane="0" />
        </action>
    </armActions>

    <armActions arm="X3" dir="NS">
        <action lane="0" exit="X22">
            <rule entrance="X1" lane="0" />
        </action>
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X22" lane="0" state="green" />

            <inlane arm="X1" lane="0" state="red" />

            <inlane arm="X3" lane="0" state="red" />

```

```

        </phase>
        <phase num="2">
            <inlane arm="X22" lane="0" state="red" />

            <inlane arm="X1" lane="0" state="green" />

            <inlane arm="X3" lane="0" state="red" />
        </phase>
        <phase num="3">
            <inlane arm="X22" lane="0" state="red" />

            <inlane arm="X1" lane="0" state="red" />

            <inlane arm="X3" lane="0" state="green" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X3">

    <armActions arm="X1" dir="NS">
        <action lane="0" exit="X2"></action>
    </armActions>

    <armActions arm="X2" dir="NS">
        <action lane="0" exit="X1"></action>
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X1" lane="0" state="green" />
            <inlane arm="X2" lane="0" state="green" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X11">

    <armActions arm="GN" dir="NS">
        <action lane="0" exit="X1"></action>
        <action lane="0" exit="X33"></action>
    </armActions>

    <armActions arm="X1" dir="NS">
        <action lane="0" exit="GN"></action>
    </armActions>
    <armActions arm="X33" dir="NS">
        <action lane="0" exit="GN">
            <rule entrance="X1" lane="0"></rule></ac-
tion></armActions>
    <trafficLightsSchedule>
        <phase num="1">
            <inlane arm="GN" lane="0" state="green" />

            <inlane arm="X1" lane="0" state="green" />
            <inlane arm="X33" lane="0" state="red" />

```

```

        </phase>
        <phase num="2">
            <inlane arm="GN" lane="0" state="green" />
            <inlane arm="X1" lane="0" state="red" />
            <inlane arm="X33" lane="0" state="green" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X22">

    <armActions arm="GS" dir="NS">
        <action lane="0" exit="X2"></action>
        <action exit="X33" lane="0"></action>
    </armActions>

    <armActions arm="X2" dir="NS">
        <action lane="0" exit="GS"></action>
    </armActions>

    <armActions arm="X33" dir="NS">
        <action lane="0" exit="GS">
            <rule entrance="X2" lane="0"></rule></ac-
tion></armActions>

    <trafficLightsSchedule>
        <phase num="1">
            <inlane arm="GS" lane="0" state="green" />
            <inlane arm="X2" lane="0" state="green" />
            <inlane arm="X33" lane="0" state="red" />
        </phase>
        <phase num="2">
            <inlane arm="GS" lane="0" state="green" />
            <inlane arm="X2" lane="0" state="red" ></inla-
ne>
            <inlane arm="X33" lane="0" state="green" />
        </phase>
    </trafficLightsSchedule>
</intersection>

<intersection id="X33">
    <armActions arm="X11" dir="NS">
        <action lane="0" exit="X22"></action>
    </armActions><armActions arm="X22" dir="NS">
        <action lane="0" exit="X11"></action>
    </armActions><trafficLightsSchedule>
        <phase num="1">
            <inlane arm="X11" lane="0" state="green" />

            <inlane arm="X22" lane="0" state="green" />
        </phase>
    </trafficLightsSchedule>

</intersection>
</intersectionDescriptions>
</RoadNet>

```

## 7.4. Załącznik 4: traffic-D-med.xml

```
<?xml version="1.0"?>
<!--
czas trwania schematu: 6h = 500

pomiedzy każda para węzłów generowane jest 750 samochodów
z rozkładem jednostajnym
-->
<traffic>

  <scheme count="750">
    <gateway id="GN">
      <uniform a="0" b="500"/>
    </gateway>
    <gateway id="GS"/>
  </scheme>

  <scheme count="750">
    <gateway id="GS">
      <uniform a="0" b="500"/>
    </gateway>
    <gateway id="GN"/>
  </scheme>

</traffic>
```

## 7.5. Załącznik 5: model-krakow.xml

```
<?xml version="1.0"?>
<RoadNet>

  <nodes>
    <gateway id="G1" x="100" y="25"></gateway>
    <gateway id="G2" x="371" y="20"></gateway>
    <gateway id="G3" x="569" y="20"></gateway>
    <gateway id="G4" x="780" y="130"></gateway>
    <gateway id="G5" x="830" y="400"></gateway>
    <gateway id="G6" x="2" y="110"></gateway>
    <gateway id="G7" x="2" y="300"></gateway>
    <gateway id="G8" x="150" y="750"></gateway>
    <gateway id="G9" x="866" y="700"></gateway>
    <gateway id="G10" x="400" y="750"></gateway>

    <intersection id="R1" x="140" y="25"></intersection>
    <intersection id="R2" x="371" y="65"></intersection>
    <intersection id="R3" x="569" y="65"></intersection>
    <intersection id="R4" x="690" y="130"></intersection>
    <intersection id="R5" x="750" y="400"></intersection>
    <intersection id="R6" x="2" y="167"></intersection>
    <intersection id="R7" x="2" y="400"></intersection>
    <intersection id="R8" x="200" y="690"></intersection>
    <intersection id="R9" x="866" y="639"></intersection>
    <intersection id="R10" x="450" y="730"></intersection>

    <intersection id="X1" x="199" y="37"></intersection>
    <intersection id="X2" x="376" y="130"></intersection>
    <intersection id="X3" x="560" y="140"></intersection>
    <intersection id="X4" x="650" y="130"></intersection>
    <intersection id="X5" x="80" y="156"></intersection>
    <intersection id="X6" x="487" y="240"></intersection>
    <intersection id="X7" x="497" y="315"></intersection>
    <intersection id="X8" x="84" y="400"></intersection>
    <intersection id="X9" x="560" y="433"></intersection>
    <intersection id="X10" x="240" y="628"></intersection>
    <intersection id="X11" x="489" y="571"></intersection>
    <intersection id="X12" x="815" y="585"></intersection>
    <intersection id="X13" x="464" y="654"></intersection>

    <intersection id="P26" x="376" y="210"></intersection>
    <intersection id="P1011" x="290" y="581"></intersection>
  </nodes>

  <roads>
    <road id="G1R1" from="G1" to="R1">
      <uplink>
        <main length="100" numberOfLanes="3" />
      </uplink>
      <downlink>
        <main length="100" numberOfLanes="3" />
      </downlink>
    </road>

    <road id="G2R2" from="G2" to="R2">
```

```

        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="G3R3" from="G3" to="R3">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="G4R4" from="G4" to="R4">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="G5R5" from="G5" to="R5">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="G6R6" from="G6" to="R6">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="G7R7" from="G7" to="R7">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="G8R8" from="G8" to="R8">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

```

```

<road id="G9R9" from="G9" to="R9">
  <uplink>
    <main length="100" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

<road id="G10R10" from="G10" to="R10">
  <uplink>
    <main length="100" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

<road id="R1X1" from="R1" to="X1">
  <uplink>
    <main length="100" numberOfLanes="3" />
    <right length="20" />
  </uplink>
  <downlink>
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

<road id="X5X1" from="X5" to="X1">
  <uplink>
    <left length="20" />
    <main length="250" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="250" numberOfLanes="3" />
    <right length="20" />
  </downlink>
</road>

<road id="X1X2" from="X1" to="X2">
  <uplink>
    <left length="20" />
    <main length="250" numberOfLanes="3" />
  </uplink>
  <downlink>
    <left length="40" />
    <main length="250" numberOfLanes="3" />
  </downlink>
</road>

<road id="R2X2" from="R2" to="X2">
  <uplink>
    <left length="20" />
    <main length="100" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

```

```

<road id="X2X3" from="X2" to="X3">
  <uplink>
    <left length="20" />
    <main length="150" numberOfLanes="3" />
  </uplink>
  <downlink>
    <left length="20" />
    <main length="150" numberOfLanes="3" />
  </downlink>
</road>

<road id="X2P26" from="X2" to="P26">
  <uplink>
    <main length="150" numberOfLanes="3" />
  </uplink>
  <downlink>
    <left length="20" />
    <main length="150" numberOfLanes="3" />
  </downlink>
</road>

<road id="P26X6" from="P26" to="X6">
  <uplink>
    <left length="20" />
    <main length="200" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="200" numberOfLanes="3" />
  </downlink>
</road>

<road id="R3X3" from="R3" to="X3">
  <uplink>
    <left length="20" />
    <main length="100" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

<road id="R4X4" from="R4" to="X4">
  <uplink>
    <main length="100" numberOfLanes="3" />
  </uplink>
  <downlink>
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

<road id="X3X4" from="X3" to="X4">
  <uplink>
    <main length="100" numberOfLanes="3" />
  </uplink>
  <downlink>
    <left length="60" />
    <main length="100" numberOfLanes="3" />
  </downlink>
</road>

<road id="X3X6" from="X3" to="X6">

```



```

        <uplink>
            <main length="250" numberOfLanes="3" />
            <right length="20" />
        </uplink>
        <downlink>
            <left length="20" />
            <main length="250" numberOfLanes="3" />
        </downlink>
    </road>

<road id="R6X5" from="R6" to="X5">
    <uplink>
        <left length="20" />
        <main length="100" numberOfLanes="3" />
    </uplink>
    <downlink>
        <main length="100" numberOfLanes="3" />
    </downlink>
</road>

<road id="X5X8" from="X5" to="X8">
    <uplink>
        <left length="20" />
        <main length="450" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="40" />
        <main length="450" numberOfLanes="3" />
    </downlink>
</road>

<road id="X6X7" from="X6" to="X7">
    <uplink>
        <main length="150" numberOfLanes="3" />
        <right length="40" />
    </uplink>
    <downlink>
        <left length="40" />
        <main length="150" numberOfLanes="3" />
    </downlink>
</road>

<road id="X8X7" from="X8" to="X7">
    <uplink>
        <left length="20" />
        <main length="650" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="650" numberOfLanes="3" />
    </downlink>
</road>

<road id="X7X9" from="X7" to="X9">
    <uplink>
        <left length="40" />
        <main length="200" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="40" />
        <main length="200" numberOfLanes="3" />
    </downlink>
</road>

```

```

        </downlink>
</road>

<road id="R7X8" from="R7" to="X8">
    <uplink>
        <left length="20" />
        <main length="100" numberOfLanes="3" />
    </uplink>
    <downlink>
        <main length="100" numberOfLanes="3" />
    </downlink>
</road>

<road id="X8X10" from="X8" to="X10">
    <uplink>
        <left length="20" />
        <main length="350" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="350" numberOfLanes="3" />
    </downlink>
</road>

<road id="X9R5" from="X9" to="R5">
    <uplink>
        <main length="350" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="350" numberOfLanes="3" />
    </downlink>
</road>

<road id="X9X12" from="X9" to="X12">
    <uplink>
        <left length="20" />
        <main length="550" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="550" numberOfLanes="3" />
    </downlink>
</road>

<road id="X9X11" from="X9" to="X11">
    <uplink>
        <main length="250" numberOfLanes="3" />
        <right length="60" />
    </uplink>
    <downlink>
        <left length="60" />
        <main length="250" numberOfLanes="3" />
    </downlink>
</road>

<road id="X10P1011" from="X10" to="P1011">
    <uplink>
        <main length="100" numberOfLanes="3" />
    </uplink>
    <downlink>

```

```

        <left length="20" />
        <main length="100" numberOfLanes="3" />
    </downlink>
</road>

<road id="P1011X11" from="P1011" to="X11">
    <uplink>
        <left length="20" />
        <main length="350" numberOfLanes="3" />
    </uplink>
    <downlink>
        <main length="350" numberOfLanes="3" />
    </downlink>
</road>

<road id="X10X13" from="X10" to="X13">
    <uplink>
        <left length="20" />
        <main length="450" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="450" numberOfLanes="3" />
    </downlink>
</road>

<road id="X10R8" from="X10" to="R8">
    <uplink>
        <main length="100" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="100" numberOfLanes="3" />
    </downlink>
</road>

<road id="X11X13" from="X11" to="X13">
    <uplink>
        <left length="20" />
        <main length="150" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="40" />
        <main length="150" numberOfLanes="3" />
    </downlink>
</road>

<road id="X13X12" from="X13" to="X12">
    <uplink>
        <left length="20" />
        <main length="550" numberOfLanes="3" />
    </uplink>
    <downlink>
        <left length="20" />
        <main length="550" numberOfLanes="3" />
    </downlink>
</road>

<road id="X12R9" from="X12" to="R9">
    <uplink>
        <main length="100" numberOfLanes="3" />
    </uplink>
    <downlink>
        <main length="100" numberOfLanes="3" />
    </downlink>
</road>

```

```

        </uplink>
        <downlink>
            <left length="20" />
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="X13R10" from="X13" to="R10">
        <uplink>
            <main length="100" numberOfLanes="3" />
        </uplink>
        <downlink>
            <left length="20" />
            <main length="100" numberOfLanes="3" />
        </downlink>
    </road>

    <road id="X12X4" from="X12" to="X4">
        <uplink>
            <main length="1050" numberOfLanes="3" />
        </uplink>
        <downlink>
            <left length="20" />
            <main length="1050" numberOfLanes="3" />
        </downlink>
    </road>
</roads>

<intersectionDescriptions>
    <intersection id="X1">

        <armActions arm="X2">
            <action lane="-1" exit="X5">
                <rule entrance="R1" lane="0" />
                <rule entrance="R1" lane="1" />
            </action>
            <action lane="0" exit="R1"></action>
        </armActions>

        <armActions arm="X5">
            <action lane="0" exit="X2">
                <rule entrance="R1" lane="0" />
            </action>
            <action lane="-1" exit="R1">
                <rule entrance="X2" lane="0" />
                <rule entrance="X2" lane="-1" />
                <rule entrance="R1" lane="0" />
            </action>
        </armActions>

        <armActions arm="R1">
            <action lane="0" exit="X2" />
            <action lane="1" exit="X5" />
        </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X2" lane="0" state="green" />
            <inlane arm="X2" lane="-1" state="red" />

```

```

        <inlane arm="R1" lane="0" state="green" />
        <inlane arm="R1" lane="1" state="green" />

        <inlane arm="X5" lane="0" state="red" />
        <inlane arm="X5" lane="-1" state="red" />
    </phase>
    <phase num="2">
        <inlane arm="X2" lane="0" state="red" />
        <inlane arm="X2" lane="-1" state="green" />

        <inlane arm="R1" lane="0" state="red" />
        <inlane arm="R1" lane="1" state="red" />

        <inlane arm="X5" lane="0" state="green" />
        <inlane arm="X5" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="X2" lane="0" state="red" />
        <inlane arm="X2" lane="-1" state="red" />

        <inlane arm="R1" lane="0" state="red" />
        <inlane arm="R1" lane="1" state="green" />

        <inlane arm="X5" lane="0" state="red" />
        <inlane arm="X5" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>
</intersection>

<intersection id="X2">

    <armActions arm="R2">
        <action lane="-1" exit="X3">
            <rule entrance="X3" lane="0" />
            <rule entrance="P26" lane="0" />
            <rule entrance="X1" lane="0" />
        </action>
        <action lane="0" exit="P26">
            <rule entrance="X1" lane="0" />
        </action>
        <action lane="0" exit="X1" />
    </armActions>

    <armActions arm="X3">
        <action lane="-1" exit="P26">
            <rule entrance="R2" lane="0" />
            <rule entrance="P26" lane="0" />
            <rule entrance="X1" lane="0" />
        </action>
        <action lane="0" exit="X1">
            <rule entrance="R2" lane="0" />
        </action>
        <action lane="0" exit="R2" />
    </armActions>

    <armActions arm="P26">
        <action lane="-1" exit="X1">
            <rule entrance="X3" lane="0" />
            <rule entrance="R2" lane="0" />
        </action>
    </armActions>

```

```

        <rule entrance="X1" lane="0" />
    </action>
    <action lane="0" exit="R2">
        <rule entrance="X3" lane="0" />
    </action>
    <action lane="0" exit="X3" />
</armActions>

<armActions arm="X1">
    <action lane="-1" exit="R2">
        <rule entrance="R2" lane="0" />
        <rule entrance="X3" lane="0" />
        <rule entrance="P26" lane="0" />
    </action>
    <action lane="0" exit="X3">
        <rule entrance="P26" lane="0" />
    </action>
    <action lane="0" exit="P26" />
</armActions>

<trafficLightsSchedule>

    <phase num="1">
        <inlane arm="R2" lane="0" state="green" />
        <inlane arm="R2" lane="-1" state="red" />

        <inlane arm="P26" lane="0" state="green" />
        <inlane arm="P26" lane="-1" state="red" />

        <inlane arm="X1" lane="0" state="red" />
        <inlane arm="X1" lane="-1" state="red" />

        <inlane arm="X3" lane="0" state="red" />
        <inlane arm="X3" lane="-1" state="red" />
    </phase>
    <phase num="2">
        <inlane arm="R2" lane="0" state="red" />
        <inlane arm="R2" lane="-1" state="red" />

        <inlane arm="P26" lane="0" state="red" />
        <inlane arm="P26" lane="-1" state="red" />

        <inlane arm="X1" lane="0" state="green" />
        <inlane arm="X1" lane="-1" state="red" />

        <inlane arm="X3" lane="0" state="green" />
        <inlane arm="X3" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="R2" lane="0" state="red" />
        <inlane arm="R2" lane="-1" state="green" />

        <inlane arm="P26" lane="0" state="red" />
        <inlane arm="P26" lane="-1" state="green" />

        <inlane arm="X1" lane="0" state="red" />
        <inlane arm="X1" lane="-1" state="red" />

        <inlane arm="X3" lane="0" state="red" />
        <inlane arm="X3" lane="-1" state="red" />
    </phase>

```

```

        <phase num="4">
            <inlane arm="R2" lane="0" state="red" />
            <inlane arm="R2" lane="-1" state="red" />

            <inlane arm="P26" lane="0" state="red" />
            <inlane arm="P26" lane="-1" state="red" />

            <inlane arm="X1" lane="0" state="red" />
            <inlane arm="X1" lane="-1" state="green" />

            <inlane arm="X3" lane="0" state="red" />
            <inlane arm="X3" lane="-1" state="green" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X3">
    <armActions arm="R3">
        <action lane="-1" exit="X4">
            <rule entrance="X4" lane="0" />
            <rule entrance="X6" lane="0" />
            <rule entrance="X2" lane="0" />
        </action>
        <action lane="0" exit="X6">
            <rule entrance="X2" lane="0" />
        </action>
        <action lane="0" exit="X2" />
    </armActions>

    <armActions arm="X4">
        <action lane="-1" exit="X6">
            <rule entrance="R3" lane="0" />
            <rule entrance="X6" lane="0" />
            <rule entrance="X2" lane="0" />
        </action>
        <action lane="0" exit="X2">
            <rule entrance="R3" lane="0" />
        </action>
        <action lane="0" exit="R3" />
    </armActions>

    <armActions arm="X6">
        <action lane="-1" exit="X2">
            <rule entrance="X4" lane="0" />
            <rule entrance="R3" lane="0" />
            <rule entrance="X2" lane="0" />
        </action>
        <action lane="0" exit="R3">
            <rule entrance="X4" lane="0" />
        </action>
        <action lane="0" exit="X4" />
    </armActions>

    <armActions arm="X2">
        <action lane="-1" exit="R3">
            <rule entrance="R3" lane="0" />
            <rule entrance="X4" lane="0" />
            <rule entrance="X6" lane="0" />
        </action>
    </armActions>

```

```

    <action lane="0" exit="X4">
      <rule entrance="X6" lane="0" />
    </action>
    <action lane="0" exit="X6" />
  </armActions>

  <trafficLightsSchedule>

    <phase num="1">
      <inlane arm="R3" lane="0" state="green" />
      <inlane arm="R3" lane="-1" state="red" />

      <inlane arm="X6" lane="0" state="green" />
      <inlane arm="X6" lane="-1" state="red" />

      <inlane arm="X2" lane="0" state="red" />
      <inlane arm="X2" lane="-1" state="red" />

      <inlane arm="X4" lane="0" state="red" />
      <inlane arm="X4" lane="-1" state="red" />
    </phase>
    <phase num="2">
      <inlane arm="R3" lane="0" state="red" />
      <inlane arm="R3" lane="-1" state="red" />

      <inlane arm="X6" lane="0" state="red" />
      <inlane arm="X6" lane="-1" state="red" />

      <inlane arm="X2" lane="0" state="green" />
      <inlane arm="X2" lane="-1" state="red" />

      <inlane arm="X4" lane="0" state="green" />
      <inlane arm="X4" lane="-1" state="red" />
    </phase>
    <phase num="3">
      <inlane arm="R3" lane="0" state="red" />
      <inlane arm="R3" lane="-1" state="green" />

      <inlane arm="X6" lane="0" state="red" />
      <inlane arm="X6" lane="-1" state="green" />

      <inlane arm="X2" lane="0" state="red" />
      <inlane arm="X2" lane="-1" state="red" />

      <inlane arm="X4" lane="0" state="red" />
      <inlane arm="X4" lane="-1" state="red" />
    </phase>
    <phase num="4">
      <inlane arm="R3" lane="0" state="red" />
      <inlane arm="R3" lane="-1" state="red" />

      <inlane arm="X6" lane="0" state="red" />
      <inlane arm="X6" lane="-1" state="red" />

      <inlane arm="X2" lane="0" state="red" />
      <inlane arm="X2" lane="-1" state="green" />

      <inlane arm="X4" lane="0" state="red" />
      <inlane arm="X4" lane="-1" state="green" />
    </phase>
  </trafficLightsSchedule>

```



```

        </trafficLightsSchedule>

</intersection>

<intersection id="X4">

    <armActions arm="X3">
        <action lane="0" exit="X12"></action>
        <action lane="0" exit="R4"></action>
    </armActions>

    <armActions arm="X12">
        <action lane="0" exit="R4">
            <rule entrance="X3" lane="0" />
        </action>
        <action lane="0" exit="X3">
            <rule entrance="X3" lane="0" />
            <rule entrance="R4" lane="0" />
        </action>
    </armActions>

    <armActions arm="R4">
        <action lane="0" exit="X3" />
        <action lane="0" exit="X12">
            <rule entrance="X3" lane="0" />
        </action>
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X3" lane="0" state="green" />
            <inlane arm="R4" lane="0" state="red" />
            <inlane arm="X12" lane="0" state="red" />
        </phase>
        <phase num="2">
            <inlane arm="X3" lane="0" state="red" />
            <inlane arm="R4" lane="0" state="red" />
            <inlane arm="X12" lane="0" state="green" />
        </phase>
        <phase num="3">
            <inlane arm="X3" lane="0" state="red" />
            <inlane arm="R4" lane="0" state="green" />
            <inlane arm="X12" lane="0" state="red" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X5">

    <armActions arm="X8">
        <action lane="-1" exit="R6">
            <rule entrance="X1" lane="0" />
            <rule entrance="X1" lane="1" />
        </action>
        <action lane="0" exit="X1"></action>
    </armActions>

    <armActions arm="R6">
        <action lane="0" exit="X8">

```

```

        <rule entrance="X1" lane="0" />
    </action>
    <action lane="-1" exit="X1">
        <rule entrance="X8" lane="0" />
        <rule entrance="X8" lane="-1" />
        <rule entrance="X1" lane="0" />
    </action>
</armActions>

<armActions arm="X1">
    <action lane="0" exit="X8" />
    <action lane="1" exit="R6" />
</armActions>

<trafficLightsSchedule>

    <phase num="1">
        <inlane arm="X8" lane="0" state="green" />
        <inlane arm="X8" lane="-1" state="red" />

        <inlane arm="X1" lane="0" state="green" />
        <inlane arm="X1" lane="1" state="green" />

        <inlane arm="R6" lane="0" state="red" />
        <inlane arm="R6" lane="-1" state="red" />
    </phase>
    <phase num="2">
        <inlane arm="X8" lane="0" state="red" />
        <inlane arm="X8" lane="-1" state="green" />

        <inlane arm="X1" lane="0" state="red" />
        <inlane arm="X1" lane="1" state="red" />

        <inlane arm="R6" lane="0" state="green" />
        <inlane arm="R6" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="X8" lane="0" state="red" />
        <inlane arm="X8" lane="-1" state="red" />

        <inlane arm="X1" lane="0" state="red" />
        <inlane arm="X1" lane="1" state="green" />

        <inlane arm="R6" lane="0" state="red" />
        <inlane arm="R6" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>

</intersection>

<intersection id="X6">

    <armActions arm="X7">
        <action lane="-1" exit="P26">
            <rule entrance="X3" lane="0" />
            <rule entrance="X3" lane="1" />
        </action>
        <action lane="0" exit="X3"></action>
    </armActions>

```

```

<armActions arm="P26">
  <action lane="0" exit="X7">
    <rule entrance="X3" lane="0" />
  </action>
  <action lane="-1" exit="X3">
    <rule entrance="X7" lane="0" />
    <rule entrance="X7" lane="-1" />
    <rule entrance="X3" lane="0" />
  </action>
</armActions>

<armActions arm="X3">
  <action lane="0" exit="X7" />
  <action lane="1" exit="P26" />
</armActions>

<trafficLightsSchedule>

  <phase num="1">
    <inlane arm="X7" lane="0" state="green" />
    <inlane arm="X7" lane="-1" state="red" />

    <inlane arm="X3" lane="0" state="green" />
    <inlane arm="X3" lane="1" state="green" />

    <inlane arm="P26" lane="0" state="red" />
    <inlane arm="P26" lane="-1" state="red" />
  </phase>
  <phase num="2">
    <inlane arm="X7" lane="0" state="red" />
    <inlane arm="X7" lane="-1" state="green" />

    <inlane arm="X3" lane="0" state="red" />
    <inlane arm="X3" lane="1" state="red" />

    <inlane arm="P26" lane="0" state="green" />
    <inlane arm="P26" lane="-1" state="red" />
  </phase>
  <phase num="3">
    <inlane arm="X7" lane="0" state="red" />
    <inlane arm="X7" lane="-1" state="red" />

    <inlane arm="X3" lane="0" state="red" />
    <inlane arm="X3" lane="1" state="green" />

    <inlane arm="P26" lane="0" state="red" />
    <inlane arm="P26" lane="-1" state="green" />
  </phase>

</trafficLightsSchedule>

</intersection>
<intersection id="X7">

  <armActions arm="X9">
    <action lane="-1" exit="X8">
      <rule entrance="X6" lane="0" />
      <rule entrance="X6" lane="1" />
    </action>
    <action lane="0" exit="X6"></action>
  </armActions>

```

```

<armActions arm="X8">
  <action lane="0" exit="X9">
    <rule entrance="X6" lane="0" />
  </action>
  <action lane="-1" exit="X6">
    <rule entrance="X9" lane="0" />
    <rule entrance="X9" lane="-1" />
    <rule entrance="X6" lane="0" />
  </action>
</armActions>

<armActions arm="X6">
  <action lane="0" exit="X9" />
  <action lane="1" exit="X8" />
</armActions>

<trafficLightsSchedule>

  <phase num="1">
    <inlane arm="X9" lane="0" state="green" />
    <inlane arm="X9" lane="-1" state="red" />

    <inlane arm="X6" lane="0" state="green" />
    <inlane arm="X6" lane="1" state="green" />

    <inlane arm="X8" lane="0" state="red" />
    <inlane arm="X8" lane="-1" state="red" />
  </phase>
  <phase num="2">
    <inlane arm="X9" lane="0" state="red" />
    <inlane arm="X9" lane="-1" state="green" />

    <inlane arm="X6" lane="0" state="red" />
    <inlane arm="X6" lane="1" state="red" />

    <inlane arm="X8" lane="0" state="green" />
    <inlane arm="X8" lane="-1" state="red" />
  </phase>
  <phase num="3">
    <inlane arm="X9" lane="0" state="red" />
    <inlane arm="X9" lane="-1" state="red" />

    <inlane arm="X6" lane="0" state="red" />
    <inlane arm="X6" lane="1" state="green" />

    <inlane arm="X8" lane="0" state="red" />
    <inlane arm="X8" lane="-1" state="green" />
  </phase>

</trafficLightsSchedule>

</intersection>

<intersection id="X8">
  <armActions arm="X5">
    <action lane="-1" exit="X7">
      <rule entrance="X7" lane="0" />
      <rule entrance="X10" lane="0" />
      <rule entrance="R7" lane="0" />
    </action>
  </armActions>
</intersection>

```

```

        <action lane="0" exit="X10">
            <rule entrance="R7" lane="0" />
        </action>
        <action lane="0" exit="R7" />
    </armActions>

    <armActions arm="X7">
        <action lane="-1" exit="X10">
            <rule entrance="X5" lane="0" />
            <rule entrance="X10" lane="0" />
            <rule entrance="R7" lane="0" />
        </action>
        <action lane="0" exit="R7">
            <rule entrance="X5" lane="0" />
        </action>
        <action lane="0" exit="X5" />
    </armActions>

    <armActions arm="X10">
        <action lane="-1" exit="R7">
            <rule entrance="X7" lane="0" />
            <rule entrance="X5" lane="0" />
            <rule entrance="R7" lane="0" />
        </action>
        <action lane="0" exit="X5">
            <rule entrance="X7" lane="0" />
        </action>
        <action lane="0" exit="X7" />
    </armActions>

    <armActions arm="R7">
        <action lane="-1" exit="X5">
            <rule entrance="X5" lane="0" />
            <rule entrance="X7" lane="0" />
            <rule entrance="X10" lane="0" />
        </action>
        <action lane="0" exit="X7">
            <rule entrance="X10" lane="0" />
        </action>
        <action lane="0" exit="X10" />
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X5" lane="0" state="green" />
            <inlane arm="X5" lane="-1" state="red" />

            <inlane arm="X10" lane="0" state="green" />
            <inlane arm="X10" lane="-1" state="red" />

            <inlane arm="R7" lane="0" state="red" />
            <inlane arm="R7" lane="-1" state="red" />

            <inlane arm="X7" lane="0" state="red" />
            <inlane arm="X7" lane="-1" state="red" />
        </phase>
        <phase num="2">
            <inlane arm="X5" lane="0" state="red" />
            <inlane arm="X5" lane="-1" state="red" />

```

```

        <inlane arm="X10" lane="0" state="red" />
        <inlane arm="X10" lane="-1" state="red" />

        <inlane arm="R7" lane="0" state="green" />
        <inlane arm="R7" lane="-1" state="red" />

        <inlane arm="X7" lane="0" state="green" />
        <inlane arm="X7" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="X5" lane="0" state="red" />
        <inlane arm="X5" lane="-1" state="green" />

        <inlane arm="X10" lane="0" state="red" />
        <inlane arm="X10" lane="-1" state="green" />

        <inlane arm="R7" lane="0" state="red" />
        <inlane arm="R7" lane="-1" state="red" />

        <inlane arm="X7" lane="0" state="red" />
        <inlane arm="X7" lane="-1" state="red" />
    </phase>
    <phase num="4">
        <inlane arm="X5" lane="0" state="red" />
        <inlane arm="X5" lane="-1" state="red" />

        <inlane arm="X10" lane="0" state="red" />
        <inlane arm="X10" lane="-1" state="red" />

        <inlane arm="R7" lane="0" state="red" />
        <inlane arm="R7" lane="-1" state="green" />

        <inlane arm="X7" lane="0" state="red" />
        <inlane arm="X7" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>

</intersection>

<intersection id="X9">

    <armActions arm="X7">
        <action lane="-1" exit="R5">
            <rule entrance="R5" lane="0" />
            <rule entrance="X12" lane="0" />
            <rule entrance="X11" lane="0" />
        </action>
        <action lane="0" exit="X12">
            <rule entrance="X11" lane="0" />
        </action>
        <action lane="0" exit="X11" />
    </armActions>

    <armActions arm="R5">
        <action lane="-1" exit="X12">
            <rule entrance="X7" lane="0" />
            <rule entrance="X12" lane="0" />
            <rule entrance="X11" lane="0" />
        </action>
        <action lane="0" exit="X11">

```

```

        <rule entrance="X7" lane="0" />
    </action>
    <action lane="0" exit="X7" />
</armActions>

<armActions arm="X12">
    <action lane="-1" exit="X11">
        <rule entrance="R5" lane="0" />
        <rule entrance="X7" lane="0" />
        <rule entrance="X11" lane="0" />
    </action>
    <action lane="0" exit="X7">
        <rule entrance="R5" lane="0" />
    </action>
    <action lane="0" exit="R5" />
</armActions>

<armActions arm="X11">
    <action lane="-1" exit="X7">
        <rule entrance="X7" lane="0" />
        <rule entrance="R5" lane="0" />
        <rule entrance="X12" lane="0" />
    </action>
    <action lane="0" exit="R5">
        <rule entrance="X12" lane="0" />
    </action>
    <action lane="0" exit="X12" />
</armActions>

<trafficLightsSchedule>

    <phase num="1">
        <inlane arm="X7" lane="0" state="green" />
        <inlane arm="X7" lane="-1" state="red" />

        <inlane arm="X12" lane="0" state="green" />
        <inlane arm="X12" lane="-1" state="red" />

        <inlane arm="X11" lane="0" state="red" />
        <inlane arm="X11" lane="-1" state="red" />

        <inlane arm="R5" lane="0" state="red" />
        <inlane arm="R5" lane="-1" state="red" />
    </phase>
    <phase num="2">
        <inlane arm="X7" lane="0" state="red" />
        <inlane arm="X7" lane="-1" state="red" />

        <inlane arm="X12" lane="0" state="red" />
        <inlane arm="X12" lane="-1" state="red" />

        <inlane arm="X11" lane="0" state="green" />
        <inlane arm="X11" lane="-1" state="red" />

        <inlane arm="R5" lane="0" state="green" />
        <inlane arm="R5" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="X7" lane="0" state="red" />
        <inlane arm="X7" lane="-1" state="green" />

```

```

        <inlane arm="X12" lane="0" state="red" />
        <inlane arm="X12" lane="-1" state="green" />

        <inlane arm="X11" lane="0" state="red" />
        <inlane arm="X11" lane="-1" state="red" />

        <inlane arm="R5" lane="0" state="red" />
        <inlane arm="R5" lane="-1" state="red" />
    </phase>
    <phase num="4">
        <inlane arm="X7" lane="0" state="red" />
        <inlane arm="X7" lane="-1" state="red" />

        <inlane arm="X12" lane="0" state="red" />
        <inlane arm="X12" lane="-1" state="red" />

        <inlane arm="X11" lane="0" state="red" />
        <inlane arm="X11" lane="-1" state="green" />

        <inlane arm="R5" lane="0" state="red" />
        <inlane arm="R5" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>

</intersection>

<intersection id="X10">

    <armActions arm="P1011">
        <action lane="-1" exit="X13">
            <rule entrance="X13" lane="0" />
            <rule entrance="R8" lane="0" />
            <rule entrance="X8" lane="0" />
        </action>
        <action lane="0" exit="R8">
            <rule entrance="X8" lane="0" />
        </action>
        <action lane="0" exit="X8" />
    </armActions>

    <armActions arm="X13">
        <action lane="-1" exit="R8">
            <rule entrance="P1011" lane="0" />
            <rule entrance="R8" lane="0" />
            <rule entrance="X8" lane="0" />
        </action>
        <action lane="0" exit="X8">
            <rule entrance="P1011" lane="0" />
        </action>
        <action lane="0" exit="P1011" />
    </armActions>

    <armActions arm="R8">
        <action lane="-1" exit="X8">
            <rule entrance="X13" lane="0" />
            <rule entrance="P1011" lane="0" />
            <rule entrance="X8" lane="0" />
        </action>
        <action lane="0" exit="P1011">
            <rule entrance="X13" lane="0" />
        </action>
    </armActions>

```



```

        </action>
        <action lane="0" exit="X13" />
    </armActions>

    <armActions arm="X8">
        <action lane="-1" exit="P1011">
            <rule entrance="P1011" lane="0" />
            <rule entrance="X13" lane="0" />
            <rule entrance="R8" lane="0" />
        </action>
        <action lane="0" exit="X13">
            <rule entrance="R8" lane="0" />
        </action>
        <action lane="0" exit="R8" />
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="P1011" lane="0" state="green" />
            <inlane arm="P1011" lane="-1" state="red" />

            <inlane arm="R8" lane="0" state="green" />
            <inlane arm="R8" lane="-1" state="red" />

            <inlane arm="X8" lane="0" state="red" />
            <inlane arm="X8" lane="-1" state="red" />

            <inlane arm="X13" lane="0" state="red" />
            <inlane arm="X13" lane="-1" state="red" />
        </phase>
        <phase num="2">
            <inlane arm="P1011" lane="0" state="red" />
            <inlane arm="P1011" lane="-1" state="red" />

            <inlane arm="R8" lane="0" state="red" />
            <inlane arm="R8" lane="-1" state="red" />

            <inlane arm="X8" lane="0" state="green" />
            <inlane arm="X8" lane="-1" state="red" />

            <inlane arm="X13" lane="0" state="green" />
            <inlane arm="X13" lane="-1" state="red" />
        </phase>
        <phase num="3">
            <inlane arm="P1011" lane="0" state="red" />
            <inlane arm="P1011" lane="-1" state="green" />

            <inlane arm="R8" lane="0" state="red" />
            <inlane arm="R8" lane="-1" state="green" />

            <inlane arm="X8" lane="0" state="red" />
            <inlane arm="X8" lane="-1" state="red" />

            <inlane arm="X13" lane="0" state="red" />
            <inlane arm="X13" lane="-1" state="red" />
        </phase>
        <phase num="4">
            <inlane arm="P1011" lane="0" state="red" />
            <inlane arm="P1011" lane="-1" state="red" />

```

```

        <inlane arm="R8" lane="0" state="red" />
        <inlane arm="R8" lane="-1" state="red" />

        <inlane arm="X8" lane="0" state="red" />
        <inlane arm="X8" lane="-1" state="green" />

        <inlane arm="X13" lane="0" state="red" />
        <inlane arm="X13" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>

</intersection>

<intersection id="X11">

    <armActions arm="X13">
        <action lane="-1" exit="P1011">
            <rule entrance="X9" lane="0" />
            <rule entrance="X9" lane="1" />
        </action>
        <action lane="0" exit="X9"></action>
    </armActions>

    <armActions arm="P1011">
        <action lane="0" exit="X13">
            <rule entrance="X9" lane="0" />
        </action>
        <action lane="-1" exit="X9">
            <rule entrance="X13" lane="0" />
            <rule entrance="X13" lane="-1" />
            <rule entrance="X9" lane="0" />
        </action>
    </armActions>

    <armActions arm="X9">
        <action lane="0" exit="X13" />
        <action lane="1" exit="P1011" />
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X13" lane="0" state="green" />
            <inlane arm="X13" lane="-1" state="red" />

            <inlane arm="X9" lane="0" state="green" />
            <inlane arm="X9" lane="1" state="green" />

            <inlane arm="P1011" lane="0" state="red" />
            <inlane arm="P1011" lane="-1" state="red" />
        </phase>
        <phase num="2">
            <inlane arm="X13" lane="0" state="red" />
            <inlane arm="X13" lane="-1" state="green" />

            <inlane arm="X9" lane="0" state="red" />
            <inlane arm="X9" lane="1" state="red" />

            <inlane arm="P1011" lane="0" state="green" />
            <inlane arm="P1011" lane="-1" state="red" />
        </phase>
    </trafficLightsSchedule>
</intersection>

```

```

        </phase>
        <phase num="3">
            <inlane arm="X13" lane="0" state="red" />
            <inlane arm="X13" lane="-1" state="red" />

            <inlane arm="X9" lane="0" state="red" />
            <inlane arm="X9" lane="1" state="green" />

            <inlane arm="P1011" lane="0" state="red" />
            <inlane arm="P1011" lane="-1" state="green" />
        </phase>

    </trafficLightsSchedule>

</intersection>

<intersection id="X12">

    <armActions arm="X9">
        <action lane="-1" exit="X4">
            <rule entrance="X4" lane="0" />
            <rule entrance="R9" lane="0" />
            <rule entrance="X13" lane="0" />
        </action>
        <action lane="0" exit="R9">
            <rule entrance="X13" lane="0" />
        </action>
        <action lane="0" exit="X13" />
    </armActions>

    <armActions arm="X4">
        <action lane="-1" exit="R9">
            <rule entrance="X9" lane="0" />
            <rule entrance="R9" lane="0" />
            <rule entrance="X13" lane="0" />
        </action>
        <action lane="0" exit="X13">
            <rule entrance="X9" lane="0" />
        </action>
        <action lane="0" exit="X9" />
    </armActions>

    <armActions arm="R9">
        <action lane="-1" exit="X13">
            <rule entrance="X4" lane="0" />
            <rule entrance="X9" lane="0" />
            <rule entrance="X13" lane="0" />
        </action>
        <action lane="0" exit="X9">
            <rule entrance="X4" lane="0" />
        </action>
        <action lane="0" exit="X4" />
    </armActions>

    <armActions arm="X13">
        <action lane="-1" exit="X9">
            <rule entrance="X9" lane="0" />
            <rule entrance="X4" lane="0" />
            <rule entrance="R9" lane="0" />
        </action>
        <action lane="0" exit="X4">

```

```

        <rule entrance="R9" lane="0" />
    </action>
    <action lane="0" exit="R9" />
</armActions>

<trafficLightsSchedule>

    <phase num="1">
        <inlane arm="X9" lane="0" state="green" />
        <inlane arm="X9" lane="-1" state="red" />

        <inlane arm="R9" lane="0" state="green" />
        <inlane arm="R9" lane="-1" state="red" />

        <inlane arm="X13" lane="0" state="red" />
        <inlane arm="X13" lane="-1" state="red" />

        <inlane arm="X4" lane="0" state="red" />
        <inlane arm="X4" lane="-1" state="red" />
    </phase>
    <phase num="2">
        <inlane arm="X9" lane="0" state="red" />
        <inlane arm="X9" lane="-1" state="red" />

        <inlane arm="R9" lane="0" state="red" />
        <inlane arm="R9" lane="-1" state="red" />

        <inlane arm="X13" lane="0" state="green" />
        <inlane arm="X13" lane="-1" state="red" />

        <inlane arm="X4" lane="0" state="green" />
        <inlane arm="X4" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="X9" lane="0" state="red" />
        <inlane arm="X9" lane="-1" state="green" />

        <inlane arm="R9" lane="0" state="red" />
        <inlane arm="R9" lane="-1" state="green" />

        <inlane arm="X13" lane="0" state="red" />
        <inlane arm="X13" lane="-1" state="red" />

        <inlane arm="X4" lane="0" state="red" />
        <inlane arm="X4" lane="-1" state="red" />
    </phase>
    <phase num="4">
        <inlane arm="X9" lane="0" state="red" />
        <inlane arm="X9" lane="-1" state="red" />

        <inlane arm="R9" lane="0" state="red" />
        <inlane arm="R9" lane="-1" state="red" />

        <inlane arm="X13" lane="0" state="red" />
        <inlane arm="X13" lane="-1" state="green" />

        <inlane arm="X4" lane="0" state="red" />
        <inlane arm="X4" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>

```

```

</intersection>

<intersection id="X13">

    <armActions arm="X11">
        <action lane="-1" exit="X12">
            <rule entrance="X12" lane="0" />
            <rule entrance="R10" lane="0" />
            <rule entrance="X10" lane="0" />
        </action>
        <action lane="0" exit="R10">
            <rule entrance="X10" lane="0" />
        </action>
        <action lane="0" exit="X10" />
    </armActions>

    <armActions arm="X12">
        <action lane="-1" exit="R10">
            <rule entrance="X11" lane="0" />
            <rule entrance="R10" lane="0" />
            <rule entrance="X10" lane="0" />
        </action>
        <action lane="0" exit="X10">
            <rule entrance="X11" lane="0" />
        </action>
        <action lane="0" exit="X11" />
    </armActions>

    <armActions arm="R10">
        <action lane="-1" exit="X10">
            <rule entrance="X12" lane="0" />
            <rule entrance="X11" lane="0" />
            <rule entrance="X10" lane="0" />
        </action>
        <action lane="0" exit="X11">
            <rule entrance="X12" lane="0" />
        </action>
        <action lane="0" exit="X12" />
    </armActions>

    <armActions arm="X10">
        <action lane="-1" exit="X11">
            <rule entrance="X11" lane="0" />
            <rule entrance="X12" lane="0" />
            <rule entrance="R10" lane="0" />
        </action>
        <action lane="0" exit="X12">
            <rule entrance="R10" lane="0" />
        </action>
        <action lane="0" exit="R10" />
    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X11" lane="0" state="green" />
            <inlane arm="X11" lane="-1" state="red" />

            <inlane arm="R10" lane="0" state="green" />
            <inlane arm="R10" lane="-1" state="red" />
        </phase>
    </trafficLightsSchedule>

```

```

        <inlane arm="X10" lane="0" state="red" />
        <inlane arm="X10" lane="-1" state="red" />

        <inlane arm="X12" lane="0" state="red" />
        <inlane arm="X12" lane="-1" state="red" />
    </phase>
    <phase num="2">
        <inlane arm="X11" lane="0" state="red" />
        <inlane arm="X11" lane="-1" state="red" />

        <inlane arm="R10" lane="0" state="red" />
        <inlane arm="R10" lane="-1" state="red" />

        <inlane arm="X10" lane="0" state="green" />
        <inlane arm="X10" lane="-1" state="red" />

        <inlane arm="X12" lane="0" state="green" />
        <inlane arm="X12" lane="-1" state="red" />
    </phase>
    <phase num="3">
        <inlane arm="X11" lane="0" state="red" />
        <inlane arm="X11" lane="-1" state="green" />

        <inlane arm="R10" lane="0" state="red" />
        <inlane arm="R10" lane="-1" state="green" />

        <inlane arm="X10" lane="0" state="red" />
        <inlane arm="X10" lane="-1" state="red" />

        <inlane arm="X12" lane="0" state="red" />
        <inlane arm="X12" lane="-1" state="red" />
    </phase>
    <phase num="4">
        <inlane arm="X11" lane="0" state="red" />
        <inlane arm="X11" lane="-1" state="red" />

        <inlane arm="R10" lane="0" state="red" />
        <inlane arm="R10" lane="-1" state="red" />

        <inlane arm="X10" lane="0" state="red" />
        <inlane arm="X10" lane="-1" state="green" />

        <inlane arm="X12" lane="0" state="red" />
        <inlane arm="X12" lane="-1" state="green" />
    </phase>

</trafficLightsSchedule>

</intersection>

<intersection id="P26">

    <armActions arm="X2">
        <action lane="0" exit="X6"></action>
    </armActions>

    <armActions arm="X6">
        <action lane="0" exit="X2">
            </action>

```

```

    </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X2" lane="0" state="green" />
            <inlane arm="X6" lane="0" state="green" />
        </phase>

    </trafficLightsSchedule>
</intersection>

<intersection id="P1011">

    <armActions arm="X10">
        <action lane="0" exit="X11"></action>
    </armActions>

    <armActions arm="X11">
        <action lane="0" exit="X10">
            </action>
        </armActions>

    <trafficLightsSchedule>

        <phase num="1">
            <inlane arm="X11" lane="0" state="green" />
            <inlane arm="X10" lane="0" state="green" />
        </phase>

    </trafficLightsSchedule>
</intersection>

<intersection id="R1">
    <armActions arm="G1">
        <action lane="0" exit="X1" />
    </armActions>
    <armActions arm="X1">
        <action lane="0" exit="G1" />
    </armActions>

    <trafficLightsSchedule>
        <phase num="1">
            <inlane arm="X1" lane="0" state="green" />
            <inlane arm="G1" lane="0" state="green" />
        </phase>
    </trafficLightsSchedule>
</intersection>

<intersection id="R2">
    <armActions arm="G2">
        <action lane="0" exit="X2" />
    </armActions>
    <armActions arm="X2">
        <action lane="0" exit="G2" />
    </armActions>

    <trafficLightsSchedule>
        <phase num="1">

```

```

        <inlane arm="X2" lane="0" state="green" />
        <inlane arm="G2" lane="0" state="green" />
    </phase>
</trafficLightsSchedule>
</intersection>

<intersection id="R3">
    <armActions arm="G3">
        <action lane="0" exit="X3" />
    </armActions>
    <armActions arm="X3">
        <action lane="0" exit="G3" />
    </armActions>

    <trafficLightsSchedule>
        <phase num="1">
            <inlane arm="X3" lane="0" state="green" />
            <inlane arm="G3" lane="0" state="green" />
        </phase>
    </trafficLightsSchedule>
</intersection>

<intersection id="R4">
    <armActions arm="G4">
        <action lane="0" exit="X4" />
    </armActions>
    <armActions arm="X4">
        <action lane="0" exit="G4" />
    </armActions>

    <trafficLightsSchedule>
        <phase num="1">
            <inlane arm="X4" lane="0" state="green" />
            <inlane arm="G4" lane="0" state="green" />
        </phase>
    </trafficLightsSchedule>
</intersection>

<intersection id="R5">
    <armActions arm="G5">
        <action lane="0" exit="X9" />
    </armActions>
    <armActions arm="X9">
        <action lane="0" exit="G5" />
    </armActions>

    <trafficLightsSchedule>
        <phase num="1">
            <inlane arm="X9" lane="0" state="green" />
            <inlane arm="G5" lane="0" state="green" />
        </phase>
    </trafficLightsSchedule>
</intersection>

<intersection id="R6">
    <armActions arm="G6">
        <action lane="0" exit="X5" />
    </armActions>
    <armActions arm="X5">
        <action lane="0" exit="G6" />
    </armActions>

```



```

        <trafficLightsSchedule>
            <phase num="1">
                <inlane arm="X5" lane="0" state="green" />
                <inlane arm="G6" lane="0" state="green" />
            </phase>
        </trafficLightsSchedule>
    </intersection>

    <intersection id="R7">
        <armActions arm="G7">
            <action lane="0" exit="X8" />
        </armActions>
        <armActions arm="X8">
            <action lane="0" exit="G7" />
        </armActions>

        <trafficLightsSchedule>
            <phase num="1">
                <inlane arm="X8" lane="0" state="green" />
                <inlane arm="G7" lane="0" state="green" />
            </phase>
        </trafficLightsSchedule>
    </intersection>

    <intersection id="R8">
        <armActions arm="G8">
            <action lane="0" exit="X10" />
        </armActions>
        <armActions arm="X10">
            <action lane="0" exit="G8" />
        </armActions>

        <trafficLightsSchedule>
            <phase num="1">
                <inlane arm="X10" lane="0" state="green" />
                <inlane arm="G8" lane="0" state="green" />
            </phase>
        </trafficLightsSchedule>
    </intersection>

    <intersection id="R9">
        <armActions arm="G9">
            <action lane="0" exit="X12" />
        </armActions>
        <armActions arm="X12">
            <action lane="0" exit="G9" />
        </armActions>

        <trafficLightsSchedule>
            <phase num="1">
                <inlane arm="X12" lane="0" state="green" />
                <inlane arm="G9" lane="0" state="green" />
            </phase>
        </trafficLightsSchedule>
    </intersection>

    <intersection id="R10">
        <armActions arm="G10">
            <action lane="0" exit="X13" />
        </armActions>
    </intersection>

```

```

    <armActions arm="X13">
      <action lane="0" exit="G10" />
    </armActions>

    <trafficLightsSchedule>
      <phase num="1">
        <inlane arm="X13" lane="0" state="green" />
        <inlane arm="G10" lane="0" state="green" />
      </phase>
    </trafficLightsSchedule>
  </intersection>

</intersectionDescriptions>
</RoadNet>

```

## 7.6. Załącznik 6: traffic-krakow\_med.xml

```
<?xml version="1.0"?>
<!--
    czas trwania schematu: 12h = 21600

    pomiędzy (prawie) każdą parą węzłów generowane jest 800 samochodów
    z rozkładem jednostajnym
-->
<traffic>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G2" />
    </scheme>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G3" />
    </scheme>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G4" />
    </scheme>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G5" />
    </scheme>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G6" />
    </scheme>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G7" />
    </scheme>

    <scheme count="800">
        <gateway id="G1">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G8" />
    </scheme>

</traffic>
```

```

</scheme>

<scheme count="800">
  <gateway id="G1">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G9" />
</scheme>

<scheme count="800">
  <gateway id="G1">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G1" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G3" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G4" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G5" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G6" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G7" />
</scheme>

```

```

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G8" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G9" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="500" b="10000" />
  </gateway>
  <gateway id="G9" />
</scheme>

<scheme count="800">
  <gateway id="G2">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
  <gateway id="G3">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G1" />
</scheme>

<scheme count="800">
  <gateway id="G3">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G2" />
</scheme>

<scheme count="800">
  <gateway id="G3">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G4" />
</scheme>

<scheme count="800">
  <gateway id="G3">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G5" />
</scheme>

<scheme count="800">
  <gateway id="G3">

```

```

        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G6" />
</scheme>

<scheme count="800">
    <gateway id="G3">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G7" />
</scheme>

<scheme count="800">
    <gateway id="G3">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G8" />
</scheme>

<scheme count="1000">
    <gateway id="G3">
        <uniform a="0" b="10000" />
    </gateway>
    <gateway id="G9" />
</scheme>

<scheme count="800">
    <gateway id="G3">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G9" />
</scheme>

<scheme count="800">
    <gateway id="G3">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G10" />
</scheme>

<scheme count="800">
    <gateway id="G3">
        <uniform a="900" b="21600" />
    </gateway>
    <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G1" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>

```

```

        <gateway id="G2" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G3" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G5" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G6" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G7" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G8" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G9" />
</scheme>

<scheme count="800">
    <gateway id="G4">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
    <gateway id="G5">
        <uniform a="0" b="21600" />
    </gateway>
    <gateway id="G1" />
</scheme>

```

```

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G2" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G3" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G4" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G6" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G7" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G8" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G9" />
</scheme>

<scheme count="800">
  <gateway id="G5">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
  <gateway id="G6">
    <uniform a="0" b="21600" />

```



```

        </gateway>
        <gateway id="G1" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G2" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G3" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G4" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G5" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G7" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G8" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G9" />
    </scheme>

    <scheme count="800">
        <gateway id="G6">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G10" />
    </scheme>

    <scheme count="800">

```

```

        <gateway id="G6">
            <uniform a="10000" b="18000" />
        </gateway>
        <gateway id="G10" />
    </scheme>
    <!-->

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G1" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G2" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G3" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G4" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G5" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G6" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G8" />
    </scheme>

    <scheme count="800">
        <gateway id="G7">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G9" />
    </scheme>

```

```

</scheme>

<scheme count="800">
  <gateway id="G7">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G1" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G2" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="5000" b="21600" />
  </gateway>
  <gateway id="G2" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G3" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G4" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G5" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G6" />
</scheme>

```

```

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G7" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G9" />
</scheme>

<scheme count="800">
  <gateway id="G8">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G10" />
</scheme>

<!-->

<scheme count="800">
  <gateway id="G9">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G1" />
</scheme>

<scheme count="1000">
  <gateway id="G9">
    <uniform a="15000" b="21600" />
  </gateway>
  <gateway id="G2" />
</scheme>

<scheme count="800">
  <gateway id="G9">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G2" />
</scheme>

<scheme count="800">
  <gateway id="G9">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G3" />
</scheme>

<scheme count="800">
  <gateway id="G9">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G4" />
</scheme>

<scheme count="800">
  <gateway id="G9">
    <uniform a="0" b="21600" />

```

```

        </gateway>
        <gateway id="G5" />
    </scheme>

    <scheme count="800">
        <gateway id="G9">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G6" />
    </scheme>

    <scheme count="800">
        <gateway id="G9">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G7" />
    </scheme>

    <scheme count="800">
        <gateway id="G9">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G8" />
    </scheme>

    <scheme count="800">
        <gateway id="G9">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G10" />
    </scheme>

    <!-->

    <scheme count="800">
        <gateway id="G10">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G1" />
    </scheme>

    <scheme count="800">
        <gateway id="G10">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G2" />
    </scheme>

    <scheme count="800">
        <gateway id="G10">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G3" />
    </scheme>

    <scheme count="800">
        <gateway id="G10">
            <uniform a="0" b="21600" />
        </gateway>
        <gateway id="G4" />
    </scheme>

```

```

<scheme count="800">
  <gateway id="G10">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G5" />
</scheme>

<scheme count="800">
  <gateway id="G10">
    <uniform a="2000" b="21600" />
  </gateway>
  <gateway id="G6" />
</scheme>

<scheme count="700">
  <gateway id="G10">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G6" />
</scheme>

<scheme count="800">
  <gateway id="G10">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G7" />
</scheme>

<scheme count="800">
  <gateway id="G10">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G8" />
</scheme>

<scheme count="800">
  <gateway id="G10">
    <uniform a="0" b="21600" />
  </gateway>
  <gateway id="G9" />
</scheme>

</traffic>

```