UNIWERSYTET IM. ADAMA MICKIEWICZA
W POZNANIU

Wydział Matematyki i Informatyki

Modelowanie i symulacja zużycia paliwa i emisji spalin w ruchu miejskim

Modeling and Simulation of Fuel Consumption and Emission in City Traffic

Patryk Żywica

PRACA MAGISTERSKA
wykonana pod kierunkiem
prof. UAM dr hab. Macieja Wygralaka

Poznań 2012

Pracę dedykuję mojej najdroższej Ani.

Jednocześnie składam serdeczne podziękowania dr Janowi Wachowiakowi za pomoc w nawiązaniu współpracy z Zarządem Dróg Miejskich w Poznaniu oraz pracownikom tej instytucji za cenne wskazówki dotyczące modelu ruchu.

Poznań, dnia 11 stycznia 2015

OŚWIADCZENIE

Ja, niżej podpisany Patryk Żywica student Wydziału Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza w Poznaniu oświadczam, że przedkładaną pracę dyplomową pt:

„*Modelowanie i symulacja zużycia paliwa i emisji spalin w ruchu miejskim*"

napisałem samodzielnie. Oznacza to, że przy pisaniu pracy, poza niezbędnymi konsultacjami, nie korzystałem z pomocy innych osób, a w szczególności nie zlecałem opracowania rozprawy lub jej części innym osobom, ani nie odpisywałem tej rozprawy lub jej części od innych osób.

Oświadczam również, że egzemplarz pracy dyplomowej w formie wydruku komputerowego jest zgodny z egzemplarzem pracy dyplomowej w formie elektronicznej.

Jednocześnie przyjmuję do wiadomości, że gdyby powyższe oświadczenie okazało się nieprawdziwe, decyzja o wydaniu mi dyplomu zostanie cofnięta.

Streszczenie

Praca poświęcona jest modelowaniu i symulacji zużycia paliwa oraz emisji spalin w ruchu miejskim. Dokonano przeglądu modeli, zarówno historycznych, jak i wykorzystywanych współcześnie. Szczególnie wiele uwagi poświęcono modelowaniu z wykorzystaniem komputerowych symulacji ruchu drogowego. Zaprezentowano oryginalny pakiet oprogramowania *ProSym*, który m.in. zawiera serwis Mapa Ruchu Drogowego (MaRuD), oferujący informacje o natężeniu ruchu, zużyciu paliwa i emisji spalin. Zaproponowano również autorską metodę aktualizacji modelu ruchu wykorzystywanego do symulacji, opartą na sterowaniu rozmytym oraz danych od kierowców.

Słowa kluczowe emisja spalin, modelowanie, zużycie paliwa, symulacja, ruch drogowy, macierz OD, sterowanie rozmyte

Summary

This work is devoted to modeling and simulation of fuel consumption and emissions in city traffic. Both historical and contemporary models of emission and fuel consumption are presented. In particular, much attention was paid to modeling using computer traffic simulations. The paper presents a new software package *ProSym*, which includes Road Map service (MaRuD), offering information about traffic, fuel consumption and emissions. Also proposed is a novel method of updating the traffic model used in simulation, based on fuzzy control and user data.

Keywords emissions, modeling, fuel consumption, simulation, traffic, OD matrix, fuzzy control

Spis treści

Streszczenie **iv**

Wstęp **ix**

1. Problematyka zużycia paliwa i emisji spalin **11**

1.1. Wprowadzenie 11

1.2. Czynniki wpływające na zużycie paliwa i emisję spalin 13

1.3. Modele zużycia paliwa i emisji spalin 16

1.3.1. Modele statyczne 16
1.3.2. Modele dynamiczne 17
1.3.2.1. Mapy emisji 18
1.3.2.2. Modele regresyjne 19
1.3.2.3. Modele oparte na zużyciu paliwa 19

2. Symulacja ruchu drogowego **20**

2.1. Elementy teorii systemów 20

2.2. Modele ruchu drogowego 21

2.2.1. Popyt transportowy 22
2.2.2. Infrastruktura drogowa 23
2.2.3. Wybór trasy 24
2.2.4. Metody modelowania potoków ruchu 25
2.2.4.1. Podejście makroskopowe 26
2.2.4.2. Podejście mikroskopowe 27
2.2.4.3. Podejście mezoskopowe 29

2.3. Komputerowa symulacja ruchu drogowego 29

2.4. Symulator SUMO 32

2.5. Symulacja w Polsce i Poznaniu 35

3. Modelowanie i sterowanie rozmyte **38**

3.1. Elementy teorii zbiorów rozmytych 38

3.1.1. Definicja zbioru rozmytego 38

3.1.2. Operacje na zbiorach rozmytych 41

3.2. Sterowanie rozmyte 43

3.2.1. FCL – Fuzzy Control Language 45

3.3. Zastosowania logiki rozmytej w modelowaniu ruchu drogowego . 46

3.3.1. Sterowanie sygnalizacją świetlną 46

3.3.2. Określanie popytu transportowego 46

3.3.3. Wybór trasy przejazdu 47

4. Pakiet oprogramowania *ProSym* **48**

4.1. Idea 48

4.2. Skład pakietu 49

4.2.1. MaRuD 49

4.2.2. KorMOD 51

4.2.3. Wersja mobilna 51

4.3. Wymagania 53

4.4. Wdrożenie i dalszy rozwój 53

4.5. *ProSym* jako projekt informatyczny 54

5. Mapa Ruchu Drogowego – MaRuD **55**

5.1. Zasada działania 55

5.2. Symulacja ruchu drogowego 59

5.3. Ocena jakości wyników symulacji 61

6. Korektor Macierzy OD – KorMOD **64**

6.1. Zasada działania 65

6.2. Mechanizm aktualizacji macierzy OD 65

6.2.1. Algorytm aktualizacji 65

6.2.2. Faza I 67

6.2.2.1. Dane wejściowe sterownika rozmytego KorMOD^{I} . . . 67

6.2.2.2. Sterownik rozmyty KorMOD^{I} 69

6.2.3. Faza II 71

6.2.3.1. Dane wejściowe sterownika rozmytego KorMOD^{II} . . . 71

6.2.3.2. Sterownik rozmyty KorMOD^{II} 72

6.3. Ocena jakości otrzymanych macierzy 73

6.3.1. Metodyka oceny 73

6.3.2. Scenariusze testowe 75

6.3.3. Wyniki testów 75

6.3.3.1. Kalibracja 76

6.3.3.2. Scenariusz „stabilizacja” 77

6.3.3.3. Scenariusz „losowy” 78

6.3.3.4. Scenariusz „osiedle” 79

6.3.4. Omówienie wyników 80

7. Podsumowanie **85**

Literatura **87**

Spis rysunków **92**

Spis tablic **93**

Dodatki **94**

A. Uwagi o zastosowanej terminologii **95**

A.1. Słownik pojęć 95

B. Architektura systemu *ProSym* **96**

B.1. Ogólna architektura systemu 96

B.2. Wykorzystane technologie 96

B.3. Architektura aplikacji MaRuD 98

B.4. Architektura modułu KorMOD 98

C. Sterownik rozmyty KorMOD **100**

Wstęp

Intensywny rozwój transportu drogowego, który miał miejsce w ostatnich dekadach, spowodował znaczący wzrost zanieczyszczenia powietrza w miastach. Stworzyło to potrzebę monitorowania i zarządzania emisją spalin. Wprowadzane w wielu krajach coraz to ostrzejsze normy dotyczące parametrów pojazdów, czy nowe przepisy ruchu drogowego, mogą okazać się nieskuteczne, jeśli wcześniej nie zostanie opracowana efektywna metoda oceny ich skuteczności.

Praca poświęcona jest modelowaniu i symulacji zużycia paliwa oraz emisji spalin. Podzielona została na część literaturową (rozdziały 1, 2 oraz 3), w której omówione zostały podstawy teoretyczne, wraz z niezbędnymi pojęciami oraz część praktyczną (rozdziały 4, 5 oraz 6), w której przedstawione zostało oprogramowanie *ProSym*, stworzone w ramach tej pracy.

Pierwszy rozdział wprowadza czytelnika w problematykę emisji spalin i zużycia paliwa. Zawarto w nim informacje o wpływie transportu, a szczególnie ruchu drogowego na środowisko naturalne. Modelowanie dowolnego zjawiska nie może się odbywać bez poznania czynników na nie oddziałujących, dlatego w dalszej części omówione zostały czynniki wpływające na emisję zanieczyszczeń. Rozdział zakończony jest klasyfikacją i przykładami popularnych modeli. Większość z nich wymaga jednak danych, których pozyskanie w świecie rzeczywistym jest trudne i kosztowne, stąd ich zastosowanie do niedawna było znacząco ograniczone.

Gwałtowny rozwój symulacji ruchu drogowego miał miejsce dopiero kilkanaście lat temu, lecz nie oznacza to, że jest to dziedzina młoda i słabo rozwinięta. Rozdział drugi opisuje zagadnienie symulacji ruchu drogowego w ujęciu teorii systemów. Przedstawione zostały również najważniejsze metody modelowania poszczególnych komponentów systemu oraz ich interakcji. Rozwój techniki spowodował znaczący wzrost mocy obliczeniowej komputerów, przy jednoczesnym spadku kosztów. Stąd obecnie symulacja komputerowa jest jedyną skuteczną metodą modelowania ruchu drogowego. Dalsza część rozdziału poświęcona jest przeglądowi oprogramowania dostępnego na rynku, zarówno darmowego, jak i komercyjnego.

Rozdział trzeci stanowi wprowadzenie do teorii zbiorów rozmytych. Omówiono w nim podstawowe pojęcia. Następnie przedstawiono zagadnienie sterowania rozmytego oraz język FCL, stosowany do jego implementacji. Rozdział kończy się prezenta-

cją kilku przykładów zastosowania logiki rozmytej do modelowania i kontrolowania ruchu drogowego.

W czwartym rozdziale zaprezentowano pakiet oprogramowania *ProSym*. Omówiony został obszar jego zastosowań, podstawowe funkcjonalności oraz wymagania. Następne w rozdziałach piątym i szóstym szczegółowo przedstawiono zasadę działania dwóch najważniejszych składników pakietu – systemu MaRuD oraz komponentu KorMOD.

Najistotniejszym rozdziałem pracy jest rozdział szósty, w którym przedstawiona została autorska metoda aktualizacji modelu ruchu drogowego oparta na sterowaniu rozmytym oraz informacjach pochodzących od kierowców. Opisane zostały również wyniki badań nad skutecznością zaproponowanego rozwiązania.

Rozdział 1

Problematyka zużycia paliwa i emisji spalin

Niniejszy rozdział poświęcony jest tematyce zużycia paliwa i emisji spalin. Po krótkim wprowadzeniu oraz przedstawieniu bieżących statystyk, omówione zostaną czynniki mające wpływ na produkcję zanieczyszczeń. Następnie zaprezentowane zostaną różne podejścia do modelowania emisji spalin i zużycia paliwa wraz z najpopularniejszymi modelami.

1.1. Wprowadzenie

Transport jest jednym z głównych czynników mających wpływ na poziom emisji spalin do atmosfery. Ogólnie zanieczyszczenia powietrza mogą być podzielone na pochodzące z czterech głównych źródeł [Hor82]:

- transport – głównie ruch drogowy,
- spalanie paliw kopalnych - głównie elektrownie,
- procesy przemysłowe,
- składowanie odpadów.

Szacuje się, że w Stanach Zjednoczonych zanieczyszczenia pochodzenia transportowego stanowią około 45 procent całkowitej emisji, z czego sam ruch drogowy to ponad $\frac{1}{3}$[NMM94].

Z badań Europejskiej Agencji Ochrony Środowiska wynika, że w 2004 roku transport przyczynia się do emisji aż 45% związków ozonotwórczych, z czego transport

drogowy stanowi 33%. Ponadto 22% całkowitej emisji substancji drobnocząsteczkowych oraz 14% substancji zakwaszających wytwarzane jest przez poruszające się po drogach pojazdy[eea07a].

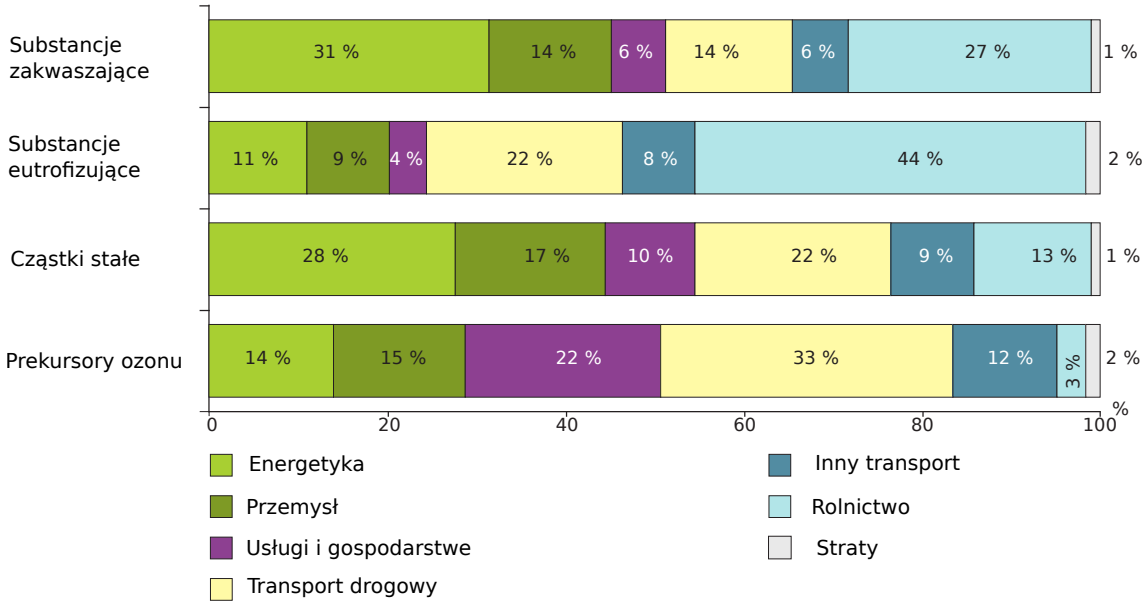
Większość zanieczyszczeń powstaje w procesie spalania paliwa oraz przez samoczynne jego odparowywanie. Benzyna oraz olej napędowy składają się węglowodorów oraz innych związków wodoru i węgla. W procesie idealnego spalania, wodór zawarty w paliwie wchodzi w reakcję z tlenem tworząc wodę. Podobnie węgiel w reakcji z tlenem tworzy dwutlenek węgla:

$$CH_4 + 2O_2 \xrightarrow{spalanie} 2H_2O + CO_2$$

$$2CH_4 + 3O_2 \xrightarrow{spalanie} 4H_2O + 2CO\,.$$

Niestety warunki idealnego spalania są nieosiągalne w rzeczywistości, a co za tym idzie w procesie spalania powstaje wiele produktów ubocznych [epa94, eea05].

- Tlenek węgla (CO) - powszechnie nazywany czadem, powstaje w procesie niecałkowitego spalania przy niedoborze tlenu, jest to bezbarwny i bezwonny trujący gaz. Jego toksyczne działanie polega na trwałym wiązaniu hemoglobiny we krwi, co w konsekwencji uniemożliwia transport tlenu w organizmie,
- Węglowodory (HC) - grupa związków chemicznych zawierających wyłącznie atomy węgla i wodoru. Pod wpływem promieni słonecznych wchodzą w reakcję z tlenkami azotu przyczyniając się do powstawania ozonu, głównego składnika smogu. Ponadto długotrwały kontakt z niektórymi węglowodorami może mieć działanie kancerogenne[epa, Deg95],



Rys. 1.1.1. Udział poszczególnych sektorów w emisji wybranych zanieczyszczeń. Źródło [eea07a].

- Związki azotu (NOx) - głównie dwutlenek azotu (NO_2), kwas azotowy III (HNO_2) oraz kwas azotowy V (HNO_3). Powstają one podczas reakcji tlenu i azotu w warunkach wysokiego ciśnienia i temperatury. Przyczyniają się do powstawania ozonu oraz są jedną z głównych przyczyn kwaśnych deszczy. Powodują zaburzenia pracy układu oddechowego,

- Pył - mieszanina różnych cząsteczek o średnicy poniżej 10 mikrometrów, w skład której wchodzą kwasy, związki organiczne, metale i popiół. U człowieka powoduje głównie kaszel, problemy z oddychaniem oraz zaburzenia pracy serca. Ich duże stężenie powoduje ograniczenie widoczności.

Dwutlenek węgla, powstający wskutek całkowitego spalania paliwa, również nie jest obojętny dla atmosfery. Mimo, że jest naturalnym jej składnikiem, a co za tym idzie nie jest uważany za zanieczyszczenie[epa01], to jest on głównym gazem cieplarnianym (około 80 procent).

Współczynnik udziału powietrza i paliwa w spalaniu jest jednym z najważniejszych czynników wpływających na procesy spalania w silniku i reaktorze katalitycznym. W warunkach nadmiaru paliwa produkowane są tlenki węgla i węglowodory, natomiast największa produkcja związków azotowych zachodzi przy nadmiarze tlenu. Stąd emisja CO i HC jest największa podczas zimnego startu silnika oraz przy dużym jego obciążeniu. Oznacza to, że związki te emitowane są głównie podczas jazdy po zatłoczonej drodze.

1.2. Czynniki wpływające na zużycie paliwa i emisję spalin

Czynniki wpływające na emisję spalin, jak również sama ich wielkość są blisko powiązane z czynnikami zużycia paliwa. Zostały one podzielone na cztery grupy. W każdej grupie omówiono główne oraz mniej istotne czynniki. Należy zwrócić uwagę na to, że na niektóre z wymienionych czynników nie mamy wpływu (np. pogoda, uwarunkowania geograficzne). Inne są całkowicie zależne od kierowców, technologii, prawodawstwa czy mody.

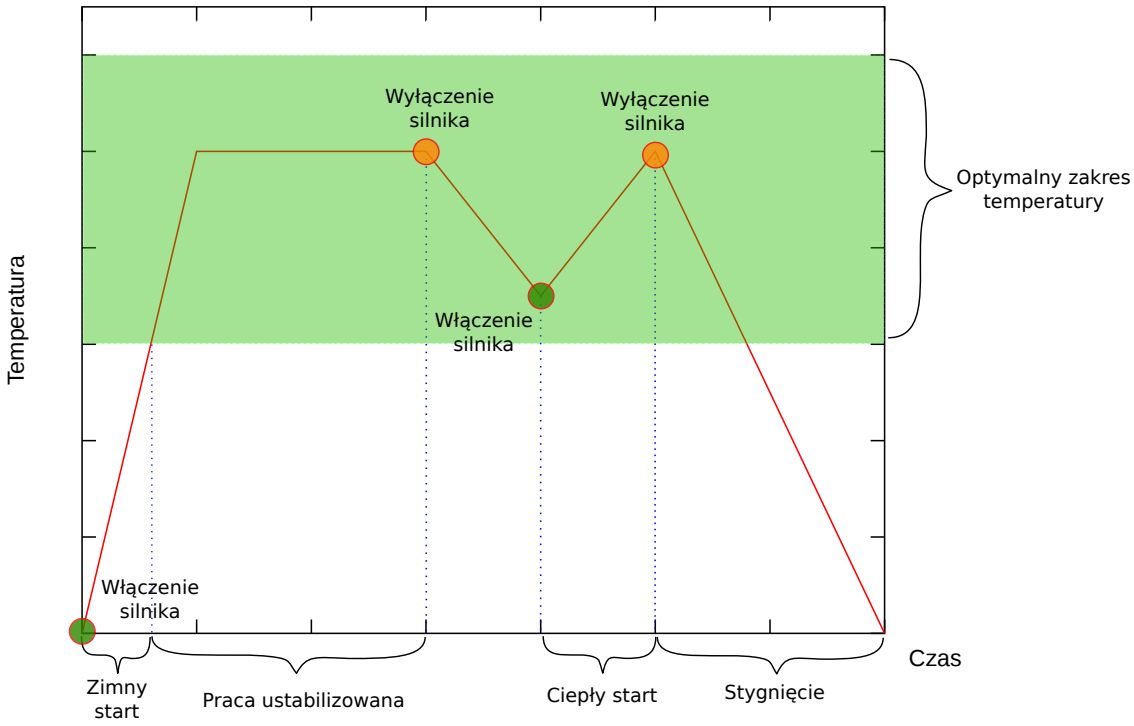
Czynniki mające wpływ na zużycie paliwa i emisję spalin przedstawiono poniżej [eea07b].

1. Czynniki zależne od podróży

   - łączna przebyta droga - kluczowy czynnik tej grupy. Przebycie większej drogi wymaga zużycia większej ilości paliwa, a co za tym idzie powoduje wyemitowanie większej ilości spalin. Jednak podczas porównywania różnych tras przejazdu pomiędzy punktem A i B, nie powinien pełnić kluczowej roli, gdyż przy niewielkich różnicach w długości trasy, znacznie większy jest wpływ pozostałych czynników.
   - prędkość jazdy - silniki współczesnych samochodów projektowane są do pracy w określonych warunkach, w których mają najlepsze osiągi. Poruszanie się ze zbyt małą lub dużą prędkością zmusza silnik do pracy poza zakresem optymalnych warunków. Co więcej, wraz ze wzrostem prędkości rośnie opór tarcia, zarówno kół o jezdnię, jak i całego pojazdu i powietrza.
   - charakterystyka zmian prędkości - częste hamowanie i przyspieszanie negatywnie wpływa na zużycie paliwa, na skutek utraty energii kinetycznej podczas hamowania. Zmiana prędkości jazdy ma jednak kluczowy wpływ na emisję spalin. Jak zostało wcześniej wspomniane, wielkość oraz skład spalin w dużej mierze zależą od stosunku powietrza do paliwa w silniku. Podczas hamowania zawartość powietrza gwałtownie wzrasta, a pojazd emituje więcej związków azotu. Podczas intensywnego przyspieszania występuje niedobór powietrza, a zatem produkowane są węglowodory i tlenek węgla.
   - udział poszczególnych faz pracy silnika (rysunek 1.2.1) – każda z faz pracy silnika posiada własną charakterystykę zużycia paliwa i emisji spalin. Podczas zimnego startu silnik zużywa więcej paliwa, gdyż pracuje w nieoptymalnych warunkach. Dlatego im większy udział tej fazy, tym emisja i zużycie paliwa będą większe. Podczas stygnięcia zużycie paliwa jest śladowe (ale zachodzi wskutek jego odparowania w ciepłym silniku). Emisja spalin w tej fazie zachodzi głównie w rozgrzanym silniku, gdyż jeszcze przez pewien czas po zakończeniu pracy zachodzą przemiany pozostałości paliwa i produktów jego spalania. Chociaż ich udział w łącznej emisji nie jest duży, należy również o nich pamiętać.

2. Charakterystyka infrastruktury drogowej

   - typ i jakość nawierzchni - nawierzchnie o nieodpowiednim współczynniku tarcia, czy drogi nieodpowiednio utwardzone, zwiększają straty energii podczas jazdy. Według badań w [Bak94] zła jakość nawierzchni zwiększa zużycie paliwa średnio o 5 procent.
   - ukształtowanie terenu - teren górzysty, kręte drogi powodują częste hamowanie i przyspieszanie.



Rys. 1.2.1. Cztery podstawowe fazy pracy silnika: zimny start - okres pracy od uruchomienia silnika do osiągnięcia przez niego optymalnej temperatury, ciepły start - uruchomienie silnika po krótkotrwałej przerwie, zanim temperatura spadnie poniżej optymalnej, praca ustabilizowana - temperatura silnika jest optymalna, stygnięcie - okres od wyłączenia silnika do uzyskania przez niego temperatury otoczenia (opracowanie własne).

- organizacja ruchu - skrzyżowania, sygnalizacje świetlne zmuszają pojazdy do zatrzymywania się [JY12]. Podobnie ograniczenia prędkości, konstrukcję zmniejszające prędkość powodują konieczność hamowania i przyspieszania. Należy zwrócić uwagę, że największy wpływ czynników tej grupy, ma miejsce na terenach miejskich.

3. Charakterystyka pojazdu

- masa, rozmiar i kształt pojazdu - im większa masa tym większą pracę musi wykonać silnik, aby zmienić prędkość pojazdu. Rozmiar i kształt mają wpływ na aerodynamikę zmniejszając, bądź zwiększając opór powietrza.
- wyposażenie dodatkowe - współcześnie samochody wyposażone są w coraz więcej systemów wspomagających oraz zwiększających komfort jazdy takich, jak automatyczne szyby, ogrzewanie, klimatyzacja. Większość tych urządzeń zasilanych jest prądem elektrycznym, obciążając dodatkowo alternator, co powoduje wzrost obciążenia silnika i zużycia paliwa.

- konserwacja pojazdu - nieodpowiednia konserwacja pojazdu może znaczenie wpłynąć na parametry jego eksploatacji. Przykładowo niewłaściwe ustawienia silnika zwiększają średnio zużycie paliwa o 10 procent, a zwiększenie szerokości opon o 2 mm, podnosi zużycie paliwa o 3 procent[Bak94].
- wiek - producenci samochodów zobligowani są do przestrzegania coraz bardziej surowych norm emisji spalin. Stąd nowsze pojazdy emitują mniej spalin. Należy jednak zwrócić uwagę, że zastosowane w tych pojazdach systemy zmniejszające emisję spalin, z wiekiem tracą wydajność.

4. Inne

- warunki atmosferyczne - wiatr, deszcz czy śnieg zwiększają opór jazdy, oraz straty energii. Niska temperatura natomiast wydłuża czas zimnego startu silnika. Zbyt wysoka temperatura powoduje z kolei jego przegrzanie. W Europie średnie zużycie paliwa zimą jest o 15 do 20 procent większe niż latem [Bak94].
- natężenie ruchu drogowego - podczas jazdy w korku konieczne jest częste zatrzymywanie się i ruszanie, przez co zużywane jest znacznie więcej paliwa niż podczas swobodnej jazdy. Ogólnie im większe natężenie ruchu, tym większa jest emisja spalin i zużycie paliwa.
- niespodziewane zdarzenia drogowe - wypadki drogowe, uszkodzenia jezdni, remonty, objazdy wpływają negatywnie na wiele wymienionych wcześniej czynników.

1.3. Modele zużycia paliwa i emisji spalin

Problem modelowania zużycia paliwa i emisji spalin jest bardzo skomplikowany. Powodów jest wiele. Po pierwsze, należy wymienić bardzo dużą liczbę czynników, z których wiele jest trudnych do obiektywnego zmierzenia, co powoduje konieczność wprowadzenia uproszczeń. Ponadto brak jest jednoznacznych badań pozwalających wyznaczyć odpowiednie zależności. Po drugie, emisja spalin i zużycie paliwa w dużym stopniu zależą od konkretnego pojazdu czy zachowań kierowcy.

Ze względu na podejście do problemu, modele można podzielić na dwie grupy: modele statyczne oparte na średniej prędkości oraz modele dynamiczne. Druga grupa pod względem zastosowanych metod dzieli się na trzy podgrupy.

1.3.1. Modele statyczne

Modele statyczne w celu wyznaczenia emisji spalin korzystają z zależności pomiędzy całkowitą drogą przebytą przez pojazdy w sieci oraz ich średnią prędkością.

Przyjmując oznaczenie E_i na całkowitą emisję spalin typu i (lub zużycia paliwa E_f) w pewnym ustalonym przedziale czasu i sieci drogowej, ogólny wzór na tę wartość w modelach statycznych przyjmuje postać

$$E_i = \sum_c \sum_l KP_l \cdot p_c \cdot WE_i(\bar{s}_l, c)\,,$$

gdzie:

l – indeks podsieci (np: drogi, skrzyżowania, lub fragmentu sieci),

c – kategoria pojazdu (np: samochód osobowy, dostawczy),

KP_l – łączna liczba kilometrów przejechanych w danej podsieci l,

p_c – procentowa ilość pojazdów kategorii c,

$\bar{s}_l$ – średnia prędkość w podsieci l,

$WE_i(\bar{s}_l, c)$ – współczynnik emisji spalin typu i dla pojazdów typu c przy średniej prędkości $\bar{s}_l$.

Modele te są stosunkowo proste, gdyż wymagają tylko zagregowanych wartości przebytej drogi, prędkości oraz rozkładu typów pojazdów. Modele tego typu nie nadają się do szacowania chwilowych wahań emisji (zużycia paliwa) i dlatego wykorzystywane są głównie w warunkach ustabilizowanego ruchu na dużych obszarach. To z kolei prowadzi do dużych przeszacowań lub niedoszacowań wynikających ze zmienności warunków drogowych na tych obszarach.

Przykładami powszechnie wykorzystywanych modeli statycznych są:

- MOBILE6 - opracowany przez United States Environmental Protection Agency [U.S03],
- Motor Vehicle Emission Inventory (MVEI) - opracowany przez California Air Resources Board [Cal96].

1.3.2. Modele dynamiczne

Modele dynamiczne uzależniają emisję spalin (lub zużycie paliwa) od chwilowych wartości wektora parametrów oraz typu pojazdu. Każdy pojazd z osobna posiada swój wkład w łączną wartość dla całej sieci. Najczęściej stosowany wektor parametrów składa się z obecnej prędkości oraz przyspieszenia.

Niech $E_i(t)$ będzie całkowitą emisją spalin typu i (lub zużyciem paliwa E_f) w ustalonej sieci drogowej w czasie t. Ogólny wzór na tę wartość ma postać

$$E_i(t) = \sum_j e_i(c_j, x_j(t))\,,$$

gdzie:

j – identyfikator pojazdu,

c_j – kategoria pojazdu j,

$x_j(t)$ – wektor wartości parametrów w chwili t,

$e_i(c_j, x_j(t))$ – emisja typu i dla pojazdu kategorii c_j i wartości parametrów $x_j(t)$.

Całkowitą emisję spalin E_i (lub zużycie paliwa) oblicza się jako

$$E_i = \int_{t_p}^{t_k} E_i(t)\mathrm{d}t\,,$$

gdzie t_p i t_k to czas początkowy oraz końcowy,

Całkę tę najczęściej wylicza się numerycznie metodą prostokątów stosując stałą szerokość prostokąta równą 1 sekundzie. Wtedy wzór przybiera następującą postać:

$$E_i = \sum_{t=t_p}^{t_k} E_i(t)\,.$$

W porównaniu z modelami statycznymi są one znacznie bardziej dokładne. Pozwalają uwzględnić chwilowe zmiany zużycia paliwa, przyspieszenia czy postój pojazdu w korku. Ich wadą przede wszystkim są wymagania wobec danych wejściowych. Wartości wektora parametrów muszą być znane w każdej sekundzie. Zbieranie tych danych jest dość trudne w świecie rzeczywistym, ponadto do ich przechowywania potrzeba niemałej ilości pamięci. Z tego też powodu obliczenia z wykorzystaniem modeli dynamicznych wymagają sporego nakładu obliczeniowego i do niedawna wykorzystywane były tylko do symulowania emisji (lub zużycia paliwa) pojedynczych pojazdów lub skrzyżowań. Obecnie wraz ze wzrostem mocy obliczeniowej komputerów oraz dostępności danych, możliwe jest wykorzystywanie coraz bardziej dokładnych modeli na większą skalę.

Ze względu na metodę wyrażenia zależności oraz dobór parametrów wejściowych, modele dynamiczne można podzielić na trzy grupy.

1.3.2.1. Mapy emisji

Mapy emisji najczęściej są realizowane w postaci dwuwymiarowych macierzy. Podają wartość emisji (lub zużycia paliwa) w zależności od prędkości oraz przyspieszenia. Tabele te są bardzo proste w przygotowaniu oraz użyciu i dlatego są często stosowane. Posiadają jednak kilka wad, których należy być świadomym. Przede wszystkim wartości emisji są zależne od warunków, w jakich zostały przeprowadzone badania (np: zimny start, testowany samochód itp.). Dlatego powszechne jest tworzenie katalogów map emisji uzyskanych dla różnych pojazdów w różnych, ale ustandaryzowanych warunkach [HHJ99, SBdH88].

Przyspieszenie (m/s)	Prędkość (km/h)									
	0	5	15	25	35	45	55	65	75	85
-15	-	-	66	56	63	69	59	76	92	115
-10	-	-	57	61	63	84	94	141	129	134
-5	-	53	53	73	85	102	130	204	194	325
0	33	59	74	116	123	131	196	193	274	152
5	-	142	163	192	192	207	275	263	350	211
10	-	-	274	301	295	357	330	454	403	275
15	-	-	-	469	568	603	779	706	1041	308

Tab. 1.3.1. Przykładowa mapa emisji CO (g/h). Źródło [HHJ99].

1.3.2.2. Modele regresyjne

Główną wadą map emisji jest dyskretność oferowanego modelu, co w znacznym stopniu utrudnia ich wykorzystanie w symulacjach komputerowych. Modele regresyjne nie posiadają tej niedogodności. Uzyskiwane są przez interpolowanie (najczęściej liniowe lub wielomianowe) danych uzyskanych analogicznie do map emisji. Rozwiązują problem konieczności tworzenia wielu map (katalogu), poprzez uzależnienie modelu od większej ilości parametrów.

Równania często jednak zawierają współczynniki, których wartości nie mają jasnej interpretacji. Konieczna jest zatem dokładna kalibracja modelu celem dobrania ich optymalnych wartości. Trudniejsze jest również zrozumienie takiego modelu.

Przykładowy model emisji spalin wykorzystywany w symulatorze SUMO, powstały przez interpolację mapy emisji HBEFA [Ger12]:

$$E_i(t,x) = \begin{cases} 0 & x_1 < 0, \\ \max\left(0, a + bx_0 \arcsin \frac{x_1}{9{,}81} + cx_0 \arcsin^2 \frac{x_1}{9{,}81} + dv_0 + ex_0^2 + fx_0^3\right) & x_1 \geq 0, \end{cases}$$

gdzie $x = (x_0, x_1)$ oraz x_0 oznacza aktualną prędkość w $\frac{m}{s}$, a x_1 to przyspieszenie w $\frac{m}{s^2}$. Współczynniki a, b, c, d, e, f są wyznaczone dla różnych klas pojazdów wykorzystywanych w symulacji.

1.3.2.3. Modele oparte na zużyciu paliwa

Poprzednie dwa podejścia uzależniały emisję (zużycie paliwa) w głównej mierze od prędkości i przyspieszenia. Opisywane tu modele traktują zużycie paliwa jako parametr wejściowy, natomiast emisja spalin uzyskiwana jest przez uwzględnienie charakteru przemian chemicznych, jakim ulega paliwo. Wspomniane modele są bardzo dokładne, jednak zarazem bardzo specyficzne dla konkretnego pojazdu [NBS02].

Rozdział 2

Symulacja ruchu drogowego

Przedstawione w poprzednim rozdziale modele zużycia paliwa i emisji spalin, za wyjątkiem modeli statycznych, wymagają dodatkowych danych. Oczywiście możliwe jest ich pozyskanie z rzeczywistych pojazdów przez instalację urządzeń pomiarowych, jednak będą one dotyczyły tylko bardzo małej frakcji pojazdów, która z dużym prawdopodobieństwem nie będzie odzwierciedlała całości ruchu na danym obszarze.

Wraz z rozwojem techniki, symulacje komputerowe cieszą coraz większą popularnością. Trend ten szczególnie dotyczy transportu. Zastosowanie symulacji do ruchu drogowego spowodowało przełom. Możliwość zbierania danych dotyczących niemal każdego aspektu systemu transportowego, przy zachowaniu dużej ich jakości, pozwoliło na rozwój, jak i praktyczne wykorzystanie wielu modeli emisji spalin i zużycia paliwa.

Poniższy rozdział poświęcony jest modelowaniu i symulacji ruchu drogowego. Na początku przedstawione zostaną elementy teorii systemów. Następnie omówione zostaną podstawowe komponenty niemal każdego modelu ruchu drogowego. Trzecia i czwarta sekcja poświęcona jest dostępnemu na rynku oprogramowaniu, które służy do symulacji. Na końcu znajduje się krótkie omówienie tego, gdzie i w jaki sposób wykorzystywane są symulacje ruchu drogowego w Polsce i Poznaniu.

2.1. Elementy teorii systemów

Teoria systemów, mająca swe początki w latach 30-tych XX wieku, wielokrotnie udowodniła swą przydatność w badaniu złożonych zjawisk. W jej ujęciu system opisywany jest jako całość składająca się z połączonych ze sobą komponentów o powiązanej funkcjonalności. Takie holistyczne ujęcie znacznie różni się od wprowadzonego w XVI wieku ujęcia redukcjonistycznego, zakładającego badanie całości poprzez rozkład na części składowe, które po pewnych uproszczeniach poddawane były ana-

lizie. W teorii systemów system jest czymś więcej niż tylko sumą swoich składowych. Kluczową rolę pełnią połączenia i interakcje pomiędzy jego składowymi[Cem05].

Badanie zachowania systemu wymaga stworzenia jego modelu. Model służy przewidywaniu skutków działania systemu w ustalonych warunkach, bez wykorzystywania rzeczywistego systemu. Wyróżniamy dwa rodzaje modeli:

- modele fizyczne,
- modele abstrakcyjne,
  - modele jakościowe (opisowe),
  - modele ilościowe (kwantytatywne).

Modelem fizycznym jest przykładowo makieta w skali 1:10. Stosowanie takich modeli jest utrudnione, a nawet niemożliwe w przypadku dużych i złożonych systemów. Modele abstrakcyjne dzielimy na jakościowe, czyli takie, które starają się wyjaśnić, jakie są składowe systemu i jakie są ich zależności oraz modele ilościowe, które umożliwiają przewidywanie rezultatów działania systemu[Cem05].

Gdy już dysponujemy modelem, kolejnym etapem badania systemu jest symulacja, czyli manipulowanie modelem w taki sposób, że działa on w zmienionej skali w czasie i/lub w przestrzeni, umożliwiając nam uchwycenie oddziaływań i zachowań, które w innym przypadku byłyby nieuchwytne z tytułu ich oddalenia w czasie i przestrzeni[Cem05]. Obecnie najczęściej symulację przeprowadza się z wykorzystaniem techniki komputerowej. Wymaga to, aby modele opisujące system były wyrażone w sposób możliwie ścisły i zrozumiały dla komputera[Bar10].

Należy zwrócić uwagę, że dla danego systemu nie istnieje jeden uniwersalny model. W trakcie budowy dowolnego modelu, przyjmowane są pewne założenia, często podyktowane celem badań. Dlatego też model nie jest niezależny od problemu, który ma być za jego pomocą analizowany. Modele ruchu drogowego są doskonałym przykładem tego zróżnicowania[Bar10].

2.2. Modele ruchu drogowego

Do stworzenia modelu ruchu drogowego konieczne jest zrozumienie jego przyczyny oraz tego w jaki sposób się on odbywa. Ruch drogowy należy rozpatrywać jako zjawisko społeczne i ekonomiczne, będące konsekwencją aktywności ludzi zlokalizowane w różnych miejscach i czasie. Aktywności te tworzą potrzebę przemieszczania osób i rzeczy, ta z kolei generuje podróże.

Model ruchu drogowego składa się wielu komponentów. Podstawowe to definicja popytu transportowego, infrastruktury drogowej oraz opis ich wzajemnego oddziaływania.

2.2.1. Popyt transportowy

Modelując ruch drogowy od strony użytkownika, głównym celem jest zrozumienie, jak podejmowane są decyzję dotyczące wyboru trasy i godziny przejazdu oraz środka transportu (samochód osobowy, komunikacja miejska, pociąg itp.). Każdy z użytkowników ma swoje wymagania i preferencje, które również muszą zostać uwzględnione. Miejsca rozpoczęcia i zakończenia podróży są określone przez rozlokowanie punktów zainteresowania, które jest konsekwencją polityki zagospodarowania przestrzennego miasta. Łącznie wszystkie wspomniane do tej pory czynniki są podstawą do określenia tak zwanego popytu transportowego - jednego z głównych składowych każdego modelu ruchu drogowego[Bar10].

W powszechnym użyciu są trzy metody opisu popytu transportowego. Wybór odpowiedniej metody uwarunkowany jest posiadanymi danymi. Popyt może być określony na podstawie danych z detektorów ruchu umieszczonych na skrzyżowaniach. Podsiadamy wtedy informację o ilości pojazdów, które przejechały przez dane skrzyżowanie w pewnym okresie czasu. Tak zdefiniowany popyt jest w dużej mierze losowy i niedokładny. W celu poprawienia tej sytuacji dodaje się do modelu również informację o prawdopodobieństwie skrętu dla głównych skrzyżowań. Najczęściej tak zdefiniowany popyt wykorzystuje się w celu weryfikacji innych modeli[Ger12].

Jako druga zostanie omówiona metoda zadaniowa oparta o statystyki populacji. Popyt transportowy określa się na podstawie rozmieszczenia przestrzennego punktów zainteresowań (dom, szkoła, praca, rekreacja, sklep, itp.), oraz ilości użytkowników danego obiektu. W ten sposób uwzględniając przeciętne zachowania ludności można przewidzieć podróże, jakie będą miały miejsce w sieci transportowej oraz ich przybliżony czas. Metoda ta jest znacznie bardziej skuteczna od poprzedniej, jednak jej skuteczność w dużym stopniu zależy od przyjętego algorytmu przydzielania tras podróży oraz punktów zainteresowań [Ger12].

Trzecią i zarazem najpowszechniej stosowaną jest metoda macierzy OD (ang. origin-destination - źródło-cel). Polega ona na podzieleniu modelowanego obszaru na rejony transportowe, a następnie określeniu dla każdej uporządkowanej pary rejonów, ilości pojazdów danego typu, które przejadą z pierwszego do drugiego w ustalonym okresie czasu. Dla każdego rejonu określa się możliwe miejsca rozpoczęcia i zakończenia podróży, co pozwala na wygenerowanie tras przejazdu pomiędzy rejonami. Popyt przedstawiany jest jako macierz kwadratowa $T^{\tau p}$

$$T^{\tau p} = (t_{i,j}^{\tau p}),$$

gdzie:

τ – przedział czasu,

p – środek transportu,

$t_{i,j}^{\tau p}$ – ilość pojazdów typu p, które przejadą z rejonu i do rejonu j w czasie τ.

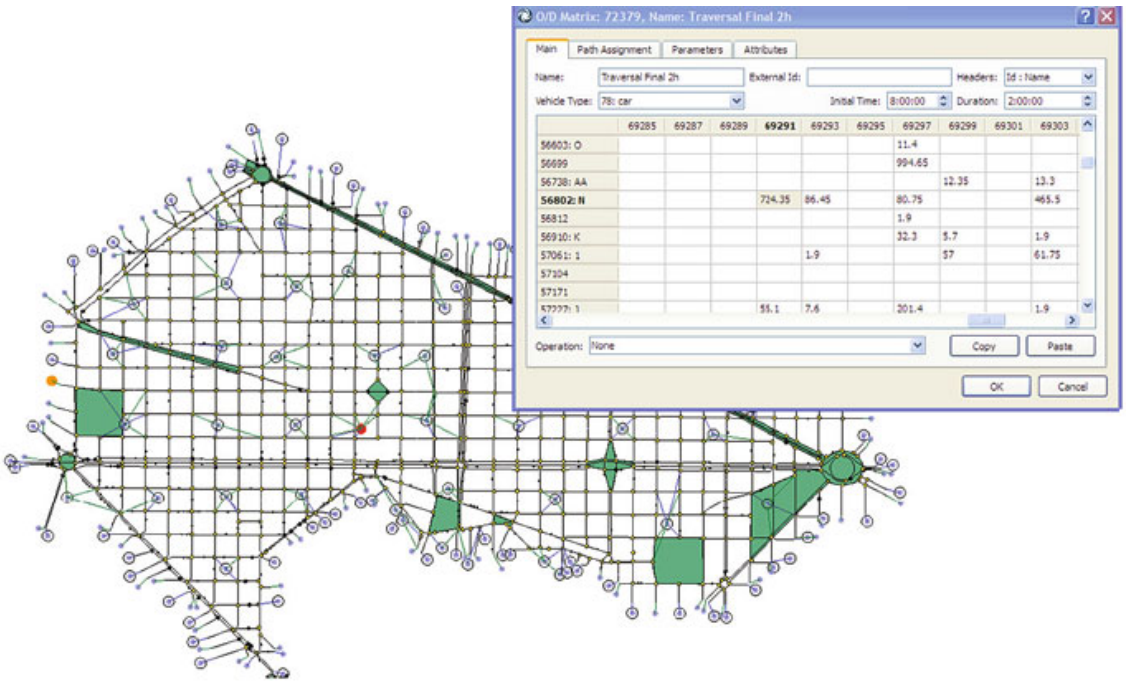
Najczęściej dokonuje się podziału na trzy podstawowe środki transportu: pojazdy osobowe, dostawcze i ciężarowe. Najbardziej ogólny zarys ruchu w mieście dostarcza macierz dobowa ($\tau = 24$ godziny). Często stosuje się również macierze reprezentujące godziny szczytu [kbr].

Zastosowanie tej metody najczęściej wymaga przeprowadzenia kompleksowych badań ruchu (KBR), które są bardzo czasochłonne i kosztowne. Jednak fakt, że jest ona stosowana w wielu miastach na świecie i w Polsce, świadczy o tym, że popyt otrzymany tą metodą musi być wysokiej jakości.

2.2.2. Infrastruktura drogowa

Infrastruktura drogowa to ten komponent systemu, który pozwala zaspokoić popyt transportowy. Większość modeli sieci drogowych składa się z węzłów i krawędzi je łączących. Cechą odróżniającą je od siebie jest ich stopień szczegółowości. Najbardziej ogólne modele składają się tylko z węzłów i krawędzi, z którymi powiązane są informacje o przepustowości i dozwolonej szybkości. Modele te idealnie nadają się do modelowania dużych obszarów, sieci dróg krajowych i międzynarodowych, czyli wszędzie tam, gdzie konieczne jest modelowanie bardzo dużej ilości pojazdów, a udział ruchu miejskiego jest minimalny.

W celu symulacji ruchu miejskiego konieczne jest wzbogacenie modelu o więcej informacji takich jak ilość pasów, dopuszczalne typy pojazdów, zależności pomiędzy



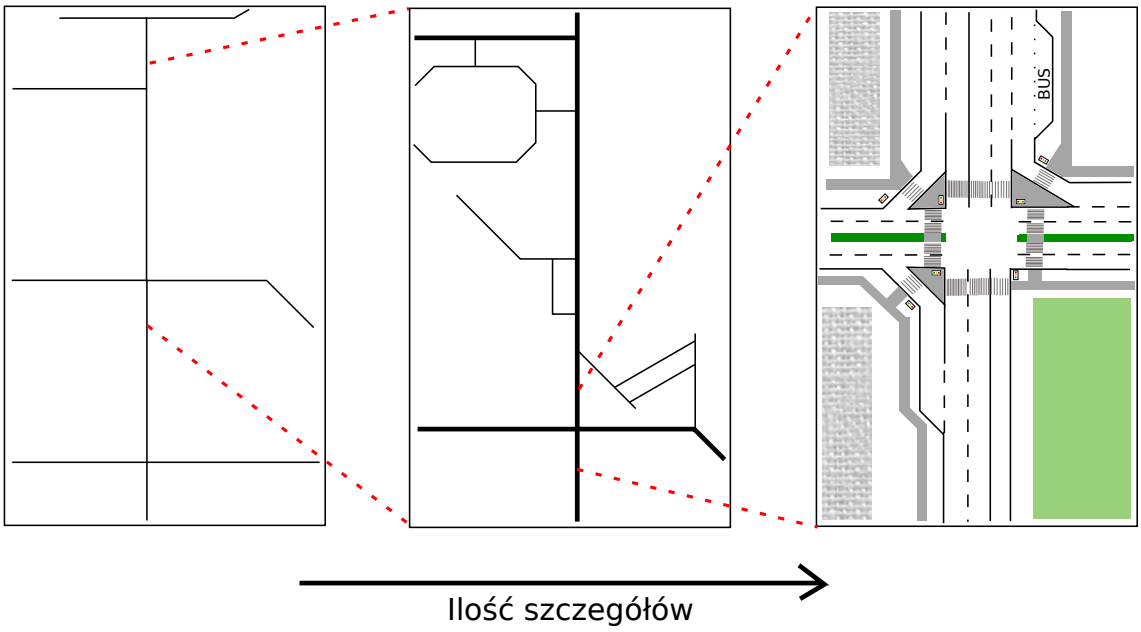
Rys. 2.2.1. Przykładowa macierz OD. Źródło [Bar10].

natężeniem, a czasem przejazdu, czy natężeniem, a prędkością. Jednak te informacje nadal nie umożliwiają dokładnego modelowania zachowania pojazdów na skrzyżowaniach. Dlatego często wprowadza się do modelu nowe obiekty opisujące wewnętrzną strukturę skrzyżowań - pasy skrętu, organizację ruchu oraz sygnalizacje świetlne, połączenia pomiędzy pasami, ich szerokość czy nawet przejścia dla pieszych[Bar10]. Na rysunku 2.2.2 przedstawiono modele infrastruktury drogowej o różnym stopniu szczegółowości.

2.2.3. Wybór trasy

Choć model popytu oraz infrastruktury stanowią dwa główne komponenty systemu transportowego, to zgodnie z definicją jest to za mało. Konieczne jest jeszcze określenie zależności pomiędzy nimi, czyli tego, w jaki sposób popyt jest zaspokajany przez sieć drogową. Kluczowym problemem jest określenie sposobu, w jaki użytkownicy wybierają trasę przejazdu[SŻ12].

W większości modeli zakłada się, że proces ten odbywa się w stanie równowagi użytkownika. Pierwszy zasadę równowagi sformułował angielski badacz transportu, John Wardrop w 1952 roku. Mówi ona, że *czas przejazdu dowolną spośród wybranych tras, jest mniejszy lub równy od czasu przejazdu dowolną trasą niewybraną* [War52]. Podstawowym jej ograniczeniem jest brak uwzględnienia zmienności warunków drogowych w czasie. Dlatego też w celu dokładniejszego modelowania konieczne jest jej rozszerzenie: *Sieć jest w stanie dynamicznej równowagi użytkownika (DUE) jeśli, dla każdej pary źródło-cel, czasy przejazdu użytkowników wyjeżdżających w tej*



Rys. 2.2.2. Modele infrastruktury drogowej o różnym stopniu szczegółowości. Opracowanie własne.

samej chwili są równe i minimalne [Bar10]. Istnieje wiele algorytmów wyznaczania DUE, od czysto analitycznych, po heurystyczne. Podejście analityczne polega na rozwiązaniu poniższego problemu optymalizacyjnego[Bar10]:

$$\begin{aligned}
(\tau_{rsp}(t) - \theta_{rs}(t)) f_{rsp}(t) = 0 \quad & \forall_{p \in P_{rs}(t)}, \forall_{(r,s)\in\mathscr{I}}, t \in [0,T]\,, \\
\tau_{rsp}(t) - \theta_{rs}(t) \geq 0 \quad & \forall_{p \in P_{rs}(t)}, \forall_{(r,s)\in\mathscr{I}}, t \in [0,T]\,, \\
\sum_{p \in P_{rs}(t)} f_{rsp}(t) = d_{rs}(t) \quad & \forall_{(r,s)\in\mathscr{I}}, t \in [0,T]\,, \\
\tau_{rsp}(t), \theta_{rs}(t), f_{rsp}(t) \geq 0\,, &
\end{aligned}$$

gdzie:

$f_{rsp}(t)$ – ilość przejazdów trasą p z r do s, rozpoczynająca się w przedziale czasu t,

$\tau_{rsp}(t)$ – aktualny koszt przejazdu z r do s trasą p w przedziale czasu t,

$\theta_{rs}(t)$ – koszt najkrótszej trasy z r do s w przedziale czasu t,

$P_{rs}(t)$ – zbiór wszystkich możliwych tras z r do s w przedziale czasu t,

$\mathscr{I}$ – zbiór wszystkich par źródło-cel sieci drogowej,

$d_{rs}(t)$ – liczba podróży z r do s w przedziale czasu t,

T – czas zakończenia symulacji.

Metody heurystyczne najczęściej opierają się na iteracyjnym powtarzaniu procesu przydzielania tras oraz wyliczania kosztów przejazdu. Taki proces nie gwarantuje osiągnięcia DUE, lecz często wyznaczane jest jego dobre przybliżenie.

Spośród metod niekorzystających z zasady równowagi warto zwrócić uwagę na rozwiązania oparte na teorii decyzji. W najprostszym ujęciu zakładamy, że zbiór P_{rs} wszystkich tras przejazdu z r do s jest skończony. Z każdą trasą skojarzona jest jej użyteczność. Rozwiązanie problemu przydziału tras przejazdu otrzymujemy jako rozwiązanie problemu decyzyjnego, które maksymalizuje sumę wartości użyteczności wszystkich tras wybranych przez użytkowników[Bar10].

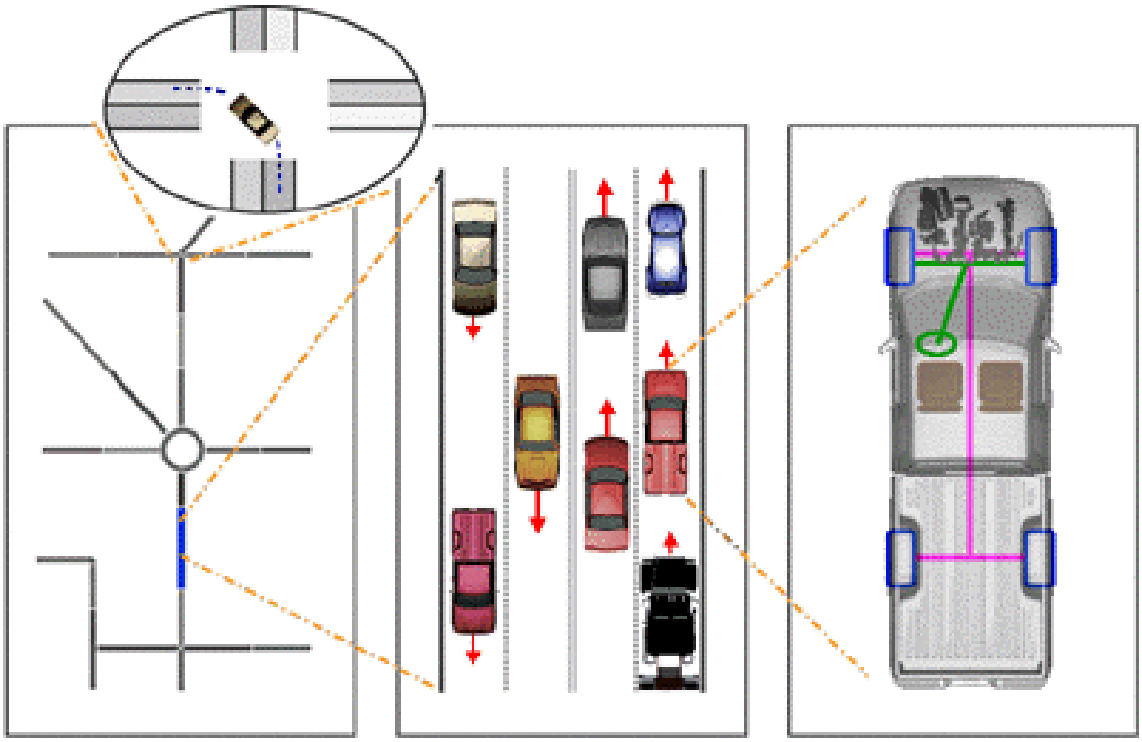
2.2.4. Metody modelowania potoków ruchu

Kolejnym bardzo ważnym aspektem każdego modelu ruchu drogowego jest określenie, w jaki sposób rozpatrywany będzie proces przemieszczania się pojazdów w sieci drogowej. Dwa skrajne rozwiązania, mikroskopowe i makroskopowe, modelują ten problem w zupełnie odmienny sposób. Modele makroskopowe operują na zagregowanych danych, w przeciwieństwie do modeli mikroskopowych, operujących na pojedynczych pojazdach.

Najpierw jednak konieczne jest zdefiniowanie trzech podstawowych stanów w jakich może znajdować się pojazd podczas jazdy[Kra98]. Pierwszy z nich, mający zarazem największy wpływ na zużycie paliwa i emisję, to stan zakorkowania, w którym to gęstość jest duża, a prędkość jazdy bardzo mała. Często występuje konieczność chwilowego zatrzymania się. Drugi to stan jazdy zsynchronizowanej, podczas której pojazdy poruszają się szybciej niż w korku. Na zachowanie kierowców główny wpływ mają jednak sąsiednie pojazdy. Trzeci stan to swobodna jazda, podczas której interakcje pomiędzy pojazdami są śladowe. Prędkość jazdy jest największa, a gęstość minimalna.

2.2.4.1. Podejście makroskopowe

Modele makroskopowe najczęściej korzystają z wyników teorii ciągłości potoków ruchu. Zajmuje się ona badaniem zmienności w miejscu i czasie oraz wzajemnych zależności pomiędzy podstawowymi parametrami potoków makroskopowych, do których zaliczamy przede wszystkim objętość q (ilość), prędkość v, oraz gęstość k. Wartości te określone są dla każdego punktu x, oraz chwili symulacji t. W modelach tych nie występuje pojęcie pojedynczego pojazdu, a opis ruchu drogowego odbywa się tylko za pomocą wspomnianych parametrów.



Rys. 2.2.3. Różne poziomy szczegółowości modelowania ruchu pojazdów. Od lewej do prawej: makroskopowy, mikroskopowy, submikroskopowy. W kółku zaznaczono poziom mezoskopowy. Źródło [Ger12].

Podstawowym równaniem tej teorii jest równianie zachowania ciągłości

$$\frac{\partial q}{\partial x} + \frac{\partial k}{\partial t} = g(x,t)\,,$$

gdzie $g(x,t)$, to liczba pojazdów dodawanych lub usuwanych sieci drogowej w chwili t i miejscu x. Mówi ono, że pomiędzy dwoma sąsiadującymi punktami symulacji liczba pojazdów może zmienić się tylko poprzez wjechanie lub wyjechanie pojazdu z sieci, zgodnie z wartością $g(x,t)$. Jeśli wartość $g(x,t) = 0$ to liczba pojazdów pomiędzy sąsiednimi punktami nie zmienia się. Punkty, w których $g(x,t) > 0$ stanowią źródła - punkty wejściowe, w których pojazdy są dodawane do symulacji. Jeśli $g(x,t) < 0$, to w danym punkcie pojazdy opuszczają sieć.

W celu rozwiązania tego równania konieczne jest przyjęcie pewnych dodatkowych założeń. Najczęściej przyjmowanym ograniczeniem jest uzależnienie wartości prędkości od gęstości w danym punkcie i czasie w poniższy sposób

$$u(x,t) = u_{max}(x)\left[1 - \left(\frac{k}{k_{max}(x)}\right)^{\alpha}\right]^{\beta}\,,$$

gdzie $u_{max}(x)$ jest maksymalną prędkością przejazdu (osiąganą podczas jazdy swobodnej), a k_{max} maksymalną gęstością (osiąganą podczas jazdy w stanie korku) w punkcie x.

Rozwiązania tych równań otrzymuje się metodami numerycznymi. Dzięki dużemu podobieństwu do równań dynamiki płynów, istnieje wiele dobrze zbadanych metod, które dają się zastosować do rozwiązywania problemów ruchu drogowego[Bar10]. Dlatego też wydajność metod makroskopowych jest bardzo duża, dzięki czemu doskonale nadają się do symulowania bardzo dużych sieci drogowych.

2.2.4.2. Podejście mikroskopowe

W podejściu mikroskopowym ruch każdego pojazdu modelowany jest osobno. Reakcje kierowców na otoczenie i sąsiednie pojazdy, takie jak przyspieszenie, hamowanie, czy zmiana pasa ruchu stanowią kluczowy element modelu. Pierwsze teorie opisujące sposób w jaki pojazdy podążają drogą powstały w pierwszej połowie lat 50-tych dwudziestego wieku. Skupiały się one na wyznaczeniu minimalnej bezpieczniej odległości pomiędzy dwoma jadącymi za sobą pojazdami. Badania teoretyczne, jak i doświadczalne pozwoliły osiągnąć zadowalający ówcześnie model, w którym odległość rosła liniowo wraz ze wzrostem prędkości.

Intensywne badania prowadzone przez firmę General Motors Group doprowadziły do powstania fundamentów obecnie wykorzystywanego modelu matematycznego. Wprowadzono pojęcie *reakcji*, *wrażliwości* i *bodźca*, które powiązane są ze sobą następującą zależnością:

$$\text{Reakcja}(t+T) = \text{Wrażliwość}(t) \times \text{Bodziec}(t)\,.$$

Reakcja w czasie $t + T$ zależy od wartości bodźca w czasie t oraz wrażliwości kierowcy. Parametr T oznacza opóźnienie reakcji. Kluczowym odkryciem okazała się obserwacja, że kierowca pojazdu stara się dostosować swoją prędkość do pojazdu go poprzedzającego, nazywanego *liderem*.

Najprostsze modele jako *bodziec* przyjmowały zmianę prędkości pomiędzy pojazdami, a jako *reakcję* przyspieszenie lub zwolnienie. Niech $x_n(t)$ oraz $x_{n+1}(t)$ oznaczają odpowiednio pozycję na drodze w chwili t pojazdu *lidera* oraz pojazdu podążającego za nim. Wtedy model ten można zapisać przy pomocy następującego wzoru:

$$x''_{n+1}(t+T) = \lambda \left[x'_n(t) - x'_{n+1}(t)\right] .$$

Oczywiście zapis $x'_n(t)$ oznacza prędkość, a $x''_n(t)$ przyspieszenie pojazdu n. Przyjmuje się, że λ jest stałą. Model ten nosi nazwę modelu liniowego, ze względu na to, że reakcja jest liniową funkcją bodźca. Dalsze badania wykazały, iż jest on niezgodny z danymi obserwowanymi w rzeczywistości, szczególnie podczas ruchu po zatłoczonych drogach.

W roku 1960 Edie zaproponował modyfikację modelu liniowego, uzależniając wrażliwość od odległości pomiędzy pojazdami

$$x''_{n+1}(t+T) = \lambda \frac{x'_{n+1}(t)}{(x_n(t) - x_{n+1}(t))^2} \left[x'_n(t) - x'_{n+1}(t)\right] .$$

Model ten został w 1961 roku uogólniony przez General Motors Group

$$x''_{n+1}(t+T) = \lambda \frac{(x'_{n+1}(t))^k}{(x_n(t) - x_{n+1}(t))^l} \left[x'_n(t) - x'_{n+1}(t)\right] .$$

W roku 1967 May i Keller na podstawie danych empirycznych określili, że najlepsze wyniki uzyskiwane są przy wartościach $k = 0,8$ oraz $l = 2,8$[Bar10].

Należy zwrócić uwagę, że modele oparte na zależności bodziec-reakcja nie są jedynym możliwym podejściem do problemu. W 1981 roku Gipps wprowadził model behawioralny, uwzględniający zachowanie i preferencje kierowcy[Bar10]. Dzięki prostocie i wydajności dużą popularnością, głównie w symulacjach dużej skali, cieszą się również modele oparte na automatach komórkowych[Kra98].

Dotychczas zaprezentowane modele dotyczyły pojazdów poruszających się po jednym pasie ruchu. Problem zmiany pasa ruchu, jest w znacznie mniejszym stopniu zbadany. Początkowe prace o tej tematyce, których autorem jest Sparmann dotyczyły analizy zachowania kierowców na drogach o dwóch pasach. Leutzbach i Busch przeprowadzili analogiczne badania dla dróg o trzech pasach[Kra98]. Brak jest jednak powszechnie przyjętych modeli. W badaniach często używa się autorskich modeli powstałych w czasie implementacji[BBEK11].

2.2.4.3. Podejście mezoskopowe

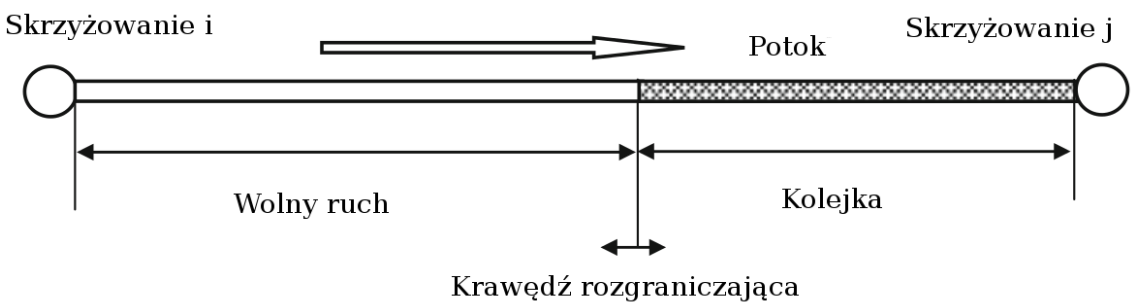
Rozwiązania mezoskopowe łączą w sobie elementy podejścia mikroskopowego oraz makroskopowego. Do modeli mikroskopowych podobne są pod względem tego, że rozpatrują ruch pojedynczych pojazdów lub ich grup. Z makroskopowymi łączy je to, że w pewnych warunkach rozpatrują ruch jako potok pojazdów. Choć istnieje wiele typów modeli mezoskopowych, większość dokonuje rozdzielenia każdej drogi na dwie części: kolejkę i cześć wolnego ruchu (patrz rysunek 2.2.4). Część wolnego ruchu symulowana jest, z wykorzystaniem uproszczonych modeli podążania pojazdów, analogicznie jak w modelach mikroskopowych. Kolejka natomiast modelowana jest jako potok, podobnie jak w modelach makroskopowych.

2.3. Komputerowa symulacja ruchu drogowego

Ze względu na duży nakład obliczeniowy, za wyjątkiem najprostszych systemów drogowych, zaprezentowane wcześniej modele nadają się jedynie do symulacji komputerowej. Na rynku istnieje wiele pakietów oprogramowania do symulacji ruchu drogowego, z czego większość z nich to rozwiązania komercyjne, istnieją jednak również rozwiązania darmowe. Jeśli przyjrzeć się zastosowanych w nich modelach, okazuje się, że przynależą one do wszystkich zaprezentowanych wcześniej typów i kategorii. Nie ma jednoznacznie dominującego rozwiązania. Należy jednak zauważyć, że dzięki postępowi technologicznemu i wzrostowi wydajności komputerów obserwuje się trend w kierunku symulacji mikroskopowych.

Dzięki takiej różnorodności możliwe jest dobranie symulatora, który najlepiej będzie spełniał wymagania badanego problemu. Poniższe zestawienie przedstawia listę najważniejszych i najczęściej wykorzystywanych symulatorów[KH09, Bar10].

PTV VISSIM symulator mikroskopowy, oparty o model behawioralny. Pozwala na modelowanie zarówno transportu prywatnego jak i publicznego. Symulator pozwala na trójwymiarową wizualizację w czasie rzeczywistym.



Rys. 2.2.4. Schemat drogi w modelu mezoskopowym. Źródło [Bar10].

Symulator	Producent	Licencja	System operacyjny	Wizualizacja	Typ symulacji
VISSIM	PTV	płatna	Windows	trójwymiarowa	mikroskopowy
VISUM	PTV	płatna	Windows	dwuwymiarowa	makroskopowy
Paramics	SIAS	płatna	Windows	trójwymiarowa	mikrospokowy
SUMO	DLR	darmowa	Windows, Linux, Mac	dwuwymiarowa	mikrospokowy
Aimsun	UPC	płatna	Windows, Linux, Mac	trójwymiarowa	mikroskopowy, makroskopowy, mezoskopowy
MITSIMLab	MIT	darmowa	Linux	dwuwymiarowa	mikroskopowy
DRACULA	University of Leeds	płatna	Windows	dwuwymiarowa	mikroskopowy
Dynameq	Inro	płatna	Windows	dwuwymiarowa	mikroskopowy
DynaMIT	MIT	płatna	Linux	dwuwymiarowa	mikroskopowy

Tab. 2.3.1. Zestawienie najpopularniejszych symulatorów ruchu. Opracowanie własne na podstawie [KH09, Bar10].

PTV VISUM symulator makroskopowy. Jest kompleksowym narzędziem do modelowania i planowania popytu transportowego w skali od miejskiej po ogólnokrajową. Posiada zaawansowane narzędzia do budowania sieci i określania popytu w postaci macierzy OD.

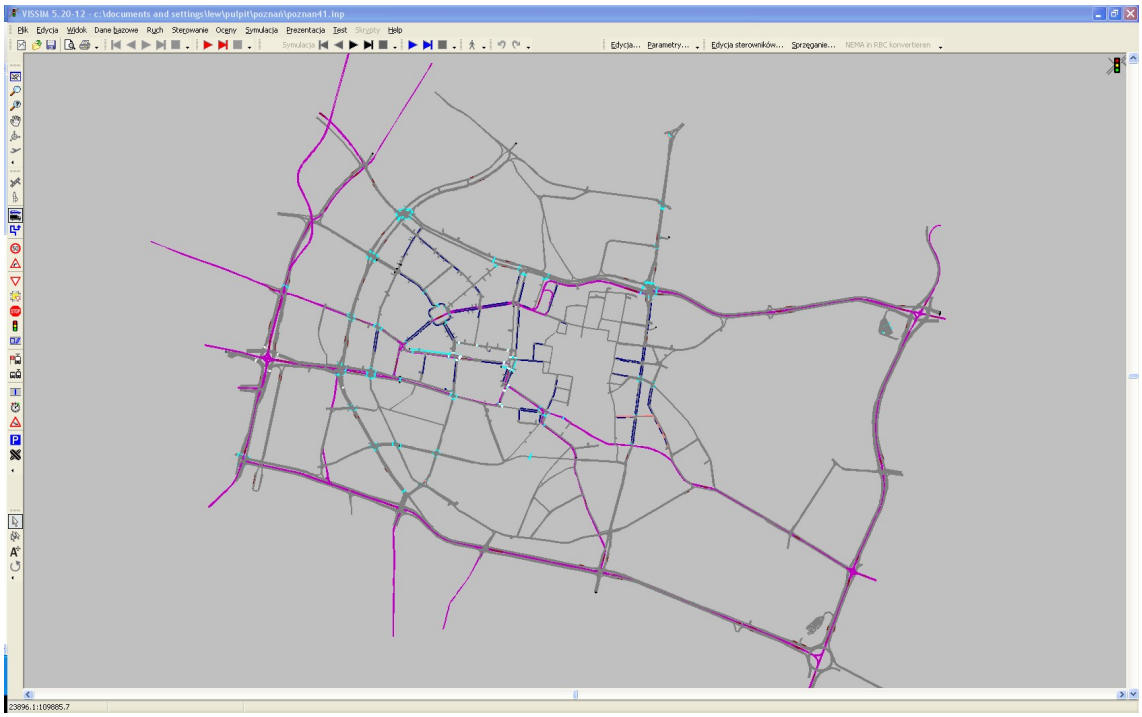
SIAS Paramics symulator mikroskopowy. Rozwijany od 1986 roku, umożliwia przeprowadzanie bardzo dokładnych symulacji (sygnalizacja świetlna, piesi) zarówno małej jak i dużej skali. Posiada narzędzia umożliwiające analizowanie wyników całej symulacji jak i pojedynczych skrzyżowań lub dróg.

SUMO otwarty symulator mikroskopowy rozwijany od 2000 roku. Znalazł zastosowanie w badaniach ruchu oraz inteligentnych systemów świateł drogowych. Oferuje bardzo potężny interfejs konsolowy oraz prostego klienta graficznego.

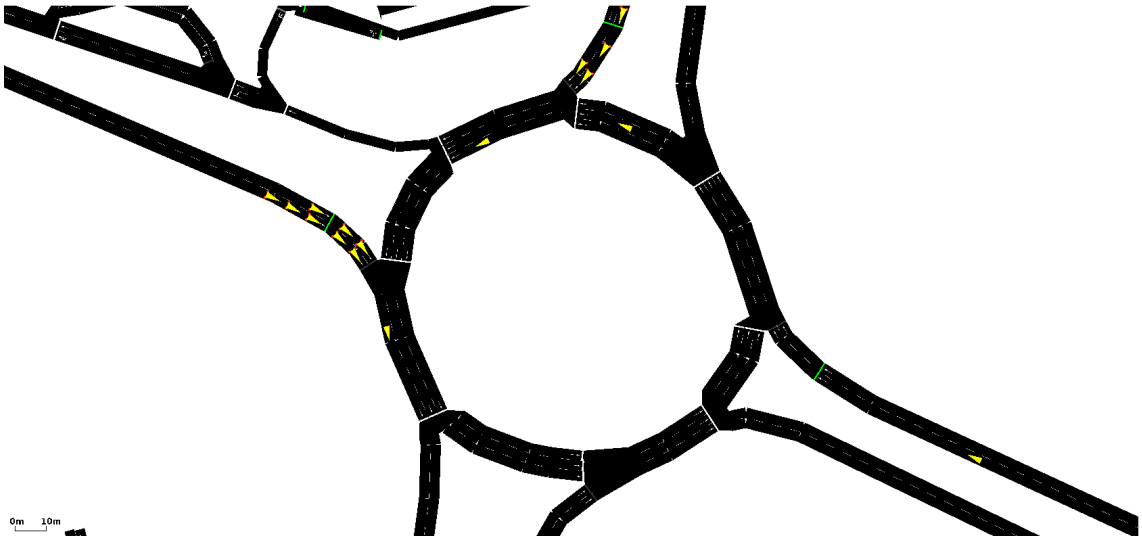
Aimsun bogaty pakiet narzędzi służących do modelowania ruchu drogowego. Jeden z niewielu symulatorów integrujący zarówno podejścia mikro, jak i makroskopowe.

MITSIMLab symulator, który przeznaczony jest do modelowania wpływu infra-

Rys. 2.3.1. Symulator VISSIM. Źródło materiały promocyjne firmy PTV.



Rys. 2.3.2. Symulator VISUM. Źródło materiały promocyjne firmy PTV.



Rys. 2.3.3. Symulator SUMO. Źródło [Ger12].

.

struktury drogowej, świateł drogowych i transportu publicznego.

DRACULA symulator mikroskopowy przeznaczony głównie do badania zachowania pojedynczych pojazdów.

Dynameq symulator mikroskopowy stworzony z myślą o modelowaniu zjawiska dynamicznej równowagi użytkowników.

DynaMIT symulator stworzony z myślą o przewidywaniu warunków drogowych. Jego główne zastosowania to modelowanie i obsługa inteligentnych systemów drogowych i informacyjnych.

2.4. Symulator SUMO

SUMO - Simulation of Urban MObility jest pakietem oprogramowania udostępnianego na licencji General Public License (GPL). Rozwój tego oprogramowania rozpoczął się w 2000 roku. Od roku 2001 rozwijany jest przez German Aerospace Center (DLR). W jego skład wchodzą wszystkie narzędzia niezbędne do przygotowania i przeprowadzenia symulacji. Mimo, że formaty danych wykorzystywane przez SUMO nie są powszechnie stosowane, udostępnia on narzędzia umożliwiające import map (np: Open Street Maps, VISSIM, VISUM), popytu (macierze OD) oraz generowanie symulacji na podstawie bardziej ogólnych informacji takich, jak dane z detektorów ruchu, czy opisu populacji[BBEK11].

SUMO jest czystym symulatorem mikroskopowym, co oznacza, że każdy pojazd modelowany jest osobno. Konieczne jest zatem określenie zarówno czasu wyjazdu, jak i dokładnej trasy przejazdu. Ponadto z każdym pojazdem związany jest szereg

atrybutów takich, jak typ: pojazdu, jego wymiary, charakterystyka przyspieszeń, zużycia paliwa czy emisji spalin. Do symulowania ruchu, wykorzystano model autorstwa Stefana Kraußa, w którym dyskretny czas rozpatrywany jest z dokładnością $\Delta t = 1\,\text{sekunda}$[Kra98]:

$$\begin{aligned} v_{bezp}(t) &= v_1(t) + \frac{g(t) - g_{poż}(t)}{\tau_b + \tau}\,, \\ v_{poż}(t) &= \min(v_{max}, v(t) + a(v)\Delta t, v_{bezp}(t))\,, \\ v(t + \Delta t) &= \max(0, v_{poż}(t) - \eta)\,, \\ x(t + \Delta t) &= x(t) + v(t)\Delta t\,, \end{aligned}$$

gdzie:

$v(t)$ – prędkość pojazdu w chwili t,

$x(t)$ – położenie pojazdu na drodze w chwili t,

$g(t)$ – odległość pojazdu od lidera w chwili t,

$g_{poż}(t)$ – pożądana odległość od lidera w chwili t,

$v_{poż}(t)$ – pożądana prędkość z jaką kierowca chciałby jechać w chwili t,

v_{max} – maksymalna prędkość z jaką pojazd może się poruszać,

$v_1(t)$ – prędkość pojazdu lidera w chwili t,

τ – czas reakcji kierowcy,

τ_b – czas konieczny na dostosowanie prędkości pojazdu podążającego do prędkości lidera,

η – czynnik losowy.

Do modelowania zużycia paliwa i emisji spalin wykorzystano powszechnie uznaną w Europie (w szczególności w Niemczech) mapę emisji HBEFA (The Handbook Emission Factors for Road Transport). Stworzenie tego modelu zostało zainicjowane przez organizacje ekologiczne Niemiec, Szwajcarii i Austrii. Później dołączyły do nich Szwecja, Norwegia i Francja. Obecnie w najnowszej wersji model został gruntownie przebudowany i zaktualizowany[Hał10].

Model wykorzystywany w SUMO powstał poprzez interpolację mapy HBEFA. Symulowane jest zużycie paliwa oraz emisja CO, CO_2, węglowodorów, związków azotu oraz substancji małocząsteczkowych[Ger12].

Oprócz wizualizacji, SUMO potrafi przygotować różne zestawienia wyników symulacji. Do podstawowych zestawień należą: pozycje wszystkich symulowanych pojazdów w czasie symulacji, dane z symulowanych detektorów ruchu, uśrednione dane

dla poszczególnych dróg lub pasów ruchu, informacje o wybieranych trasach oraz światłach drogowych. Możliwe jest określenie częstości próbkowania, jak i okresu agregacji informacji wynikowych.

Ogólny schemat przygotowania symulacji z wykorzystaniem SUMO jest następujący [Ger12].

1. **Przygotowanie sieci drogowej**. Dostępne są narzędzia: netgen służący do generowania losowych sieci, oraz netconvert służący do importowania map zapisanych w innych formatach. Możliwe jest również ręczne przygotowanie opisu sieci drogowej.

2. **Przygotowanie definicji podróży**. Możliwe są dwa podejścia: importowanie macierzy OD (program od2trips) oraz generowanie podróży w oparciu o aktywności (activitygen).

3. **Obliczenie tras przejazdu**. SUMO obsługuje wiele metod wyznaczenia tras. Najczęściej stosowana metoda, oparta o algorytm najkrótszych ścieżek Dijkstry, daje najlepsze wyniki w połączeniu z iteracyjnym algorytmem wyznaczania dynamicznej równowagi użytkowników. Inne dostępne metody to wyznaczanie tras przejazdu na podstawie prawdopodobieństw skrętu na skrzyżowaniach lub ilości pojazdów mijających dany punkt w sieci.

4. **Symulacja**. W oparciu o przygotowaną wcześniej sieć drogową oraz trasy przejazdu, można wykonać symulację. Dostępny jest tryb tekstowy (sumo) lub graficzny (sumo-gui). W trybie graficznym prezentowana jest dwuwymiarowa animacja ruchu w sieci, z możliwością analizowania na bieżąco większości parametrów symulacji.

5. **Analiza wyników**. Rezultatem symulacji są pliki z wynikami, których format opisany jest w dokumentacji pakietu.

Należy zwrócić uwagę na to, że symulator SUMO znacząco różni się od innego dostępnego na rynku oprogramowania tego typu. Większość czynności wykonywanych jest z wykorzystaniem linii komend systemu operacyjnego, a konfiguracja odbywa się przez edycję odpowiednich plików konfiguracyjnych. Wszystkie dane zarówno wejściowe, jak i wyjściowe przechowywane są w plikach xml. Takie podejście zostało spowodowane wieloma czynnikami, a jednym z głównych jest grupa użytkowników, dla której ten symulator jest przeznaczony. Są to głównie osoby z doświadczeniem informatycznym, badacze, doktoranci i studenci. Symulator SUMO natomiast rzadko jest stosowany przez firmy lub instytucje publiczne do codziennej pracy[Bar10].

2.5. Symulacja w Polsce i Poznaniu

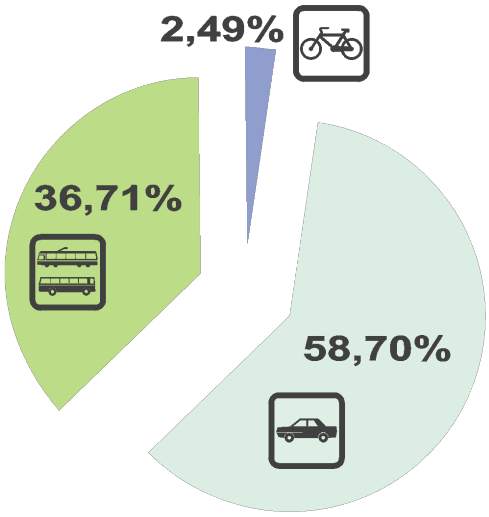
Choć wydawać by się mogło, że w Polsce symulacje ruchu nie są powszechne, to jak wskazują zalecenia Ministerstwa Rozwoju Regionalnego [mrr08], wszystkie inwestycje drogowe wymagają przeprowadzenia symulacji ich wpływu na otoczenie. Dlatego też Generalna Dyrekcja Dróg i Autostrad publikuje szczegółowe wyniki Generalnego Pomiaru Ruchu. Ostatnie badanie miało miejsce w 2010 roku [Opo10]. Dane uzyskane z tych badań pozwalają na określenie globalnego (ogólnokrajowego) modelu ruchu i ewaluacji wpływu wielkich inwestycji drogowych na transport drogowy w Polsce. Ponadto oprócz pomiarów ruchu, GDDKA prowadzi badania nad oddziaływaniem inwestycji w infrastrukturę drogową na środowisko [gdd11]. W raporcie uwzględniono poziom emisji spalin, wpływ na miasta oraz lokalne ekosystemy.

W terenie miejskim istnienie jak i dokładność modelu ruchu zależą natomiast od lokalnego zarządu dróg miejskich. Zgodnie z zaleceniami Niebieskiej Księgi [mrr08], duże inwestycje w miastach powyżej 300 000 mieszkańców, należy poprzedzić symulacją całego obszaru miasta wraz z przyległymi miejscowościami.

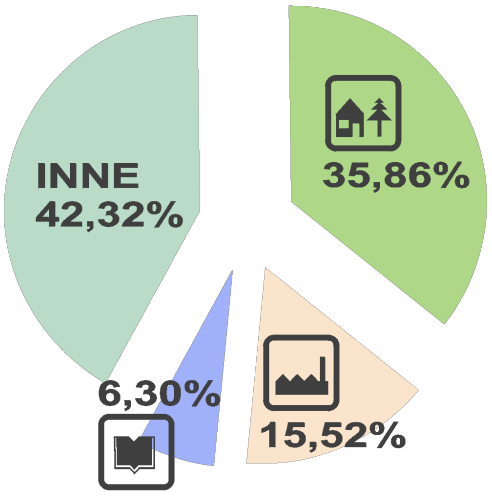
W Poznaniu Zarząd Dróg Miejskich dysponuje modelem ruchu otrzymanym w wyniku Kompleksowego Badania Ruchu (KBR) [kbr]. Model ten jest regularnie aktualizowany (rok 2000, 2006 i 2009). KBR w 2000 roku obejmowało okres dwóch tygodni. Pomiarem objęto natężenie ruchu na najważniejszych węzłach komunikacyjnych miasta. Dokonano również pomiarów natężeń ruchu na głównych drogach w powiecie. Kolejnym bardzo ważnym elementem Poznańskiego modelu ruchu, są wyniki badań ankietowych w gospodarstwach domowych (800 ankiet) oraz na drogach wylotowych z miasta (21991 ankiet, co stanowi 45,3 procent całkowitego ruchu wylotowego z powiatu Poznańskiego).

W wyniku badań otrzymano dane o strukturze ruchu w Poznaniu. Podział pomiędzy środki transportu wynosi: 56,02% – samochody, 41,42% – komunikacja zbiorowa, 2,49% – rower. Najczęstszym celem podróży w obszarze powiatu był dom (35,9%), następnie praca (15,5%), nauka (6,3%), natomiast w Poznaniu kolejno 31,95% dom, 15,71% praca i 7,43% nauka. W wyniku symulacji modelu ruchu określono również średnią długość podróży samochodowej na 10,2 km, natomiast w komunikacji zbiorowej na 12,6 km.

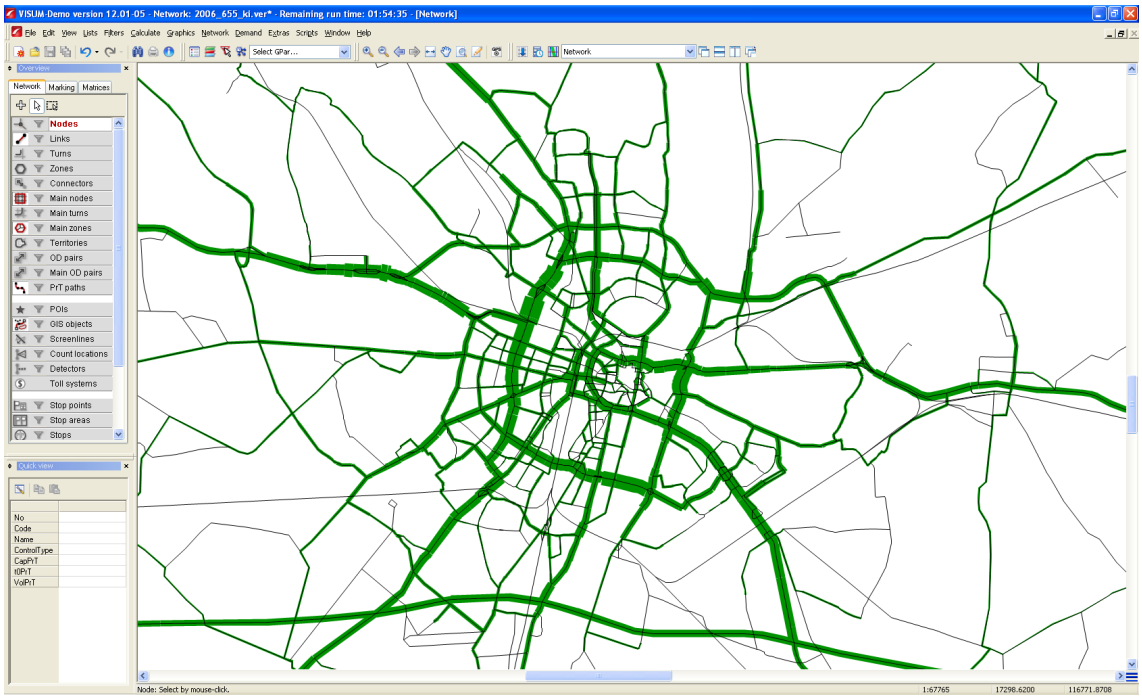
Do symulacji ruchu w Poznaniu wykorzystywany jest dobowy model ruchu z 2006 roku oraz model szczytu porannego z 2009 roku. Model ten jest zapisany w formacie macierzy OD oraz mapy z podziałem na 456 rejonów transportowych. Do symulacji wykorzystuje się oprogramowanie VISUM w wersji 8.



Rys. 2.5.1. Udział poszczególnych środków transportu w Poznaniu. Źródło [kbr].



Rys. 2.5.2. Najczęstsze cele podróży w Poznaniu. Źródło [kbr].



Rys. 2.5.3. Model ruchu miasta Poznań w programie VISUM.

Rozdział 3

Modelowanie i sterowanie rozmyte

Teoria zbiorów rozmytych, ciesząca się obecnie ogromną popularnością, została zapoczątkowana w 1965 roku przez Loftiego Zadeha w artykule pod tytułem „Fuzzy sets". Jednak dynamiczny rozwój, początkowo nie zauważanej teorii, nastąpił dopiero w drugiej połowie lat 70-tych, kiedy udało się rozwiązać problem sterowania piecem do wytwarzania cementu – z wykorzystaniem logiki rozmytej (Mamdani). Jednak kluczowe okazały się spektakularne sukcesy w Japonii. Najgłośniejszym z nich było opracowanie układu sterowania metrem w mieście Sendai. Dzięki zastosowaniu wnioskowania rozmytego, wykorzystano wiedzę doświadczonych motorniczych, która w połączeniu z możliwościami obliczeniowymi komputerów, pozwoliła między innymi zmniejszyć czasy opóźnień oraz koszty utrzymania metra.

Dalszy rozwój spowodował zastosowanie logiki rozmytej nie tylko w urządzeniach przemysłowych, lecz również codziennego użytku. Jednym z ciekawszych przypadków jest stabilizacja obrazu w kamerach. Problem ten jest bardzo trudny, chociażby ze względu na konieczność odróżnienia zamierzonego ruchu od drgań ręki operatora. Sama teoria również została znacznie rozwinięta, umożliwiając w ten sposób zastosowanie w kolejnych obszarach przemysłu i nauki.

Poniższy rozdział stanowi krótkie wprowadzenie do teorii zbiorów rozmytych. Następnie omówione zostanie samo sterowanie rozmyte w oryginalnym ujęciu zaproponowanym przez Zadeha. Na samym końcu przedstawione zostanie kilka przykładowych zastosowań logiki rozmytej do modelowania ruchu drogowego.

3.1. Elementy teorii zbiorów rozmytych

3.1.1. Definicja zbioru rozmytego

Podstawowym pojęciem opisywanej tu teorii jest zbiór rozmyty. W zastosowaniach matematyki, najczęściej celem definiowania zbioru jest określenie lub spre-

cyzowanie pewnego pojęcia, takiego jak na przykład liczby parzyste czy wartości dopuszczalne pewnego procesu. Jako ilustracja opisywanych tu pojęć posłuży zbiór określający pojęcie „niewielka względna zmiana", który mógłby zostać przedstawiony w następującej postaci: $\{x \in \mathbb{R} : -0.2 \leq x \leq 0.2\}$. Jest to podzbiór zbioru liczb rzeczywistych, który w tym przypadku stanowi tak zwane *uniwersum* (obszar rozważań).

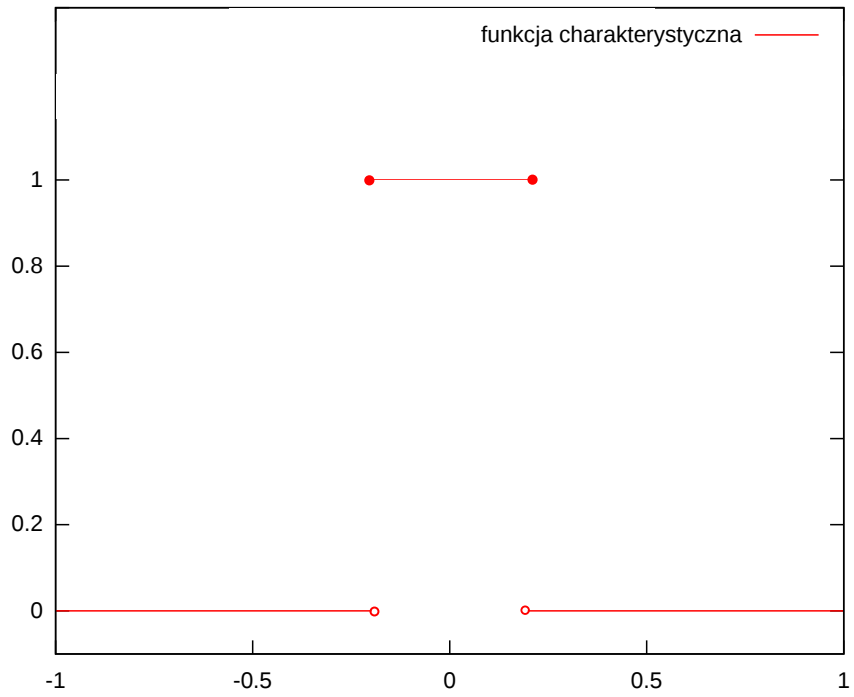
W klasycznej teorii mnogości, z każdym zbiorem A powiązana jest *funkcja charakterystyczna*

$$\chi_A : X \to \{0, 1\},$$

która dla dowolnego elementu x w uniwersum X przyjmuje wartość 1, jeśli x należy do zbioru A, oraz 0 w przeciwnym przypadku. Wykres funkcji charakterystycznej, która odpowiada zbiorowi z przykładu przedstawiony jest na rysunku 3.1.1.

Zastosowanie zbiorów do reprezentacji pojęć (wartości) precyzyjnych, z jakimi mamy do czynienia w matematyce, jest naturalne i nie stanowi problemu. Okazuje się jednak, że człowiek rzadko operuje informacją precyzyjną. Jeśli zapytaćby pasażerów tramwaju, ile czasu trwała podróż, uzyskamy odpowiedzi: około 10 minut, mniej niż kwadrans, dłużej niż zwykle, a nawet nie wiem. Wszystkie odpowiedzi opisują tę samą wartość rzeczywistego czasu przejazdu.

Pomimo tego, że człowiek doskonale radzi sobie z informacją nieprecyzyjną, w wielu przypadkach próba jej wyrażenia za pomocą precyzyjnych zbiorów skazana jest na porażkę. Podstawowym ograniczeniem zbiorów, wykorzystywanych do mo-



Rys. 3.1.1. Funkcja charakterystyczna zbioru $\{x \in \mathbb{R} : -0.2 \leq x \leq 0.2\}$.

delowania pojęć nieprecyzyjnych, jest to, że dla każdego elementu możliwe są tylko dwa stany – może on należeć do zbioru lub nie. Obserwacja ta była podstawą wprowadzenia przez Zadaha pojęcia zbioru nieostrego (rozmytego)[Zad65]. Funkcję charakterystyczną zbioru χ_A rozszerzono do *funkcji przynależności* μ_A:

$$\mu_A : X \to [0, 1],$$

która przyjmuje dowolną wartość pomiędzy 0 (całkowite nieprzynależenie) oraz 1 (pełne przynależenie).

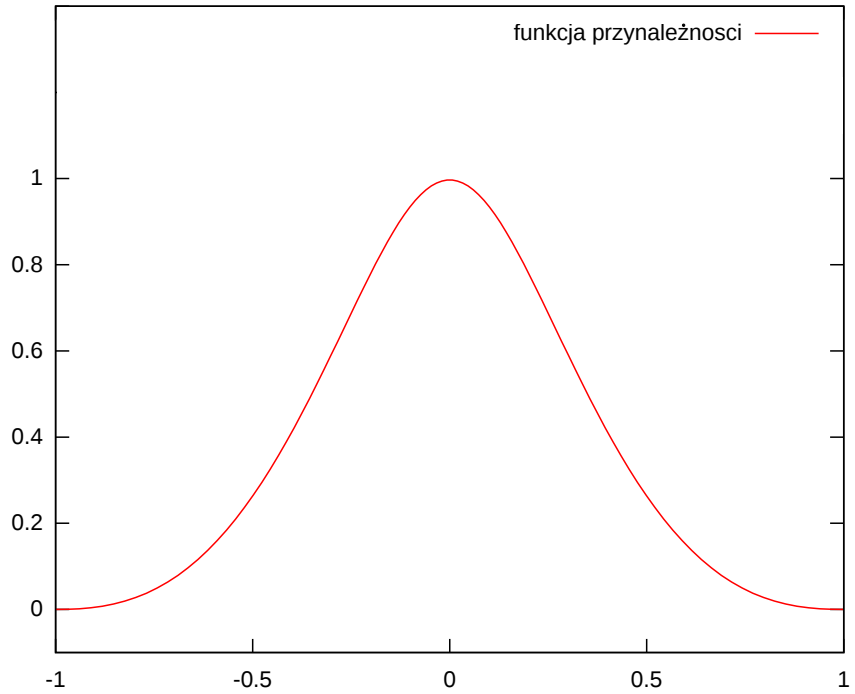
W przypadku przykładowego pojęcia „małych zmian względnych" odpowiedni zbiór rozmyty może zostać przedstawiony jak na rysunku 3.1.2.

Choć funkcje przynależności mogą być (i z reguły są), analogicznie jak w przypadku klasycznym, utożsamiane z odpowiadającymi im zbiorami rozmytymi, najczęściej przyjmuje się jednak inną formalną definicję zbioru rozmytego[Kac01]. Zbiór rozmyty A w uniwersum X określa się jako zbiór (klasyczny) par uporządkowanych

$$A = \{(\mu_A(x), x) : x \in X\},$$

gdzie μ_A jest funkcją przynależności zbioru rozmytego A, a $\mu_A(x) \in [0, 1]$ nazywany jest stopniem przynależności elementu x do zbioru rozmytego A. Tak zdefiniowany zbiór rozmyty jest tożsamy z funkcją przynależności, gdyż ta w świetle teorii mnogości określona jest jako zbiór par uporządkowanych.

W praktyce najczęściej wykorzystywane są dwie reprezentacje zbioru rozmytego: za pomocą funkcji przynależności lub w *postaci singletonowej*. Pierwsza z nich zo-



Rys. 3.1.2. Funkcja przynależności zbioru przykładowego.

stała już omówiona. Druga jest szczególnie przydatna do reprezentacji zbiorów rozmytych, dla których zbiór $supp(A) = \{x \in X : \mu_A(x) > 0\}$, nazywany *nośnikiem*, jest skończony lub przeliczalny. Poniższy zapis odpowiada zbiorowi rozmytemu z rysunku 3.1.3

$$A = {}^{0.1}/_{-0.2} + {}^{0.5}/_{-0.1} + {}^{1.0}/_{0.0} + {}^{0.5}/_{0.1} + {}^{0.1}/_{0.2}\,.$$

Korzystając z tej notacji można przedstawić dowolny zbiór klasyczny B jako zbiór rozmyty $\tilde{B}$

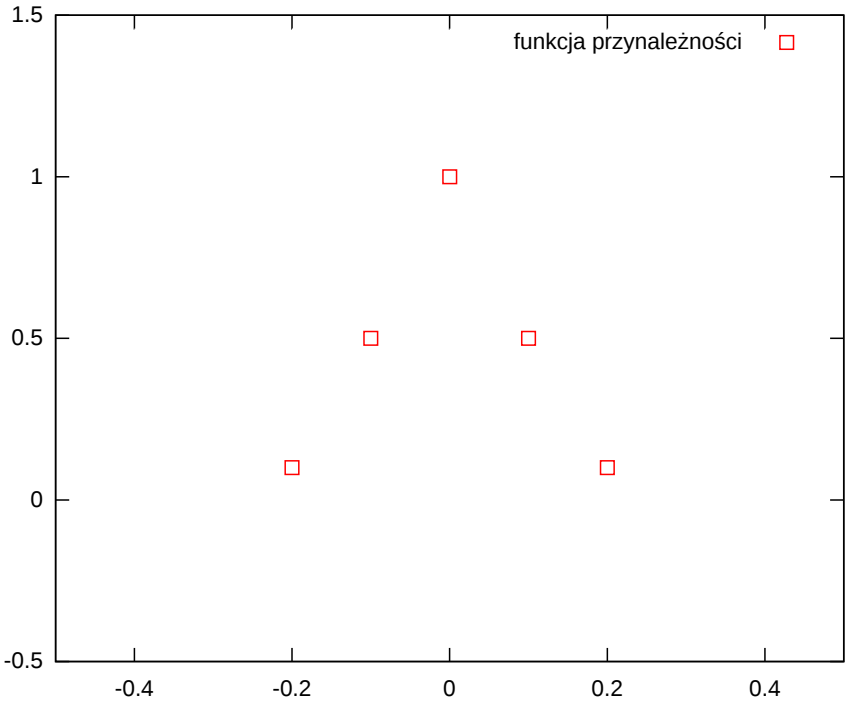
$$\tilde{B} = \sum_{b \in B} {}^{1.0}/_{b}\,.$$

3.1.2. Operacje na zbiorach rozmytych

W klasycznej teorii mnogości podstawowe operacje na zbiorach to: dopełnienie, suma oraz iloczyn. W przypadku zbiorów rozmytych, jak zostanie to dalej opisane, nie są to operację określone w jeden możliwy sposób. Na początku jednak zostanie omówiony przypadek najprostszy, który jest zarazem najczęściej wykorzystywany w praktyce.

Dopełnienie zbioru rozmytego A określa się jako zbiór rozmyty, którego funkcja przynależności dana jest w następujący sposób:

$$\mu_{\neg A}(x) = 1 - \mu_A(x)\,.$$



Rys. 3.1.3. Zbiór rozmyty $A = {}^{0.1}/_{-0.2} + {}^{0.5}/_{-0.1} + {}^{1.0}/_{0.0} + {}^{0.5}/_{0.1} + {}^{0.1}/_{0.2}$.

Suma dwóch zbiorów rozmytych A i B stanowi zbiór rozmyty, którego funkcja przynależności ma postać

$$\mu_{A\cup B}(x) = \max(\mu_A(x), \mu_B(x))\,,$$

natomiast przekrój (iloczyn) zbiorów określa następujący wzór:

$$\mu_{A\cap B}(x) = \min(\mu_A(x), \mu_B(x))\,.$$

Powyższe definicje sumy i przekroju, mimo że powszechnie stosowane i zgodne z intuicją, nie są jedynym możliwym sposobem określenia tych operacji. Poniżej zostaną opisane dwie rodziny funkcji, które dzięki spełnieniu pewnych warunków, mogą zastąpić operację maksimum i minimum w definicjach sumy i przekroju zbioru.

Funkcję $t : [0,1] \times [0,1] \to [0,1]$ taką, że dla każdego $a, b, c \in [0,1]$:

$t(a,1) = 1$	(element neutralny 1),
$a \le b \Rightarrow t(a,c) \le t(b,c)$	(monotoniczność),
$t(a,b) = t(b,a)$	(przemienność),
$t(a,t(b,c)) = t(t(a,b),c)$	(łączność)

nazywamy *t-normą*.

Funkcję $s : [0,1] \times [0,1] \to [0,1]$ taką, że dla każdego $a, b, c \in [0,1]$:

$s(a,0) = a$	(element neutralny 0),
$a \le b \Rightarrow s(a,c) \le s(b,c)$	(monotoniczność),
$s(a,b) = s(b,a)$	(przemienność),
$s(a,s(b,c)) = s(s(a,b),c)$	(łączność)

nazywamy *t-konormą*.

Podstawowymi przykładami t-norm są:

- minimum
$$t(a,b) = \min(a,b)\,,$$
- iloczyn algebraiczny
$$t(a,b) = a \cdot b\,,$$
- t-norma Łukasiewicza
$$t(a,b) = \max(0, a+b-1)\,.$$

Podstawowymi przykładami t-konorm są:

- maksimum
$$s(a,b) = \max(a,b)\,,$$

- suma probabilistyczna

$$s(a,b) = a + b - ab\,,$$

- t-konorma Łukasiewicza

$$s(a,b) = \min(a+b, 1)\,.$$

Ogólne definicje sumy i przekroju zbiorów mają postać:

$$\mu_{A\cup B}(x) = s(\mu_A(x), \mu_B(x))\,,$$

$$\mu_{A\cap B}(x) = t(\mu_A(x), \mu_B(x))\,.$$

W praktyce ważne jest, aby t-norma t i t-konorma s były odpowiednio dopasowane, tj. by spełniony był następujący warunek de Morgana:

$$\forall_{a,b\in\mathbb{R}} \quad s(a,b) = 1 - t(1-a, 1-b)\,.$$

Zapewnia to poprawne zachowanie operacji przekroju i sumy, gdy są one wykonywane łącznie.

3.2. Sterowanie rozmyte

Poniżej zostanie opisane klasyczne podejście do sterowania rozmytego zaproponowane przez Zadeha [Zad73]. Do jego najważniejszych zalet niewątpliwie należy możliwość opisu działania systemu bez znajomości jego dokładnego modelu, tylko na podstawie obserwacji i doświadczenia. Jest ono nazywane również podejściem lingwistycznym, z tego powodu, że podstawowym jego elementem jest pojęcie *zmiennej lingwistycznej*. Przykładem zmiennej lingwistycznej jest *temperatura*, przyjmująca wartości takie jak *wysoka, niska czy średnia*, które nazywane są wartościami lingwistycznymi. Wartości te są utożsamiane z pewnymi zbiorami rozmytymi, określonymi w uniwersum dopuszczalnych wartości zmiennej.

Wykorzystując zmienne lingwistyczne można następnie określić *rozmyte zdania warunkowe*. Przykładowo, niech x i y będą zmiennymi lingwistycznymi oraz A i B są wartościami lingwistycznymi tych zmiennych. Wtedy rozmyte zdanie warunkowe (regułę rozmytą) można zapisać jako

$$(i) \quad \text{IF } \underbrace{x = A}_{\text{przesłanka}} \text{ THEN } \underbrace{y = B}_{konkluzja}\,.$$

Reguła ta posłuży w dalszym toku rozdziału do ilustracji opisywanych tam zagadnień.

Wnioskowanie odbywa się z wykorzystaniem uogólnionej reguły *Modus Ponens*:

Fakt	$x = A^*$
Implikacja	IF $x = A$ THEN $y = B$
Wniosek	$y = B^*$

Różnica względem tradycyjnej reguły Modus Ponens polega na tym, że fakt nie musi być dokładnie zgodny z przesłanką implikacji – przesłanka jest spełniona w pewnym stopniu. Określenie jego wartości to pierwszy etap tak zwanego wnioskowania rozmytego (inferencji). Schemat 3.2.1 przedstawia cały proces. Poniżej podstawowe jego etapy zostaną dokładniej omówione.

Pierwszy etap to określanie stopnia spełnienia przesłanek a_i reguły i, czyli *aktywacja*. W przypadku prostych przesłanek takich, jak w poprzednim przykładzie stopnień spełnienia przesłanek jest tożsamy z stopniem przynależności wartości zmiennej x^* do zbioru A, $a_i = \mu_A(x^*)$. Możliwe jest jednak określenie złożonych przesłanek. W przypadku, gdy ma ona postać NOT $x = A$, przesłanki są spełnione w stopniu $a_i = 1 - \mu_A(x^*)$. Natomiast jeśli przesłanka jest złożona z dwóch przesłanek połączonych spójnikiem logicznym (koniunkcja lub alternatywa) $P_1 \, {}^{AND}/_{OR} \, P_2$, to

$$a_i = t(a_{P_1}, a_{P_2}) \qquad \text{(w przypadku koniunkcji)}$$
$$a_i = s(a_{P_1}, a_{P_2}) \qquad \text{(w przypadku alternatywy)}$$

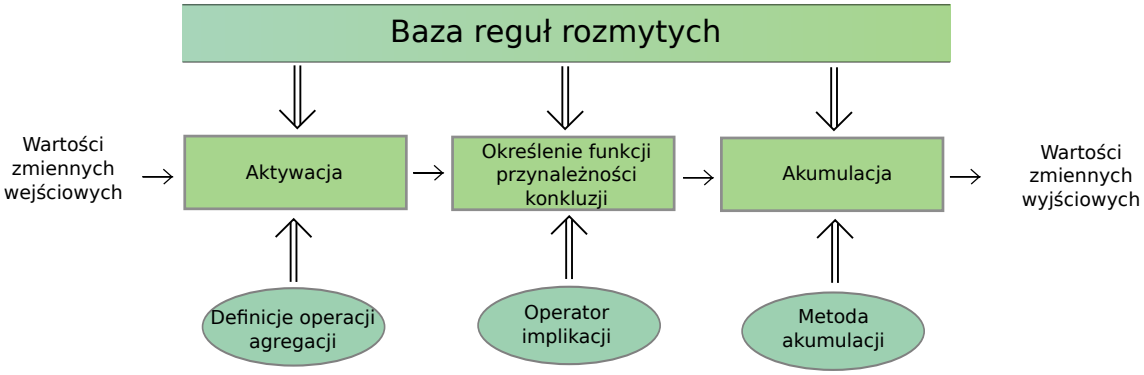
gdzie t i s oznaczają odpowiednio t-norme i t-konormę. W literaturze proces ten nazywany jest *agregacją*. W przypadku bardziej złożonych przesłanek, zawierających wiele składowych, agregację wykonuje się zgodnie z priorytetami operatorów.

Drugi etap to określenie funkcji przynależności konkluzji $y = B$ reguły i dla danych wartości wejściowych, oznaczanej przez μ_{y^*}. W tym celu wykorzystuje się operator implikacji rozmytej. Najczęściej stosowany jest *operator Mamdaniego*, w którego przypadku funkcja przynależności konkluzji określona jest jako

$$\mu^i_{y^*}(x) = \min(a_i, \mu_B(x)) \, .$$

Drugim powszechnie stosowanym operatorem jest *implikacja produktowa*

$$\mu^i_{y^*}(x) = a_i \cdot \mu_B(x) \, .$$



Rys. 3.2.1. Schemat wnioskowania rozmytego.

Dysponujemy teraz wynikową funkcją przynależności określoną dla każdej reguły z osobna. W następnym etapie, zwanym *akumulacją*, dla każdej zmiennej lingwistycznej y znajdującej się w konkluzji jakiejś reguły, należy ustalić wynikową funkcję przynależności μ_{y^*}. Najczęściej wykorzystuje się w tym celu operator maksimum

$$\mu_y^* = \max_i \mu_{y^*}^i \,,$$

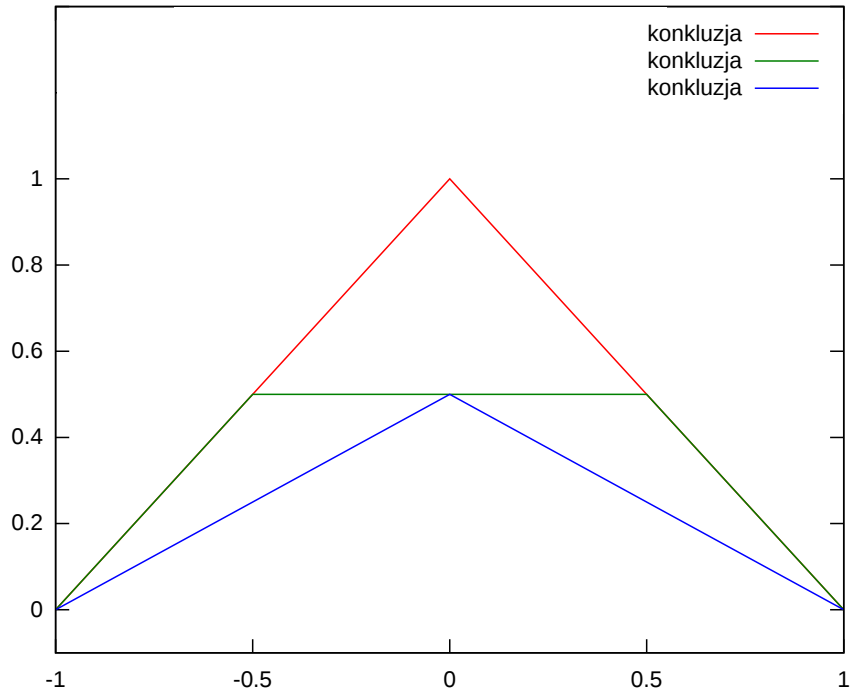
chociaż możliwe jest zastosowanie dowolnej t-normy.

Następnie rozmyty zbiór wynikowy przekształca się na wartość liczbową w procesie *defuzyfikacji*, dlatego też wyjście sterownika rozmytego pomimo zastosowania w nim wartości nieprecyzyjnych, jest liczbą w tradycyjnym znaczeniu.

3.2.1. FCL – Fuzzy Control Language

Fuzzy Control Language jest językiem przeznaczonym do implementacji sterowników rozmytych. Został on zdefiniowany przez International Electrotechnical Commission (IEC) w dokumencie IEC 61131-7[fcl97].

Dzięki prostocie oraz dostępności wielu narzędzi umożliwiających przetwarzanie sterowników FCL, możliwe jest tworzenie i testowanie sterowników, bez posiadania specjalistycznej wiedzy na temat wnioskowania rozmytego.



Rys. 3.2.2. Funkcja przynależności konkluzji przy wartości $a_i = 0.5$.

3.3. Zastosowania logiki rozmytej w modelowaniu ruchu drogowego

Modelowanie systemów transportowych do niedawna odbywało się z wykorzystaniem rozwiązań deterministycznych i/lub stochastycznych. Wykorzystywana do ich konstrukcji obiektywna i precyzyjna wiedza, nie jest jednak jedynym dostępnym źródłem informacji. Subiektywne dane pochodzące od użytkowników systemu transportowego, mimo iż odgrywają znaczącą rolę w rzeczywistości, w wielu modelach są pomijane.

Kierowcy pojazdów operują w środowisku nieprecyzyjnym, z pewnością każdy kierowca wie, jaka jest odległość od poprzedzającego go pojazdu (bezpieczna, mała, duża, ...). Jednak mało kto potrafiłby odpowiedzieć na precyzyjne pytanie ile metrów dzieli jego pojazd od pojazdu go poprzedzającego. Dla człowieka nieprecyzyjność jest czymś zupełnie naturalnym, nie przysparzającym kłopotów.

Ruch drogowy jest doskonałym przykładem na to, że modelowanie z wykorzystaniem metod precyzyjnych, procesów bardzo silnie zależnych od człowieka, jest zadaniem trudnym. Naturalnym zdaje się zatem wykorzystanie aparatu logiki rozmytej, dzięki któremu należyta uwaga zostanie poświęcona nieprecyzyjności i subiektywności informacji zawartej w modelu.

Poniższy podrozdział ma na celu zaprezentowanie przykładów zastosowania logiki rozmytej do modelowania i zarządzania ruchem drogowym, zużyciem paliwa i emisją spalin. Więcej informacji na ten temat można znaleźć w pracy [Teo99].

3.3.1. Sterowanie sygnalizacją świetlną

Historycznie pierwsze zastosowanie logiki rozmytej do zagadnień transportowych miało miejsce w 1977 roku[Teo99]. Pappis i Mamdani wykorzystali sterowanie rozmyte do kontroli sygnalizacji świetlnej na pojedynczym skrzyżowaniu dwóch dróg jednokierunkowych. Od tego czasu powstało wiele rozwiązań obejmujących coraz to większy obszar. W pracy [KSXL11] zaprezentowano kompleksowy system zarządzania światłami drogowymi oparty o architekturę neuronowo–rozmytą. Dzięki koordynacji tak zwanej „zielonej fali", system ten umożliwia zarówno zmniejszenie czasu przejazdu, jak i emisji spalin oraz zużycia paliwa.

3.3.2. Określanie popytu transportowego

Jak zostało wspomniane w rozdziale drugim macierze OD stanowią kluczowy element wielu modeli ruchu drogowego. Jednak zarówno ich stworzenie jak i późniejsza aktualizacja są zadaniami bardzo czasochłonnymi i kosztownymi. Z tego powodu

powstało wiele metod ich aktualizacji w oparciu o dane, których pozyskanie jest mniej kosztowne od przeprowadzania kompleksowych badań ruchu. Jedną z najpopularniejszych jest aktualizacja macierzy na podstawie danych z detektorów ruchu umieszczonych na skrzyżowaniach. Do poprawnego działania wymagają one jednak wielu zewnętrznych informacji o systemie transportowym i rozkładzie popytu.

W pracy [RC98] zaproponowano metodę aktualizacji macierzy OD na podstawie danych z detektorów opartą na wnioskowaniu rozmytym. System działa dwuetapowo, w pierwszym korzystając z bazy reguł pozyskanych od ekspertów, ustalane są niezbędne wartości parametrów zewnętrznych. Następnie stosowana jest rekurencyjna procedura aktualizacji. Badania przeprowadzone na testowych sieciach drogowych, pokazały, że wyniki uzyskane tą metodą są bardzo bliskie otrzymanym w sposób klasyczny, przy czym wymagają znacznie mniejszej ilości informacji.

3.3.3. Wybór trasy przejazdu

Wybór trasy przejazdu pomiędzy dwoma punktami miasta, dokonywany przez kierowcę, jest w dużym stopniu subiektywny. Decyzja oparta o poprzednie doświadczenia, własne przekonania, jest często trudna do modelowania. W rozdziale drugim zostało opisane pojęcie równowagi użytkowników, w której to użytkownik zawsze wybiera trasę przejazdu o najmniejszym koszcie.

Pierwsze prace, które wykorzystały logikę rozmytą do modelowania procesu wyboru trasy, powstały w latach 90-tych XX wieku. W swoim rozwiązaniu Teodorović oraz Kikuchi, stosują wnioskowanie rozmyte w celu modelowania procesu podejmowania decyzji przez kierowcę. W następnych latach powstało wiele analogicznych modeli, starających się rozwiązać ten problem[Teo99].

Rozdział 4

Pakiet oprogramowania *ProSym*

Rozdział ten zawiera prezentację autorskiego oprogramowania *ProSym*, która skierowana jest do potencjalnych użytkowników systemu. Pierwsza część opisuje, dlaczego powstanie takiego systemu jest potrzebne, oraz jakie są korzyści z tego płynące. Druga część zawiera opis komponentów wchodzących w jego skład. Dzięki stylowi zbliżonemu do podręcznika użytkownika, możliwe jest poznanie systemu od jego strony. Część trzecia opisuje wymagania systemu, zarówno techniczne, jak i względem danych o infrastrukturze drogowej oraz modelu ruchu. Ostatnia część poświęcona jest dotychczasowym wdrożeniom systemu oraz planom rozwoju na przyszłość.

4.1. Idea

W ostatnich latach nastąpił dynamiczny rozwój rynku oprogramowania wspomagającego kierowcę w wyborze trasy przejazdu. W systemy nawigacji satelitarnej wyposażone są już nie tylko samochody luksusowe, lecz również telefony komórkowe, dzięki czemu niemal każdy może znaleźć „najlepszą" trasę dojazdu do celu. „Najlepsza" z reguły oznacza trasę spełniająca zadane kryteria, o najmniejszej długości. Niekoniecznie oznacza to, że czas przejazdu będzie najkrótszy. Kierowcy marnują wiele godzin, czekając w korkach, których mogliby uniknąć poprzez wybranie innej drogi. Sytuację pogarsza fakt, że wybór nieoptymalnej trasy, oprócz dłuższego czasu przejazdu, powoduje większe zużycie paliwa i emisję spalin. Współczesne systemy nawigacji w bardzo wąskim zakresie pomagają kierowcom. Dane o korkach są albo statyczne (i przez to często nieaktualne), albo pojawiają się z opóźnieniem (w przypadku systemów operujących na danych zgłaszanych przez kierowców).

Tę niekorzystną sytuację można by poprawić poprzez zastosowanie symulacji ruchu drogowego. Jest to jednak zadanie trudne i czasochłonne, a często też kosztowne. Konieczna jest znajomość oprogramowania do symulacji ruchu, a także stworzenie

odpowiedniego interfejsu, który pozwoli dotrzeć do kierowców. Co więcej, należy przełożyć wyniki symulacji na opis tego, gdzie i kiedy powstaną korki oraz jakie będzie zużycie paliwa i emisja spalin.

Celem powstania oprogramowania *ProSym* (*Prosta Symulacja*) jest uproszczenie, w jak największym stopniu, procesu tworzenia interaktywnej symulacji ruchu. Dzięki niemu osoby nieposiadające zaawansowanej wiedzy z zakresu informatyki, mogą udostępnić mieszkańcom miasta (lub innego wybranego obszaru) aplikację, która przyczyni się nie tylko do poprawy komfortu życia, lecz również pozwoli oszczędzić zarówno pieniądze, jak i środowisko. *ProSym* skierowany jest przede wszystkim do władz miejskich oraz organizacji pozarządowych chcących, niewielkim nakładem pracy, przyczynić się do ochrony środowiska.

4.2. Skład pakietu

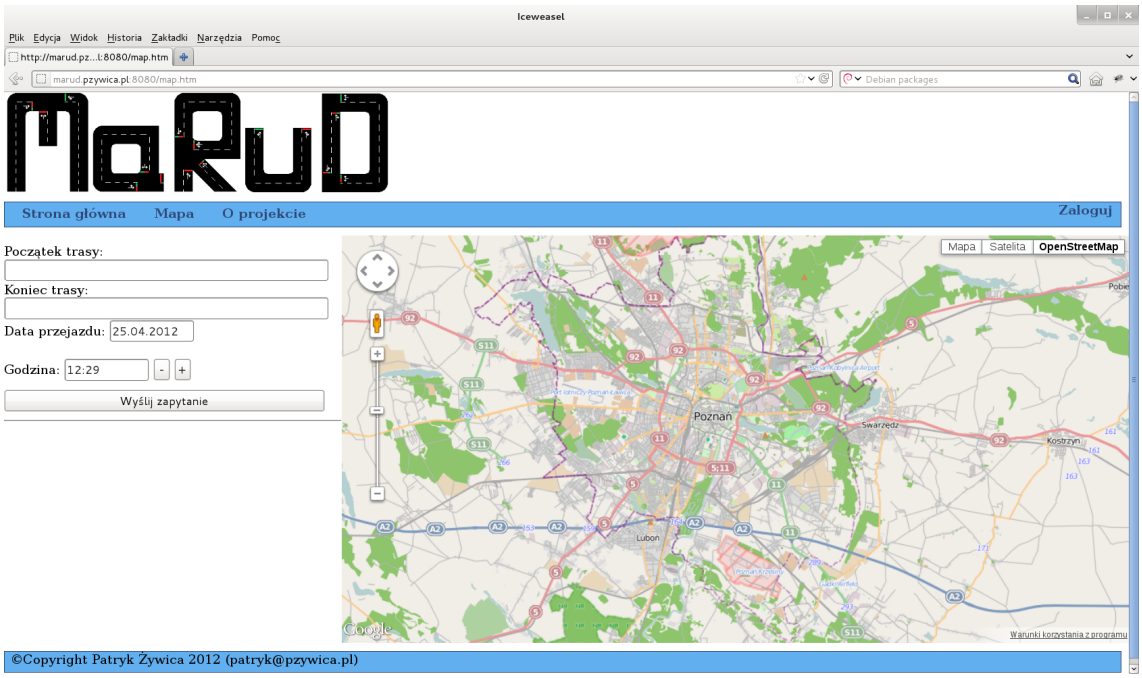
Oprogramowanie *ProSym* to zbiór narzędzi, których zadaniem jest ułatwienie przygotowania symulacji ruchu. Obecnie pakiet składa się z dwóch komponentów: Mapy Ruchu Drogowego (MaRuD) oraz Korektora Macierzy OD (KorMOD). Pierwszy pełni rolę wspomnianego wcześniej interfejsu, za pomocą którego użytkownicy – kierowcy, mogą wyszukać trasy przejazdu oraz zapoznać się z przewidywanymi warunkami drogowymi i spodziewanym zużyciem paliwa oraz emisją zanieczyszczeń.

KorMOD natomiast odpowiedzialny jest za aktualność prezentowanych wyników. Jak wiadomo miasta rozwijają się, zmianie ulega również infrastruktura drogowa. Do tego, aby wyniki symulacji dobrze odzwierciedlały sytuację drogową, niezbędne jest, aby model ruchu miasta był jak najbardziej aktualny. KorMOD wykorzystuje wnioskowanie rozmyte oraz dane zebrane przez aplikację MaRuD do nieustannego aktualizowania macierzy OD wykorzystywanej do symulacji.

Poniżej oba programy zostaną opisane z punktu widzenia użytkownika, czyli kierowcy chcącego znaleźć odpowiednią trasę podróży. W osobnej sekcji omówiony zostanie również klient mobilny MaRuD4Android.

4.2.1. MaRuD

Mapa Ruchu Drogowego udostępniana jest jako strona internetowa. Za jej pomocą użytkownik może wprowadzić informacje o planowanej podróży: miejsce rozpoczęcia, zakończenia oraz dzień i godzinę. Punkt początkowy i końcowy może zostać wprowadzony jako adres, nazwa własna, lub współrzędne geograficzne. W przypadku błędnych danych (takich jak „literówka" w nazwie ulicy) system stara się odgadnąć intencje użytkownika. Ze względu na zabezpieczenie przed zbytnim obciążeniem serwera możliwe jest wybranie daty przejazdu z co najwyżej trzydniowym



Rys. 4.2.1. Strona główna aplikacji MaRuD Poznań.

wyprzedzeniem.

W odpowiedzi na zapytanie, wyświetlana jest informacja o kilku proponowanych trasach przejazdu. Użytkownik może prześledzić ich przebieg na mapie. W razie potrzeby możliwa jest edycja zaproponowanych tras. Odbywa się to techniką „przeciągnij i upuść". Podobnie możliwa jest zmiana punktu początkowego i końcowego poprzez przesunięcie ich w dowolne miejsce na mapie.

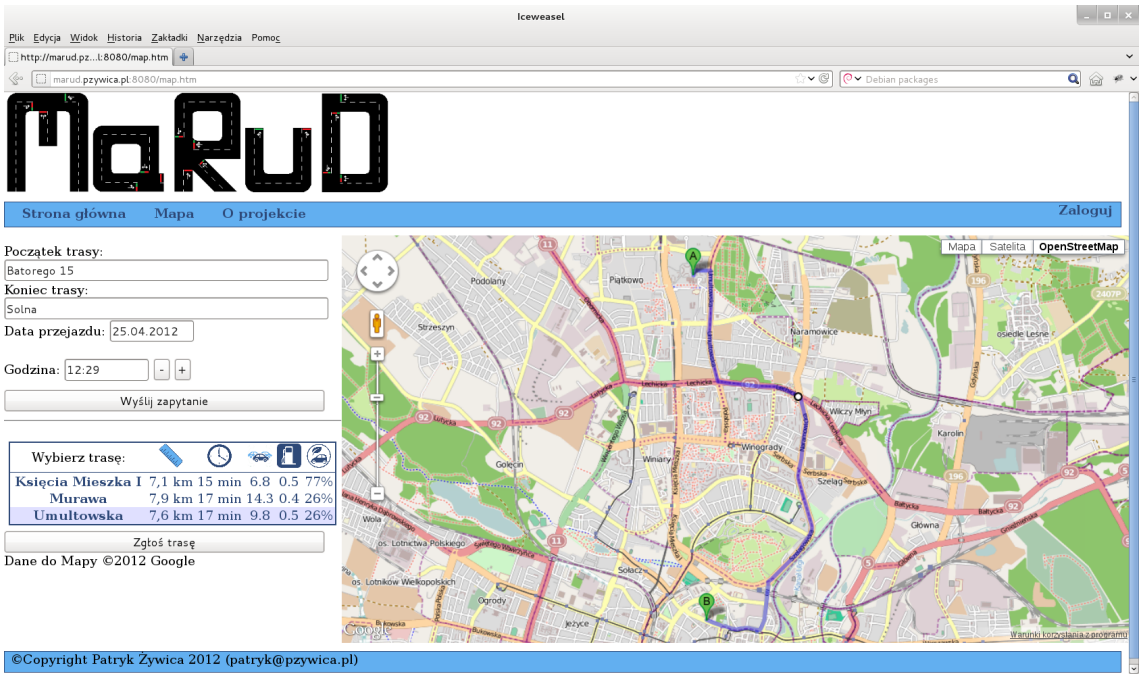
Tabela wyboru trasy zawiera ponadto informacje, które odróżniają ten system od innych dostępnych na rynku. Oprócz standardowej informacji, jaką jest długość trasy, udostępniane są również przewidywane informacje o czasie przejazdu, średniej liczbie pojazdów na drodze, łącznym zużyciu paliwa oraz emisji spalin. Znaczenie prezentowanych wartości jest następujące:

średnia ilość pojazdów – wyrażona w samochodach na minutę, dotyczy całości trasy,

łączne zużycie paliwa – wyrażone w litrach, wartość dotyczy teoretycznego przeciętnego europejskiego samochodu zdefiniowanego w modelu HBEFA,

emisja spalin – wyrażona jako procent o jaki emisja spalin będzie większa niż w przypadku przejazdu tą trasą w warunkach optymalnych.

Wyświetlane informację oparte są o aktualne wyniki symulacji, przeprowadzanej na serwerze MaRUD.



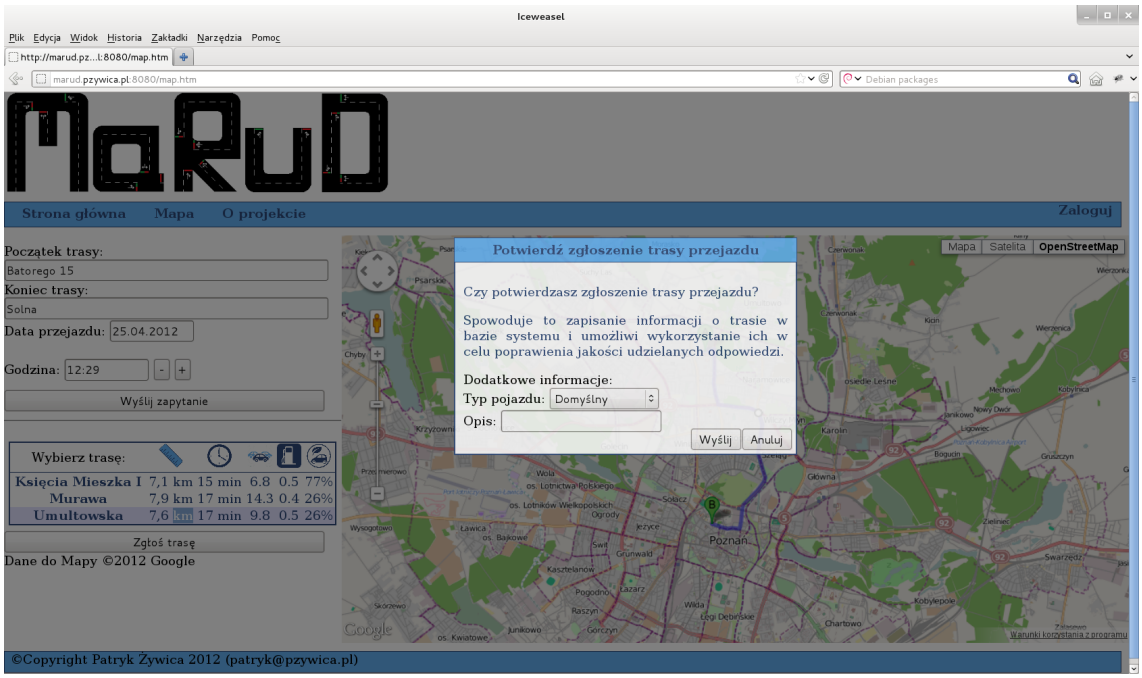
Rys. 4.2.2. Widok wyboru trasy aplikacji MaRuD Poznań.

4.2.2. KorMOD

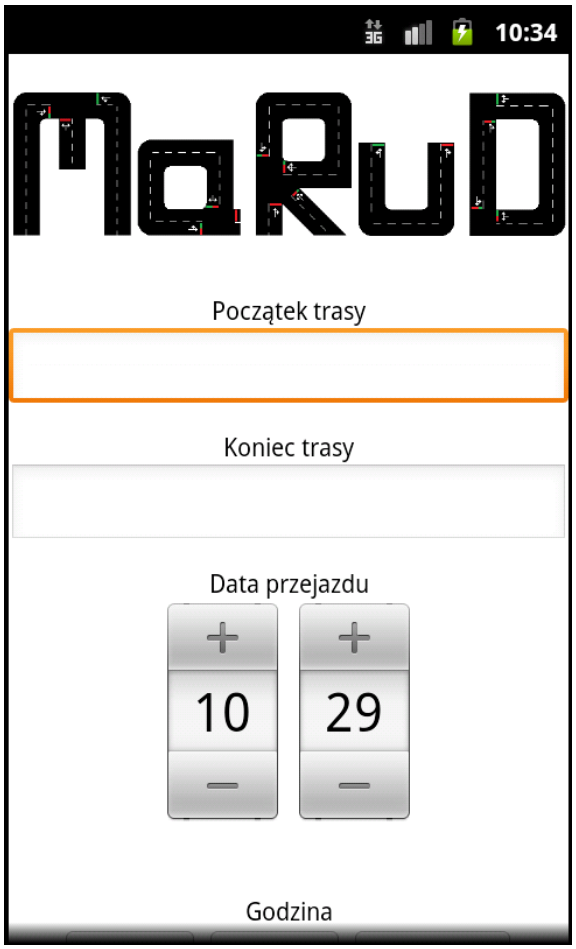
KorMOD jest komponentem działającym po stronie serwera. To on odpowiada za aktualność prezentowanych wyników symulacji. Użytkownik systemu powinien być jednak świadomy jego istnienia, gdyż do poprawnego funkcjonowania wymaga danych o trasach przejazdów. Osoby korzystające z aplikacji MaRuD, mają możliwość dodania swojej trasy do bazy przejazdów. W tym celu po wybraniu odpowiedniej trasy należy ją „zgłosić" (rys. 4.2.3). Zapisane zostaną godzina i dzień przejazdu, typ pojazdu oraz krótki opis pozwalający na późniejszą identyfikację lub zawierający dodatkowy komentarz użytkownika. Administrator systemu ma możliwość zarządzania bazą zgłoszonych tras z poziomu przeglądarki internetowej.

4.2.3. Wersja mobilna

Jak zostało to wcześniej wspomniane, urządzenia mobilne, w szczególności telefony, wykorzystywane są przez kierowców jako systemy nawigacji satelitarnej. Dlatego też w skład pakietu *ProSym* wchodzi również aplikacja mobilna, której funkcjonalność odpowiada aplikacji MaRuD. W obecnej wersji dostępna jest ona na platformę Android. Interfejs użytkownika został odpowiednio dostosowany, aby spełniać wymagania stawiane aplikacjom mobilnym (rysunek 4.2.4).



Rys. 4.2.3. Widok zgłaszania trasy przejazdu systemu KorMOD.



Rys. 4.2.4. Aplikacja MaRuD4Android, ekran początkowy.

4.3. Wymagania

Wymagania systemu *ProSym* względem użytkowników są minimalne. Do przeglądania strony internetowej potrzebny jest komputer klasy PC połączony z Internetem i wyposażony w przeglądarkę internetową obsługującą JavaScript. Do wdrożenia systemu *ProSym* niezbędne jest spełnienie pewnych wymagań, które można podzielić na techniczne i dotyczące danych wejściowych. Kluczowe z nich zostaną omówione poniżej.

- Wymagania techniczne

  Serwer: do uruchomienia oprogramowania *ProSym* niezbędny jest komputer z systemem operacyjnym Windows XP lub nowszym, albo Linux (testowane na dystrybucji Debian 6), posiadający przynajmniej 512mb pamięci operacyjnej oraz 10gb pamięci dyskowej (20gb w przypadku dużych obszarów symulacji).

  Dostęp do modułu mapy: wizualizacja mapy korzysta z Google Maps API udostępnianego przez firmę Google nieodpłatnie. Konieczne jest jednak posiadanie klucza dostępu. Opis procesu przyznawania klucza znajduje się na stronach Google Maps `http://developers.google.com/maps/`.

- Wymagania wobec danych wejściowych

  Mapa: do przeprowadzana symulacji konieczne jest posiadanie mapy symulowanego obszaru. Zalecanym źródłem map jest portal OpenStreetMaps (OSM) `http://openstreetmaps.org`. Umożliwia on pobranie szczegółowej mapy dowolnego wybranego obszaru. Obsługiwane są jednak również inne formaty wspierane przez symulator SUMO.

  Model ruchu: w obecnej wersji pakiet *ProSym* wspiera modele ruchu w formie macierzy OD. Macierz musi być powiązana z mapą, zawierającą definicje rejonów transportowych.

4.4. Wdrożenie i dalszy rozwój

Na chwilę obecną system *ProSym* został uruchomiony dla Poznania i okolic (`http://marud.pzywica.pl/poznan`). Korzysta on z map OSM oraz z modelu ruchu powstałego na podstawie danych uzyskanych od Zarządu Dróg Miejskich w Poznaniu. Serwer zlokalizowany jest na Wydziale Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza.

Trwają prace nad uruchomieniem systemu dla czwartego co do wielkości miasta w województwie wielkopolskim – Piły. Dalszy rozwój najprawdopodobniej spowoduje objęcie symulacją wszystkich większych miast Wielkopolski.

4.5. *ProSym* jako projekt informatyczny

ProSym jest projektem otwartym udostępnianym na licencji GPLv3. Podczas jego tworzenia wykorzystano wiele technologii i aplikacji, bez których nie mógłby on powstać lub nie oferowałby pewnych funkcjonalności. W szczególności są to:

- symulator SUMO,
- Google Maps API,
- serwis OpenStreetMaps,
- Spring Framework,
- jQuery,
- Java.

Następne dwa rozdziały zawierają bardziej szczegółowe informację o programach wchodzących w skład pakietu *ProSym*. Ponadto aktualne informacje o projekcie dostępne są na stronie `http://prosym.sourceforge.net/` oraz `http://prosym.pzywica.pl/`.

Rozdział 5

Mapa Ruchu Drogowego – MaRuD

Poprzedni rozdział zawierał podstawowe informacje o funkcjonalnościach aplikacji wchodzących w skład pakietu *ProSym*. Bieżący oraz następny opisują, w jaki sposób są one realizowane.

Jako pierwszy zostanie omówiony sposób w jaki realizowane są zapytania użytkownika. Następnie wyjaśniony zostanie sam proces przeprowadzania symulacji wraz z dyskusją o jakości uzyskanych wyników.

5.1. Zasada działania

Wiadomo już jak działa aplikacja MaRuD z punktu widzenia użytkownika. Obecnie zostanie omówiony proces realizowania zapytań użytkownika, wykonywany każdorazowo, nie tylko gdy kierowca zada zapytanie w postaci punktu docelowego i końcowego, lecz również, gdy dokona modyfikacji trasy.

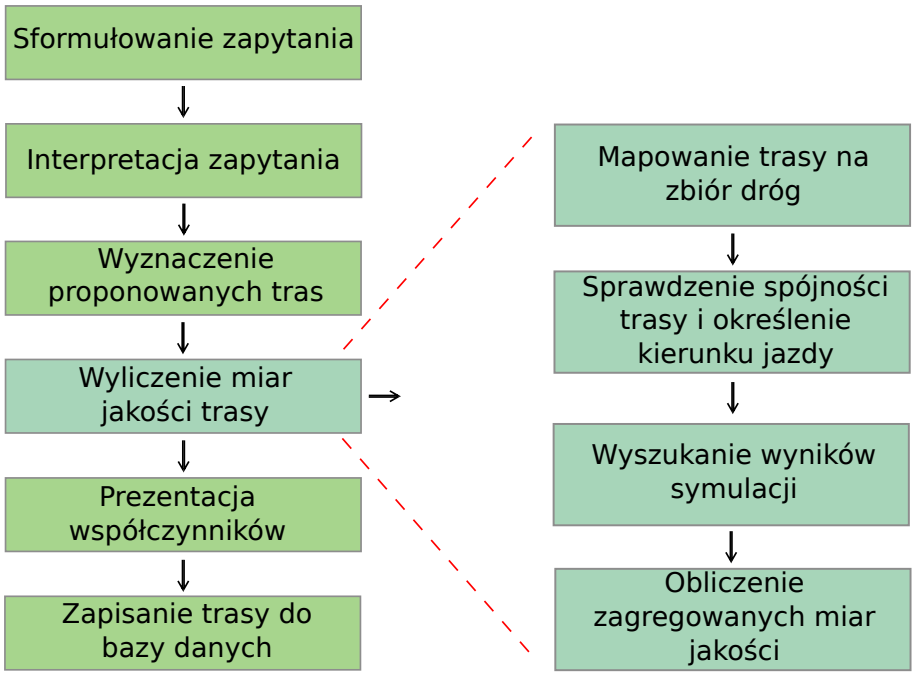
Podstawowy proces realizacji zapytania został przedstawiony na rysunku 5.1.1. Każdy z sześciu etapów zostanie omówiony poniżej.

1. Sformułowanie zapytania

   Użytkownik formułuje zapytanie poprzez podanie podstawowych informacji o podróży: jej początku i końcu, godzinie oraz dniu. Określenie punktu początkowego i końcowego może odbywać się w jeden z następujących sposobów: poprzez podanie nazwy ulicy i opcjonalnie numeru domu, podanie współrzędnych geograficznych lub nazwy własnej (np. Urząd Miasta Poznań).

2. Interpretacja zapytania

   O ile godzina i dzień przejazdu określone są jednoznacznie i po wstępnej walidacji nie wymagają dalszego przetwarzania, o tyle miejsce startu i zakończenia przed dalszym przetwarzaniem muszą zostać ujednoznacznione. W tym celu



Rys. 5.1.1. Schemat realizacji zapytania przez aplikację MaRuD.

w pierwszej kolejności zakres poszukiwania jest zawężany do obszaru objętego symulacją. Następnie zapytanie przekazywane jest do serwisu Google Maps API, który dokonuje tak zwanego geokodowania, czyli przekształcenia nazw i adresów z postaci opisowej na współrzędne geograficzne.

3. Wyznaczenie i wyświetlenie proponowanych tras

   Obecna wersja systemu MaRuD używa do wyznaczania i wyświetlania proponowanych tras serwisu Google Maps API, który po przeprowadzonym geokodowaniu udziela informacji o możliwych trasach przejazdu pomiędzy dwoma punktami. Otrzymane w ten sposób trasy przejazdu są ciągiem punktów (par współrzędnych geograficznych), które mogą być bezpośrednio naniesione na mapę w celu wizualizacji drogi.

4. Wyliczenie miar jakości trasy

   Poprzez miarę jakości trasy rozumie się tu wartość liczbową umożliwiającą ocenę danej trasy zgodnie z pewnym kryterium. W obecnej wersji systemu *ProSym* dostępne są następujące miary: czasu przejazdu, zużycia paliwa, natężenia ruchu i emisji spalin.

   Proces wyznaczania miar można podzielić na cztery etapy. Każdy z nich zostanie poniżej opisany.

   (a) Mapowanie trasy na zbiór dróg

   W celu wyliczenia wartości poszczególnych miar, w pierwszej kolejności konieczne jest przetłumaczenie ciągu punktów, opisujących trasę wybraną

przez użytkownika, na zbiór dróg będących elementami mapy wykorzystywanej w symulacji. Rozwiązanie tego problemu oparte jest o następujący algorytm:

Wejście: ciąg punktów (P_i) długości n oraz mapa drogowa w postaci grafu skierowanego G

Wyjście: zbiór dróg, które potencjalnie mogą budować trasę wybraną przez użytkownika

$D \leftarrow \emptyset$
for all krawędź $e \in G$ **do**
  if odległość krawędzi e od dowolnego punktu P_i jest mniejsza niż 200 metrów **then**
    $D \leftarrow D \cup \{e\}$
  end if
end for
$F \leftarrow \{\}$
for all $k \in D$ **do**
  $F[k] \leftarrow 0$
end for
for all $i \in \{2..n\}$ **do**
  for all $k \in D$ **do**
    $d \leftarrow$ odległość krawędzi k od punktu P_i
    $a \leftarrow$ kąt pomiędzy krawędzią k oraz odcinkiem (P_{i-1}, P_i)
    $F[k] \leftarrow \max\{\text{funkcjaOceny}(d, a), F[k]\}$
  end for
end for
$R \leftarrow \emptyset$
for all $k \in D$ **do**
  if $F[k] \geq 0.5$ **then**
    $R \leftarrow R \cup \{k\}$
  end if
end for
return R

W pierwszej pętli wybierane są krawędzie, które znajdują się w sąsiedztwie punktów określonych przez użytkownika. Następnie każda krawędź oceniana jest z uwzględnieniem jej odległość i kąta jaki tworzy z krzywą nakreśloną przez zadany ciąg punktów. Krawędzie, dla których wartość funkcji oceny przekracza wartość progową zostają zwrócone jako wynik.

Ponadto w tym kroku wyznaczane są krawędzie początkowe i końcowe, jako te, dla których ocena względem punktów początkowych i końcowych była najwyższa.

(b) Sprawdzenie spójności i określenie kierunku jazdy

W poprzednim kroku wyznaczony został zbiór potencjalnych krawędzi, które wchodzą w skład trasy. Kolejnym etapem przetwarzania jest sprawdzenie spójności tzn. określenie czy istnieje zgodna z kierunkiem krawędzi ścieżka (trasa) od krawędzi początkowej do końcowej. W tym celu wykorzystywany jest standardowy algorytm przeszukiwania w głąb z nawrotami. W przypadku odpowiedzi negatywnej (również, gdy krawędź początkowa lub końcowa nie została znaleziona), program kończy działanie, informując o prawdopodobnym błędzie w określeniu trasy.

Gdy znalezionych tras jest więcej niż jedna, wybierana jest ta, której długość jest najbardziej zbliżona do długości krzywej nakreślonej przez zadany ciąg punktów.

Poprawność metody ustalania trasy (etapy *a* oraz *b*) została sprawdzona testami przeprowadzonymi na zestawie 50 ciągów punktów, z których 30 reprezentowało trasy poprawne, 15 powstało przez zmodyfikowanie poprawnej trasy w sposób uniemożliwiający przejazd samochodem. Pozostałe zestawy testowe to losowe ciągi punktów. Otrzymano satysfakcjonujące wyniki. 27 tras z pierwszej grupy zostało poprawnie rozpoznanych. Spośród tras zmodyfikowanych błędnie została rozpoznana tylko jedna (system znalazł trasę przejazdu, korzystając z dróg pobliskich miejscu modyfikacji). Żaden z ciągów losowych nie został rozpoznany jako poprawna trasa.

(c) Wyszukanie wyników symulacji

Dla każdej krawędzi (drogi), która przynależy do wybranej trasy, w bazie danych wyszukiwane są aktualne wyniki symulacji dla zadanego czasu przejazdu. Jeśli nie zostaną odnalezione (co może mieć miejsce dla rzadko uczęszczanych dróg w godzinach nocnych lub przed zakończeniem pierwszej symulacji), przyjmowane są wartości optymalne, czyli takie, w których ruch pojazdu nie jest wstrzymywany przez inne samochody.

Dla każdego 15 minutowego przedziału czasowego oraz dla każdej drogi w sieci w bazie dostępne są następujące wyniki symulacji:

- unormowane wartości emisji spalin i zużycia paliwa pojedynczego pojazdu (wyrażone w miligramach na kilometr),
- ilość pojazdów, która przejechała daną drogą,
- średnia prędkość,
- średni czas oczekiwania.

(d) Obliczenie zagregowanych miar jakości

Na podstawie dostępnych informacji dla każdej drogi obliczane są zagregowane miary dla całej trasy. W przypadku miary zużycia paliwa jest to

suma wartości dla wszystkich krawędzi. Miara natężenia ruchu wyliczana jest jako średnia ważona natężeń ruchu dróg wchodzących w skład trasy. Wagi są proporcjonalne do długości krawędzi.

Bardziej skomplikowane obliczania mają miejsce w przypadku emisji spalin. Symulator SUMO umożliwia modelowanie pięciu podstawowych zanieczyszczeń emitowanych przez samochody. Natomiast użytkownicy systemu – kierowcy rzadko posiadają wiedzę pozwalającą określić, czy podana wartość jest duża czy mała. Dlatego zdecydowano, aby miarę emisji spalin prezentować jako procentowy wzrost emisji w stosunku do jazdy w warunkach optymalnych. Stąd wartość 0% oznacza, że pojazd będzie jechał w warunkach, w których otaczający go ruch uliczny nie ma na niego negatywnego wpływu.

Przy wyliczaniu miary emisji uwzględniana jest emisja wszystkich pięciu dostępnych grup substancji. Wartości te są sumowane po uprzednim znormalizowaniu do przedziału od 0 do 1. Tak wyliczona wartość bierze udział w wyliczeniu procentowej miary względem wartości optymalnej.

5. Prezentacja miar jakości

   Wyliczone miary wyświetlane są jako integralna część ekranu wyboru trasy. Zastosowano tu przetwarzanie asynchroniczne, gdyż obliczenia mogą być czasochłonne, szczególnie w przypadku długich tras.

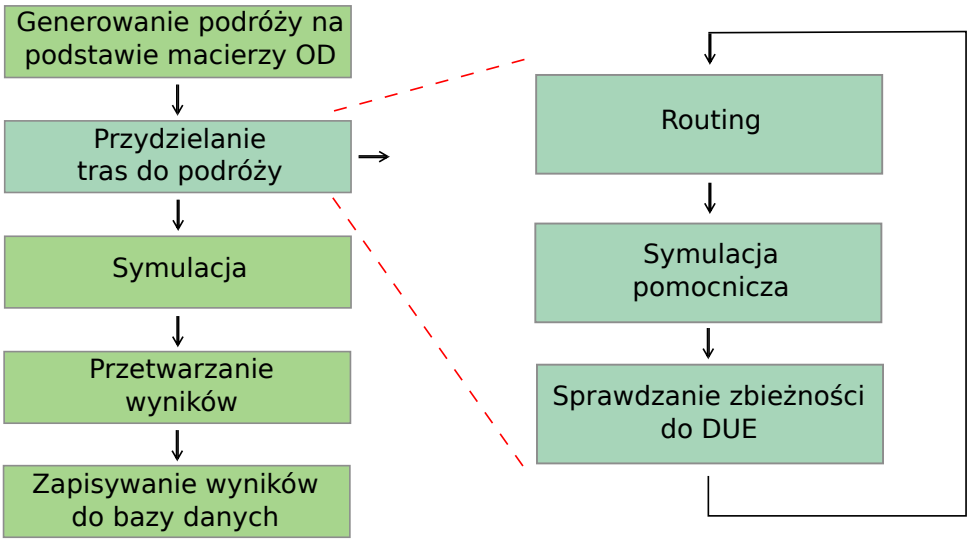
6. Zapisanie trasy do bazy

   Jeśli użytkownik systemu podejmie decyzję o wybraniu zaproponowanej trasy, ma on możliwość zapisania jej w bazie danych. Baza ta jest następnie wykorzystywana przez moduł KorMOD, w celu aktualizacji macierzy OD wykorzystywanej do przygotowywania symulacji. Oprócz informacji o początku, końcu oraz dacie przejazdu zapamiętywany jest również ciąg punktów reprezentujący trasę. Umożliwia to późniejsze powtórzenie wszystkich obliczeń w przypadku zmiany mapy, lub dostępności nowych wyników symulacji.

5.2. Symulacja ruchu drogowego

Cały pakiet *ProSym* skupia się przede wszystkim na symulacji ruchu drogowego, dlatego niniejszy podrozdział został w całości poświęcony temu zagadnieniu.

Do symulacji wykorzystywane jest oprogramowanie SUMO w wersji 0.15. Opis symulatora, wraz z zastosowanym w nim modelem, znajduje się w rozdziale drugim. Poniżej zostaną omówione zagadnienia związane z jego wykorzystaniem w celu modelowania zużycia paliwa i emisji spalin, na przykładzie oprogramowania *ProSym*.



Rys. 5.2.1. Schemat przeprowadzania symulacji w systemie *ProSym*

Pakiet *ProSym* wykorzystuje dobową symulację całości obszaru. Proces jej przygotowywania i przeprowadzania został przedstawiony na rysunku 5.2.1. Każdy etap zostanie poniżej omówiony.

1. Generowanie podróży na podstawie macierzy OD

   Jak zostało wcześniej wspomniane, macierz OD określa liczbę pojazdów danego typu, które przejadą pomiędzy dwoma dowolnymi rejonami transportowymi. W symulacjach mikroskopowych operuje się na pojedynczych pojazdach, które dysponują swoja indywidualną trasą przejazdu. Stąd konieczne jest wygenerowanie na podstawie macierzy OD miejsca rozpoczęcia i zakończenia oraz godziny podróży. Proces ten odbywa się z wykorzystaniem generatora liczb pseudolosowych. Należy zauważyć, że istnieje wiele typów macierzy OD: od tygodniowych, przez dobowe, godzin szczytu, do jednogodzinnych. Aby proces generowania macierzy dał poprawny rezultat, należy zdefiniować rozkład pojazdów w czasie doby. Precyzując, rozkład ten określa, jaka część liczby pojazdów określonej w macierzy OD rozpocznie swoją podróż o określonej godzinie. Dobre opracowanie na ten temat wraz z przykładami znajduje się w [ST96].

   Pojedyncza macierz opisuje jeden typ pojazdu. Powyższy proces należy przeprowadzić dla każdego typu z osobna.

2. Przydzielanie tras

   Poprzedni krok wykonywany jest jednorazowo. Jego wynik w uproszczeniu to lista czwórek postaci: początek, koniec, godzina podróży oraz typ pojazdu. Do symulacji mikroskopowej potrzeba jednak znaczenie więcej. Konieczne jest ustalenie dla każdego pojazdu trasy, będącej ciągiem krawędzi (dróg).

Najprostsze rozwiązanie, polegające na kierowaniu pojazdów najkrótszą drogą, nie prowadzi do dobrych rezultatów. Powstaje wiele korków, które nie mają w rzeczywistości miejsca. Spowodowane jest to tym, że najkrótsza trasa często przebiega przez drogi nieprzystosowane do dużego ruchu (np. drogi osiedlowe).

W rozdziale drugim opisano równowagę użytkownika, jako kluczowy element procesu przydzielania tras. W systemie *ProSym* wyznaczanie równowagi użytkownika odbywa się metodą iteracyjną. W każdej iteracji, dowolny pojazd porusza się najtańsza w danej chwili dostępną trasą. Przed pierwszą iteracją wagi krawędzi (koszty) w grafie sieci drogowej ustalane są proporcjonalnie do ich długości, niezależnie od momentu symulacji. Następnie uruchamiana jest symulacja pomocnicza, której wynikiem są czasy przejazdu na poszczególnych krawędziach zagregowane co 900 sekund. Na ich podstawie wyliczane są nowe wagi dla każdej chwili symulacji. Po aktualizacji wag następuje kolejna iteracja. Jako kryterium stopu przyjęto wartość sumarycznego czasu przejazdu wszystkich pojazdów. Jeśli różnica pomiędzy kolejnymi iteracjami będzie dostatecznie mała, proces kończy się, a otrzymany przydział tras zostaje uznany za dostatecznie bliski równowadze użytkownika.

3. Symulacja właściwa

   Do symulacji właściwej wykorzystywane są trasy przejazdu uzyskane w poprzednim kroku. W odróżnieniu od symulacji pomocniczej, gdzie oczekiwanym wynikiem są tylko czasy przejazdu danymi drogami, tu obliczane są wszystkie współczynniki dotyczące parametrów ruchu, zużycia paliwa oraz emisji spalin.

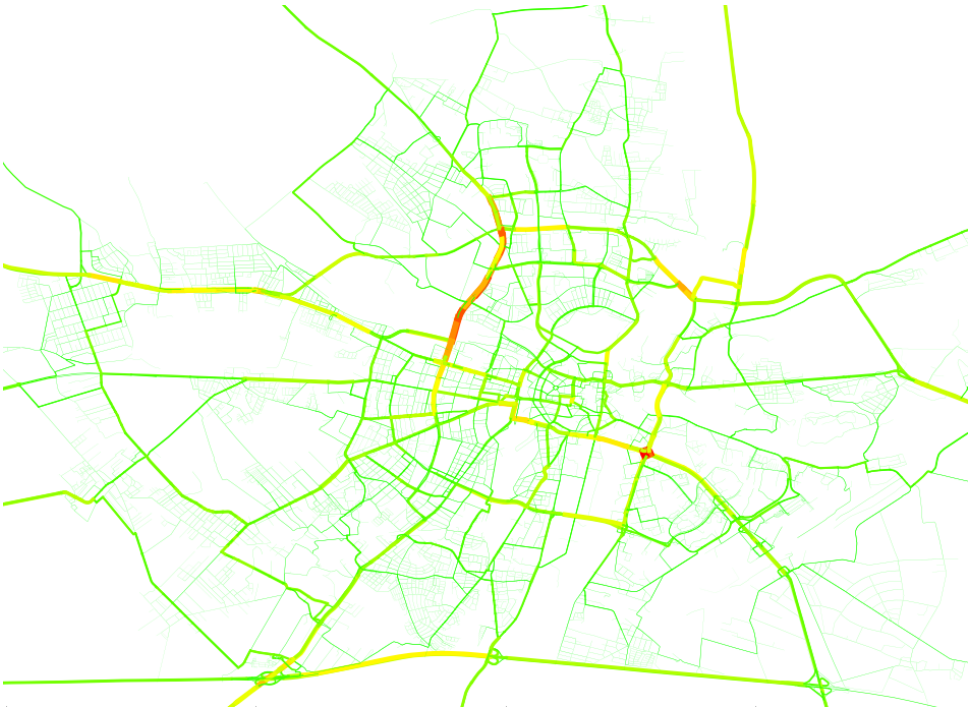
4. Przetwarzanie wyników i zapis do bazy danych

   Wyniki symulacji rozlokowane są w kliku plikach. W celu przygotowania ich do formatu obsługiwanego przez aplikację MaRuD, należy przetworzyć je, wyliczając niezbędne wartości, a następnie zapisać je do bazy danych.

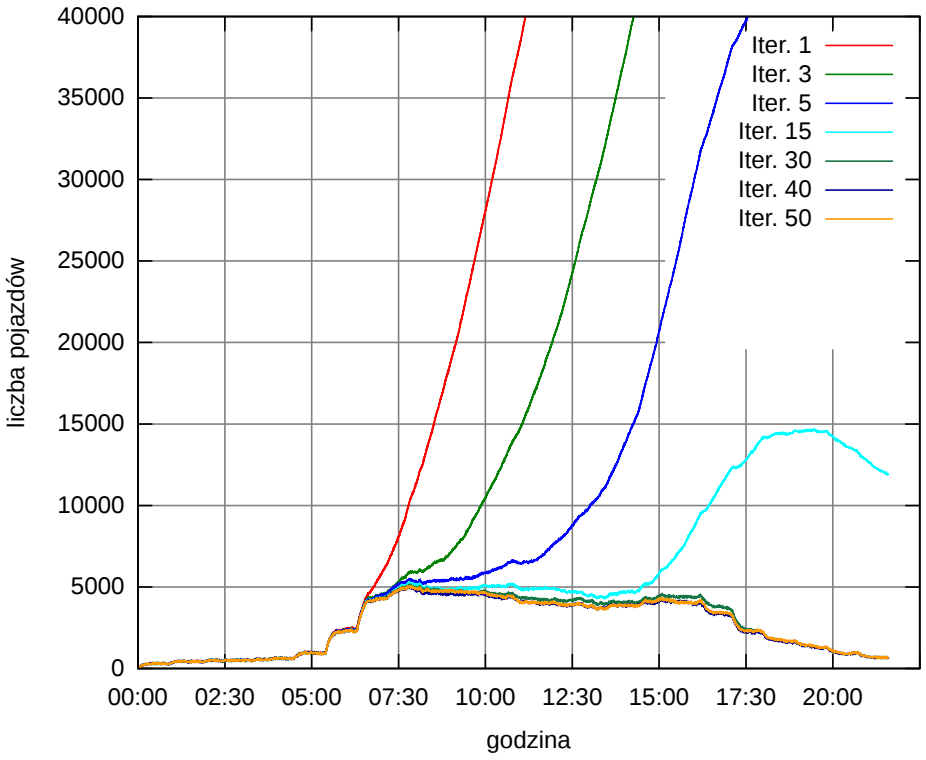
5.3. Ocena jakości wyników symulacji

Zaprezentowany powyżej schemat przygotowywania symulacji został przeprowadzony dla modelu ruchu miasta Poznań. Do obliczeń wykorzystano komputer wyposażony w procesor *Intel Core i7* oraz 8GB pamięci RAM. Obliczenia trwały 30 dni, podczas których udało się przeprowadzić symulację 50 iteracji algorytmu znajdywania dynamicznego punktu równowagi.

Otrzymane wyniki wiernie odwzorowują warunki drogowe panujące na ulicach Poznania. Ponadto natężenia ruchu na głównych drogach, są zgodne z wartościami wyliczonymi przez oprogramowanie VISUM, na oryginalnym modelu pochodzącym od Zarządu Dróg Miejskich (rysunek 5.3.1).



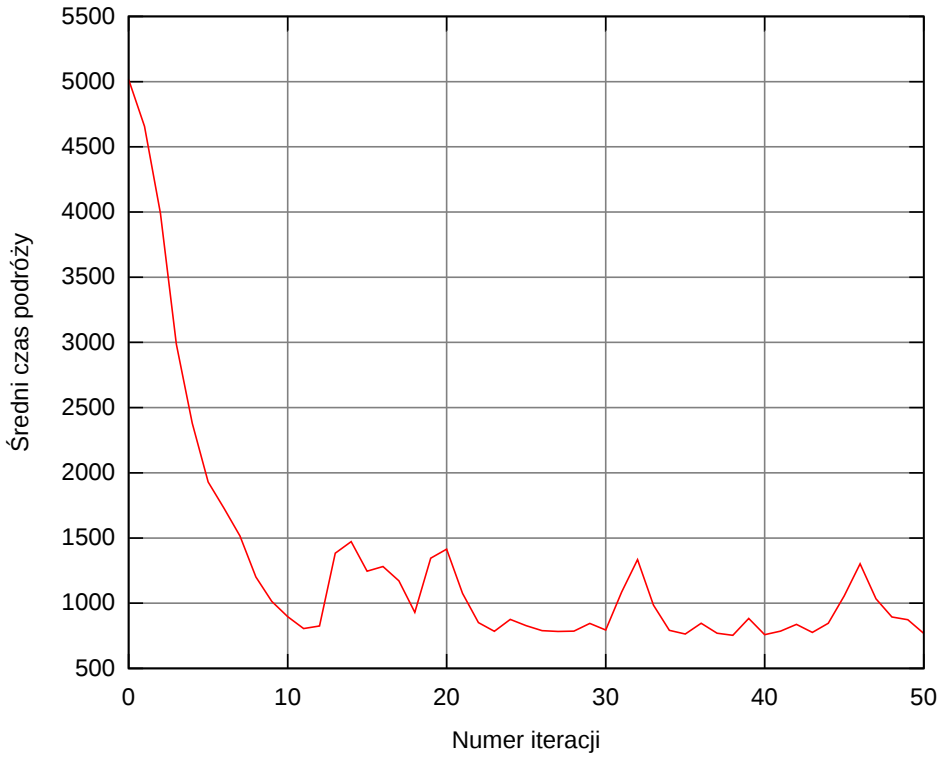
Rys. 5.3.1. Średni rozkład natężenia ruchu drogowego.



Rys. 5.3.2. Liczba pojazdów poruszających się w sieci drogowej w zależności od numeru iteracji procesu wyznaczania dynamicznej równowagi użytkownika.

W symulacji nie występuje zjawisko sztucznych korków. Rozkład liczby pojazdów poruszających się w sieci drogowej został przedstawiony na rysunku 5.3.2. Jak widać liczba ta początkowo rośnie monotonicznie, co świadczy o dużej liczbie korków, które

paraliżują ruch. Ostatnie iteracje niemalże nie różnią się między sobą, stąd wniosek, że otrzymane trasy przejazdu są bliskie punktowi równowagi.



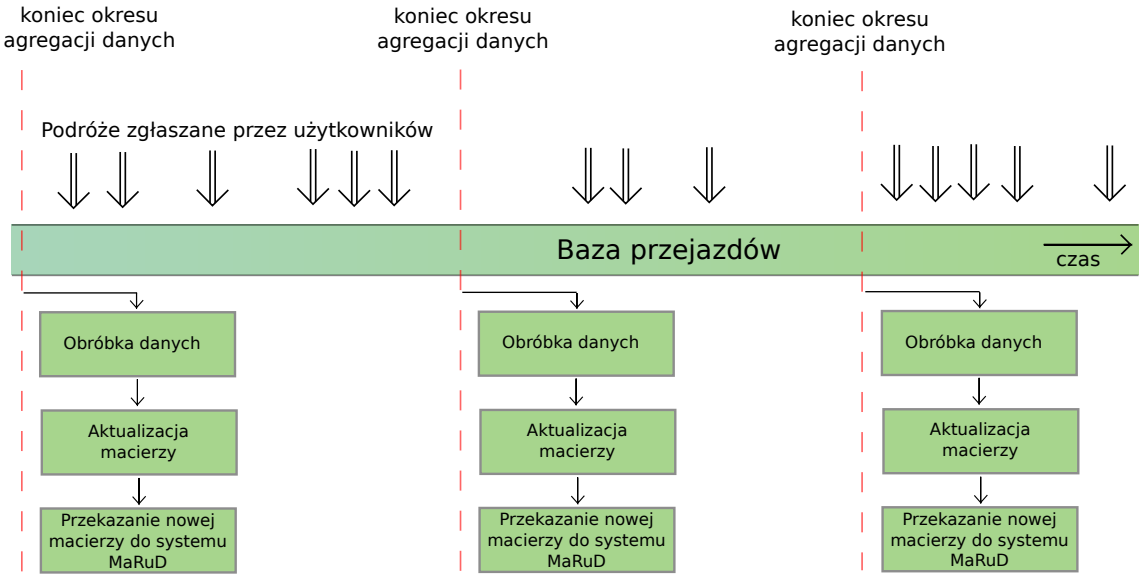
Rys. 5.3.3. Średni czas przejazdu pojazdów poruszających się w sieci drogowej w zależności od numeru iteracji procesu wyznaczania dynamicznej równowagi użytkownika.

Trzecim kryterium jakości, wynikającym niemal z definicji punktu równowagi użytkowników, jest średni czas przejazdu wszystkich pojazdów. Jego wartość w zależności od numeru iteracji przedstawiona jest na rysunku 5.3.3. Jak widać po 50 iteracjach wartość ta ustabilizowała się, co świadczy o jakości otrzymanego przydziału tras.

Rozdział 6

Korektor Macierzy OD – KorMOD

System KorMOD, stworzony w ramach tej pracy, odpowiedzialny jest za aktualność modelu wykorzystywanego do symulacji, a co za tym idzie za jakość jej wyników. Jego zasada działania opiera się na innowacyjnej idei zbierania informacji od użytkowników – kierowców, za pomocą internetu. W związku z rozwojem systemów wspomagania wyboru trasy oraz znaczącym wzrostem ich użycia, celowym zdaje się wykorzystanie danych pozyskanych w ten sposób. Aplikacja MaRuD dostępna zarówno na komputery jak i urządzena mobilne, wspomaga proces zbierania danych. KorMOD odpowiedzialny jest za ich analizę oraz aktualizację wykorzystywanego modelu symulacji.



Rys. 6.1.1. Schemat działania systemu KorMOD.

6.1. Zasada działania

Schemat działania systemu KorMOD, przedstawiony został na rysunku 6.1.1. Dane są nieustannie gromadzone w bazie przejazdów. Gdy ich ilość jest wystarczająca do przeprowadzenia procesu aktualizacji, następuje agregacja danych. Uzyskane informacje przekazywane są do wbudowanego w Korektor Macierzy OD sterownika rozmytego, który na ich podstawie dokonuje aktualizacji obecnej macierzy OD. Otrzymana w ten sposób macierz przekazywana jest do systemu MaRuD, który od tej pory będzie ją wykorzystywał we wszystkich przeprowadzanych symulacjach.

6.2. Mechanizm aktualizacji macierzy OD

W tym podrozdziale zostanie dokładnie opisany sposób, w jaki KorMOD dokonuje aktualizacji macierzy OD. Jak zostało to wcześniej opisane, głównym komponentem są sterowniki rozmyte KorMOD^I oraz KorMOD^{II}. Pozostała cześć systemu KorMOD odpowiedzialna jest za przygotowanie zagregowanych i znormalizowanych danych wejściowych, oraz przetworzenie informacji wyjściowych ze sterownika.

Ogólny algorytm aktualizacji zostanie omówiony jako pierwszy, dzięki czemu łatwiejsze będzie zrozumienie znaczenia poszczególnych wejść i wyjść sterownika rozmytego. Następnie zostaną omówione wszystkie wejścia wraz z metodą obliczenia ich wartości oraz same sterowniki rozmyte wykorzystywane w systemie.

6.2.1. Algorytm aktualizacji

Aktualizacja przebiega w trzech etapach. Podczas pierwszego etapu dla każdej pary rejonów transportowych, na podstawie zagregowanych danych, wyliczana jest para wartości (względna_zmiana, ufność). Druga faza natomiast przyjmuje jako swoje wejście wynik działania pierwszej fazy z obecnego i poprzedniego wykonania. Następnie, w oparciu o informacje uzyskane w drugiej fazie, przeprowadzona zostaje aktualizacja macierzy OD. Obie fazy oraz proces aktualizacji zostaną opisane poniżej.

Faza pierwsza realizowana jest przez następujący algorytm.

Wejście: macierz OD $M = (m_{i,j})$, zagregowane dane z bazy przejazdów
Wyjście: macierz $R^I = (r_{i,j})$, par postaci (względna_zmiana, ufność).

$R^I \leftarrow \theta$
for all wiersz i w macierzy M **do**
  for all kolumna j w macierzy M **do**
    $\vec{p} \leftarrow$ wektor parametrów dla rejonu $t_{i,j}$
    $r_{i,j} \leftarrow \text{KorMOD}^I(\vec{p})$

end for
end for
return R^I

Wylicza on dla każdej pary rejonów transportowych w macierzy OD wektor parametrów wymagany przez sterownik rozmyty KorMOD^I. Następnie uruchamia go, wynik jego działania zapisując w macierzy wynikowej.

Jak łatwo zauważyć algorytm ten jest bardzo prosty, a jedyne znaczące operacje wykonywane są podczas obliczania wartości wejściowych oraz przez sam sterownik. Zostaną one dokładnie opisane w dalszym toku pracy.

Faza druga, przebiega analogicznie jak pierwsza, z tą różnicą, że wektor parametrów generowany jest na podstawie macierzy R_b^I oraz R_p^I. Proces obliczania wektora parametrów oraz sterownik fazy drugiej zostaną opisane w następnych podrozdziałach.

Wejście: macierz $R_b^I = (r_{i,j}^b)$, będąca wynikiem pierwszej fazy bieżącej aktualizacji, macierz $R_p^I = (r_{i,j}^p)$, będąca wynikiem pierwszej fazy poprzedniej aktualizacji
Wyjście: macierz $R^{II} = (r_{i,j})$, par postaci (względna_zmiana, ufność).
$R^{II} \leftarrow \theta$
for all wiersz i w macierzy M **do**
for all kolumna j w macierzy M **do**
$\vec{p} \leftarrow$ wektor parametrów dla rejonu $t_{i,j}$
$r_{i,j} \leftarrow$ KorMOD$^{II}(\vec{p})$
end for
end for
return R^{II}

Główną zaletą rozgraniczenia procesu aktualizacji na dwie fazy, jest możliwość wykorzystania w pierwszej fazie uśrednionych wartości, dzięki czemu wyniki zostają uniezależnione od chwilowych wahań. Z kolei druga faza umożliwia analizę danych historycznych, dzięki czemu możliwe jest uwzględnienie ogólnego trendu zmian.

Nie bez znaczenia jest również wykorzystanie danych o ufności informacji niesionej przez wynik fazy pierwszej. Dzięki temu w fazie drugiej, możliwe jest wcześniejsza reakcja na trend, który podczas poprzedniej aktualizacji nie był wystarczająco potwierdzony.

Dane o względnej zmianie wartości macierzy OD wykorzystywane są następnie do obliczenia nowych wartości i stworzenia zaktualizowanej macierzy OD. Macierz wynikowa fazy drugiej zawiera informację o względniej zmianie wartości oraz ufności tej informacji w danej komórce macierzy. Niech $M = (m_{i,j})$ oznacza pierwotną macierz OD, a $R = ((w_{i,j}, u_{i,j}))$ będzie macierzą wynikową drugiej fazy. Wtedy

nowa macierz OD M' wyliczana jest jako:

$$m'_{i,j} = C \cdot m_{i,j} \cdot (1 + w_{i,j}) \cdot u_{i,j}^{\alpha} ,$$

gdzie α jest stałą, której wpływ zostanie omówiony wraz z wynikami badań nad skutecznością modułu KodMOD.

6.2.2. Faza I

6.2.2.1. Dane wejściowe sterownika rozmytego $\mathbf{KorMOD}^I$

W celu zdefiniowania danych wejściowych do sterownika KorMOD^I, niezbędne jest wprowadzenie kilku pojęć. Mapę sieci drogowej będziemy teraz rozpatrywać, odmiennie niż poprzednio, jako zbiór n rejonów transportowych $T = \{t_i : 1 \leq i \leq n\}$. Zbiór $S_1(t_i) = \{t_j \in T : t_i \text{ oraz } t_j \text{ sąsiadują ze sobą}\}$ oznacza sąsiedztwo rejonu t_i. Ponadto niech

$$S_2(t_i) = \{t_j \in T : t_j \neq t_i \text{ oraz } S_1(t_i) \cap S_1(t_j) \neq \emptyset\}$$

będzie oznaczało dalsze sąsiedztwo. Należy zwrócić uwagę na to, że $t_i \notin S_1(t_i)$, $t_i \notin S_2(t_i)$ oraz $S_1(t_i) \subset S_2(t_i)$.

Niech M oraz R będą, odpowiednio, bieżącą macierzą OD oraz macierzą OD powstałą na podstawie podróży zapisanych w bazie MaRuD. Wtedy $m_{i,j}$ (odpowiednio $r_{i,j}$) określa liczbę pojazdów, które przejechały z rejonu t_i do rejonu t_j. Wyjaśnienie niektórych wprowadzonych tu pojęć znajduje się na rysunku 6.2.1.

Macierz R określana jest na podstawie danych w bazie KorMOD w następujący sposób. Dla każdej podróży pobierana jest lokalizacja jej początku i końca. Następnie wyszukiwane są rejony transportowe zawierające oba punkty – t_i oraz t_j, a odpowiadająca im wartość $r_{i,j}$ zwiększana jest o jeden.

Dla ustalonych i oraz j sterownik KorMOD^I przyjmuje następujące parametry:

- współczynnik łącznej ilości w,

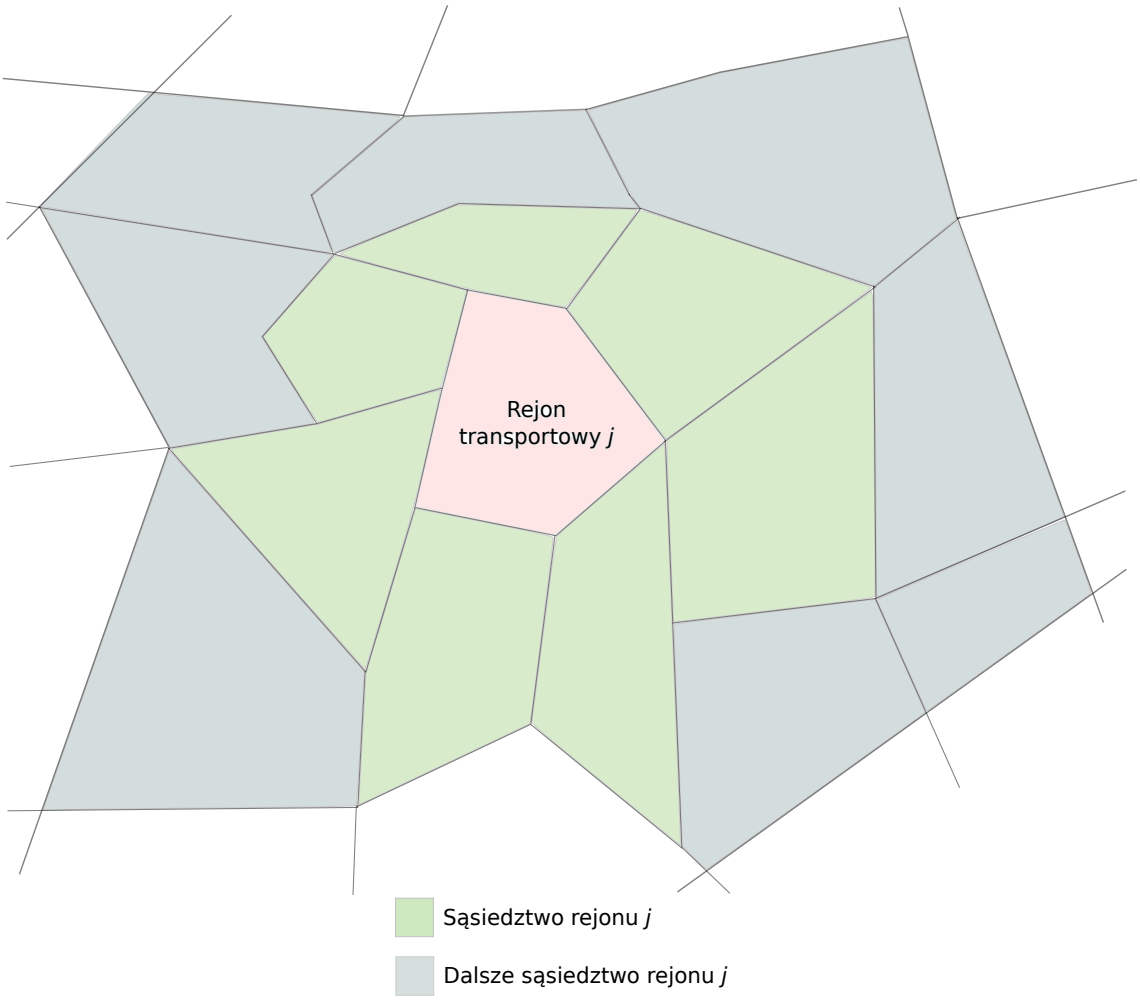
  Jest obliczany jako

  $$w = \frac{\sum_{k=1}^{n} \sum_{l=1}^{n} r_{k,l}}{\sum_{k=1}^{n} \sum_{l=1}^{n} m_{k,l}} ,$$

  wyraża stosunek ilości podróży pochodzących od użytkowników, względem tych z bieżącej macierzy OD.

- współczynnik komórki $k_{i,j}$,

  Jest obliczany jako $k_{i,j} = \frac{r_{i,j}}{w \cdot m_{i,j}}$, wyraża względna zmianę wartości danej komórki macierzy OD.



Rys. 6.2.1. Rejony transportowe oraz ich sąsiedztwa na mapie.

- współczynnik odwróconej komórki $k'_{i,j}$,

  Podobny do współczynnika komórki z tą różnicą, że uwzględnia podróże w przeciwnym kierunku. $k'_{i,j} = \frac{r_{j,i}}{w \cdot m_{j,i}}$

- współczynnik sąsiedztwa $s_{i,j}$,

$$s_{i,j} = \frac{\sum\limits_{t_k \in S_1(t_i)} \sum\limits_{t_l \in S_1(t_j)} r_{k,l}}{w \cdot \sum\limits_{t_k \in S_1(t_i)} \sum\limits_{t_l \in S_1(t_j)} m_{k,l}}$$

  Wyraża uśrednioną względną zmianę wartości całego sąsiedztwa rejonu t_i względem sąsiedztwa rejonu t_j

- współczynnik odwróconego sąsiedztwa $s'_{i,j}$,

$$s'_{i,j} = \frac{\sum\limits_{t_k \in S_1(t_i)} \sum\limits_{t_l \in S_1(t_j)} r_{l,k}}{w \cdot \sum\limits_{t_k \in S_1(t_i)} \sum\limits_{t_l \in S_1(t_j)} m_{l,k}}$$

Analogicznie jak współczynnik sąsiedztwa, lecz zlicza podróże w przeciwnym kierunku.

- współczynnik dalszego sąsiedztwa $d_{i,j}$,

$$d_{i,j} = \frac{\sum\limits_{t_k \in S_2(t_i)} \sum\limits_{t_l \in S_2(t_j)} r_{k,l}}{w \cdot \sum\limits_{t_k \in S_2(t_i)} \sum\limits_{t_l \in S_2(t_j)} m_{k,l}}$$

Wyraża uśrednioną względną zmianę wartości całego dalszego sąsiedztwa rejonu t_i względem dalszego sąsiedztwa rejonu t_j

- współczynnik odwróconego dalszego sąsiedztwa $d'_{i,j}$,

$$d'_{i,j} = \frac{\sum\limits_{t_k \in S_2(t_i)} \sum\limits_{t_l \in S_2(t_j)} r_{l,k}}{w \cdot \sum\limits_{t_k \in S_2(t_i)} \sum\limits_{t_l \in S_2(t_j)} m_{l,k}}$$

Analogicznie jak współczynnik dalszego sąsiedztwa, lecz zlicza podróże w przeciwnym kierunku.

6.2.2.2. Sterownik rozmyty KorMOD^I

Sterownik fazy pierwszej uwzględnia siedem opisanych powyżej parametrów wejściowych. Na ich podstawie wylicza względną zmianę oraz współczynnik zaufania. Poniżej zostaną zdefiniowane wykorzystane w nim zmienne oraz zestaw reguł. Kod źródłowy sterownika w języku FCL znajduje się w dodatku C.

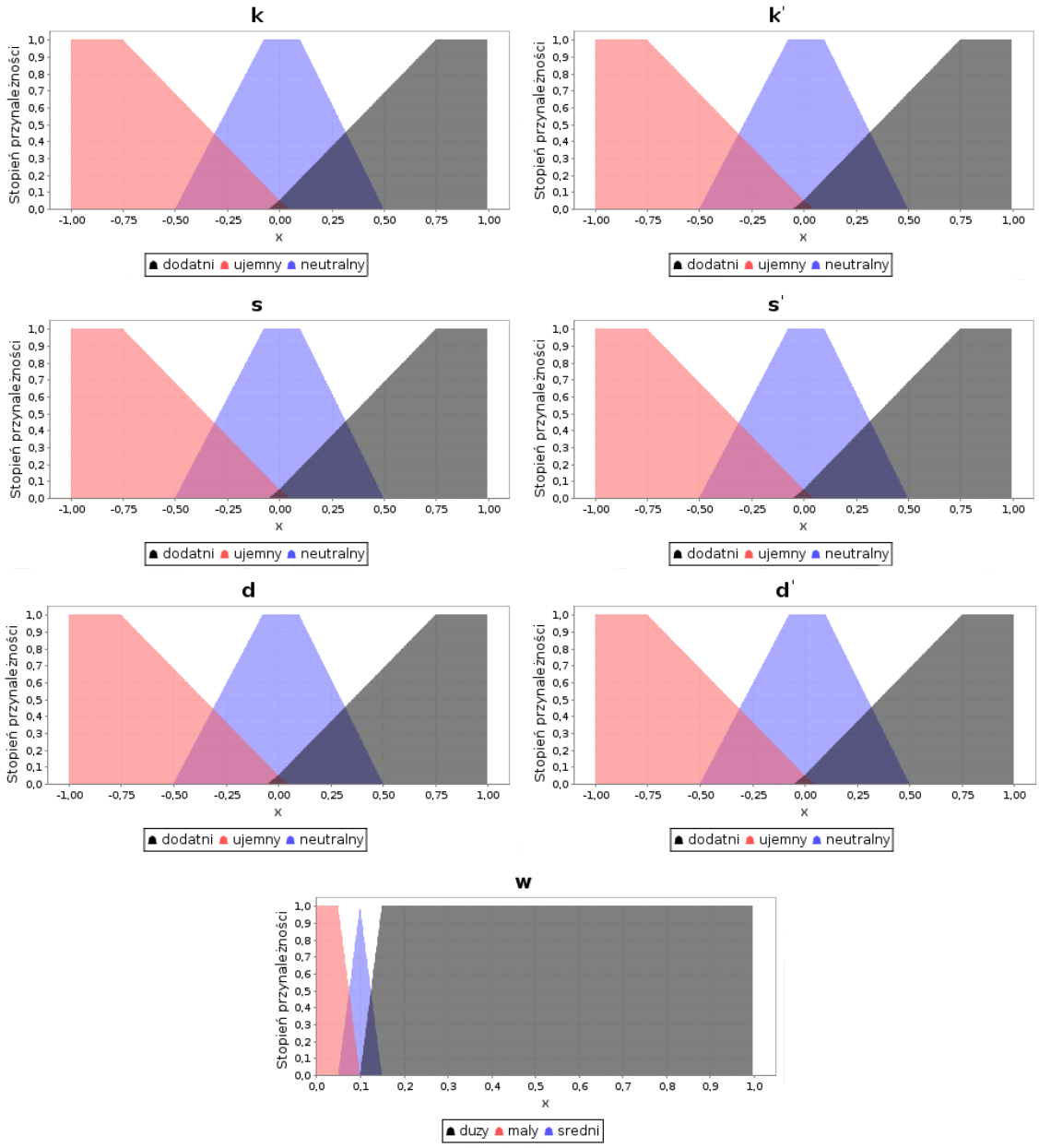
Wartości każdego z parametrów wejściowych zostały opisane przez trzy zbiory rozmyte: *dodatni*, *neutralny* i *ujemny*. Wyjątek stanowi zmienna w opisana przez zbiory: *mały*, *średni* i *duży*. Kształt tych zbiorów został przedstawiony na rysunku 6.2.2.

Wyjście sterownika stanowią dwie zmienne. Wartość względnej zmiany opisana jest przez zbiory: *dodatni*, *neutralny* oraz *ujemny*. Ufność, natomiast przez: *mała*, *średnia* i *duża*. Kształt tych zbiorów zaprezentowany jest na rysunku 6.2.3

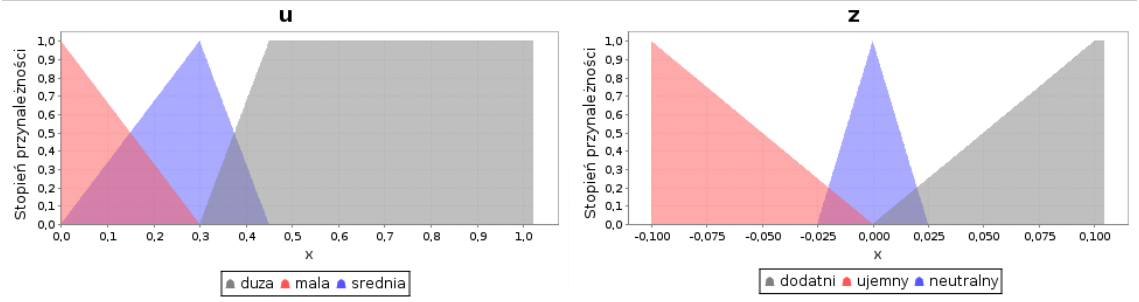
Pierwsza reguła uwzględnia współczynnik łącznej ilości w. Im jest on większy tym liczba podróży zgłoszonych w systemie MaRuD bliższa jest ilości pojazdów poruszających się po danym obszarze. Dlatego też wartość w bezpośrednio wpływa na współczynnik ufności. Odpowiednia reguła przyjmuje postać:

```
IF w = duże THEN ufność = duża.
```

Kolejne reguły odzwierciedlają intuicję mówiącą, że im obszar, którego dotyczy zmiana jest większy, tym mniejsza szansa na to, że zmiana jest tylko chwilowym zaburzeniem. Reguły odpowiadające za te zachowanie mają następującą postać:



Rys. 6.2.2. Zbiory rozmyte opisujące parametry wejściowe sterownika KorMOD^I.



Rys. 6.2.3. Zbiory rozmyte opisujące zmienne wyjściowe sterownika KorMOD^I.

```
IF k = dodatnie
    THEN zmiana = dodatnia, ufność = mała,
IF k = dodatnie AND s = dodatnie
    THEN zmiana = dodatnia, ufność = średnia,
IF k = dodatnie AND s = dodatnie AND s = dodatnie
    THEN zmiana = dodatnia, ufność = duża.
```

Kolejna grupa reguł stara się wykryć sprzeczne informacje, powstałe na wskutek chwilowych wahań:

```
IF k = dodatnie AND s = ujemne
    THEN zmiana = neutralna,
IF k = dodatnie AND s = ujemne AND k = ujemne
    THEN zmiana = neutralna, ufność = duża.
```

Ostatnia grupa reguł, korzysta ze współczynników odwróconych, czyli tych dotyczących podróży w przeciwnym kierunku. Ponieważ KorMOD obsługuje macierze dobowe, zatem w przypadku, w którym wartości w obu kierunkach wykazują przeciwne wartości zmian, rośnie ryzyko zarejestrowania chwilowych wahań. Dlatego też, w takiej sytuacji ufność zostaje obniżona:

```
IF k = dodatnie AND ( k' = ujemne OR s' = ujemne OR d' = ujemne)
       THEN ufność = mała.
```

Zbiór reguł zawiera również analogiczne reguły dla ujemnych i neutralnych wartości parametrów.

6.2.3. Faza II

6.2.3.1. Dane wejściowe sterownika rozmytego KorMOD^{II}

Jak już zostało to opisane wcześniej, wejście sterownika KorMOD^{II} stanową wyniki działania bieżącego oraz poprzedniego wykonania pierwszej fazy.

Dla ustalonych i oraz j sterownik KorMOD^{II} uwzględnia następujące parametry:

- względna zmiana wartości z poprzedniego wykonania pierwszej fazy w_p,
- ufność z poprzedniego wykonania pierwszej fazy u_b,
- względna zmiana wartości z bieżącego wykonania pierwszej fazy w_p,
- ufność z bieżącego wykonania pierwszej fazy u_b.

6.2.3.2. Sterownik rozmyty KorMOD^{II}

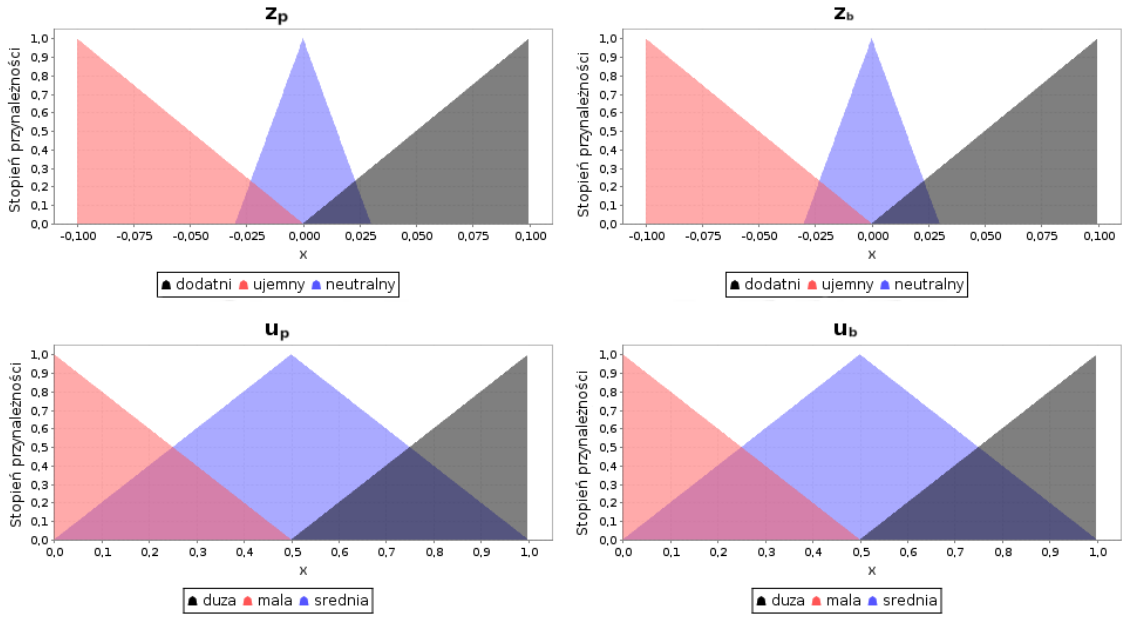
Sterownik rozmyty drugiej fazy przyjmuje cztery opisane powyżej parametry wejściowe. Na ich podstawie wylicza względną zmianę oraz współczynnik zaufania do tej informacji. Kod sterownika w języku FCL znajduje się w dodatku C.

Parametry wejściowe reprezentowane są przez zbiory rozmyte *dodatni*, *neutralny* i *ujemny* w przypadku względnej zmiany oraz *mała*, *średnia* i *duża* dla ufności. Kształt tych zbiorów został przedstawiony na rysunku 6.2.4. Definicja wyjścia sterownika jest identyczna jak w przypadku pierwszej fazy (rysunek 6.2.3).

Pierwsza grupa reguł odpowiada za utrzymanie trendu zmiany. Jeśli bieżąca zmiana jest zgodna ze zmianą poprzednią, to macierz zostanie zgodnie z nią zaktualizowana.

```
IF zb = dodatni AND zp = dodatni
            THEN z = dodatni
IF zb = ujemny AND zp = ujemny
            THEN z = ujemny
IF zb = neutralny AND zp = neutralny
            THEN z = neutralny
```

Druga grupa odpowiada za określenie ufności. Jeśli wyniki bieżącej i poprzedniej pierwszej fazy są pewne to współczynnik ufności jest duży. Natomiast jeśli dowolna z wartości ufności u_b lub u_p jest mała, to i obecna ufność powinna być mała.



Rys. 6.2.4. Zbiory rozmyte opisujące wartości wejściowe sterownika KorMOD^{II}.

```
IF ub = duza AND up = duza
            THEN u = duza;
IF ub = srednia AND up = srednia
            THEN u = srednia;

IF ub = mala OR up = mala
            THEN u = mala, z = neutralny;
```

Ostatnia grupa reguł odpowiada za wykrywanie niezgodności zmian poprzedniego i bieżącego wykonania pierwszej fazy. Jeśli zmiana w bieżącej pierwszej fazie jest dodatnia, a poprzednio była ujemna to istnieje duża szansa na to, że jest to wynik chwilowych zaburzeń, dlatego też w tej sytuacji ostateczna zmiana powinna być neutralna, a ufność mała. Analogicznie dla ujemnej wartości zmiany.

```
IF zb = dodatni AND zp = neutralny
            THEN z IS neutralny;
IF zb = dodatni AND zp = ujemny
            THEN z = neutralny, u = mala;
```

6.3. Ocena jakości otrzymanych macierzy

W celu oceny sprawności modułu KorMOD w aktualizowaniu macierzy OD, dokonano ewaluacji systemu na danych pochodzących od Zarządu Dróg Miejskich w Poznaniu. Podstawowymi kryteriami oceny była sama zbieżność, jak również jej szybkość w różnych warunkach.

6.3.1. Metodyka oceny

Eksperyment polegał na symulowaniu zmian zachodzących w popycie transportowym. Dysponując dwoma modelami ruchu A oraz B, powstałymi w różnych okresach czasu, można wygenerować dane, które w pewien sposób przybliżać będą trasy zgłoszone przez użytkowników systemu *ProSym* w tym okresie.

Metoda konstruowania zbioru testowego X opierała się na założeniu, że w dowolnym dniu d pomiędzy datą d_0 powstania modelu A, a datą d_k powstania modelu B, obecny popyt transportowy można w przybliżeniu opisać w następujący sposób:

$$X_d = \frac{d - d_0}{d_k - d_0} \cdot A + \frac{d_k - d}{d_k - d_0} \cdot B,$$

czyli jako kombinację liniową modeli A oraz B.

Wiadomo, że dysponując bazą wszystkich podróży, wyliczenie obecnej macierzy OD byłoby zadaniem trywialnym. Dlatego też, podczas generowania danych

testowych uwzględniany jest parametr s określający, jaki procent ruchu z modelu przybliżonego zostanie wykorzystany jako dane wejściowe systemu *ProSym*.

Sam proces konstruowania listy podróży dla dnia d można zapisać w następujący sposób:

Wejście: A – macierz OD z dnia d_0 oraz B – macierz OD z dnia d_k.
Wyjście: lista podróży w postaci trójek (źródło, cel, czas rozpoczęcia)

$L \leftarrow$ pusta lista
for all para rejonów transportowych (t_i, t_j) **do**
  if $d \leq d_0$ **then**
    dodaj $a_{i,j}$ wystąpień pary (t_i, t_j) do L
  else if $d \leq d_k$ **then**
    dodaj $\frac{d-d_0}{d_k-d_0} \cdot a_{i,j} + \frac{d_k-d}{d_k-d_0} \cdot b_{i,j}$ wystąpień pary (t_i, t_j) do L do L
  else
    dodaj $b_{i,j}$ wystąpień pary (t_i, t_j) do L
  end if
end for
$D \leftarrow$ losuj $\alpha \cdot$ długość(L) elementów z listy L
$R \leftarrow$ pusta lista
for all para $(t_i, t_j) \in D$ **do**
  $g \leftarrow$ losuj godzinę rozpoczęcia podróży zgodnie z dobowym rozkładem liczby pojazdów
  dodaj do R trójkę (t_i, t_j, g)
end for
return R

Dni w okresie pomiędzy d_0 oraz d_k modelowane są jako kombinacja liniowa modeli A oraz B. Natomiast dni przed d_0 (lub po d_k) tylko na podstawie modelu A (odpowiednio B). Dzięki temu można sprawdzić, jak system KorMOD radzi sobie ze zmianami w macierzy, oraz po jakim czasie nowa macierz stabilizuje się.

Proces oceny polega na generowaniu nowych danych wejściowych zgodnie z powyższym schematem, które następnie przekazywane są do komponentu KorMOD. Dokonuje on aktualizacji macierzy. Otrzymane macierze są następnie porównywane między sobą, jak i z oczekiwaną macierzą modelu B. Za kryterium zgodności macierzy przyjęto sumę kwadratów różnic:

$$\text{zgodność}(A, B) = \sum_{i=1}^{N} \sum_{j=0}^{N} (a_{i,j} - b_{i,j})^2$$

Wartość ta jest jednak trudna w interpretacji, dlatego w celu prezentacji wyników konieczne było jej znormalizowanie. Jako punkt odniesienia przyjęto sumę kwadratów odchyleń macierzy powstałej przez wprowadzenie losowych zaburzeń (szumów) o różnym stopniu intensywności. Otrzymane wartości poddane zostały regresji li-

niowej, w rezultacie utrzymano następujące równanie:

$$\text{szum}(A, B) = 9.51876 \cdot 10^{-5} \cdot \text{zgodność}(A, B) + 0.0002,$$

przy współczynniku R^2 równym 0.9999. Otrzymany współczynnik nazywany będzie poziomem szumu. Interpretacja jego wartości jest następująca: jeśli wartość $\text{szum}(A, B) = a$, to macierze A oraz B różnią się od siebie tak, jak macierz zaburzona losowo w stopniu a względem swojej pierwotnej postaci. W dalszym toku pracy, gdzie to będzie możliwe do porównywania macierzy będzie wykorzystywany poziom szumu.

W celu ułatwienia oceny, zaimplementowano również prostą metodę aktualizacji, która w każdym kroku zmienia macierz OD tak, aby odzwierciedlała ona dokładnie dane z bazy systemu KorMOD. Korzystając z poprzednio wprowadzonej notacji aktualizacja przebiega zgodnie ze wzorem:

$$m'_{i,j} = \beta \cdot m_{i,j} \cdot (1 + k_{i,j}),$$

gdzie $k_{i,j}$ oznacza wcześniej zdefiniowany współczynnik komórki, a β współczynnik uczenia omówiony w dalszym toku pracy, wraz ze stałą α. Metoda ta mimo swojej prostoty daje zaskakująco dobre wyniki.

6.3.2. Scenariusze testowe

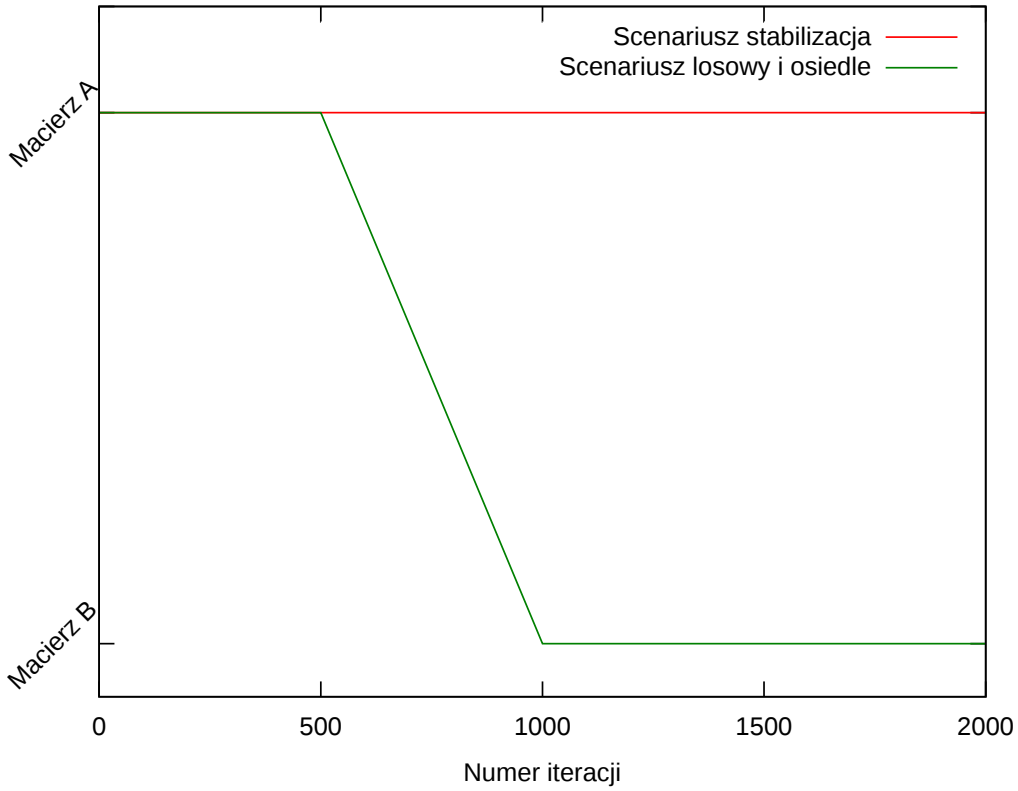
Badania przeprowadzono z wykorzystaniem rzeczywistych modeli ruchu miasta Poznań. Testy zostały podzielone na trzy scenariusze: „stabilizacja", „losowy" oraz „osiedle".

Pierwszy z nich miał na celu sprawdzenie stabilności sterownika KorMOD. Sprawdzono jak sterownik zachowuje się w warunkach braku zmian w macierzy OD, oraz jaki jest poziom szumu przez niego wprowadzany. Znaczna część testów wykonanych zgodnie z tym scenariuszem posłużyła do kalibracji obu metod względem siebie. Testy z wykorzystaniem drugiego scenariusza pozwoliły zbadać zdolność modułu KorMOD do odwzorowywania losowych zmian w macierzy. Trzecia grupa testów obejmowała symulowane zmiany w strukturze popytu transportowego polegające na wzroście lub spadku zainteresowania kierowców pewnymi obszarami miasta.

Ze względu na ograniczenia obliczeniowe we wszystkich testach jako macierz bazową, wykorzystano model ruchu pojazdów osobowych z 2005 roku. Ponadto przyjęto założenie, że jedna iteracja to jeden dzień rzeczywistej pracy systemu.

6.3.3. Wyniki testów

Zaprezentowane poniżej wyniki, zostały uzyskane korzystając z komputerów Studenckich Laboratoriów Komputerowych Wydziału Matematyki i Informatyki UAM.



Rys. 6.3.1. Schematy scenariuszy testowych.

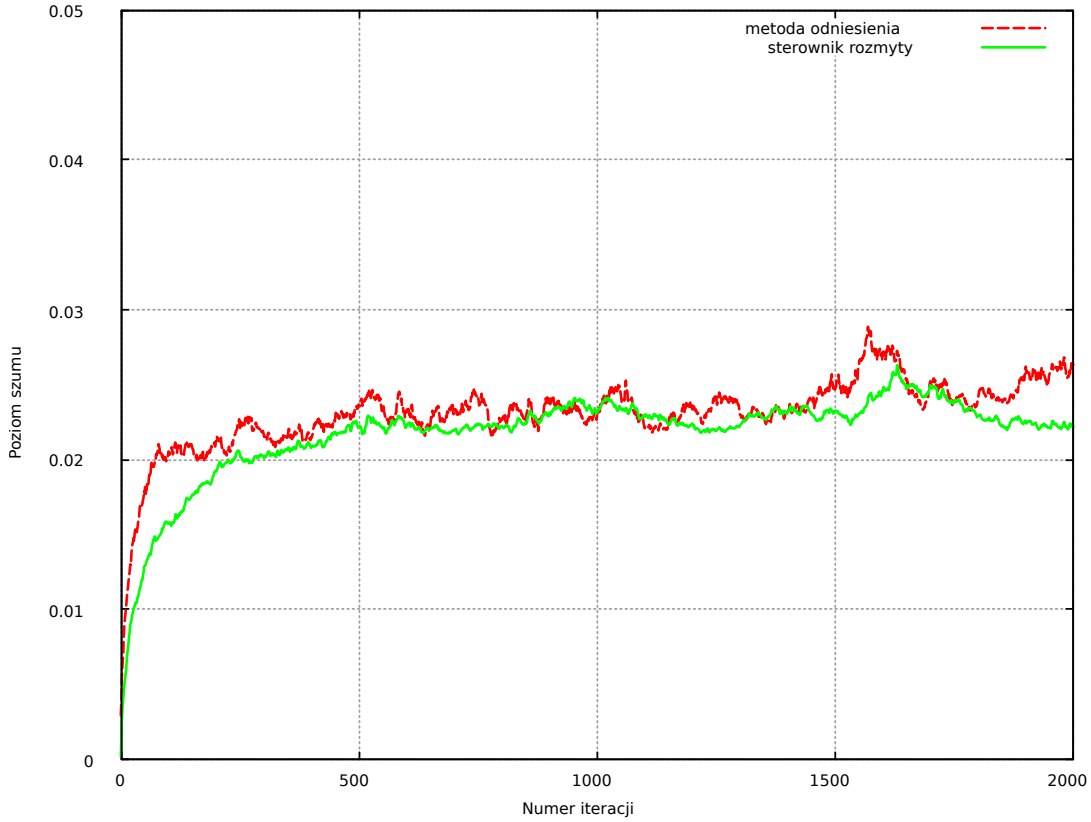
W obliczeniach brało udział 56 maszyn (procesor Intel Core i3, 4GB pamięci RAM). Łącznie z próbami kalibracyjnymi przeprowadzono symulację ponad 500 tysięcy iteracji, co przy uwzględnieniu średniego czasu trwania jednej iteracji wynoszącego 44 sekundy, daje około 250 dni obliczeń na pojedynczym komputerze.

6.3.3.1. Kalibracja

Poniższa sekcja zostanie poświęcona doborowi parametrów α i β. Ich znaczenie jest zbliżone – oba wpływają na zbieżność (i jej szybkość) metody aktualizacji macierzy. Przy nieodpowiednim doborze parametrów obie metody stają się rozbieżne lub generują bardzo duży poziom szumu. Z testów wynika, że zarówno wartości $\alpha < 1.0$, jak i $\beta > 0.2$ dyskwalifikują obie metody ze względu na poziom szumu znacznie przekraczający 100%.

Kalibracja metody odniesienia i sterownika rozmytego ma na celu umożliwienie porównania ich ze sobą. Jako kryterium przyjęto poziom szumu generowany w optymalnych warunkach tj. gdy ilość użytkowników systemu *ProSym* jest znacząca i wynosi 10% mieszkańców miasta ($s = 0.1$).

Zadowalający poziom szumu, wynoszący średnio $2.3 - 2.4\%$ osiągany jest dla wartości $\alpha = 2.25$ oraz $\beta = 0.0085$. Wykres dokładnego przebiegu testu przy tych wartościach parametrów znajduje się na rysunku 6.3.2.



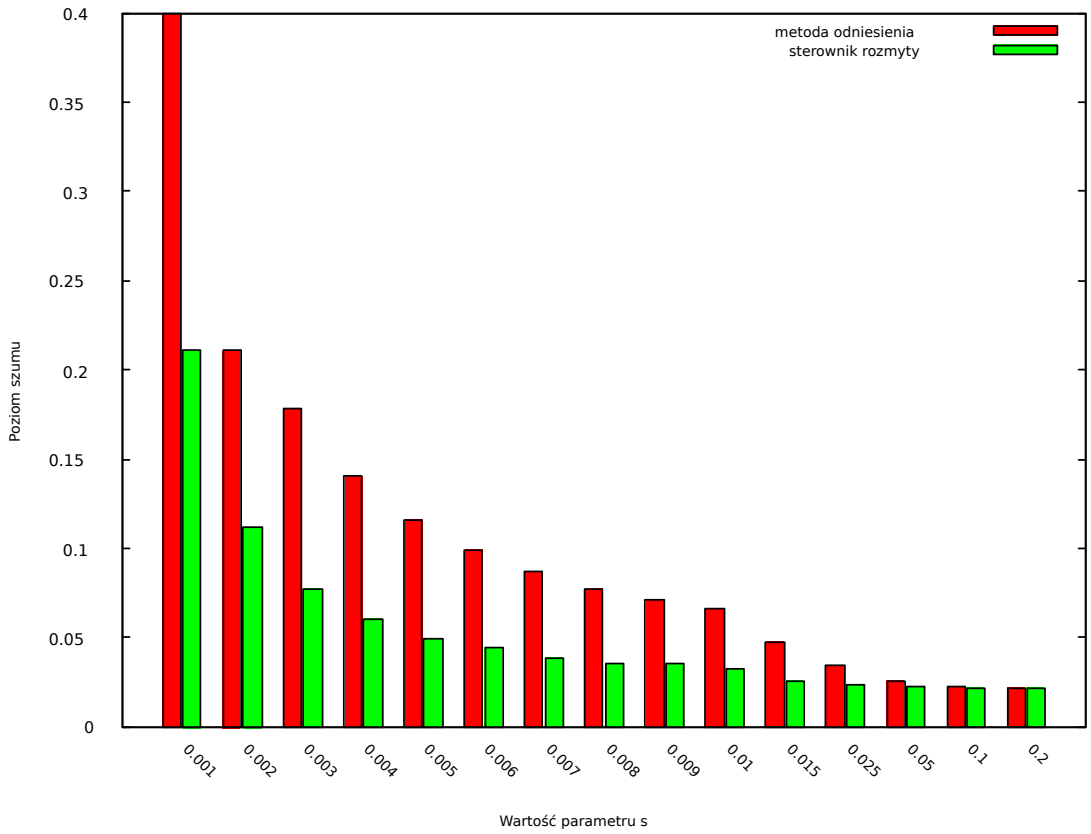
Rys. 6.3.2. Poziom szumu przy $\alpha = 2.25$ oraz $\beta = 0.0085$.

6.3.3.2. Scenariusz „stabilizacja”

Testy wykonane zgodnie z tym scenariuszem miały na celu, sprawdzenie stabilności zaproponowanego sterownika rozmytego. Przez okres 2000 iteracji moduł KorMOD był uruchamiany na danych generowanych na podstawie tej samej macierzy OD przy różnych wartościach parametru s. Posłużyło to określeniu minimalnego procentu podróży jaki musi zostać zgłoszony do systemu, aby mógł on efektywnie funkcjonować.

Zaimplementowana metoda odniesienia jest zbieżna (tj. poziom generowanego szumu stabilizuje się) dla wartości $s \geq 0.05$. Przy wartościach niższych, poziom szumu wprowadzanego do macierzy OD rośnie nieograniczenie. Otrzymane wyniki pozwalają stwierdzić, że sterownik rozmyty działa poprawnie już przy $s \geq 0.015$, a optimum osiąga dla $s \geq 0.025$. Uśrednione poziomy szumu w zależności od wartości s przedstawione są na rysunku 6.3.3, natomiast dokładny przebieg testów dla wartości $s = 0.1$ oraz $s = 0.025$ na rysunkach 6.3.2 i 6.3.4.

Pozwala to wnioskować, że zaproponowana metoda aktualizacji macierzy jest znacznie mniej podatna na niedobór informacji oraz spowodowane tym zaburzenia losowe. Dzięki temu wymagana liczba użytkowników systemu *ProSym* może być nawet trzykrotnie mniejsza, niż przy zastosowaniu metody tradycyjnej, przy zachowaniu takiej samej jakości aktualizowanej macierzy.



Rys. 6.3.3. Porównanie średniego poziomu szumu w zależności od wartości s. Scenariusz „stabilizacja".

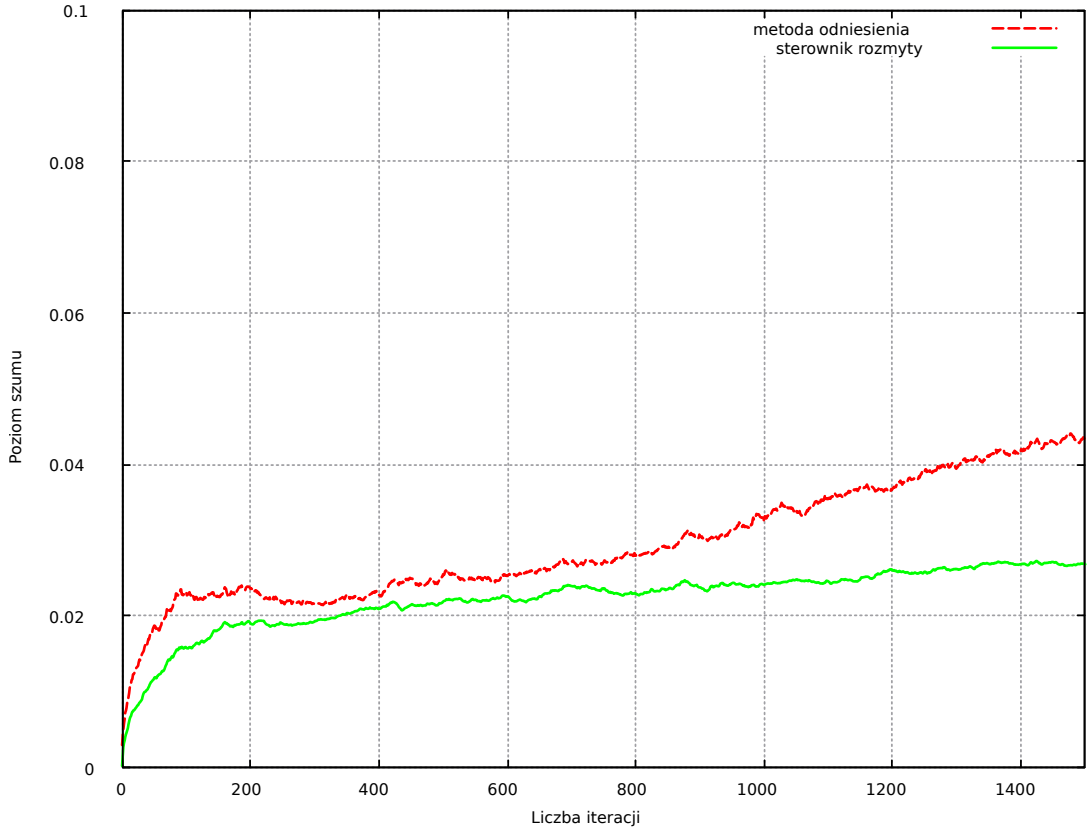
6.3.3.3. Scenariusz „losowy"

Scenariusz losowy obejmował testy, w których model ruchu został zmieniony w sposób przypadkowy. Stopień zmian macierzy końcowej określa parametr m. Zgodność macierzy OD, wyliczano jako odległość pomiędzy macierzą wynikową oraz obecną macierzą przybliżoną używaną do generowania danych wejściowych.

Symulacja zmian odbywa się w warunkach ustabilizowanych tj. stopniowa zmiana macierzy odbywa się po 500 iteracjach stabilizacji. Czas jej trwania to 500 iteracji. Po tym następuje okres 1000 iteracji, w którym macierz ponownie jest niezmienna. Schemat scenariusza przedstawiony jest na rysunku 6.3.1.

Wyniki pomiarów pozwalają stwierdzić, że obie metody dobrze radzą sobie z losowymi zmianami w macierzy nawet, gdy są one stosunkowo duże ($m = 0.2$). Na tej podstawie można stwierdzić, że sterownik rozmyty jest bardziej odporny na zbyt małą ilość informacji. Metoda odniesienia przy wartościach $s \leq 0.05$ wprowadza bardzo duży błąd do macierzy, praktycznie ignorując zmiany w niej zachodzące, podczas gdy sterownik KorMOD zachowuje swoją sprawność nawet przy wartości $s = 0.015$.

W warunkach pełnej dostępności informacji tj. gdy $s \geq 0.05$ sterownik rozmyty wykazuje nieco wolniejszą reakcję na zmiany, w porównaniu z metodą odniesienia.



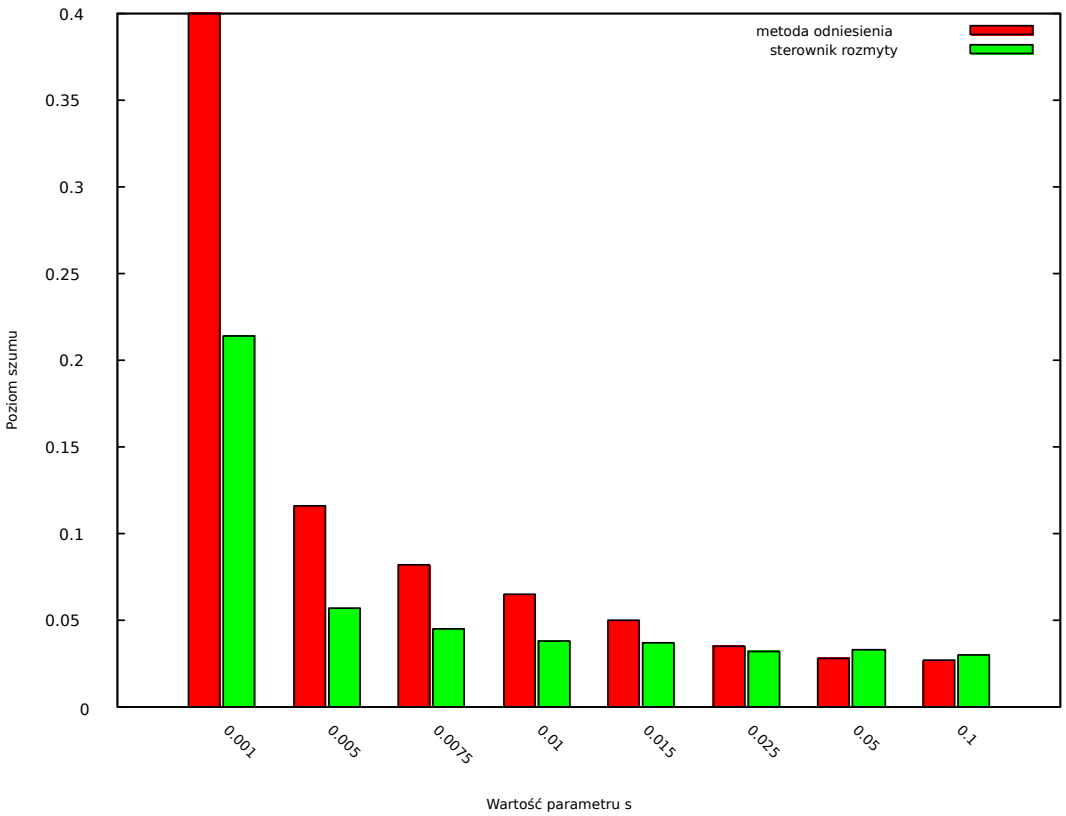
Rys. 6.3.4. Przebieg testu „stabilizacja” dla $s = 0.025$.

Jednak KorMOD, w okresie stabilizacji następującym po zmianie, osiąga znacząco niższy poziom szumu w porównaniu z metodą odniesienia.

6.3.3.4. Scenariusz „osiedle”

Scenariusz ten zbliżony jest do „losowego” z tą różnicą, że zmiany wprowadzane do macierzy końcowej nie są losowe. Symulują one rzeczywiste procesy zachodzące w popycie transportowym. Liczba pojazdów przejeżdżających pomiędzy wybranymi rejonami transportowymi została odpowiednio zwiększona lub zmniejszona, wraz z sąsiedztwem. Zmiany takie mogą być spowodowane na przykład rozbudową miasta, powstaniem nowego osiedla, czy likwidacją lub przeniesieniem pewnych punktów zainteresowań.

Metodyka oceny jest taka sama jak w przypadku testów „losowych”. Jak wynika z pomiarów, w tym przypadku sterownik KorMOD dorównuje szybkością reakcji na zmiany metodzie odniesienia w warunkach pełnej dostępności informacji. Gdy ilość informacji pochodzącej od użytkowników jest ograniczona, sterownik rozmyty generuje poprawne macierze OD podczas, gdy metoda odniesienia podobnie jak w przypadku „losowym” generuje szum przewyższający zadany poziom zmian.



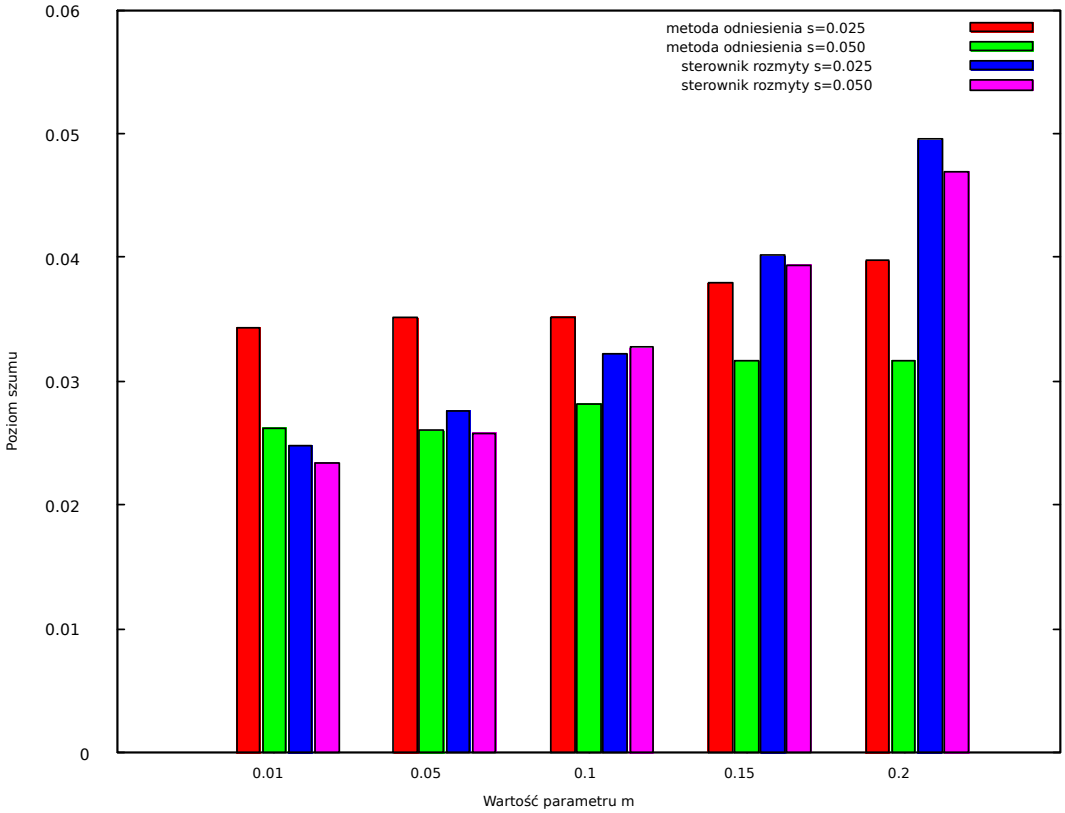
Rys. 6.3.5. Porównanie średniego poziomu szumu w zależności od wartości s. Scenariusz „losowy", $m = 0.1$.

6.3.4. Omówienie wyników

Podczas testów z wykorzystaniem wszystkich trzech scenariuszy, sterownik KorMOD wykazywał znacznie większą odporność na brak lub błędy w dostarczonej informacji. Poprawne działanie sterownika już przy wartości $s = 0.015$ daje nadzieję na możliwość zastosowania tego systemu w rzeczywistości. W przypadku miasta wielkości Poznania, oznaczałoby to, że system *ProSym* wymagałby około 8000 zgłoszonych tras dziennie, w porównaniu z metodą odniesienia, wymagającą prawie 30000.

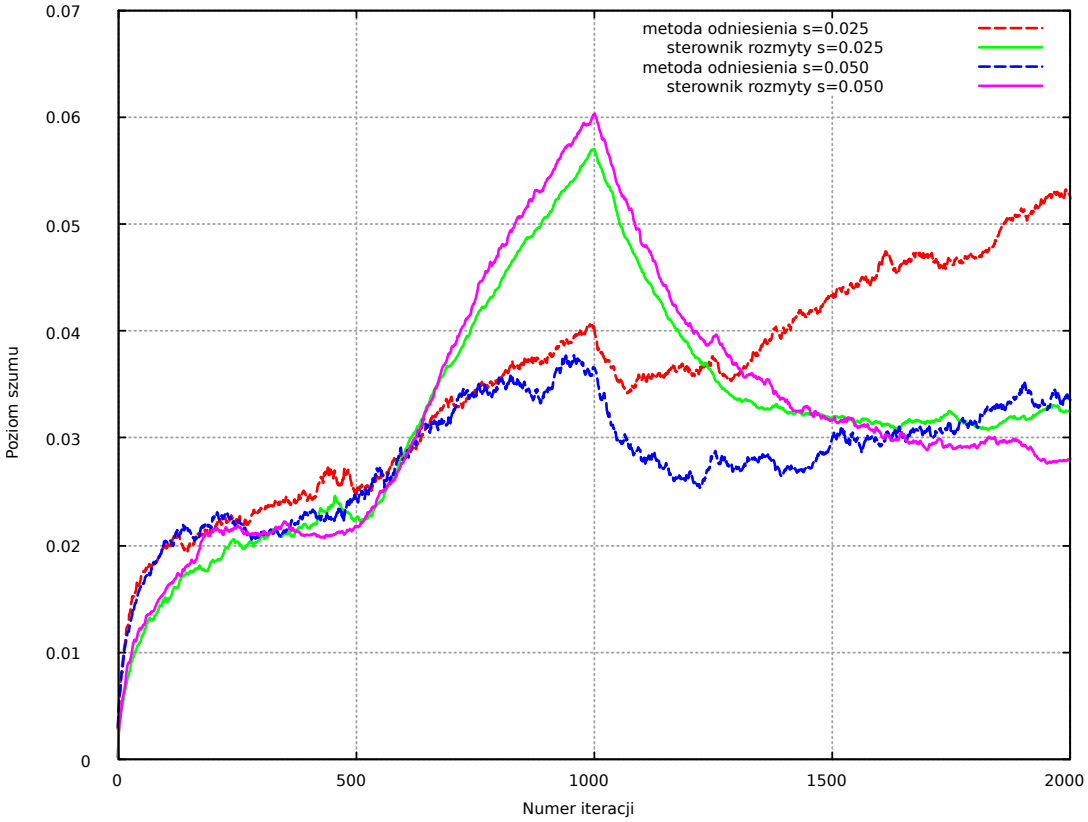
Optymistycznie wypadły również testy szybkości reakcji na zmiany. Mimo, iż w testach losowych jest ona mniejsza, to w testach odzwierciedlających rzeczywiste procesy zachodzące w popycie transportowym, tempo odzwierciedlania zmian w macierzy jest porównywalne z metodą odniesienia. Ponadto końcowy poziom szumu jest istotnie mniejszy, co pozwala stwierdzić, że sterownik KorMOD potrafi uwzględnić znacznie mniejsze zmiany w macierzy.

Zaprezentowane tu wyniki, otrzymano korzystając z pewnej konfiguracji sterownika (parametry zbiorów rozmytych, wykorzystane stałe), uzyskanej na podstawie wiedzy o charakterystyce modelowanego procesu. Niestety ze względu na duży nakład obliczeniowy, niemożliwe było przeprowadzenie procesu optymalizacji z wykorzystaniem współczesnych metod takich, jak sieci neuronowe, czy algorytmy gene-

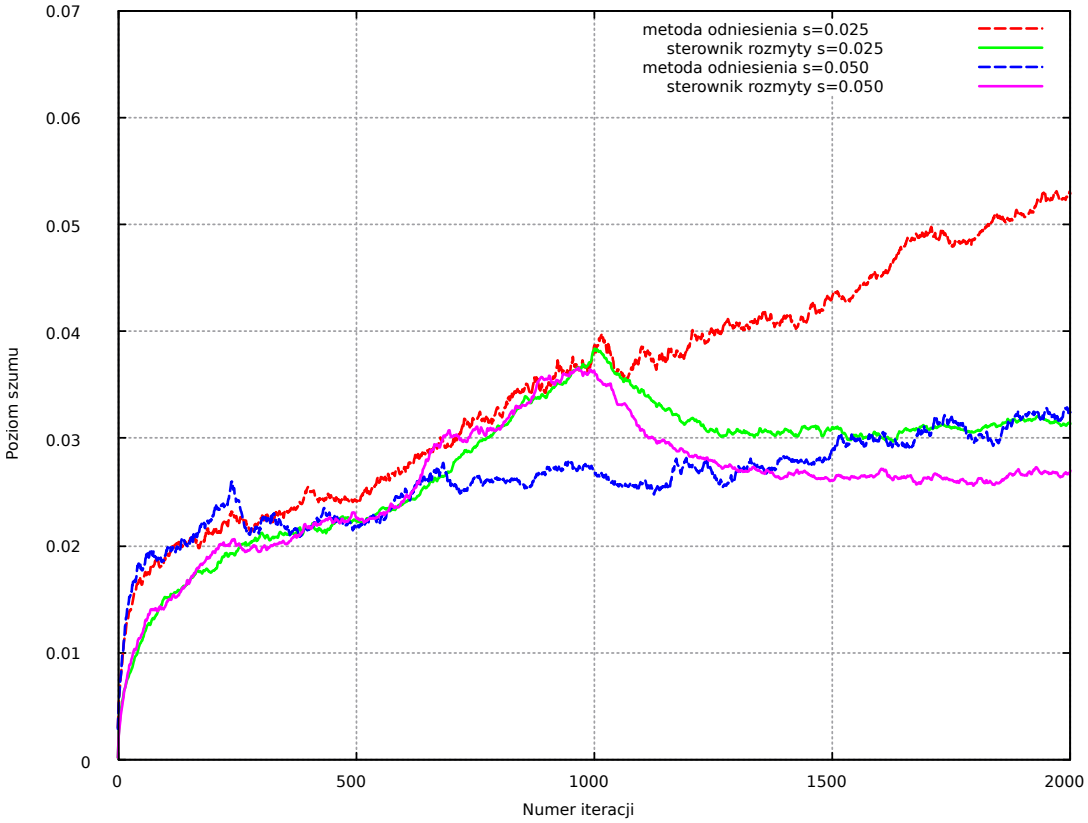


Rys. 6.3.6. Porównanie średniego poziomu szumu w zależności od wartości m, scenariusz „losowy".

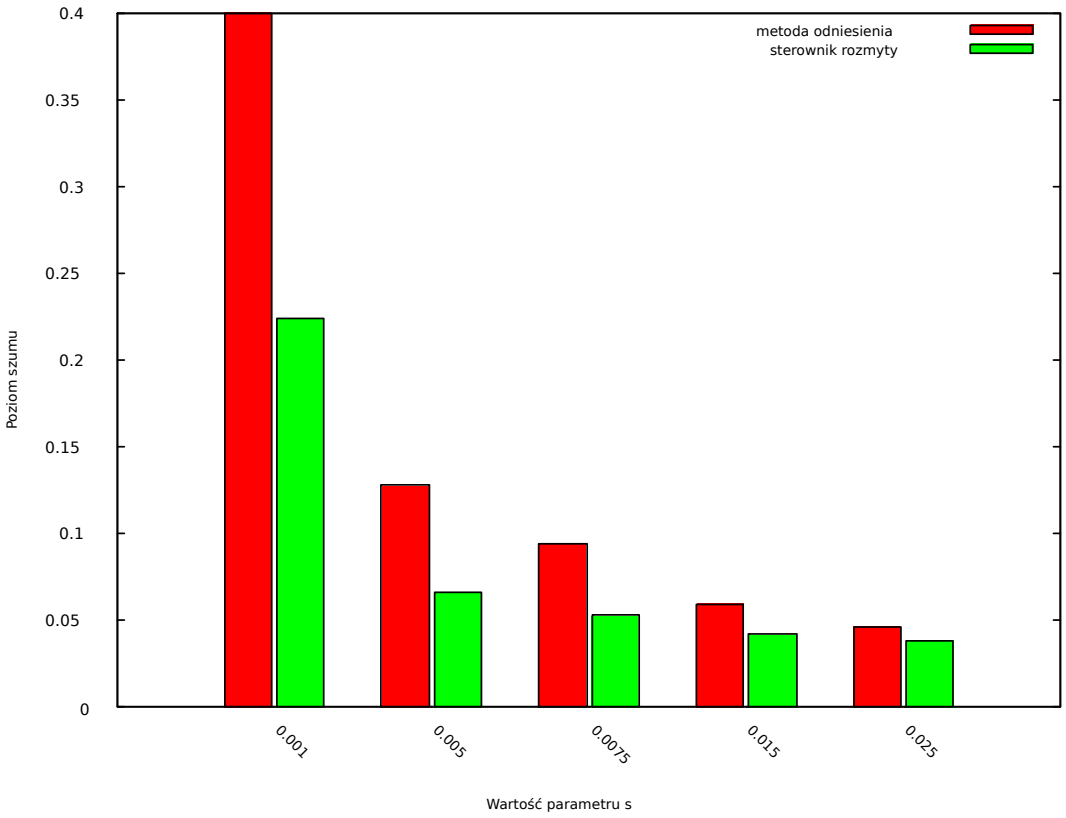
tyczne. Wskazane jest przeprowadzenie optymalizacji, co z pewnością przyczyni się do uzyskania jeszcze lepszych wyników.



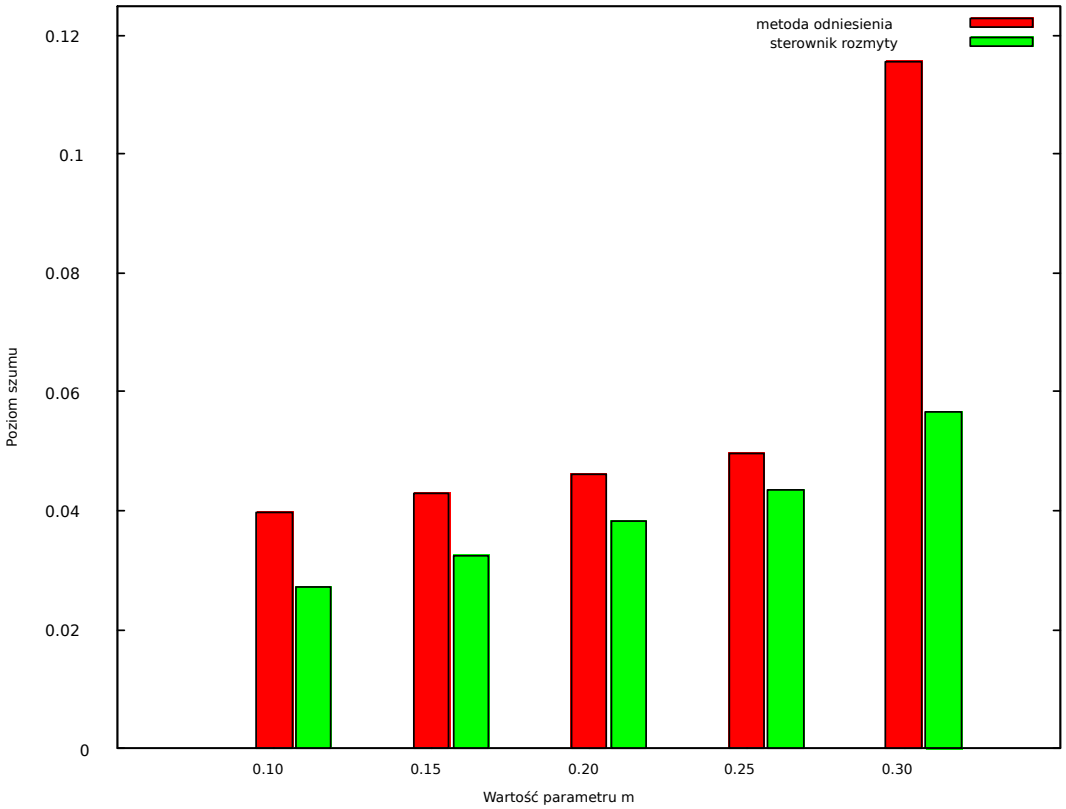
Rys. 6.3.7. Przebieg testu „losowy” dla $m = 0.1$.



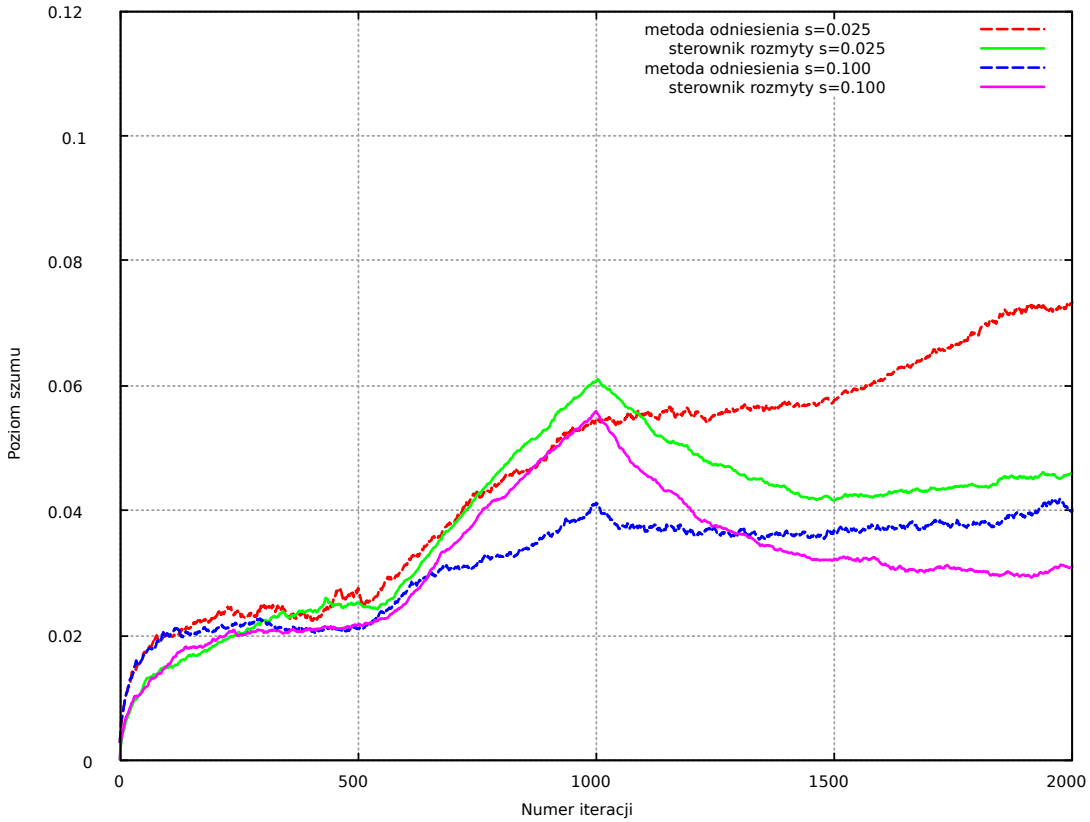
Rys. 6.3.8. Przebieg testu „losowy” dla $m = 0.05$.



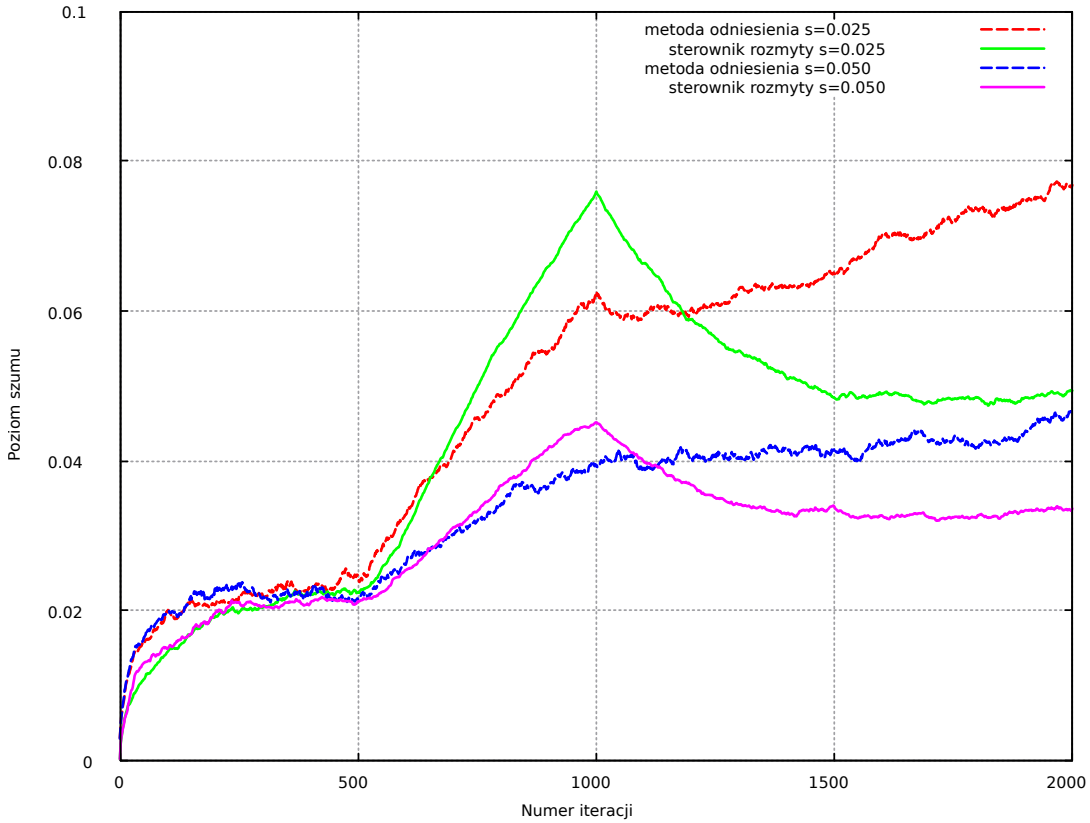
Rys. 6.3.9. Porównanie średniego poziomu szumu w zależności od wartości s. Scenariusz „osiedle”, $m = 0.2$.



Rys. 6.3.10. Porównanie średniego poziomu szumu w zależności od wartości m, scenariusz „osiedle”, $s = 0.025$.



Rys. 6.3.11. Przebieg testu „osiedle” dla $m = 0.2$.



Rys. 6.3.12. Przebieg testu „osiedle” dla $m = 0.25$.

Rozdział 7

Podsumowanie

W niniejszej pracy przedstawione zostały różne podejścia do modelowania emisji spalin i zużycia paliwa w ruchu drogowym. Wraz z rozwojem techniki i wzrostem mocy obliczeniowej komputerów, rośnie zainteresowanie metodami modelowania opartymi na symulacji komputerowej. Stąd też większość pracy poświęcona jest temu podejściu.

Przedstawiono podstawowe informacje na temat symulacji ruchu drogowego, następnie dokonano przeglądu dostępnego oprogramowania. Należy zauważyć, że nie wszystkie dostępne symulatory oferują funkcję modelowania emisji spalin i zużycia paliwa. Dużo uwagi poświęcono, wykorzystanemu w projekcie *ProSym*, symulatorowi SUMO.

W ramach pracy został zaimplementowany pakiet oprogramowania *ProSym*, którego zadaniem jest zminimalizowanie nakładu pracy potrzebnego do uruchomienia symulacji ruchu, wraz z modelowaniem emisji spalin i zużycia paliwa. Informacje pozyskane z symulacji są na bieżąco udostępnianie wszystkim zainteresowanym za pośrednictwem strony internetowej. Dzięki temu kierowcy mają możliwość porównania różnych tras przejazdu nie tylko pod kątem długości, lecz również emisji zanieczyszczeń i zużycia paliwa, co pozwoli to na dobór optymalnej trasy przejazdu. Co więcej, z pewnością zwiększy świadomość kierowców na temat emisji spalin i zużycia paliwa.

Bardzo istotnym komponentem oprogramowania *ProSym* jest Korektor Macierzy OD. Jest on odpowiedzialny za aktualizację modelu wykorzystywanego do symulacji, a zatem za jakość informacji prezentowanych użytkownikom systemu. Jego działanie oparte jest na autorskim podejściu wykorzystującym sterowanie rozmyte oraz informacje pochodzące od użytkowników – kierowców. Przeprowadzone testy wskazują jednoznacznie, iż zaproponowane podejście spełnia stawiane przed nim wymagania.

Oprogramowanie *ProSym* udostępniane jest na otwartej licencji, co w połączeniu z odpowiednią architekturą systemu, otwiera szerokie możliwości dalszego rozwoju.

Szczególny nacisk powinien zostać położony na optymalizację wykorzystywanych parametrów, dodanie nowych funkcjonalności oraz dalsze uproszczenie sposobu uruchomienia i utrzymania aplikacji.

Podsumowując, oryginalny, autorski charakter mają następujące elementy niniejszej pracy magisterskiej:

- pakiet oprogramowania *ProSym*,
- oparty na symulacji komputerowej serwis Mapy Ruchu Drogowego (MaRuD),
- metoda aktualizacji modelu ruchu drogowego oparta na sterowaniu rozmytym oraz danych pochodzących od kierowców (KorMOD).

Bibliografia

[Bak94] M. Baker. Fuel Consumption and Emission Models for Evaluating Traffic Control and Route Guidance Strategies. Praca magisterska, Queen's University, Kingston, 1994.

[Bar10] J. Barceló, redaktor. *Fundamentals of Traffic Simulation*. Springer, New York, 2010.

[BBEK11] M. Behrisch, L. Bieker, J. Erdmann, D. Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, Spain, 2011.

[Cal96] California Air Resources Board. *Methodology for estimating emissions from on-road motor vehicles*, 1996.

[Cem05] C. Cempel, redaktor. *Teoria i inżynieria systemów – zasady i zastosowania myślenia systemowego*. Instytut Technologii Eksploatacji Państwowy Instytut Badawczy, Radom, 2005.

[Deg95] P. Degobert. *Automobile and pollution*. Society of Automotive Engineers, 1995.

[eea05] The European environment - State and outlook 2005. Raport instytutowy, Eurorean Environment Agency, 2005.

[eea07a] Air pollution in Europe 1990–2004. Raport instytutowy, European Environment Agency, 2007.

[eea07b] EMEP/CORINAIR Emission Inventory Guidebook. Raport instytutowy, Eurorean Environment Agency, 2007.

[epa] Strona internetowa U.S. Environmental Protection Agency. `http://www.epa.gov/air/urbanair/`. Dostęp: 17.04.2012.

[epa94] Automobile Emissions: An Overview. Raport instytutowy EPA 400-F-92-007, U.S. Environmental Protection Agency, 1994.

[epa01] Inventory of US greenhouse gas emissions and sinks: 1990–1999. Raport instytutowy EPA236-R- 01-001, U.S. Environmental Protection Agency, 2001.

[fcl97] IEC 1131 - Programmable Controllers, Part 7 – Fuzzy Control Programming. Raport instytutowy, International Electrotechnical Commission, 1997.

[gdd11] Prognoza oddziaływania na środowisko skutków realizacji Programu Budowy Dróg Krajowych na lata 2011 – 2015. Raport instytutowy, Generalna Dyrekcja Dróg Krajowych i Autostrad, 2011.

[Ger12] German Aerospace Center. *SUMO User's Guide*, 2012.

[Hał10] Ilona Hałucha. 2 w 1: modelowanie ruchu i obliczanie emisji zanieczyszczeń. *e-droga.pl - portal drogowy*, 2010.

[HHJ99] J. Hickman, D. Hassel, R. Joumard. Methodology for calculating transport emissions and energy consumption. Raport instytutowy, Transport and Road Research Laboratory, 1999.

[Hor82] J.L. Horowitz. *Air Quality Analysis for Urban Transportation Planning*. MIT Press, 1982.

[JY12] Lv Jinpeng, Zhang Yunlong. Effect of signal coordination on traffic emission. *Transportation Research Part D*, 17:149–153, 2012.

[Kac01] J. Kacprzyk. *Wieloetapowe sterowanie rozmyte*. WNT, Warszawa, 2001.

[kbr] Kompleksowe Badania Ruchu: Poznań. `http://www.bit-poznan.com.pl/kbr.htm`. Dostęp: 21-04-2012.

[KH09] G. Kotusevski, K.A. Hawick. A Review of Traffic Simulation Software. Raport instytutowy, Computer Science, Institute of Information and Mathematical Sciences, Massey University, Albany, 2009.

[Kra98] Stefan Krauß. *Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics*. Praca doktorska, Hauptabteilung Mobilität und Systemtechnik des DLR, Köln, 1998.

[Kry10] A. Krych. Słownictwo kompleksowych badań i modelowania ruchu. *Modelowanie podróży i prognozowanie ruchu*, 2010.

[KSXL11] Xiangjie Kong, Guojiang Shen, Feng Xia, Chuang Lin. Urbal Arterial Traffic Two-direction Green Wave Intelligent Coordination Control Technique and Its Application. *Journal of Automation, Control and Systems*, 9:60–68, 2011.

[mrr08] Niebieska Księga. Infrastruktura drogowa. Raport instytutowy, Ministerstwo Rozwoju Regionalnego, 2008.

[NBS02] E. K. Nam, H. Brazil, S. Sutulo. The integration of a fuel rate based modal emissions model (CMEM modified) into the VISSIM microscopic traffic model – inventory comparison with MOBILE6 in Southfield Michigan. *Proceedings of the 12th CRC On-Road Vehicle Emissions Workshop*, San Diego, California, 2002.

[NMM94] S. Nizich, T. McMullen, D. Misenheimer. *National Air Pollutant Emissions Trends, 1900-1993*. Office of Air Quality Planning and Standards, Research Triangle Park, 1994.

[Opo10] Krzysztof Opoczyński. Synteza wyników Generalnego Pomiaru Ruchu 2010. Raport instytutowy, Generalna Dyrekcja Dróg Krajowych i Autostrad, 2010.

[Pie03] A. Piegat. *Modelowanie i sterowanie rozmyte*. EXIT, Warszawa, 2003.

[RC98] Harikishan Reddy, Partha Chakroborty. A fuzzy inference based assignment algorithm to estimate O–D matrix from link volume counts. *Computers, Environment and Urban Systems*, 22:409–423, 1998.

[SBdH88] P.J. Sturm, P. Boulter, P. de Haan. Instantaneous emission data and their use in estimating passenger car emissions. Raport instytutowy, Transport and Road Research Laboratory, 1988.

[ST96] G. Schmidt, B. Thomas. *Hochrechnungsfaktoren für manuelle und automatische Kurzzeitzählungen im Innerortsbereich*. Hrsg.: Bundesministerium für Verkehr, Abteilung Straßenbau, Godesberg, 1996.

[SŻ12] Andrzej Szarata, Joanna Żygłowicz. Dynamczne modele podróży. *e-droga.pl - portal drogowy*, 2012.

[Teo99] Dušan Teodorović. Fuzzy logic systems for transportation engineering: the state of the art. *Transportation Research Part A*, 33:337–364, 1999.

[U.S03] U.S. Environmental Protection Agency. *User's Guide to MOBILE6.1 and MOBILE6.2*, 2003.

[War52] J. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers, Part II*, 1(36):352–362, 1952.

[Zad65] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[Zad73] L. Zadeh. Outline of new approach to analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, 2:28–44, 1973.

Spis rysunków

1.1.1. Udział poszczególnych sektorów w emisji wybranych zanieczyszczeń 12
1.2.1. Wykres faz pracy silnika 15

2.2.1. Przykładowa macierz OD 23
2.2.2. Modele infrastruktury drogowej o różnym stopniu szczegółowości . . 24
2.2.3. Różne poziomy szczegółowości modelowania ruchu pojazdów 26
2.2.4. Schemat drogi w modelu mezoskopowym 29
2.3.1. Symulator VISSIM 31
2.3.2. Symulator VISUM 31
2.3.3. Symulator SUMO 32
2.5.1. Udział poszczególnych środków transportu w Poznaniu 36
2.5.2. Najczęstsze cele podróży w Poznaniu 36
2.5.3. Model ruchu miasta Poznań w programie VISUM 37

3.1.1. Funkcja charakterystyczna zbioru $\{x \in \mathbb{R} : -0.2 \leq x \leq 0.2\}$ 39
3.1.2. Przykładowa funkcja przynależności 40
3.1.3. Zbiór rozmyty jako suma singletonów 41
3.2.1. Schemat wnioskowania rozmytego 44
3.2.2. Funkcja przynależności konkluzji 45

4.2.1. Strona główna aplikacji MaRuD Poznań 50
4.2.2. Widok wyboru trasy aplikacji MaRuD Poznań 51
4.2.3. Widok zgłaszania trasy przejazdu systemu KorMOD 52
4.2.4. Aplikacja MaRuD4Android 52

5.1.1. Schemat realizacji zapytania przez aplikację MaRuD 56
5.2.1. Schemat przeprowadzania symulacji w systemie *ProSym* 60
5.3.1. Średni rozkład natężenia ruchu drogowego 62
5.3.2. Liczba pojazdów poruszających się w sieci drogowej 62
5.3.3. Średni czas przejazdu pojazdów poruszających się w sieci drogowej . 63

6.1.1. Schemat działania systemu KorMOD 64
6.2.1. Rejony transportowe oraz ich sąsiedztwa na mapie 68
6.2.2. Zbiory rozmyte opisujące wartości wejściowe sterownika KorMOD^{I} . 70
6.2.3. Zbiory rozmyte opisujące wartości wyjściowe sterownika KorMOD^{I} . 70
6.2.4. Zbiory rozmyte opisujące wartości wejściowe sterownika KorMOD^{II} 72
6.3.1. Schematy scenariuszy testowych 76
6.3.2. Poziom szumu przy $\alpha = 2.25$ oraz $\beta = 0.0085$ 77
6.3.3. Porównanie średniego poziomu szumu w zależności od wartości s, scenariusz „stabilizacja” 78
6.3.4. Przebieg testu „stabilizacja” dla $s = 0.025$ 79
6.3.5. Porównanie średniego poziomu szumu w zależności od wartości s, scenariusz „losowy” 80
6.3.6. Porównanie średniego poziomu szumu w zależności od wartości m, scenariusz „losowy” 81
6.3.7. Przebieg testu „losowy” dla $m = 0.1$ 82
6.3.8. Przebieg testu „losowy” dla $m = 0.05$ 82
6.3.9. Porównanie średniego poziomu szumu w zależności od wartości s, scenariusz „osiedle” 83
6.3.10. Porównanie średniego poziomu szumu w zależności od wartości m, scenariusz „osiedle” 83
6.3.11. Przebieg testu „osiedle” dla $m = 0.2$ 84
6.3.12. Przebieg testu „osiedle” dla $m = 0.25$ 84

B.1.1. Ogólna architektura systemu *ProSym* 97
B.3.1. Architektura komponentu MaRuD 99
B.4.1. Architektura komponentu KorMOD 99

Spis tablic

1.3.1. Przykładowa mapa emisji CO (g/h) 19

2.3.1. Zestawienie najpopularniejszych symulatorów ruchu 30

Dodatki

Dodatek A

Uwagi o zastosowanej terminologii

Terminologia dotycząca ruchu drogowego zastosowana w tej pracy pochodzi głównie z [Kry10] oraz [kbr]. Jak jednak można zauważyć, cześć teoretyczna pracy napisana została głównie w oparciu o literaturę anglojęzyczną. Dlatego też część terminów musiała zostać przetłumaczona na język polski w sposób dosłowny. Celem tego dodatku jest podanie ich oryginalnej formy, co może ułatwić doprecyzowanie znaczenia.

A.1. Słownik pojęć

macierz OD (ang. origin–destination matrix, OD matrix)

popyt transportowy (ang. traffic demand)

potok ruchu (ang. traffic flow)

trasa (droga) przejazdu (ang. route)

podróż (ang. trip)

modele oparte na zużyciu paliwa (ang. load based models)

metoda zadaniowa określania popytu transportowego (ang. activity–based demand generation)

rejon transportowy (ang. zone, district)

wybór (przydział) tras przejazdu (ang. route choose, routing)

(dynamiczna) równowaga użytkownika (ang. (dynamic) user equilibrium, DUE)

modele podążania pojazdów (ang. car following models)

Dodatek B

Architektura systemu *ProSym*

B.1. Ogólna architektura systemu

ProSym wykonany jest w architekturze klient–serwer. Serwer udostępnia poprzez sieć serwisy, które wykorzystywane są przez aplikacje klienckie.

Podczas projektowania położono duży nacisk na modularność. Dzięki temu dalszy rozwój aplikacji i rozbudowa o kolejne funkcjonalności nie stanowi problemu. Obecnie dostępne są trzy główne moduły: MaRuD, KorMOD oraz narzędzia pomocnicze. Dwa pierwsze zostaną opisane poniżej. W skład narzędzi pomocniczych wchodzą różne skrypty ułatwiające wykonywanie podstawowych zadań konfiguracyjnych i administracyjnych systemu.

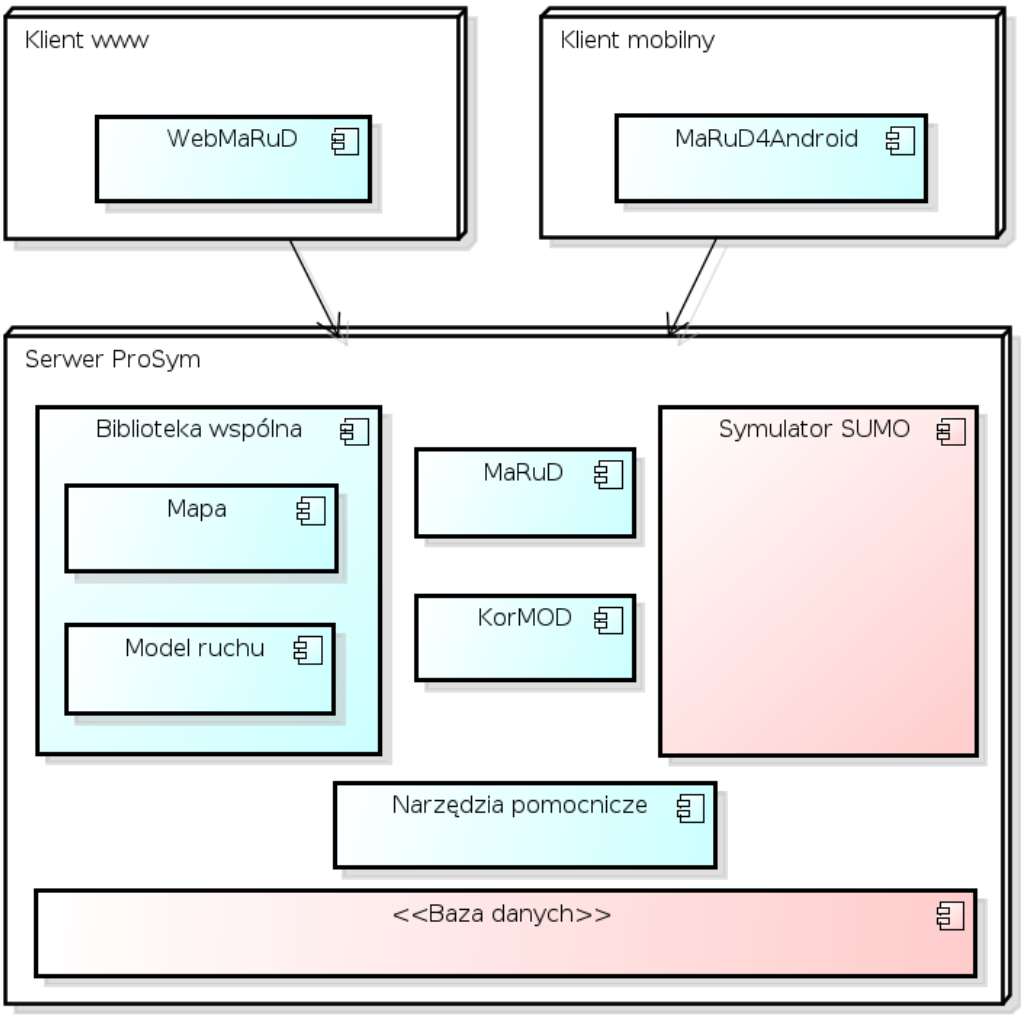
Pomimo tego, że większość komponentów pakietu korzysta z symulatora SUMO, jest on oddzielną i niezależną aplikacją, która jest uruchamiana przez aplikację MaRuD. Co więcej do działania oprogramowania nie jest konieczne jego posiadanie, o ile wyniki symulacji zostaną dostarczone w inny sposób (np. statyczne wyniki symulacji wgrane do systemu w postaci plików).

Poglądowy diagram budowy pakietu *ProSym* znajduje się na rysunku B.1.1.

B.2. Wykorzystane technologie

Pakiet oprogramowania *ProSym* jest złożonym systemem integrującym wiele dostępnych na rynku darmowych technologii. Poniższa sekcja przedstawia zestawienie najważniejszych z nich.

Głównym kryterium doboru wykorzystywanego oprogramowania było to, aby mógł on działać na jak największej liczbie powszechnie używanych komputerów (i telefonów). Dlatego też zdecydowano się na język Java po stronie serwera oraz format strony internetowej po stronie klienta. Należy również zwrócić uwagę, że wszystkie



Rys. B.1.1. Ogólna architektura systemu *ProSym*.

wybrane technologie (za wyjątkiem Google Maps API) udostępniane są na otwartych licencjach (*open source*).

Opis zastosowanych technologii zostanie podany z podziałem na części aplikacji:

- klient www
  - JQuery 1.7.1
  - Google Maps API 3 – wizualizacja tras przejazdu
- klient mobilny – MaRuD4Android
  - Android SDK 10
  - Google APIs SDK 10
- serwer *ProSym*
  - Spring Framework 3.1.1

– myBatis 3.1.0 – mapowanie relacyjno–obiektowe
– jFuzzyLogic 2.1 – wykonywanie sterownika rozmytego KorMOD
– Apache Tomcat 6
– Open Street Maps – format map
– SUMO 0.15 – symulator ruchu drogowego
– Python 2.7.2 – skrypty konfiguracyjne
– PostgreSQL 9.1 – baza danych

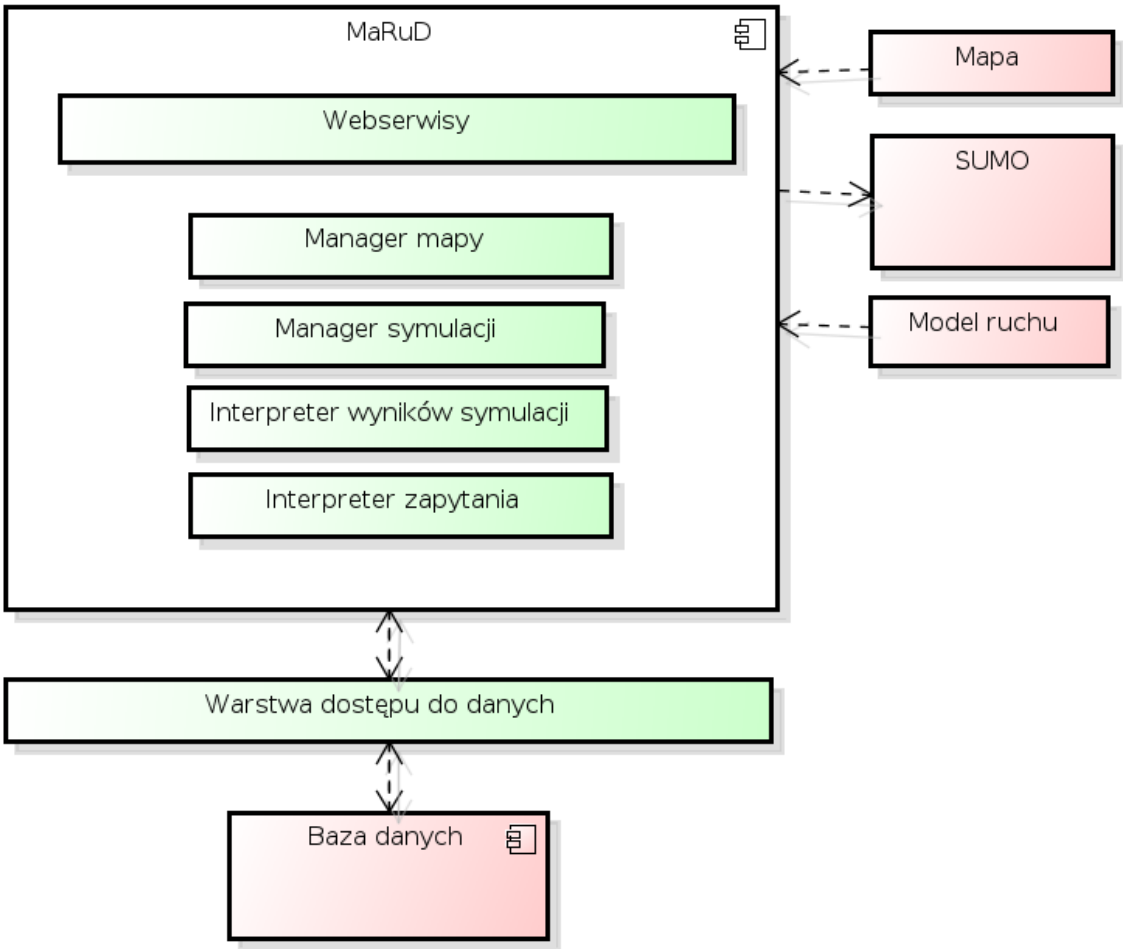
B.3. Architektura aplikacji MaRuD

Diagram B.3.1 przedstawia architekturę części serwerowej aplikacji MaRuD. Wyszczególniono na nim główne elementy wewnętrzne (manager mapy, symulacji, interpreter wyników oraz interpreter zapytania) i zewnętrzne (symulator SUMO, mapa oraz model ruchu). Dostęp do bazy danych, współdzielonej przez cały pakiet *ProSym*, odbywa się za pomocą warstwy dostępu do danych, która zapewnia odpowiedni poziom bezpieczeństwa.

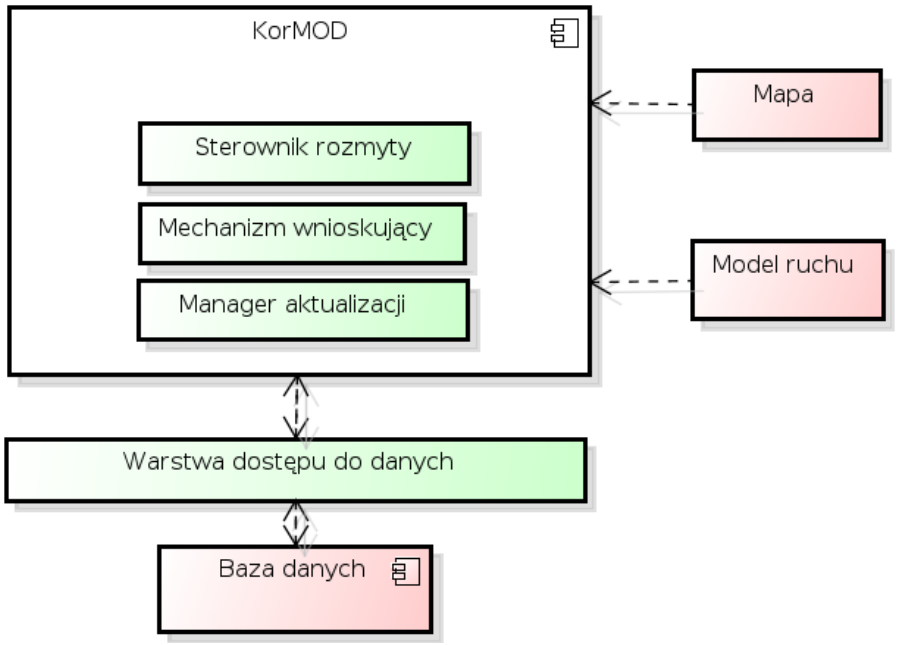
Webserwisy udostępniane przez serwer zaprojektowane są zgodnie z ideą REST, dzięki czemu możliwe jest ich wykorzystanie w wielu różnych aplikacjach.

B.4. Architektura modułu KorMOD

Moduł KorMOD jest stosunkowo prosty w swej budowie. Trzy główne jego elementy to: sterownik rozmyty, mechanizm wnioskujący oraz manager aktualizacji odpowiedzialny za zmianę macierzy OD wykorzystywanej do symulacji. Zewnętrzne do programu są: mapa i model ruchu. Podobnie jak aplikacja MaRuD, KorMOD korzysta ze współdzielonej bazy danych za pośrednictwem warstwy dostępu do danych.



Rys. B.3.1. Architektura komponentu MaRuD.



Rys. B.4.1. Architektura komponentu KorMOD.

Dodatek C

Sterownik rozmyty KorMOD

Poniższy kod sterownika w języku FCL zawiera dwa bloki, każdy odpowiedzialny za inną fazę procesu aktualizacji macierzy OD.

```
FUNCTION_BLOCK phase1

VAR_INPUT
    w : REAL;
    k : REAL;
    kp: REAL;
    s : REAL;
    sp: REAL;
    d : REAL;
    dp: REAL;
END_VAR

VAR_OUTPUT
    z : REAL;
    u : REAL;
END_VAR

FUZZIFY w
    TERM maly := (0, 1.0) (0.05, 1) (0.1, 0);
    TERM sredni := (0.05, 0) (0.1, 1) (0.15, 0);
    TERM duzy := (0.1, 0) (0.15, 1) (1.0, 1);
END_FUZZIFY

FUZZIFY k
    TERM ujemny := (-1, 1.0) (-0.75, 1.0) (0, 0.05) (0.05, 0);
```

```
    TERM neutralny := (-0.5, 0) (-0.075, 1) (0.1, 1) (0.5, 0);
    TERM dodatni := (-0.05, 0) (0, 0.05) (0.75, 1.0) (1, 1);
END_FUZZIFY

FUZZIFY kp
    TERM ujemny := (-1, 1.0) (-0.75, 1.0) (0, 0.05) (0.05, 0);
    TERM neutralny := (-0.5, 0) (-0.075, 1) (0.1, 1) (0.5, 0);
    TERM dodatni := (-0.05, 0) (0, 0.05) (0.75, 1.0) (1, 1);
END_FUZZIFY

FUZZIFY s
    TERM ujemny := (-1, 1.0) (-0.75, 1.0) (0, 0.05) (0.05, 0);
    TERM neutralny := (-0.5, 0) (-0.075, 1) (0.1, 1) (0.5, 0);
    TERM dodatni := (-0.05, 0) (0, 0.05) (0.75, 1.0) (1, 1);
END_FUZZIFY

FUZZIFY sp
    TERM ujemny := (-1, 1.0) (-0.75, 1.0) (0, 0.05) (0.05, 0);
    TERM neutralny := (-0.5, 0) (-0.075, 1) (0.1, 1) (0.5, 0);
    TERM dodatni := (-0.05, 0) (0, 0.05) (0.75, 1.0) (1, 1);
END_FUZZIFY

FUZZIFY d
    TERM ujemny := (-1, 1.0) (-0.75, 1.0) (0, 0.05) (0.05, 0);
    TERM neutralny := (-0.5, 0) (-0.075, 1) (0.1, 1) (0.5, 0);
    TERM dodatni := (-0.05, 0) (0, 0.05) (0.75, 1.0) (1, 1);
END_FUZZIFY

FUZZIFY dp
    TERM ujemny := (-1, 1.0) (-0.75, 1.0) (0, 0.05) (0.05, 0);
    TERM neutralny := (-0.5, 0) (-0.075, 1) (0.1, 1) (0.5, 0);
    TERM dodatni := (-0.05, 0) (0, 0.05) (0.75, 1.0) (1, 1);
END_FUZZIFY

DEFUZZIFY z
    TERM ujemny := (-0.1, 1.0) (-0.09, 0.9) (0, 0);
    TERM neutralny := (-0.025, 0) (0, 1) (0.025, 0);
    TERM dodatni := (0, 0) (0.09, 0.9) (0.1, 1);
    METHOD : COG;
    DEFAULT := 0;
```

```
END_DEFUZZIFY

DEFUZZIFY u
    TERM mala := (0, 1) (0.3, 0);
    TERM srednia := (0, 0) (0.3, 1) (0.45, 0);
    TERM duza := (0.3, 0) (0.45, 1) (1, 1);
    METHOD : COG;
    DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
  AND : MIN;
  ACT : MIN;
  ACCU : MAX;

  RULE 1 : IF w IS duzy
                THEN u IS duza;
  RULE 2 : IF w IS sredni
                THEN u IS srednia;
  RULE 3 : IF w IS maly
                THEN u IS mala;


  RULE 4 : IF k IS dodatni AND s IS dodatni AND d IS dodatni
                THEN z IS dodatni, u IS duza;
  RULE 5 : IF k IS dodatni AND s IS dodatni AND NOT d IS dodatni
                THEN z IS dodatni, u IS srednia;
  RULE 6 : IF k IS dodatni AND NOT ( s IS dodatni OR d IS dodatni)
                THEN z IS dodatni, u IS mala;

  RULE 7 : IF k IS neutralny AND s IS neutralny AND d IS neutralny
                THEN z IS neutralny, u IS duza;
  RULE 8 : IF k IS neutralny AND s IS neutralny AND NOT d IS neutralny
                THEN z IS neutralny, u IS srednia;
  RULE 9 : IF k IS neutralny AND NOT ( s IS neutralny OR d IS neutralny)
                THEN z IS neutralny, u IS mala;

  RULE 10 : IF k IS ujemny AND s IS ujemny AND d IS ujemny
                THEN z IS ujemny, u IS duza;
  RULE 11 : IF k IS ujemny AND s IS ujemny AND NOT d IS ujemny
```

```
                THEN z IS ujemny, u IS srednia;
  RULE 12 : IF k IS ujemny AND NOT ( s IS ujemny OR d IS ujemny)
                THEN z IS ujemny, u IS mala;


  RULE 13 :IF k IS dodatni AND s IS ujemny AND k is ujemny
                THEN z IS neutralny, u IS duza;
  RULE 14 :IF k IS dodatni AND s IS ujemny
                THEN z IS neutralny;

  RULE 15 :IF k IS ujemny AND s IS dodatni AND k is dodatni
                THEN z IS neutralny, u IS duza;
  RULE 16 :IF k IS ujemny AND s IS dodatni
                THEN z IS neutralny;


  RULE 17 :IF k IS dodatni AND kp IS ujemny AND
                             (sp IS ujemny OR dp IS ujemny)
                THEN u IS mala;

  RULE 18 :IF k IS ujemny AND kp IS dodatni AND
                             (sp IS dodatni OR dp IS dodatni)
                THEN u IS mala;
END_RULEBLOCK

END_FUNCTION_BLOCK

FUNCTION_BLOCK phase2

VAR_INPUT
    zb : REAL;
    zp : REAL;
    ub: REAL;
    up : REAL;
END_VAR

VAR_OUTPUT
    z : REAL;
    u : REAL;
END_VAR
```

```
FUZZIFY ub
    TERM mala := (0, 1) (0.5, 0);
    TERM srednia := (0, 0) (0.5, 1) (1, 0);
    TERM duza := (0.5, 0) (1, 1);
END_FUZZIFY

FUZZIFY up
    TERM mala := (0, 1) (0.5, 0);
    TERM srednia := (0, 0) (0.5, 1) (1, 0);
    TERM duza := (0.5, 0) (1, 1);
END_FUZZIFY

FUZZIFY zb
    TERM ujemny := (-0.1, 1.0) (0, 0);
    TERM neutralny := (-0.03, 0) (0, 1) (0.03, 0);
    TERM dodatni := (0, 0) (0.1, 1);
END_FUZZIFY

FUZZIFY zp
    TERM ujemny := (-0.1, 1.0) (0, 0);
    TERM neutralny := (-0.03, 0) (0, 1) (0.03, 0);
    TERM dodatni := (0, 0) (0.1, 1);
END_FUZZIFY

DEFUZZIFY z
    TERM ujemny := (-0.1, 1.0) (0, 0);
    TERM neutralny := (-0.025, 0) (0, 1) (0.025, 0);
    TERM dodatni := (0, 0) (0.09, 0.9) (0.1, 1);
    METHOD : COG;
    DEFAULT := 0;
END_DEFUZZIFY

DEFUZZIFY u
    TERM mala := (0, 1) (0.3, 0);
    TERM srednia := (0, 0) (0.3, 1) (0.45, 0);
    TERM duza := (0.3, 0) (0.45, 1) (1, 1);
    METHOD : COG;
    DEFAULT := 0;
END_DEFUZZIFY
```

```
RULEBLOCK No1
    AND : MIN;
    ACT : MIN;
    ACCU : MAX;

    RULE 1 : IF zb IS dodatni AND zp IS dodatni
                THEN z IS dodatni;
    RULE 2 : IF zb IS ujemny AND zp IS ujemny
                THEN z IS ujemny;
    RULE 3 : IF zb IS neutralny AND zp IS neutralny
                THEN z IS neutralny;


    RULE 4 : IF ub IS duza AND up IS duza
                THEN u IS duza;
    RULE 5 : IF ub IS srednia AND up IS srednia
                THEN u IS srednia;

    RULE 6 : IF ub IS mala OR up IS mala
                THEN u IS mala, z IS neutralny;


    RULE 7 : IF zb IS dodatni AND zp IS neutralny
                THEN z IS neutralny;
    RULE 8 : IF zb IS dodatni AND zp IS ujemny
                THEN z IS neutralny, u IS mala;

    RULE 9 : IF zb IS neutralny AND zp IS dodatni
                THEN z IS neutralny;
    RULE 10 : IF zb IS neutralny AND zp IS ujemny
                THEN z IS neutralny;

    RULE 11 : IF zb IS ujemny AND zp IS neutralny
                THEN z IS neutralny;
    RULE 12 : IF zb IS ujemny AND zp IS dodatni
                THEN z IS neutralny, u IS mala;

END_RULEBLOCK
END_FUNCTION_BLOCK
```