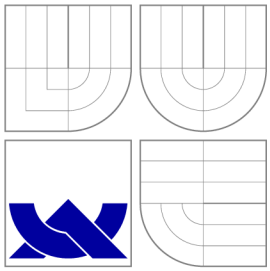
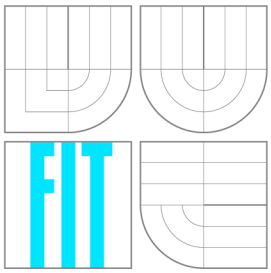


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SIMULACE ŘÍZENÍ DOPRAVY POMOCÍ SEMAFORŮ

SIMULATION OF TRAFFIC CONTROL WITH TRAFFIC LIGHTS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETER MICHALÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAROSLAV ROZMAN, Ph.D.

BRNO 2014

Abstrakt

Práce se zaobírá simulací řízení dopravy pomocí inteligentních dopravních systémů. V rámci této práce byl vytvořen simulátor řízení napsaný v jazyce Java. Simulátor využívá pro rozhodování fuzzy logiku a pracuje na základě reálných dopravních dat. V práci jsou taktéž popsány způsoby řízení dopravy a vyhodnocení počítačové simulace.

Abstract

The aim of work is traffic simulation using intelligent traffic systems. In the work was created traffic control simulator written in Java. Simulator uses fuzzy logic for deciding and operates under real traffic data. In this work are also described methods of traffic control as well as computer simulation evaluation.

Klíčová slova

simulace, inteligentní řízení dopravy, fuzzy logika

Keywords

simulation, intelligent traffic control, fuzzy logic

Citace

Peter Michalík: Simulace řízení dopravy pomocí semaforů, bakalářská práce, Brno, FIT VUT v Brně, 2014

Simulace řízení dopravy pomocí semaforů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Rozmana, PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Peter Michalík
20. května 2014

Poděkování

Děkuji svému vedoucímu, Ing. Jaroslavovi Rozmanovi, PhD., za odborné vedení při tvorbě této práce. Taky bych rád poděkoval panu Ing. Michalovi Švandovi z Brněnských komunikací a.s. za poskytnutá data.

Obsah

Kapitola 1

Úvod

Jedným z hlavných problémov súčasných miest je zlá dopravná situácia. Jej príčiny môžu byť rôzne, od sociálnych až po ekonomické. Faktom však je, že veľkosť vozového parku obyvateľstva má v posledných rokoch rastúci charakter [6]. Hlavne väčšie mestá sa preto snažia zatraktívniť hromadnú dopravu a tým odľahčiť miestne komunikácie. Situáciu však treba riešit komplexne a tak jedným zo smerov, ktorým je možné sa uberať, je budovanie inteligentných systémov pre riadenie dopravy.

Táto práca sa preto zaoberá riadením dopravy pomocou semaforov. Cieľom práce je vytvoriť simulátor, ktorý bude na základe získaných dát simulovať dopravu v centre Brna. Konkrétne sa práca zameria na úsek križovatiek v centre Brna.

Spôsob riadenia križovatiek v mestách je uvedený v druhej kapitole. Spolu so spôsobom riadenia je v kapitole uvedený prehľad jednotlivých zariadení, ktoré sa v súčasnosti pri riadení dopravy na križovatkách používajú. Takisto sú v kapitole uvedené rôzne spôsoby inteligentného riadenia dopravy. Tretia kapitola sa zameria na návrh a implementáciu simulátora, ktorý bude simulovať konkrétny dopravný úsek. Takisto v tejto kapitole bude popísaný spôsob riadenia jednotlivých križovatiek.

V predposlednej kapitole budú zhodnotené výsledky simulátora. Tie sú založené na reálnych dátach z prostredia križovatiek, zberom ktorých sa tiež bude kapitola zaoberať.

Kapitola 2

Analýza

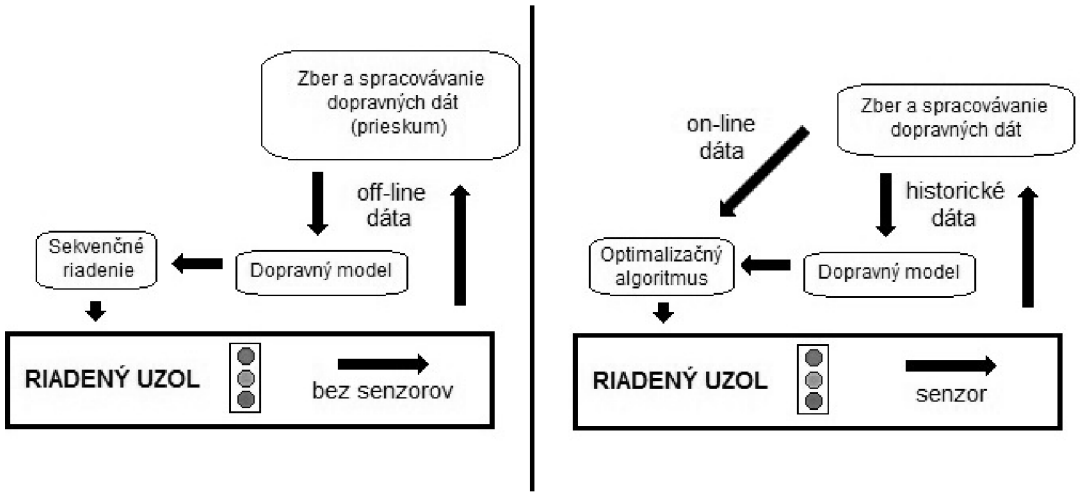
Táto kapitola poskytuje bližšie informácie o riadení dopravy. Popísané sú v nej technické prostriedky využívané pri dopravnom riadení a spomenuté sú aj používané systémy na riadenie dopravy. Bližšie budú popísané spôsoby inteligentného riadenia dopravy.

2.1 Riadenie dopravy

Riadenie dopravy je neoddeliteľnou súčasťou dopravných systémov. Hlavným prínosom je optimalizácia riadenia dopravného procesu. Medzi základné možnosti riadenia dopravy môžeme popri značkách zaradiť aj svetelné signalizačné zariadenia (ďalej SSZ). Práve tie sú hlavným prostriedkom regulácie a optimalizácie dopravy v mestách. Riadenie dopravy na križovatkách sa realizuje na základe signálnych plánov. Tie obsahujú údaje o tom, ktoré pruhy budú v istom čase spustené, poradie striedania množín pruhov a čas, ktorý bude určitá množina pruhov spustená. Existujú dva spôsoby riadenia dopravy:

- riadenie pevným signálnym plánom
- riadenie dynamickým signálnym plánom

Pri riadení pevným signálnym plánom sa systém rozhoduje vždy rovnako a to na základe dát o doprave získaných pozorovaním z prieskumu. Dynamický signálny plán umožňuje prispôsobovať jednotlivé fázy a tým reagovať na potreby aktuálnej premávky. V systéme sa využívajú senzory, tým pádom sú dáta vždy aktuálne a na ich základe sa vykonáva optimalizačný algoritmus.



Obr. 2.1: Riadenie pevným (vľavo) a dynamickým (vpravo) signálnym plánom.

2.1.1 Dopravné detektory

Dopravné senzory sú základom pre aplikovanie techniky inteligentných dopravných systémov. Detekcia slúži hlavne k získaniu informácií o prítomnosti vozidiel v daných detekčných zónach. Pomocou detektorov môžeme získať dopravné parametre a informácie, ktoré majú vplyv na riadenie a organizovanie dopravy akými sú napríklad intenzita dopravného prúdu, rýchlosť vozidiel a obsadenosť jazdných pruhov alebo klasifikácia vozidiel podľa ich kategórie. Ich využitie zefektívňuje prevádzku oblastí a umožňuje tak predchádzať dopravným kongesciám počas času dopravných špičiek. V súčasnej dobe existuje množstvo rôznych druhov senzorov, ktoré využívajú rôzne fyzikálne princípy.

Indukčné slučky

Jednými z najpoužívanejších senzorov sú indukčné slučky vo vozovke [11]. Indukčné slučky sú tvorené drôtom a majú určitý počet závitov. Sú zabudované do vozovky a tým pádom môžu zbierať údaje o intenzite dopravy, rýchlosti, skladbe dopravného prúdu, hustote dopravy, atď. Údaje sú prenášané do radiča SSZ. Ak sa požaduje meranie rýchlosti a rozlíšenie kategórií vozidiel, je nutné použiť dve slučky.

Dopravný uzol je vybavený obvykle dvoma druhmi detektorov:

- predlžovacím - je umiestnený 30 - 50 m pred stopovou čiarou
- výzvovým - je umiestnený tesne pred stopovou čiarou

Dopravný radič SSZ riadi program, ktorý neustále testuje obraz dopravy nad jednotlivými detektormi a na ich základe potom vykonáva optimalizačný algoritmus.

Okrem detektorov v dopravných uzloch, ktoré sa využívajú pre priame riadenie dopravy, sú inštalované takisto strategické detektory pre rozhodovacie procesy na úrovni oblasti alebo celého útvaru. Pre modelovanie a klasifikáciu dopravy je potrebné merať aspoň dva parametre dopravného prúdu. Najčastejšie sa meria intenzita dopravy a rýchlosť či obsadenosť v reze, ktorá je vypočítaná z hustoty dopravy.

Najväčšou nevýhodou indukčných slučiek je, že pri ich inštalácií je narušený povrch vozovky. Z toho dôvodu sa stále viac presadzujú metódy, ktoré povrch vozovky neničia.

Videodetekčné systémy

Na rozdiel od indukčných slučiek, videodetekčné systémy nenarušujú povrch vozovky. K získaniu informácií o stave vozidiel využívajú analýzu obrazu. Tieto informácie prenášajú na vstupy radiča SZZ, poprípade na ostatné dopravné zariadenia. Okrem toho, že nevyžadujú zásah do vozovky patrí medzi výhody tohto systému schopnosť detekcií prekážok v dopravnej premávke ako sú nehody, odstavené vozidlá poprípade iné prekážky znemožňujúce plynulý prejazd daným dopravným úsekom. Po detekovaní problému sú automaticky upozornení pracovníci dispečingu.

Systém je teda využívaný v dopravných systémoch ako zdroj dát vypovedajúcich o aktuálnej situácií. Systém sa využíva na miestach, kde nieje možné alebo vhodné použitie indukčných slučiek - časté rekonštrukcie, nevhodný kryt vozovky, prítomnosť električkovej dopravy, poprípade v historických centrách miest. [4]

Mikrovlné detektory

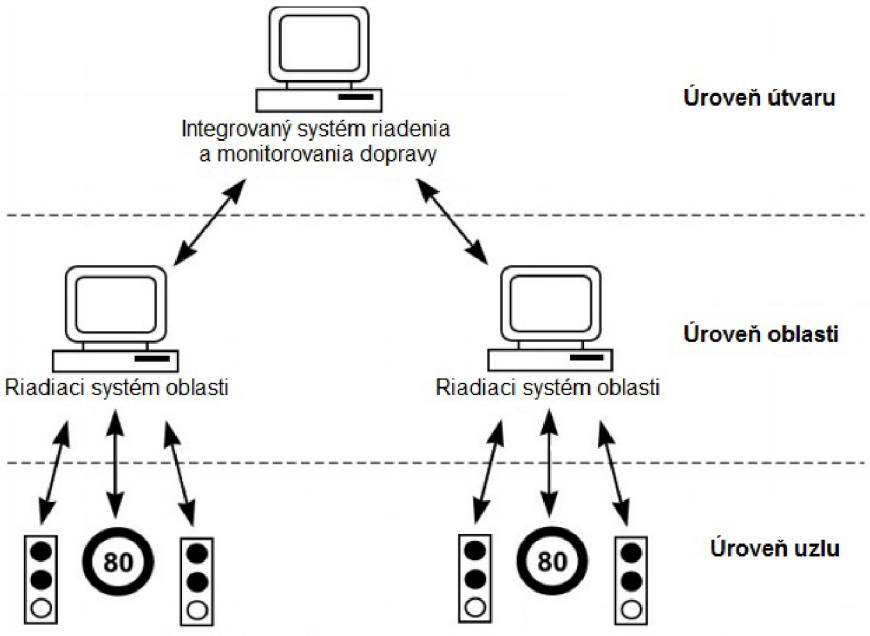
Podobne ako aj videodetektory poskytujú mikrovlné detektory možnosť zberu dopravných dát, bez nutnosti zásahu do vozovky. Pracujú na princípe vysielania a následného príjmu mikrovlných lúčov. Takisto ako už spomínané detekčné systémy aj tieto informujú radič SZZ o obsadenosti detekčnej zóny vozidlom.

Mikrovlné detektory pracujú spoľahlivo za každého počasia, klasifikujú prechádzajúce vozidla a takisto umožňujú meranie rýchlosti. Napriek spomínaným výhodám ako aj jednoduchej montáži sa na križovatkách nepoužívajú až v takej miere. [4]

2.1.2 Architektúra telematického systému

Pri riadení mesta sa zvyčajne používa trojvrstvová architektúra telematického systému. Najnižšiu úroveň, teda úroveň uzlu, tvoria zariadenia, ktoré priamo zasahujú do riadenia dopravy, teda svetelné signalizačné zariadenia a ich radiče, parkovacie systémy, svetelné tabule, systémy riadenia tunelov atď. Nad uzlovou úrovňou stojí úroveň oblastí. Oblasti by mali byť, pokiaľ je to možné, uzatvorenými celkami a takmer bez väzby na okolie. Obsahujú takisto riadiaci systém, ktorý príslušnú oblasť ovláda. Na najvyššej - mestskej úrovni sú integrované systémy riadenia jednotlivých oblastí. Systémy mestskej úrovne tiež monitorujú dopravu v mestách. Najdôležitejšie činnosti telematického systému sú:

- optimalizácia a riadenie dopravnej siete
- informovanie a navigácia
- preferencie mestskej hromadnej dopravy
- zaistenie náväznosti na ďalšie systémy a subsystémy



Obr. 2.2: Trojvrstvová architektúra telematického systému. [2]

2.1.3 Riadenie dopravy v mestách

Riadenie mestskej dopravy v reálnom čase sa rozdeľuje na centralizované a decentralizované.

Centralizované riadenie

Riadenie s centralizovanou inteligenciou využíva všetky detektory v danej oblasti. V reálnom čase vyhodnocuje ich výstupy a na základe optimalizačného výpočtu mení riadené parametere. Tento spôsob je však ekonomicky a technicky náročný.

Decentralizované riadenie

Decentralizovaná inteligencia riadenia spočíva v tom, že dopravný uzol okamžite reaguje na dopravné stavy. Na vyššej úrovni je riadiaci počítač, ktorý má funkciu koordinátora jednotlivých uzlov siete. Decentralizovaná inteligencia riadenia zbiera dáta od všetkých detektorov. Na základe aktuálnej dopravnej situácie mení dĺžky cyklu, prípadne dĺžku zelených alebo skladby jednotlivých fáz. Signalizačné zariadenia môžu byť usporiadané buď plošne, alebo sériovo a sú riadené adaptívne v určitom časovom rastre pohybujúcom sa od 10 do 30 minút. [2] Príkladom sú systémy MOTION a TASS.

TASS TASS (Traffic Actuated Sinal plan Selection) [2] je softvérový nástroj, ktorý výberom riadených signálnych plánov na základe aktuálnej dopravnej situácie z detektorov reaguje na dopravu. Aktuálna dopravná situácie je vyhodnocovaná z meraných dopravných dát ukladaných v systéme riadenia. Pomocou zoznamu podmienok a prostredníctvom rozhodovacích tabuliek sa vyberá konkrétny signálny plán. TASS pracuje na dvoch úrovniach:

- strategická úroveň - detekcia dopravnej situácie v riadenej oblasti a jej okolí

- taktická úroveň - výber signálneho plánu pre skupinu radičov v dostatočnej vzdialenosti pred oblasťou

Na strategickej úrovni sú detekované rôzne dopravné situácie. Pre každú situáciu existuje aspoň jeden signálny plán pre každú križovatku, ktorý zodpovedá charakteru dopravných podmienok danej situácie (štandardná doba cyklu, koordinácia atď). Na taktickej úrovni sa vyberajú signálne plány pre všetky radiče v skupine TASS v závislosti na aktuálnych dopravných podmienkach. Cieľom je reagovať rýchlo na fluktuácie dopravného prúdu v časti riadenej oblasti. To býva zaistené výberom alternatívneho signálneho plánu, ktorého základné charakteristiky (doba cyklu, koordinácia, a pod.) by mali korešpondovať s práve aktívnym základným signálnym plánom. Tým sa zabráni vzniku umelo vyvolaných porúch dopravného prúdu spôsobených prepínaním signálnych plánov.

MOTION Systém MOTION (Method for the Optimalisation of Traffic Signals In On-line controlled Networks) [2] je makroskopický riadiaci systém pre optimalizáciu riadenia dopravných tokov v mestskej cestnej sieti. Základnou koncepciou metódy MOTION je schopnosť kombinovať výhody účinného modelu dopravnej siete pre najdôležitejšie dopravné prúdy v sieti s možnosťou takmer okamžitej reakcie na zmenu dopravnej situácie prostredníctvom miestneho riadenia v križovatkách. Pre umožnenie tohoto pružného riadenia sa údaje o premávke zhromažďujú, kompletizujú a analyzujú. Na základe tejto analýzy sa časti signálnych programov (doba cyklu, sled fází, dĺžky zelených) optimalizujú pre všetky križovatky v sieti a vytvárajú sa nové signálne programy. MOTION pracuje v troch úrovniach.

- strategická úroveň - každých 10–15 minút sa určuje doba cyklu, rozdelenie zelených, základný sled fází a parametre koordinácie
- taktická úroveň - po 60–90 sekundách môže byť základný sled fází ovplyvňovaný metódou miestneho riadenia pre vypočítanú dĺžku cyklu napríklad vložením špeciálnej fázy (lokálny sled fází, napr. pre preferovanie verejnej hromadnej dopravy)
- operačná úroveň - zhruba každú sekundu je možné reagovať na miestne riadenie pre jednotlivé vozidlá (dĺžka zelenej, ktorí reaguje na jednotlivé vozidlá, poprípade preferenciu MHD)

V strategickej úrovni môžeme priradiť verejnej doprave vyššie priority ako individuálnej automobilovej doprave. Radiče si pri miestnom riadení zachovávajú vysokú úroveň samostatnosti pri reakciách na dopravné situácie. Povolené medze sú nastavené centrálnym riadením. Spolu so systémom MOTION sa používa aj jeho modul CIM (Congestion and Incident Management), ktorý umožňuje vykonávať management riadenia dopravy pri nehodách, kongesciách a mimoriadnych situáciach s rôznymi stratégiami riadenia. Modul CIM umožňuje na úrovni siete výber reakcií na nehody alebo kongescie v závislosti na aktuálnej dopravnej situácii.

Preferencia MHD

Aby boli prostriedky mestskej hromadnej dopravy [12] schopné dodržiavať svoj cestovný poriadok, je pri riadení dopravy zohľadniť ich prítomnosť na križovatke a na základe toho vykonať opatrenia, aby prešli daný úsek čo najrýchlejšie. Rozlišujeme dva druhy preferencí.

Pasívna preferencia Tú využívajú hlavne električky pomocou trolejových kontaktov, poprípade autobusy využívajúce slučkové detektory. Nevýhodou tejto preferencie býva nevhodné umiestnenie senzorov. Trolejové kontakty patria medzi najpoužívanejšie pasívne detektory využívané pri zvýhodňovaní MHD.

Aktívna preferencia Tento typ preferencie využíva bezdrôtový spôsob detekcie a priameho prihlásenia do radiča križovatky. Vozidlo vybavené palubným počítačom vyhodnotí signál z infračerveného majáku o polohe vozidla, Následne palubný počítač vozidla vyšle rádiovú informáciu do radiča SSZ, ktorá obsahuje informácie o linke, vzdialenosti a smeru vozidla pred križovatkou. Radič informáciu vyhodnotí a nastaví príslušný sled fází alebo fázu pre vozidlo MHD.

2.2 Spôsoby inteligentného riadenia dopravy

Matematický model riadenia SSZ sa konštruuje veľmi obtiažne. Hlavný dôvod spočíva v stochastickom charaktere príjazdu vozidiel na svetelnú križovatku. Preto sa pri popise riadenia využívajú rôzne spôsoby. Medzi najrozšírenejšie spôsoby umelej inteligencie pri riadení dopravy SSZ patrí fuzzy logika využívaná k vlastnému riadeniu, evolučné algoritmy, najčastejšie genetického charakteru ako optimalizačný nástroj, multiagentové systémy a v neposlednej rade neurónové siete.

2.2.1 Fuzzy logika

Fuzzy logika [7] je relatívne jednoduchý spôsob, ktorý umožňuje formalizovať znalosti, ktoré sú vo vágnej jazykovej forme. Pomocou Fuzzy Aproximačného Teorému môžeme ľubovoľnú spojitú funkciu aproximovať s ľubovoľnou presnosťou pomocou konečnej množiny fuzzy premenných, hodnôt a pravidiel. Zo všeobecného hľadiska patrí k najvýraznejším výhodám fuzzy logiky:

- jednoduchá implementácia
- jednoduchý a prirodzený koncept
- flexibilita
- tolerancia vágnych informácií, možnosť využitia skúseností expertov
- možnosť modelovať nelineárne systémy ľubovolného stupňa komplexnosti
- zvýšená priehľadnosť designu vďaka lingvistickej znalosti

Vo fuzzy logike sa logické výroky ohodnocujú stupňom príslušnosti, ktorého hodnoty sú v rozmedzí od 0 do 1. Vo výrokovej a predikátovej logike sú výroky buď pravdivé alebo nepravdivé (buď 1 alebo 0). Kvôli tomu je fuzzy logika vhodnejšia pre rozhodovacie úlohy. Funkcia príslušnosti umožňuje priradiť príslušnosť k množinám v rozmedzí od 0 do 1 vrátane. Vďaka tomu je možné pomocou fuzzy logiky vyjadriť pojmy ako „trochu“ „dosť“ alebo „veľa“ - umožňuje vyjadriť čiastočnú príslušnosť k množine.[1]

Fuzzy riadenie Jednou z prvých oblastí, kde našla fuzzy logika uplatnenie bolo riadenie a regulácia. Fuzzy sa tu využíva dvoma spôsobmi:

- priame fuzzy riadenie - fuzzy algoritmus priamo prijíma hodnoty z riadeného systému a reaguje na ne tak, že posiela priamo riadiace zásahy do systému
- nepriame fuzzy riadenie - fuzzy algoritmus síce prijíma hodnoty zo systému, ale spracovanie výstupov a riadiace zásahy robí klasický lineárny regulátor. Fuzzy regulátor len prepína medzi viacerými lineárnymi regulátormi, podľa toho, v ktorom pracovnom bode je systém

Proces vyhodnocovania každej entity riadenej fuzzy logikou - fuzzy blokom sa skladá z nasledujúcich častí:

- fuzzyfikácia - prevod vstupných hodnôt na fuzzy množiny
- aplikácia tabuľky pravidiel - samotný riadiaci algoritmus, ktorý nastavuje operátor podľa systému
- fuzzyfikácia - prevod vstupu na fuzzy množiny
- agregácia - spojenie výstupu pravidiel
- defuzzyfikácia - vyrába podľa výstupu pravidiel výstupné hodnoty

Pri použití fuzzy logiky sa zvyšuje efektivita pri prejazde križovatkou. Nevýhoda je však pri viacerých križovatkách. Fuzzy logika je vhodná len pre jednu križovatku a pri viacerých nastávajú problémy s definovaním rozhodujúcich pravidiel, pretože každá križovatka má svoje vlastné špecifiká a sústavu križovatiek ovplyvňuje veľké množstvo faktorov.

2.2.2 Expertné systémy

Expertným systémom [10] môžeme nazvať počítačový systém hľadajúci riešenie problému v rozsahu určitého súboru tvrdenia alebo určitého zoskupenia znalostí, ktoré boli formulované expertmi pre danú špecifickú aplikačnú oblasť. Je založený na reprezentácií poznatkov expertov, ktoré využíva pri riešení zadaných problémov. Pre riešenie vymedzenej triedy úloh v určitých problémových oblastiach využíva kooperujúce programy. Sú teda vybavené znalosťami experta danej oblasti, v ktorých rozsahu je schopný uskutočňovať rozhodnutia rýchlosťou a kvalitou sa expertovi vyrovnávajúcich.

Expertné systémy sú určené pre podporu rozhodovania pri riešení zložitých úloh. Základné triedy problémov, ktoré sú vhodné pre riešenie expertnými systémami sú:

- interpretácia - rozpoznávanie situácií z údajov, ktoré ich popisujú
- predikcia - odvodenie očakávaných dôsledkov danej situácie
- diagnostika - určenie stavu systému z pozorovateľných prejavov jeho správania
- konštruovanie - výber a zostavenie objektov do určitého funkčného celku pri daných obmedzujúcich podmienkach
- plánovanie - zostavenie postupnosti akcií za účelom dosiahnutia požadovaného cieľa

- monitorovanie - sledovanie a porovnávanie údajov zodpovedajúcich určitej situácií za účlom zisťovania a následného odstraňovania odchýliek od očakávanej situácie
- opravovanie - výber, zostavenie a uskutočnenie postupnosti akcií odstraňujúcich odchýlky alebo chybové stavy
- učenie - diagnostika, ladenie a upravovanie vedomostí študenta
- riadenie - interpretácia, predikcia, monitorovanie a úprava správania systému

Expertné systémy by teda podľa tried uplatnenia boli vhodné pre riadenie dopravy. Počítačová realizácia takéhoto riešiaceho systému však zdôrazňuje význam znalostí spojených s riešeným problémom. Preto by pre riadenie bolo potrebné definovať tie správne pravidlá, na ktoré treba expertov danej problematiky. Ich absencia by mohla znamenať, že sa celkové riadenie nezlepší, ale naopak zhorší.

2.2.3 Neurónové siete

Umelá neurónová sieť je prostriedkom pre spracovanie komplexných dát. Sú inšpirované architektúrou ľudského mozgu a tým pádom sú schopné sa učiť a analyzovať rozsiahle komplexné množiny dát, ktoré lineárne algoritmy zvládajú len ťažko.

Umelá neurónová sieť funguje tak, že vytvára spojenie medzi mnohými rôznymi procesnými prvkami, z ktorých je každý analogický so samostatným neurónom v biologickom mozgu. Každý neurón dostáva mnoho vstupných signálov. Potom, na základe vnútorného vyvažovacieho systému, produkuje jednotlivý výstupný signál, ktorý je typicky zasielaný ako vstup inému neurónu.

Model neurónu (perceptrón) prijíma vstupné signály cez synaptické váhy tvoriace váhový vektor. Vstupný vektor sa nazýva vzor alebo obrazec. Zložky vstupného vektora môžu nadobúdať reálne alebo binárne hodnoty. Zložky váhového vektora sú reálne čísla.

Dôležitou vlastnosťou neurónových sietí je schopnosť učenia, takisto ako majú aj biologické neuróny. Umelé neurónové siete typicky začínajú s náhodnými váhami pre všetky svoje neuróny. To znamená, že nemajú žiadne informácie a musia byť učené pre riešenie konkrétneho problému. Existujú dva spôsoby učenia:

- učenie bez učiteľa - vystavenie veľkému množstvu dát, smerujúci k odhaleniu zákonností a súvislostí v týchto dátach
- učenie s učiteľom - vykonávanie špeciálnych úloh, učiteľ vyhodnocuje, či je prístup správny, na základe toho sú neurónové váhy posilnené, alebo oslabené, využíva sa pre rozpoznávanie a aplikácie na riešenie problémov

Neuróny sa rozdeľujú do rôznych vrstiev. Vstupná vrstva dostáva vstupné údaje, výstupná zasa vytvára výstup. Zvyčajne sa tiež používajú vnútorné skryté vrstvy. V tejto štruktúre nie je možné predvídať presný dátový tok.

Umelé neurónové siete sa ukázali ako veľmi užitočné v rozmanitých aplikáciach reálneho sveta, ktoré pracujú s komplexnými a často neúplnými dátami ako napríklad rozpoznávanie reči, alebo obrazu. V obore dopravy by bolo možné použiť neurónové siete tiež. V procese učenia by pre ne boli vstupom informácie zo senzorov. Tým by sa priraďovali určité signálne plány. Neurónová sieť by bola po procese učenia schopná riadiť dopravu za všetkých podmienok, na aké bola naučená. [5]

2.2.4 Multiagentové systémy

Multiagentový systém [9] je systém skladajúci sa z niekoľkých softvérových agentov, ktorí sú schopní vzájomnej spolupráce, ktorá vedie k riešeniu daného problému.

Agent je samostatná entita umiestnená do určitého prostredia, ktorá môže vykonávať nejakú činnosť. Inteligentní agenti sú agenti, ktorí vykonávajú svoju činnosť za účelom splnenia svojich úloh. Inteligentný agent musí tým pádom chápať svoje okolie a na základe podnetov zvonka vykonávať príslušné akcie pre splnenie cieľa. Podnety môžu prichádzať v rôznych podobách, napr. vizuálnej, hmatovej, hovorenej, textovej atď. Rozhodovanie prebieha tiež niekoľkými spôsobmi v závislosti na tom, či má agent kompletnú, alebo len čiastočnú znalosť svojho prostredia, či jedná sám, alebo spolupracuje, poprípade súťaží s ostatnými agentmi apod. softvérový agent je inteligentný agent vsadený do sveta počítačov, ktorý plní požiadavky užívateľov.

multiagentové systémy sa podobne ako neurónové siete vedia učiť a vyvíjať. Výhodou je tiež, že dokáže plánovať určité zmeny dopredu a zároveň flexibilne reagovať na zmeny.

Pri dopravnej simulácií, ktorá využíva multiagentové systémy majú agenti role jednotlivých častí dopravného úseku. Teda v systéme sú prítomní agenti jednotlivých križovatiek, úsekov ciest a oblastí. Agenti využívajú informácie získané z prostredia, teda dopravných senzorov, a na základe týchto údajov a správ od ostatných agentov tvoria signálny plán. Takisto je využívaný predikčný model, ktorý dokáže určité dopravné situácie predpokladať podľa aktuálne prebiehajúcej situácie.

2.2.5 Evolučné algoritmy

Evolučné algoritmy [8] patria medzi globálne optimalizačné algoritmy. Ich cieľom je prehľadať priestor a tým nájsť optimálne riešenie. Ak je priestor možných riešení malý, je možné jednotlivé riešenia vyhodnotiť hrubou silou a vybrať z nich to najlepšie. Avšak vo väčšine prípadov veľkosť priestoru rastie so zložitosťou problému. Pri takých problémoch nie je efektívne a ani možné prehľadávať celý priestor riešenia a preto je nájdenie optimálneho riešenia v krátkom čase nemožné. Evolučné algoritmy však tento problém riešia.

Týmto spôsobom teda môžeme riešiť obtiažne problémy, u ktorých je veľký priestor riešenia. Tiež je možné týmto postupom riešiť ťažko formulovateľné problémy. Vedia takisto nájsť viac optimálnych riešení. Pri veľmi obtiažnych problémoch sú však schopné nájsť len približné riešenia. Aj keď zložitosť rastie len lineárne s veľkosťou populácie a počtom generácií, výpočet nemusí byť rýchly vzhľadom k tomu, že vyhodnotiť riešenie býva časovo náročné.

Evolučné algoritmy sa dajú použiť aj v doprave. Pri riešení riadenia križovatiek však treba vhodne zakódovať stav systému do genotypu, na ktorý sa algoritmus aplikuje. K nájdeniu úspešného riešenia je tiež potrebné správne určiť funkciu, ktorá určí správnosť chromozómu a spôsob určenia výsledku a tým pádom ukončenia evolúcie. Výstupom algoritmu bude teda taký genotyp, ktorý po rozkódovaní bude obsahovať najlepšie riešenie signálnych plánov pre danú situáciu.

Kapitola 3

Návrh a implementácia

Táto kapitola popisuje tvorbu simulátora ako aj tvorbu riadiacej logiky. Najskôr sa zameria na návrh konkrétnej časti a potom priblíži jej implementáciu.

3.1 Simulátor

Spočiatku bolo potrebné si určiť, aký model dopravy bude pre prácu použitý. Kvôli tomu, že modelom bude úsek križovatiek okolo centra Brna, takzvaný malý okruh, bola vybratá makroskopická simulácia ako vhodná [3]. V makroskopickej simulácií sa presne nezohľadňuje správanie vodiča a jeho neurčitosť. Túto časť by bolo možné do simulácie doplniť, avšak bolo usúdené, že rozdiely v simuláciách by boli zanedbateľné.

3.1.1 Prvky simulátora

Ďalším krokom bolo určenie častí, z ktorých sa model bude skladať. Pre simuláciu riadenia dopravného úseku križovatiek je potrebné zohľadniť nasledujúce:

- vstup a výstup vozidiel do/zo systému (generovanie)
- dopravné senzory
- signalizačné zariadenia
- jazdné pruhy pripojené na križovatku
- štatistiky
- užívateľské rozhranie

Generovanie vozidiel Vozidlá budú generované do konkrétneho jazdného pruhu križovatky podľa dát reálnej premávky. Počas simulácie budú vstupovať pomocou pruhu do jednotlivých križovatiek a po opustení križovatky sa zo systému vytratia. Ak aj pokračujú k ďalšej križovatke, ktorá je súčasťou simulovaného systému, budú pri nej vygenerované znova. Takéto riešenie je výhodné najmä z toho dôvodu, že nie je potrebné brať do úvahy situáciu, že vozidlo opustilo systém niekde medzi križovatkami. Vstup vozidiel bude teda realizovaný generátorom. Podľa získaných dát z počtu vozidiel ktoré na danom úseku do systému vstúpia bude vybratý generátor s určitým rozdelením pravdepodobnosti. Pre proces počtu príchodov za jednotku času sa používa Poissonovo rozdelenie, ktoré sa dá ľahko previesť na exponenciálne rozdelenie, ktoré udáva čas medzi príchodmi.

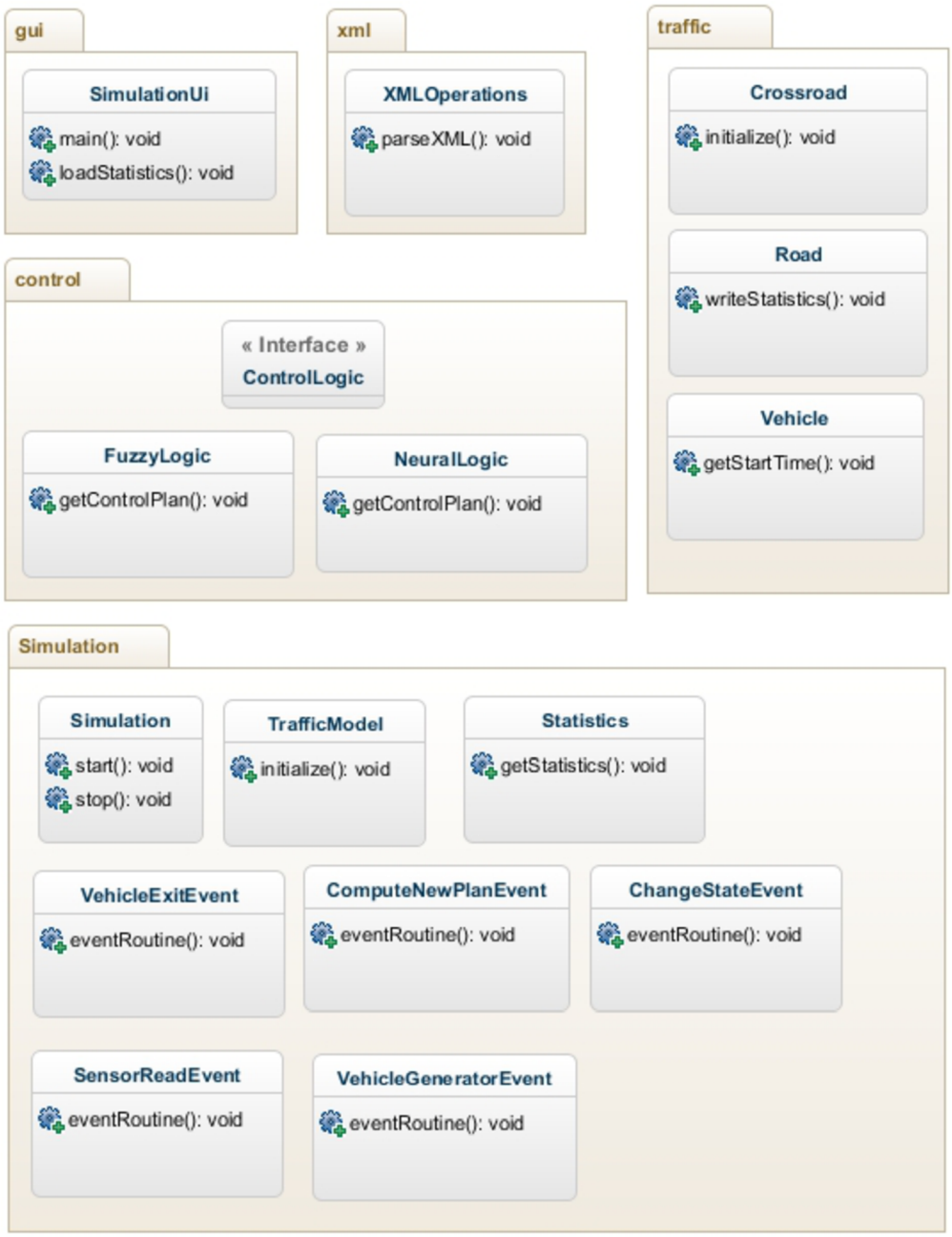
Dopravné senzory Ďalšou dôležitou súčasťou simulácie riadenia sú dopravné senzory. Keďže simulácia má čo najvernejšie napodobňovať reálne prostredie, nemôžeme vykonávať operácie, ktoré nie sú v reálnom svete možné (napr. križovatka sama o sebe nevie, koľko vozidiel čaká v ktorom pruhu). Preto je nutné do simulácie zaradiť aj skutočné prostriedky na získavanie počtu vozidiel a to vo forme dopravných čidiel. Ako bolo uvedené v kapitole 2.1 existuje veľké množstvo dopravných detektorov. Všetky však majú z hľadiska riadenia dopravy jeden účel a to zmerať počet čakajúcich vozidiel. Preto budú v simulácií použité senzory, ktoré budú zisťovať počet vozidiel, ktoré do križovatky vstúpili, teda počet čakajúcich vozidiel.

Signalizačné zariadenia Tieto zariadenia sú základným kameňom riadenia dopravy. Preto je ich súčasť v simulácií považovaná za najdôležitejšiu. Ich hlavnou úlohou bude rozhodovanie a výber signálneho plánu podľa aktuálnej dopravnej situácie. Aktuálnu dopravnú situáciu budú získavať od dopravných senzorov tej križovatky, ktorých súčasťou sú. Na základe týchto dát bude rozhodovací algoritmus vypočítavať vhodný signálny plán.

Jazdné pruhy Vozidlá vstupujúce do križovatky sa pohybujú po vozovke. Aj túto časť dopravného úseku treba teda brať do úvahy. Na vozovke sa tvorí fronta čakajúcich vozidiel. Na vozovku budú tiež napojené senzory, ktoré monitorujú aktuálny počet vozidiel, ktorý po vozovke prešiel za určitú jednotku času.

Štatistiky Kvôli možnosti merať výsledky simulácie, je potrebné zbierať rôzne štatistiky z celého meraného úseku po dobu simulácie. Medzi tieto štatistiky patrí napríklad priemerná dĺžka fronty, doba strávená vo fronte a podobne. Preto bude simulátor musieť zapisovať tieto dáta a takisto ich vedieť počas a hlavne po skončení simulácie vhodne zobraziť.

Užívateľské rozhranie Aj keď sú výsledné štatistiky hlavným výstupom zo simulácie, je viac ako vhodné používateľovi hodnoverne zobrazovať priebeh simulácie. Preto bude celý priebeh simulácie zobrazený v GUI, v ktorom bude mať možnosť prechádzať jednotlivými križovatkami a zobrazovať k nim aktuálne štatistiky.

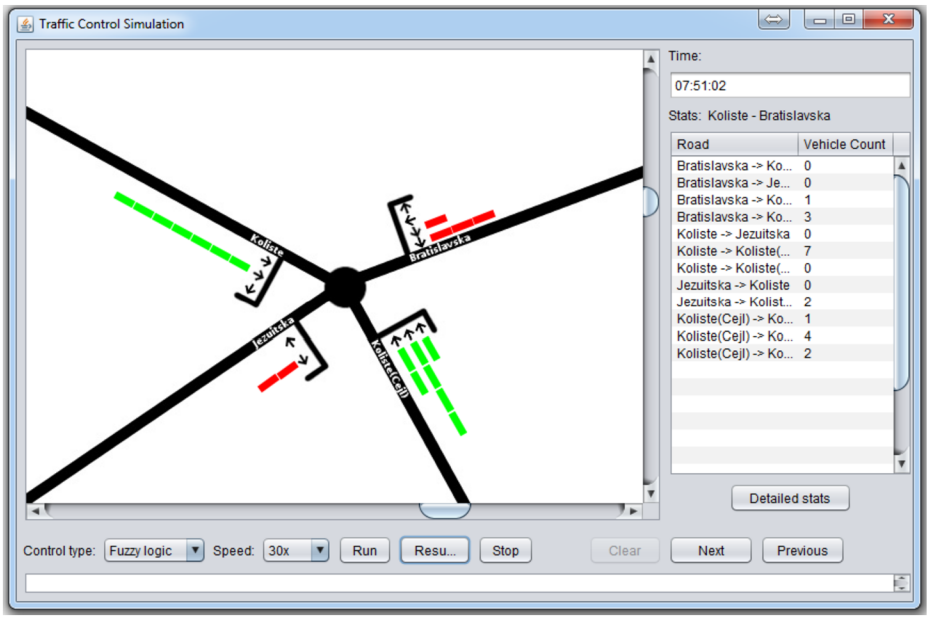


Obr. 3.1: Navrhovaný diagram tried.

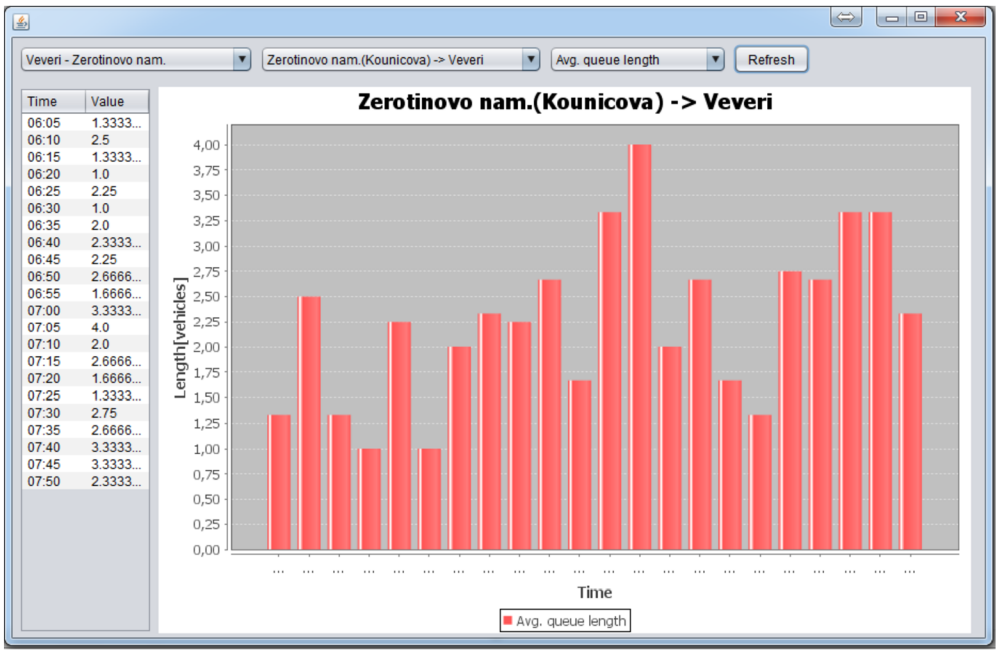
3.1.2 Implementácia simulátora

Po návrhu simulátora bolo možné pristúpiť k jeho implementácií. Implementačným jazykom bola zvolená Java 7. Hlavným kritériom pre výber jazyka bola jeho multiplatformnosť ako aj množstvo použiteľných knižníc.

Grafické užívateľské rozhranie



Obr. 3.2: Hlavné okno užívateľského rozhrania.



Obr. 3.3: Okno s detailnými štatistikami.

Pri návrhu programu sa počítalo s vytvorením grafického užívateľského rozhrania, ktoré by bolo vhodné pre zobrazovanie aktuálneho stavu na jednotlivých križovatkách. Kvalitné užívateľské rozhranie má užívateľovi priniesť presne to, čo od aplikácie požaduje, V tomto

prípade to je zobrazenie aktuálneho stavu simulácie. Tomu zodpovedajú stavy jednotlivých križovatiek.

Pri jednoduchšej aplikácií by možno stačilo vytvoriť len textové rozhranie v ktorom by sa zobrazovali dáta pre každú križovatku. V tejto práci som sa však rozhodol vytvoriť aj grafické rozhranie pre lepšie znázornenie skúmaného problému. Základná myšlienka ktorá bola nakoniec aj zrealizovaná bolo rozdelenie okna programu na zobrazovaciu časť, ktorú tvorí nákres skúmanej časti premávky - na nej sa zobrazujú jednotlivé križovatky spolu so znázornenými vozidlami a na časť, v ktorej sa zobrazujú štatistiky. Pre ovládanie simulácie boli do grafického návrhu pridané tlačidlá. Užívateľské rozhranie je zobrazené na nasledujúcich obrázkoch.

Na implementáciu grafického rozhrania boli použité knižnice `Swing` a `AWT`. Zobrazovacia časť je umiestnená v komponente `JPanel`. Kvôli rozsiahlosti nákresu križovatiek bolo potrebné použiť rolovací box `JScrollPanel`. Jeho pozadie tvorí už spomínaný podklad. Na pozadie sú vykresľované jednotlivé vozidlá. Vykresľovanie je uskutočnené pomocou funkcie `paintComponent()`. Tá, pokiaľ je simulácia spustená, prechádza všetky križovatky a na ne napojené vozovky a volá pre ne metódu `draw()`, ktorá vykresľuje aktuálny počet vozidiel na danej vozovke.

Simulácia

Pre implementáciu simulácie bola zvolená knižnica `DESMO-J`[1], ktorá poskytuje potrebné prostriedky pre diskrétnu simuláciu. Medzi potrebné prostriedky patria:

- základ pre tvorbu modelov udalostí
- základ pre tvorbu jednotlivých entít
- generátory potrebných rozdelení pravdepodobnosti
- fronty
- kalendár na plánovanie udalostí
- zber a uchovávanie štatistík

Bola teda zvolená simulácia riadená udalosťami. Základom je trieda `Simulation`, ktorá vytvára objekt triedy `Experiment` definujúci celý simulačný experiment a objekt modelovej triedy `TrafficModel`, ktorá vychádza zo všeobecnej triedy `Model`. Trieda `TrafficModel` sa teda stará o inicializáciu všetkých súčastí modelu. A teda jeho udalostí, generátorov a front. Takisto vytvára množinu križovatiek, ktoré definuje trieda `Crossroad`, a vozoviek na nich pripojených, definovaných triedou `Road`.

Keďže zvolená simulácia je riadená udalosťami bolo potrebné vytvoriť také udalosti, ktoré zodpovedajú dianiu v premávke a jej riadení. Udalosti sú pri štarte simulácie naplánované na čas 0 sekúnd. Každá z nasledujúcich udalostí sa po vykonaní naplánuje na vykonanie v budúcnosti.

`ChangeStateEvent` Udalosť ktorá vykonáva zmenu aktívnej časti cyklu signálneho plánu. Pracuje s aktuálnym signálnym plánom pre danú križovatku. Po zmene na novú časť cyklu sa zistí dĺžka, ktorú má trvať a na ten čas je naplánovaná ďalšia zmena.

[1] http://desmoj.sourceforge.net/home.html

`ComputeNewPlanEvent` Udalosť slúžiaca na výpočet nového signálneho plánu. Po uplynutí určitého času sa podľa aktuálnych údajov získaných zo senzorov vypočíta nový signálny plán, ktorý je používaný až do času ďalšieho výpočtu. Po tomto čase sa plán počíta nanovo.

`SensorReadEvent` Táto udalosť slúži na zistenie stavu dopravnej situácie, konkrétne podľa dopravných senzorov. Tie počítajú počet vozidiel, ktoré prejdú križovatkou. Po uplynutí určitej doby, keď nastane táto udalosť, sú dáta zo senzorov uložené a senzory vynulované. Uložené dáta sú potom používané pre výpočet nových signálnych plánov.

`VehicleGeneratorEvent` Generátor vozidiel sa stará o príchod vozidiel do systému. Pracuje pre každú vozovku napojenú na križovatku samostatne. Generátor vždy vytvorí jedno vozidlo a vloží ho do čakacej fronty križovatky. Aktivuje sa v čase príchodu ďalšieho vozidla. Ten je počítaný generátorom s exponenciálnym rozdelením pravdepodobnosti, ktoré vyjadruje časové rozpätie medzi príchodmi vozidiel do systému. Parameter pre generátor je pre každú vozovku iný.

`VehicleExitEvent` Udalosť odoberajúca vozidlá z fronty. Je plánovaná na každé 2 sekundy. Táto hodnota bola zvolená ako čas, ktorý potrebuje vozidlo na opustenie križovatky z jej hranice. Pri zavolaní udalosti sa vždy kontroluje, ktoré vozovky majú od riadenia nastavený stav voľno. Z tých sú následne odobrané vozidlá z ich front.

Import dát z XML súboru

Pre simulačné experimenty je dôležité, aby sa dáta potrebné pre simuláciu mohli meniť. Z tohto dôvodu bolo zvolené, že údaje o križovatkách ako počet ich pruhov, názvy pruhov a hlavne hustotu premávky v konkrétnych časoch bude možné meniť bez rekompilácie programu. Preto bol naimplementovaný import týchto dát z `XML` súborov. Pre implementáciu bola použitá knižnica knižnica `xstream`[1].

Uloženie dát v XML súbore je nasledovné:

```
<crossroad>
  <name>Veveri - Zerotinovo nam.</name>
  <planCount>3</planCount>
  <roads>
    <road>
      <id>c01_2to4</id>
      <name>Zerotinovo nam.(Kounicova) -> Veveri</name>
      <maxSize>20</maxSize>
      <active>
        <int>0</int>
      </active>
      <genTime>
        <int>17</int>
        <int>22</int>
        ...
        <int>23</int>
        <int>22</int>
```

[1] http://xstream.codehaus.org/

```
      </genTime>
    </road>
    <road>
      ...
    </road>
  </roads>
</crossroad>
```

Počiatočným tagom je `<crossroad>`. `<name>` je jej reálne meno a `<planCount>` je počet častí signálneho plánu križovatky. Každá križovatka obsahuje vozovky `<roads>`. `<road>` je potom počiatočnou značkou vozovky. Tá má svoje programové `<id>`, reálne meno `<name>`, veľkosť `<maxSize>` udávanú ako maximálny počet vozidiel, ktorý sa na danú vozovku zmestí a taktiež definíciu, na ktorú časť signálneho plánu zareaguje `<active>`. Keďže tých môže byť viac, každá je označená tagom `<int>`. Poslednou hodnotou sú počty vozidiel, ktoré danou vozovkou prejdú `<genTime>`. Používajú sa ako parameter pre generátor vozidiel. Počty sú vždy pre 15-minutový úsek a keďže ich je viac, sú znova označené tagom `<int>`.

Spracovanie štatistík

Na získavanie poznatkov zo simulovaného modelu je potrebný nejaký výstup. V prípade riadenia dopravy boli ako výstupné určené tri údaje:

Priemerná dĺžka fronty Údaj o dĺžke fronty čakajúcich vozidiel je meraný vždy v momente, kedy vozidlá dostanú povolenie na prejazd križovatkou, teda sa na signalizačnom zariadení zmení červená na zelenú. Vtedy si simulátor zapíše aktuálnu hodnotu dĺžky fronty ako aj, že vyčítanie nastalo.

Priemerný čas strávený vo fronte V momente príchodu vozidla do fronty je tomuto vozidlu zapísaný čas vstupu. Pri prejdení križovatkou sa vypočíta strávený čas pre každé jedno vozidlo. Priemer týchto časov je potom započítaný do štatistík.

Počet vozidiel Tento údaj jednoducho zaznamenáva počet vozidiel, ktoré križovatkou pre daný smer prešli.

Štatistiky sa počítajú vždy pre päťminútové časové bloky. Následne sú zobrazované ako histogram do grafu. Na zobrazovanie grafu bola využitá knižnica `JFreeChart`[1].

3.2 Riadenie dopravy

3.2.1 Návrh riadenia

Pri návrhu riadenia bolo usúdené, že riadenie bude prebiehať decentralizovane. To znamená, že signalizačné zariadenia nebudú mať medzi sebou žiadneho arbitra. Dôraz bude kladený na dynamické riadenie, čiže zariadenia nebudú pracovať podľa pevných signálnych plánov vypočítaných z predchádzajúceho pozorovania dopravnej situácie, ale budú vypočítavané dynamicky. Riadenie a výpočet signálnych plánov teda bude odvodený podľa aktuálnej situácie získanej z dopravných senzorov a správ od ostatných signalizačných zariadení.

[1] http://www.jfree.org/jfreechart/

Signálny plán Pre riadenie aktuálneho stavu na križovatke sa využíva signálny plán. Pre túto prácu je definovaný signálny plán ako postupnosť povoľovacích príkazov pre vstup do križovatky pre jednotlivé pruhy. Každý jeden príkaz má určité trvanie a reagujú naň určité jazdné pruhy. To znamená, že napríklad pre križovatku so štyrmi vstupnými jazdnými pruhmi (pre každé rameno križovatky jeden) by mohol byť plán definovaný ako $P = \{25, 10\}$ udáva, že prvý povoľovací príkaz trvá 25 sekúnd a reaguje naň napríklad severné a južné rameno križovatky (reakcia na konkrétny príkaz sa definuje pri vytváraní pruhu) a druhý povoľovací príkaz trvá 10 sekúnd a reaguje naň západné a východné rameno križovatky.

3.2.2 Implementácia výpočtu riadenia

Riadenie v simulátore abstrahuje abstraktná trieda `ControlLogic`, ktorá deklaruje hlavičku hlavnej metódy používanej na riadenie `getControlPlan`. Jej vstupom sú údaje zo všetkých senzorov danej križovatky a výstupom signálny plán, ktorý bol popísaný v predchádzajúcej kapitole. Ten je reprezentovaný dátovou štruktúrou pole typu `double[]`. Pre simuláciu boli implementované dva spôsoby riadenia dopravy.

Fuzzy riadenie

Hlavným spôsobom riadenia dopravy v simulácií je pomocou Fuzzy logiky. O výpočet riadenia sa stará trieda `FuzzyControl`, ktorá zabezpečuje rozhranie medzi programom v Jave a fuzzy blokom napísanom v jazyku FCL (Fuzzy Control Language). Každá križovatka využíva na riadenie svoj vlastný fuzzy blok. Rozhranie bolo implementované pomocou knižnice `jFuzzyLogic`[1].

Riadenie pomocou neurónových sietí

V tomto prípade výpočet zabezpečujú neurónové siete. K ich implementácií je využitá knižnica `Neuroph`[2]. Vstupom pre neurónovú sieť je pole hodnôt typu `double[]`. Výstupom je spomínaný signálny plán. Pri prvom spustení simulácie sa overí prítomnosť súborov s dátami už natrénovaných sietí. Ak neexistujú, tak sa na základe trénovacích dát uložených v CSV súboroch vypočítajú nové. Na spracovanie CSV súborov využívam knižnicu `OpenCSV`[3]. Po procese učenia sú nové siete uložené do príslušných súborov, aby sa pri najbližšom použití nemuseli nanovo vytvárať. Ak by došlo k zmene trénovacích dát a tým pádom by bolo treba vytvoriť nové siete, stačí súbory so starými vymazať a nové sa vytvoria automaticky.

3.3 Popis modelu

Model použitý v tejto simulácií modeluje sústavu 15 križovatiek okolo centra Brna. Veľkosť vytvoreného modelu je pomerná k reálnej vozovke. To znamená, že sú zachované jednotlivé dĺžky vozoviek. Nestáva sa preto, že by sa v nejakom pruhu zobrazovalo viac vozidiel, ako je v reálnom svete možné. Ak by došlo k naplneniu niektorého z pruhov, ďalšie vozidlá už nie sú do pruhu vpustené a simulátor je o tejto skutočnosti informovaný.

Takisto jednotlivé pruhy sú zobrazené tak ako v skutočnosti. Táto možnosť je pre prípad, že je pre jeden smer viac jazdných pruhov. Jediným obmedzením je fakt, že reálne viac-

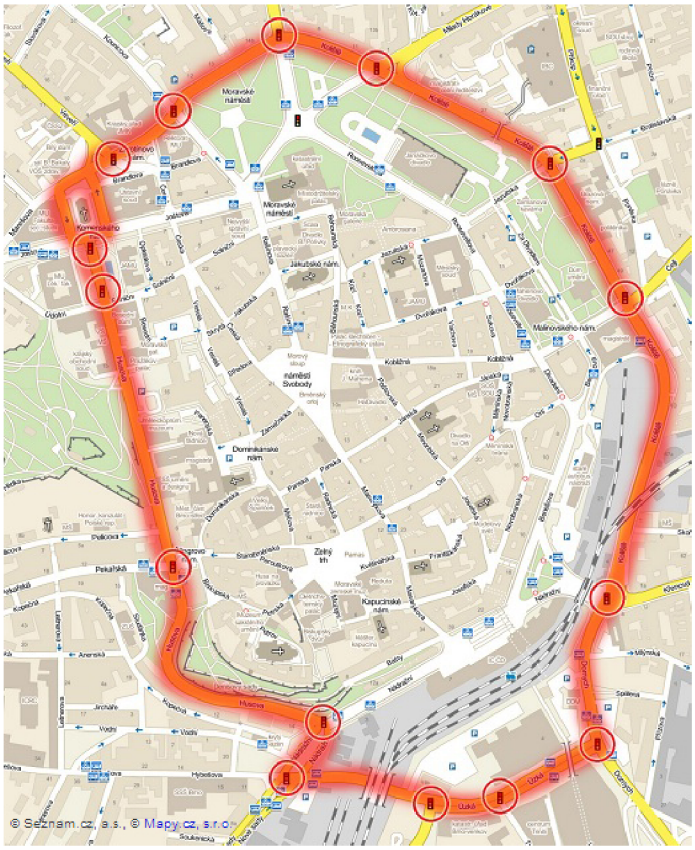
[1] http://jfuzzylogic.sourceforge.net/
[2] http://neuroph.sourceforge.net/
[3] http://opencsv.sourceforge.net/

smerové pruhy sú v modeli reprezentované samostatne. K tomuto kroku ma viedol fakt, že dáta o počte vozidiel, ktoré som získal boli pre konkrétny smer a nie pruh.

Kapitola 4

Simulácia a jej výsledky

Táto kapitola popisuje vyhodnocovanie výsledkov simulácie z vytvoreného simulátora. K tomu bolo potrebné uskutočniť radu meraní v teréne, ktorému sa táto kapitola tiež venuje.



Obr. 4.1: Mapa meraného dopravného úseku.

4.1 Zber dopravných dát

Neoddeliteľnou súčasťou procesu vytvárania určitého modelu pre simuláciu je zber dát, na základe ktorých bude model pracovať. V prípade modelu dopravy je to hlavne počet

vozidiel, ktoré prechádzajú jednotlivými úsekmi križovatky. Tieto dáta sú potom využívané ako základ pre generovanie nových vozidiel. Pre potrebu tejto práce bolo preto potrebné získať dáta z každej križovatky a z každého jej pruhu osobitne.

Možným riešením by bolo osobne prejsť všetky križovatky a v každej počítať vozidlá pre jednotlivé smery. Takéto riešenie by však malo viaceré nevýhody. Kvôli relatívne veľkému počtu križovatiek by to bolo značne časovo náročné. Takisto sčítavanie jednotlivých vozidiel pre konkrétny smer by nebolo jednoduché, kvôli absencií pozorovacej techniky.

Uvedené dôvody ma prinútili hľadať ďalšie možnosti získania dát efektívnejšou cestou. Obrátil som sa preto na Obor dopravy Magistrátu mesta Brna so žiadosťou o potrebné dáta. V Brne sa o prevádzku mestských komunikácií stará podnik Brněnské komunikace a.s. Podľa slov jedného z pracovníkov vykonávajú aj monitoring križovatiek na území Brna. Monitorujú zhruba 50-60 križovatiek, z ktorých získavajú rôzne dáta a medzi nimi aj počty prechádzajúcich motorových vozidiel, ktoré sú potrebné pre túto prácu.

Pozorovanie križovatiek sa vykonáva v závislosti na frekventovanosti križovatky raz za niekoľko rokov. Dáta sa získavajú dvoma spôsobmi. Poskytujú ich dopravné detektory, ktoré sú do systémov dôležitých križovatiek zapojené, alebo sú získavané ručne. V prípade ručného zberu sa celá križovatka 24 hodín sníma kamerou z určitého bodu, ktorý ma na situáciu dobrý výhľad. Záznam sa potom ručne analyzuje, teda sú zapisované počty vozidiel v jednotlivých smeroch.

Interné nariadenie podniku dovoľuje poskytovanie dopravných dát školám pre nekomerčné použitie, hlavne pre potreby záverečných prác. Po odovzdaní prehlásenia, že poskytnuté údaje nebudú použité mimo túto prácu mi boli potrebné dáta odovzdané v elektronickej forme. Súbor obsahoval štatistiky pre každú križovatku a to konkrétne počty jednotlivých druhov vozidiel pre konkrétny smer a hodinu začínajúc od 6:00 a končiac 18:00. Poskytnuté dáta boli zozbierané v rokoch 2010 až 2012.

4.2 Riadenie

Riadenie križovatiek je definované signálnymi plánmi. V reáli sú vytvorené signálne plány pre všetky možné scenáre, ktoré môžu na cestách nastať, napr. príjazd prednostného vozidla, dopravné obmedzenie v určitom úseku, plánovaná udalosť, ktorá si vyžaduje plynulejšiu premávku v istom smere. Pre potreby tejto práce však tieto udalosti neberieme do úvahy. Boli teda vytvorené jednoduché signálne plány s premenlivou dĺžkou jednotlivých častí cyklu.

4.2.1 Pozorovanie dopravného úseku

Aby bolo riadenie dopravy v meste efektívne, musí byť každá križovatka riadená vlastným spôsobom, preto by klasický spôsob riadenia, ktorý rozdeľuje povoľovací čas pre jednotlivé smery rovnomerne, nebol možný. Z toho dôvodu by vytvorenie triviálneho plánu nebolo riešením pre modelovaný dopravný úsek. Pre získanie povedomia o signálnych plánoch používaných v danom úseku som sa znova obrátil na Brnenské komunikace a.s. Táto žiadosť však nebola vyhovená, pretože signálne plány sú firemné know-how a nie sú poskytované ani na akademické účely.

Prvou fázou pri tvorbe riadenia simulovaného úseku teda bolo pozorovanie. Systém bolo spočiatku potrebné preskúmať a zistiť aspoň jeho základné fungovanie. Bolo teda vykonané pozorovanie fungovania systému. Pozorovanie prebiehalo ráno od 7:00 do 8:30, na obed od

12:00 do 13:30 a v čase poobednej špičky od 15:30 do 17:00. Časy boli vybrané z dôvodu obsiahnutia rôznych stavov vyťaženia dopravy.

Meranie bolo hlavne zamerané na získanie povedomia o signálnych plánoch na križovatkách. Hlavne bolo zamerané na odlišnosti v riadení jednotlivých križovatiek, akými sú počet pruhov pre daný smer a obmedzenia pohybu vyplývajúce z dopravného značenia. Princíp merania spočíval v tom, že na skúmanej križovatke bol najskôr pozorovaný systém prepínania povoľovacích signálov pre rôzne pruhy a neskôr boli zmerané časy, kedy boli jednotlivé pruhy povolené k prejazdu križovatkou. Celý okruh bol kvôli lepšiemu povedomiu o spôsobe určovania časov jednotlivých častí cyklu premeraný vždy dvakrát.

4.2.2 Tvorba a výpočet riadenia

Po skončení pozorovania bolo možné pristúpiť k tvorbe riadenia pre daný úsek križovatiek v Brne. Bol odsledovaný spôsob riadenia, teda to, ktoré pruhy a v akom poradí majú povolenie k prejazdu križovatkou. Každá križovatka sa v tomto správa špecificky a preto bolo treba každú križovatku riešiť samostatne. Popis signálnych plánov jednotlivých križovatiek, konkrétne postupnosť povoľovacích príkazov pre jednotlivé pruhy je popísaný v prílohe B.

Časť vytvoreného simulátora, ktorá má na starosti riadenie berie ako vstup počty vozidiel, ktoré prešli danou križovatkou pre daný smer. Jej výstupom sú povoľovacie časy pre jednotlivé časti signálneho plánu. Ich počet sa môže meniť v závislosti od konkrétneho plánu križovatky. Výstupný súbor povoľovacích časov by sa mal odvíjať od aktuálnej dopravnej situácie. Aktuálnu dopravnú situáciu merajú dopravné senzory. Ich výstupy teda slúžia ako vstupy pre riadenie dopravnej siete. Cyklus pre výpočet nových časov ako aj zber dát z dopravných senzorov bol stanovený na 5 minút. Táto hodnota bola určená ako kompromis medzi aktuálnosťou situácie a možnými výchylkami dopravy. Ak by bol čas moc dlhý, tak by plán nevedel pružne reagovať na zmeny. Ak by bol zasa moc krátky, mohlo by sa stať, že za daný čas príde príliš malý, alebo príliš veľký počet vozidiel a tým pádom nebude výsledný signálny plán zodpovedať realite.

Neurónové siete

Jednou z možností výpočtu povoľovacích časov sú neurónové siete. Tie na základe určitého vstupu, v tomto prípade dát zo senzorov produkujú výstup - povoľovacie časy. Pred ich spustením však musia byť natrénované na konkrétnych dátach. Po procese tréningu sú siete schopné sami rozhodovať o výstupe aj pri úplne neznámych dátach. Presnosť ich výstupu sa odvíja od množstva trénovacích dát. To znamená, že čím viac trénovacích dát majú siete k dispozícií, tým presnejší výstup produkujú. Všeobecne trénovacie dáta vyzerajú nasledovne:

Stĺpec	1	2	...	N	N+1	N+2	...	N+M
Hodnota	s_1	s_2	...	s_N	o_1	o_2	...	o_M
s - senzor križovatky, o - výstup siete								

Z trénovacích dát sú potom vytvorené neurónové siete, ktoré dokážu produkovať očakávaný výstup a ten následne priradiť riadiacej časti simulácie.

Fuzzy logika

Ďalšou možnosťou je využitie fuzzy logiky. V simulátore je vytvorené rozhranie pre použitie fuzzy blokov napísaných v jazyku FCL. Toto riešenie má napríklad výhodu v tom, že je možné fuzzy bloky meniť bez rekompilácie programu. Každá križovatka má svoj vlastný fuzzy blok. Ich vstupom sú rovnako ako aj pri neurónových sieťach, dáta z jednotlivých senzorov a výstupom rovnako povoľovacie časy.

Vstupné dáta sú fuzzyfikované, teda je im priradená určitá miera danej vlastnosti. Vlastnosťou rozumieme stupeň premávky pre daný smer, ktorý môže byť nízky, stredný a vysoký. Výstupné dáta prechádzajú opačným procesom, defuzzyfikáciou a teda im je na základe miery danej vlastnosti priradená výstupná hodnota - dĺžka povoľovacieho času. Povoľovací čas môže byť krátky, stredný a dlhý. Príklad fuzzyfikácie a defuzzyfikácie:

```
FUZZIFY line5
  TERM low := (0, 1) (8, 0);
  TERM medium := (6, 0) (19, 1) (29, 0);
  TERM high := (24, 0) (40, 1);
END_FUZZIFY

DEFUZZIFY signal2
  TERM short := (0,1) (20,0);
  TERM medium := (15,0) (25,1) (35,0);
  TERM long := (30,0) (50,1);
  METHOD : COG;
  DEFAULT := 25;
END_DEFUZZIFY
```

Výstup bloku je ovplyvňovaný rozhodovacími pravidlami. Pravidlá podmienečne priraďujú určitú vlastnosť výstupu na základe závislých vstupných vlastností. Teda hodnotu výstupu danej časti signálneho plánu ovpluvňujú tie pruhy, ktoré na túto časť reagujú. Rozhodovacie pravidlá môžu vyzerať napríklad takto:

```
  RULE 1 : IF line1 IS low AND line2 IS low AND line3 IS low then \
           signal1 IS short;
  RULE 2 : IF line2 IS medium OR line3 IS medium \
           then signal1 IS medium;
  RULE 3 : IF line1 IS high AND line2 IS high AND line3 IS high \
           then signal1 IS long;
```

Príklad celého fuzzy bloku je uvedený v prílohe C.

4.3 Experimentovanie

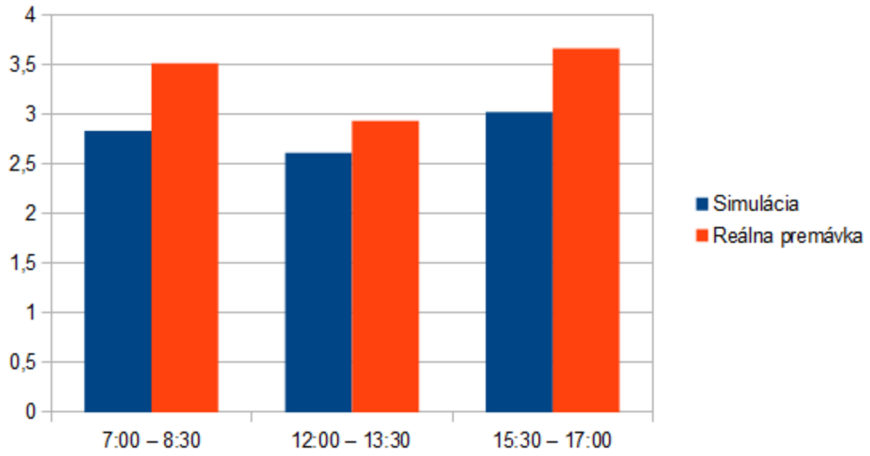
Po vytvorení riadenia pre jednotlivé križovatky bolo treba riadenie optimalizovať, aby pri simulácií nedochádzalo k úplnému zaplneniu jazdných pruhov a aby boli výsledky získané simuláciou čo najviac priblížené reálnej premávke. Kvôli absencií väčšieho množstva trénovacích dát však nie sú pre tento prípad neurónové siete vhodným nástrojom na výpočet signálneho plánu. Preto som sa rozhodol pre experimentovanie využiť fuzzy logiku.

Pre porovnanie s reálnou premávkou boli zvolené tri časti dňa a to ráno od 7:00 do 8:30, na obed od 12:00 do 13:30 a podvečer od 15:30 do 17:00. Pri experimentoch bola porovnávaná priemerná dĺžka fronty čakajúcich vozidiel. Tá bola počítaná vždy, keď riadenie križovatky udelilo povoľovací signál pre určitý smer. V tom okamihu boli započítané vozidlá, ktoré v danom pruhu čakali.

4.3.1 Výsledky experimentov

Postupnými iteráciami bolo riadenie zdokonaľované. Ako metriku zdokonalenia som bral hodnotu priemernej dĺžky fronty pre časový úsek 5 minút. V uvedenej tabuľke a grafe sú uvedené priemerné hodnoty dĺžky čakajúcej fronty pre jednotlivé smery v celej simulovanej sieti.

	Simulácia	**Reálna premávka**	**Rozdiel (v %)**
7:00 - 8:30	2,83	3,51	19,4%
12:00 - 13:30	2,61	2,93	11,0%
15:30 - 17:00	3,02	3,66	17,5%



Obr. 4.2: Graf výsledkov simulácie

Je nutné poznamenať, že je v tomto prípade nameraná hodnota nižšia, ako sú reálne dĺžky front v jednotlivých jazdných pruhoch. Uvedený spôsob merania bol zvolený z dôvodu dizajnu simulátora, kde neexistuje pruh s viacerými smermi. Tým pádom menej vyťažené smery, ktoré sú reálne spájané do jedného v tomto prípade spôsobujú nižšiu priemernú dĺžku.

Z uvedených dát vyplýva, že najväčšie zlepšenie nastalo pri rannej premávke, kde zlepšenie oproti reálnej premávke bolo 19,4%. Pritom žiadny z pruhov pri simulácií nedosiahol svoju maximálnu veľkosť.

Kapitola 5

Záver

Táto práca sa venuje simulácií riadenia dopravy pomocou semaforov. Semafory sú v tomto prípade myslené ako dômyselné zariadenia používajúce rôzne technické prostriedky, ktoré pomáhajú zlepšiť kvalitu riadenia dopravy. Riadenie za použitia týchto prostriedkov dokáže flexibilne meniť na základe aktuálneho stavu premávky bez zásahu človeka, preto sa dá označiť za inteligentné.

Práca v jednotlivých kapitolách popisuje spôsob, akým sa v súčasnosti doprava riadi, technické prostriedky ktoré slúžia na podporu riadenia, nástroje, ktoré je možné využiť na riadenie a jeho simuláciu, návrh a implementáciu simulátora. V poslednej kapitole sa venuje vytváraniu samotného riadenia potrebného pre simuláciu, ako aj experimentovaniu so simulátorom a vyhodnoteniu dosiahnutých výsledkov v závislosti k reálnej premávke.

Vo výsledku bol vytvorený simulátor, ktorý simuluje riadenie križovatiek v oblasti malého okruhu okolo centra Brna. Simulátor je riadený udalosťami a pre výpočet vhodného signálneho plánu použitého na riadenie križovatiek je možné použiť fuzzy logiku, alebo neurónové siete. Simulátor zobrazuje svoj aktuálny stav vďaka grafickému užívateľskému rozhraniu a pracuje na základe reálnych dát o premávke, ktoré boli poskytnuté firmou Brněnské komunikace a.s. a dát získaných pozorovaním. Na základe vykonaných experimentov bolo zistené, že vytvorené riadenie zlepšuje plynulosť dopravy o 15% oproti dnes používaným signálnym plánom.

V budúcnosti by sa simulátor mohol vyvíjať rôznymi smermi. Mohol by sa rozšíriť veľkosť simulovanej časti dopravy. Prípadne by mohli byť implementované ďalšie súčasti reálnej premávky, ktoré by skvalitnili výstupy zo simulácie. Pod tým si predstavujem napríklad implementáciu mestskej hromadnej dopravy a jej preferencií, ktorú simulátor v súčasnej verzií pri simulácií nezohľadňuje, alebo by mohla byť doimplementovaná časť, ktorá by simulovala nerozhodnosť pri správaní vodičov jednotlivých vozidiel. Posledným možným vylepšením by mohlo byť vytvorenie editora, vďaka ktorému by bolo možné jednoduchšie pridávať jednotlivé križovatky.

Literatura

[1] Fuzzy logika. [Online], [cit. 2014-05-01].
URL http://sk.wikipedia.org/wiki/Fuzzy_logika

[2] Inteligentní systém řízení dopravy v městské oblasti. [Online], [cit. 2013-01-12].
URL http://www.eltodo.cz/produkty-a-sluzby/vyrobni-program/Doprava_IS_RizeniDopravy.pdf

[3] Modelování dopravy na pozemních komunikacích (ČÁST 1). [Online], [cit. 2013-01-22].
URL http://projekt150.ha-vel.cz/node/94

[4] Systémy pro silniční dopravu - Dopravní detektory. [Online], [cit. 2013-01-13].
URL http://www.azd.cz/admin/files/Dokumenty/pdf/Produkty/Silnicni/Dopravni-detektory.pdf

[5] Umelé neurónové siete. [Online], [cit. 2013-05-19].
URL http://ii.fmph.uniba.sk/~benus/books/UNS_revised.pdf

[6] Ročenka dopravy České Republiky. 2011, [Online], [cit. 2013-01-10].
URL https://www.sydos.cz/cs/rocenka_pdf/Rocenka_dopravy_2011.pdf

[7] Hotmar, P.; Palatová, M.: Fuzzy řízení dopravy na světelné křižovatce. Technická zpráva, ústav Počítačové a řídící techniky, VŠCHT Praha.

[8] MACH, M.: Evolučné algoritmy - Prvky a princípy. [Online], [cit. 2013-01-20].
URL http://neuron-ai.tuke.sk/machm/publications/kniha-eapp.pdf

[9] NETRVALOVÁ, A.: Úvod do problematiky multiagentních systémů. [Online], [cit. 2013-01-20].
URL http://www.kiv.zcu.cz/~netrvalo/phd/MAS.pdf

[10] POKORNÝ, M.: *Expertní systémy*. Ostravská univerzita v Ostravě, 2004.

[11] TICHÝ, T.: *Řídicí systémy dopravy - dopravní telematika*. České vysoké učení technické v Praze, 2004.

[12] TICHÝ, T.: Řízení dopravy ve městě. 2012, [Online], [cit. 2013-01-13].
URL http://www.fd.cvut.cz/personal/kolarra5/UISp02.pdf

Dodatek A

Obsah CD

- src/ - zdrojové súbory programu
- build/ - program v spustiteľnej forme
- tex/ - zdrojové súbory technickej správy
- projekt.pdf - technická správa
- readme.txt - popis použitia simulátoru

Dodatek B

Spôsob riadenia križovatiek

Pozn.: všetkými smermi sú myslené všetky smery povolené dopravným značením.

Veveří - Žerotínovo nám.

- **1.** Žerotínovo nám. (Kounicova) všetkými smermi
- **2.** Žerotínovo nám. všetkými smermi
- **3.** Veveří všetkými smermi

Žerotínovo nám. - Husova

- **1.** Žerotínovo nám. -¿ Husova, Žerotínovo nám. -¿ Marešova
- **2.** križovatka pre vozidlá uzatvorená

Husova - Údolní

- **1.** Solniční všetkými smermi, Údolní všetkými smermi
- **2.** Komonského nám. všetkými smermi, Husova všetkými smermi

Husova - Pekařská

- **1.** Husova (Joštova) všetkými smermi, Husova (Nádražní) všetkými smermi
- **2.** Pekařská vštkými smermi, Šilingrovo nám. všetkými smermi
- **3.** Pekařská všetkými smermi

Husova - Nádražní

- **1.** Nádražní všetkými smermi, Nové sady -¿ Nádražní
- **2.** Nové sady všetkými smermi, Husova -¿ Nové sady
- **3.** Husova všetkými smermi

Nové Sady - Hybešova

- **1.** Hybešova (Úzká) všetkými smermi, Hybešova všetkými smermi
- **2.** Nádražní všetkými smermi, Nové sady všetkými smermi
- **3.** Hybešova (Úzká) všetkými smermi

Úzká - Uhelná

- **1.** Hybešova všetkými smermi, Úzká všetkými smermi
- **2.** Úzká všetkými smermi
- **3.** Uhelná všetkými smermi
- **4.** Uhelná všetkými smermi, Hybešova -¿ Uhelná

Trnitá - Úzká

- **1.** Trnitá všetkými smermi, Úzká -¿ Trnitá
- **2.** Úzká (Dornych) všetkými smermi, Úzká všetkými smermi

Dornych - Úzká

- **1.** Úzká všetkými smermi
- **2.** Dornych (Křenová) všetkými smermi, Dornych (Komárov) všetkými smermi
- **3.** Úzká -¿ Dornych (Komárov)

Dornych - Křenová

- **1.** Dornych všetkými smermi
- **2.** Dornych -¿ Křenová, Dornych -¿ Koliště, Koliště -¿ Benešova, Koliště -¿ Dornych
- **3.** Kolište všetkými smermi, Křenová -¿ Koliště
- **4.** Benešova všetkými smermi, Křenová všetkými smermi

Koliště - Cejl

- **1.** Koliště (Křenová) všetkými smermi, Malinivského nám. -¿ Koliště (Křenová)
- **2.** Koliště (Křenová) -¿ Cejl, Koliště (Křenová) -¿ Koliště (Lidická), Koliště (Lidická) všetkými smermi
- **3.** Koliště (Lidická) všetkými smermi, Cejl -¿ Koliště (Lidická)
- **4.** Cejl všetkými smermi, Malinovského nám. všetkými smermi

Koliště - Bratislavská

- **1.** Koliště (Cejl) všetkými smermi, Koliště všetkými smermi
- **2.** Bratislavská všetkými smermi, Jezuitská všetkými smermi
- **3.** Bratislavská všetkými smermi

Koliště - M.Horákové

- **1.** Koliště (Bratislavská) všetkými smermi
- **2.** Koliště (Lidická) všetkými smermi, Koliště (Bratislavská) -¿ M. Horákové, Koliště (Bratislavská) -¿ Koliště (Lidická)
- **3.** M. Horákové všetkými smermi, Moravské nám. všetkými smermi

Lidická - Koliště

- **1.** Koliště všetkými smermi, Moravské nám. všetkými smermi
- **2.** Moravské nám. (Roosveltova) všetkými smermi, Lidická všetkými smermi
- **3.** Lidická všetkými smermi, Koliště -¿ Lidická

Kounicova - Moravské nám.

- **1.** Žerotínovo nám. všetkými smermi
- **2.** Žerotínovo nám. všetkými smermi, Moravské nám. (Lidická) všetkými smermi
- **3.** Moravské nám. všetkými smermi, Kounicova všetkými smermi
- **4.** Kounicova všetkými smermi

Dodatek C

Príklad fuzzy bloku pre križovatku

Dornych - Křenová

```
// Decision logic FIS (fuzzy inference system) for crossroad c10

FUNCTION_BLOCK decision_logic

// Input variables
VAR_INPUT
  line1 : REAL;
  line2 : REAL;
  line3 : REAL;
  line4 : REAL;
  line5 : REAL;
  line6 : REAL;
  line7 : REAL;
  line8 : REAL;
  line9 : REAL;
  line10 : REAL;
  line11 : REAL;
  line12 : REAL;
  line13 : REAL;
END_VAR

VAR_OUTPUT
  signal1 : REAL;
  signal2 : REAL;
  signal3 : REAL;
  signal4 : REAL;
END_VAR

// Fuzzify input variables
//c10_1to2
FUZZIFY line1
  TERM low := (0, 1) (30, 0);
  TERM medium := (22, 0) (45, 1) (60, 0);
```

```
  TERM high := (49, 0) (82, 1);
END_FUZZIFY

//c10_1to4
FUZZIFY line2
  TERM low := (0, 1) (8, 0);
  TERM medium := (6, 0) (21, 1) (30, 0);
  TERM high := (25, 0) (42, 1);
END_FUZZIFY

//c10_2to1_1
FUZZIFY line3
  TERM low := (0, 1) (8, 0);
  TERM medium := (6, 0) (14, 1) (19, 0);
  TERM high := (16, 0) (27, 1);
END_FUZZIFY

//c10_2to1_2
FUZZIFY line4
  TERM low := (0, 1) (8, 0);
  TERM medium := (6, 0) (14, 1) (19, 0);
  TERM high := (16, 0) (27, 1);
END_FUZZIFY

//c10_2to4
FUZZIFY line5
  TERM low := (0, 1) (4, 0);
  TERM medium := (3, 0) (13, 1) (20, 0);
  TERM high := (17, 0) (28, 1);
END_FUZZIFY

//c10_2to6_1
FUZZIFY line6
  TERM low := (0, 1) (33, 0);
  TERM medium := (25, 0) (53, 1) (70, 0);
  TERM high := (57, 0) (96, 1);
END_FUZZIFY

//c10_2to6_2
FUZZIFY line7
  TERM low := (0, 1) (33, 0);
  TERM medium := (25, 0) (53, 1) (70, 0);
  TERM high := (57, 0) (96, 1);
END_FUZZIFY

//c10_4to1
FUZZIFY line8
  TERM low := (0, 1) (4, 0);
```

```
  TERM medium := (3, 0) (9, 1) (14, 0);
  TERM high := (11, 0) (19, 1);
END_FUZZIFY

//c10_4to6
FUZZIFY line9
  TERM low := (0, 1) (1, 0);
  TERM medium := (0, 0) (4, 1) (7, 0);
  TERM high := (6, 0) (10, 1);
END_FUZZIFY

//c10_6to1
FUZZIFY line10
  TERM low := (0, 1) (7, 0);
  TERM medium := (5, 0) (13, 1) (19, 0);
  TERM high := (16, 0) (27, 1);
END_FUZZIFY

//c10_6to2_1
FUZZIFY line11
  TERM low := (0, 1) (30, 0);
  TERM medium := (22, 0) (45, 1) (60, 0);
  TERM high := (49, 0) (82, 1);
END_FUZZIFY

//c10_6to2_2
FUZZIFY line12
  TERM low := (0, 1) (30, 0);
  TERM medium := (22, 0) (45, 1) (60, 0);
  TERM high := (49, 0) (82, 1);
END_FUZZIFY

//c10_6to4
FUZZIFY line13
  TERM low := (0, 1) (4, 0);
  TERM medium := (3, 0) (11, 1) (17, 0);
  TERM high := (14, 0) (24, 1);
END_FUZZIFY

// Defzzzify output variable
DEFUZZIFY signal1
  TERM short := (0,1) (11,0);
  TERM medium := (9,0) (15,1) (20,0);
  TERM long := (18,0) (30,1);
  METHOD : COG;
  DEFAULT := 15;
END_DEFUZZIFY
```

```
// Defzzzify output variable
DEFUZZIFY signal2
  TERM short := (0,1) (20,0);
  TERM medium := (18,0) (25,1) (30,0);
  TERM long := (38,0) (50,1);
  METHOD : COG;
  DEFAULT := 25;
END_DEFUZZIFY

// Defzzzify output variable
DEFUZZIFY signal3
  TERM short := (0,1) (11,0);
  TERM medium := (9,0) (15,1) (20,0);
  TERM long := (18,0) (30,1);
  METHOD : COG;
  DEFAULT := 15;
END_DEFUZZIFY

// Defzzzify output variable
DEFUZZIFY signal4
  TERM short := (0,1) (15,0);
  TERM medium := (13,0) (20,1) (25,0);
  TERM long := (23,0) (35,1);
  METHOD : COG;
  DEFAULT := 20;
END_DEFUZZIFY

// Inference rules
RULEBLOCK No1
  AND : MIN;  // Use 'min' for 'and'
  ACT : MIN;  // Use 'min' activation method
  ACCU : MAX; // Use 'max' accumulation method

  RULE 1 : IF line9 IS low AND line10 IS low AND line11 IS low AND \
           line12 IS low AND line13 IS low then signal1 IS short;
  RULE 2 : IF line9 IS medium OR line10 IS medium OR line11 IS medium \
           OR line12 IS medium OR line13 IS medium then signal1 IS medium;
  RULE 3 : IF line9 IS high AND line10 IS high AND line11 IS high \
           AND line12 IS high AND line13 IS high then signal1 IS long;

  RULE 4 : IF line5 IS low AND line6 IS low AND line7 IS low AND line10 \
           IS low AND line11 IS low AND line12 IS low then signal2 IS short;
  RULE 5 : IF line5 IS medium OR line6 IS medium OR line7 IS medium \
           OR line10 IS medium OR line11 IS medium OR line12 IS medium \
           then signal2 IS medium;
  RULE 6 : IF line5 IS high AND line6 IS high AND line7 IS high AND \
           line10 IS high AND line11 IS high AND line12 IS high \
           then signal2 IS long;
```

```
  RULE 7 : IF line1 IS low AND line3 IS low AND line4 IS low AND line5 \
           IS low AND line6 IS low AND line7 IS low then signal3 IS short;
  RULE 8 : IF line1 IS medium OR line3 IS medium OR line4 IS medium OR \
           line5 IS medium OR line6 IS medium OR line7 IS medium \
           then signal3 IS medium;
  RULE 9 : IF line1 IS high AND line3 IS high AND line4 IS high AND \
           line5 IS high AND line6 IS high AND line7 IS high \
           then signal3 IS long;

  RULE 10 : IF line1 IS low AND line2 IS low AND line8 IS low AND \
            line9 IS low then signal4 IS short;
  RULE 11 : IF line1 IS medium OR line2 IS medium OR line8 IS medium \
            OR line9 IS medium then signal4 IS medium;
  RULE 12 : IF line1 IS high AND line2 IS high AND line8 IS high AND \
            line9 IS high then signal4 IS long;
END_RULEBLOCK

END_FUNCTION_BLOCK
```