

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

## KALIBRACE MIKROSIMULAČNÍHO MODELU DOPRAVY

DIPLOMOVÁ PRÁCE

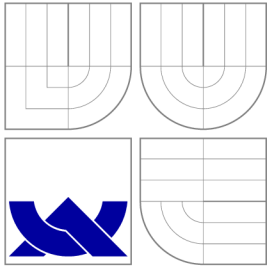
MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. PAVEL POKORNÝ

BRNO 2013



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

# **KALIBRACE MIKROSIMULAČNÍHO MODELU DOPRAVY**

MICROSCOPIC TRAFFIC SIMULATION MODEL CALIBRATION

**DIPLOMOVÁ PRÁCE**  
MASTER'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**Bc. PAVEL POKORNÝ**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. PAVOL KORČEK**

BRNO 2013

## **Abstrakt**

Práce se zabývá problematikou mikroskopické simulace dopravy. Součástí práce je návrh a implementace mikrosimulačního modelu, který je založen na celulárním automatu a vychází z modelu Nagel-Schreckenberg. Model podporuje kalibraci genetickým algoritmem. Práce obsahuje výsledky simulací s nastavením modelu, které bylo nalezeno při kalibraci.

## **Abstract**

This thesis main focus is microscopic traffic simulation. Part of this work is the design and implementation of microsimulation model based on cellular automaton. Implemented model supports calibration with genetic algorithm. The results of calibration and simulations are included.

## **Klíčová slova**

Simulace dopravy, genetický algoritmus, celulární automat, kalibrace modelu.

## **Keywords**

Traffic simulation, genetic algorithm, cellular automaton, model calibration.

## **Citace**

Pavel Pokorný: Kalibrace mikrosimulačního modelu dopravy, diplomová práce, Brno, FIT VUT v Brně, 2013

# Kalibrace mikrosimulačního modelu dopravy

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Pavla Korčeka.

.....

Pavel Pokorný

24. května 2013

## Poděkování

Rád bych na tomto místě poděkoval vedoucímu práce Ing. Pavlovi Korčekovi za jeho cenné připomínky, poskytnutí odborné literatury a za ochotu věnovat se mi mimo konzultační hodiny.

© Pavel Pokorný, 2013.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1 Úvod</b>	<b>3</b>
<b>2 Modelování a simulace dopravy</b>	<b>5</b>
2.1 Charakteristiky dopravy . . . . .	6
2.2 Druhy modelů . . . . .	8
2.2.1 Makroskopické modely . . . . .	8
2.2.2 Mikroskopické modely . . . . .	9
2.3 Fundamentální diagram dopravy . . . . .	10
<b>3 Celulární automaty</b>	<b>11</b>
3.1 Definice CA . . . . .	11
3.2 Wolframovy 1D celulární automaty . . . . .	13
3.3 Kategorie CA . . . . .	13
3.4 Použití CA v modelování dopravy . . . . .	14
3.4.1 Možná vylepšení modelu . . . . .	17
<b>4 Evoluční algoritmy</b>	<b>22</b>
4.1 Genetické algoritmy . . . . .	23
<b>5 Analýza a návrh dopravního simulátoru</b>	<b>27</b>
5.1 Požadavky na systém . . . . .	27
5.2 Předzpracování dat . . . . .	28
5.3 Kalibrace modelu . . . . .	28
5.4 Reprezentace entit . . . . .	30
5.5 Generování vozidel . . . . .	32
5.6 Vizualizace . . . . .	32
<b>6 Popis implementace</b>	<b>33</b>
6.1 Implementovaná vylepšení . . . . .	34
<b>7 Popis experimentů</b>	<b>36</b>
7.1 Analýza použitých dat . . . . .	36
7.2 Průběh simulace . . . . .	38
7.3 Kalibrace modelu . . . . .	39
<b>8 Závěr</b>	<b>46</b>
<b>A Obsah CD</b>	<b>51</b>

<b>B</b>	<b>Návod k použití</b>	<b>52</b>
B.1	Simulátor . . . . .	52
B.1.1	Vizualizace . . . . .	53
B.2	Genetický algoritmus . . . . .	54
B.3	Nástroje pro analýzu dat z Prahy . . . . .	54
B.4	Automatizované testy . . . . .	55
<b>C</b>	<b>Konfigurace simulace</b>	<b>56</b>
<b>D</b>	<b>Konfigurace evoluce</b>	<b>57</b>
<b>E</b>	<b>Ukázka nastavení simulátoru</b>	<b>58</b>

# Kapitola 1

## Úvod

V současné době je přítomnost kvalitní dopravní sítě a dostupných dopravních prostředků nutná k zajištění chodu snad všech odvětví hospodářství. Každý den je přepraveno nespočet surovin, výrobků i osob. Obchod se rozvíjí rychlým tempem a stejně tak rostou požadavky na rychlou přepravu. S rostoucí mírou urbanizace roste počet lidí přepravujících se do měst za prací i zábavou. Každá dopravní infrastruktura má však určitou kapacitu a větší hustota provozu způsobuje dopravní zácpy, které s sebou přináší spoustu problémů. Dochází k silnějšímu znečišťování ovzduší výfukovými plyny, roste opoždění průmyslové i hromadné přepravy a v neposlední řadě roste spotřeba neobnovitelných zdrojů.

Podle výzkumu z roku 2011 [3] stojí Evropu přetížení dopravy, tzn. tvoření kolon či odborně kongescí, každoročně zhruba 1% hrubého domácího produktu.

Evropská unie předpokládá, že v roce 2030 nákladní doprava v porovnání s rokem 2005 vzroste zhruba o 40% a do roku 2050 o něco více než o 80% [4]. Osobní doprava má růst poněkud pomaleji než nákladní: o 34% do roku 2030 a o 51% do roku 2050. Dle údajů mezinárodní organizace výrobců motorových vozidel roste každoročně produkce aut, v roce 2000 bylo celosvětově vyrobeno 58 miliónů vozidel a v roce 2012 to bylo už 84 miliónů [5].

Průměrný obyvatel Londýna strávil v roce 2012 každý všední den dopravou více než jednu hodinu [6], za týden to může být čas jednoho pracovního dne i víc. Není tedy divu, že v každém státě i městě jsou nebo vznikají instituce, které se starají o to, aby doprava byla plynulá a kapacita infrastruktury vyhovovala poptávce.

Z výše zmíněného je zřejmé, že se očekává zvyšování počtu vozidel na silnicích a stav dopravy bude nutné řešit čím dál více. Dopravní situaci lze vylepšovat několika způsoby.

Důležitým prvkem ovlivňující plynulost jsou pravidla dopravy. Ve většině vyspělých zemích jsou již stanovena a dodržována, zde tedy není příliš prostor pro vylepšení. Stále častěji se využívá automatické řízení dopravy pomocí zařízení jako jsou světelná signalizační zařízení nebo světelné tabule s dopravními informacemi. Plynulost závisí ve velké míře také na kvalitě povrchu vozovky, na počtu pruhů, povolené rychlosti, ukázněnosti řidičů nebo převládajícím typu vozidel.

Úprava dopravní infrastruktury je často velice nákladná a právě zde dostávají slovo počítačové simulátory dopravy. Jejich úkolem je poskytnout údaje, které mohou posloužit k lepšímu návrhu dopravní infrastruktury, k lepšímu nastavení zařízení určených k řízení provozu, nebo na základě kterých se řidič může rozhodnout vybrat nejvhodnější trasu.

Součástí této práce je návrh a implementace simulátoru, který by dokázal na základě hustoty dopravy predikovat délku projetí úseku. Simulátor bude testován na reálných datech. K dispozici jsou údaje o provozu z pražské ulice Evropská ze tří pracovních dnů, které obsahují příjezdové časy jednotlivých vozidel.

V kapitole 2 jsou uvedeny základní pojmy a rozdělení modelů používaných pro simulaci dopravy. Kapitola 3 obsahuje popis celulárních automatů, na nichž jsou postavené mikroskopické modely jednoproudé vozovky, které jsou zde podrobněji prozkoumány. Kapitola 4 obsahuje úvod do problematiky evolučních algoritmů a poté se zabývá genetickými algoritmy, které budou použity pro kalibraci vytvořeného simulačního modelu. V kapitolách 5 a 6 je uvedena analýza, návrh a popis implementace simulátoru založeném na modelu Nagel-Schreckenberg. V kapitole 7 jsou pak k nalezení výsledky analýzy dat, experimenty s nastavením simulátoru a kalibrace modelu pomocí genetického algoritmu. Závěr obsahuje shrnutí dosažených výsledků a zamyšlení nad možnými rozšířeními simulátoru.



## Kapitola 2

# Modelování a simulace dopravy

V této kapitole jsou popsány základní principy a pojmy z oblasti modelování a simulace dopravy, dále je uvedeno rozdělení modelů dopravy na základě různých kritérií a popsán fundamentální diagram dopravy.

Už od počátků rozmachu osobní automobilové přepravy se objevovaly snahy definovat pravidla, kterými se řídí dopravní tok. Existují různé teorie zabývající se vztahy mezi vozidly samotnými a infrastrukturou. Tyto teorie jsou označovány jako teorie dopravního toku (*Traffic Flow Theory*) [16]. Jejich cílem je porozumět principům dopravy a matematicky vyjádřit pozorované jevy.

S pomocí počítačové simulace je možné získat nové poznatky a odvodit správnost daných teorií. Simulace obecně je napodobení nějakého reálného systému, procesu nebo jevu. Počítačová simulace umožňuje získat relativně levně nové poznatky z experimentování s modelem daného systému. V oblasti dopravy by bylo velice nákladné si například ověřit, zda navržená topologie má předpokládanou propustnost. Výsledky simulace je možné tedy využít pro odhadnutí chování systému v různých situacích. [19]

Pro účely simulace se využívá různě detailních modelů, které jsou většinou zjednodušením reálného systému a nezahrnují všechny faktory ovlivňující jeho chování. Abychom byli schopni vytvořit model nějakého systému, musíme poznat, jak systém funguje a uvědomit si, že neexistuje pouze jeden model systému. Podoba modelu se odvíjí od toho, jaké jeho vlastnosti chce jeho tvůrce sledovat, jaké představy má o fungování systému a jaké otázky chce zodpovědět. [7]

Při vytváření modelu se ignorují vlastnosti systému, které nemají na výsledek simulace vliv a naopak se pozornost zaměřuje na ty vlastnosti, které nejvíce ovlivňují sledované chování systému.

## 2.1 Charakteristiky dopravy

Stav dopravy je možné charakterizovat pomocí základních vlastností - objem nebo tok, rychlost a hustota. Průměrné hodnoty těchto veličin lze sledovat z pohledu prostoru nebo času. [14] Objem a tok lze použít při popisu přerušované i nepřerušované dopravy. Rychlost a hustota se používá primárně pro popis nepřerušované dopravy. Nepřerušovaná doprava je na souvislých úsecích bez křižovatek a semaforů, například tedy na dálnicích [2]. Mezi těmito veličinami je následující vztah:

$$q = \rho \cdot v \quad (2.1)$$

kde:

$q$  = dopravní tok (počet vozidel/hod),  
 $v$  = průměrná rychlost (km/h),  
 $\rho$  = hustota dopravy (počet vozidel/km).

### Tok (*flow rate, q*) a objem (*volume, V*)

Tok a objem jsou veličiny, jejichž hodnoty jsou závislé na počtu vozidel, která projedou určitým místem během určitého časového intervalu.

Objem dopravy je počet vozidel, která projedou jedním místem nebo úsekem během sledovaného časového intervalu. Z nasbíraných dat je pak možné vypočítat např. kolik vozidel průměrně denně projede určitým úsekem. Jedná se tedy o předpokládaný nebo odpozorovaný počet vozidel, která projedou místem za určitý interval.

Hodnota toku dopravy se uvádí pro intervaly menší než jedna hodina a bývá v jednotkách počet vozidel za hodinu. Hodnota představuje, kolik by za hodinu projelo vozidel, kdyby byla doprava po celou hodinu stejná jako ve sledovaném intervalu. Například pozorovanému objemu 100 vozidel za 15 minut odpovídá tok 400 vozidel/hodinu.

Rozdíl mezi objemem a tokem nemusí být na první pohled zřejmý. Na následujícím příkladu bude tento rozdíl objasněn. Pro 4 po sobě jdoucí čtvrt hodiny byl pozorován objem 1000, 1200, 1100 a 1000 vozidel. Objem pro pozorovanou hodinu je 4300 vozidel/hod. Tok se ale měnil každých 15 minut a v nejvytíženějších 15 minutách byl 4800 vozidel/hod.

Brát v úvahu tok v době dopravní špičky je důležité při analýze kapacity komunikace. Pokud by např. úsek z předchozího příkladu měl kapacitu 4500 vozidel/hod, byla by tato kapacita během dopravní špičky překročena a docházelo by k tvoření kongescí i přes to, že pozorovaný objem za celou hodinu je menší než kapacita vozovky. Což je samozřejmě nežádoucí, protože může trvat i několik hodin, než se kongesce rozvolní. [2]

Tyto dvě veličiny kvantifikují poptávku, tedy počet vozidel, které chtějí v daném časovém úseku používat dopravní komunikaci. Pokud se jedná o úsek, kde se tvoří často kongesce, představují tyto hodnoty spíše kapacitní omezení této komunikace.

### Rychlost (*velocity, v*)

Rychlost obecně představuje ujetou vzdálenost za jednotku času, uvádí se v jednotkách kilometru (nebo míle) za hodinu.

Průměrná rychlost v časové doméně  $\bar{v}_t$  (*time-mean speed, TMS*) lze vypočítat jako aritmetický průměr rychlostí všech vozidel, která projela za měřený čas jedním místem.  $N$  reprezentuje počet vozidel, která projela sledovaný bod úseku.

$$\bar{v}_t = \frac{1}{N} \sum_{i=1}^N v_i \quad (2.2)$$

Pro výpočet průměrné rychlosti v prostorové doméně  $\bar{v}_t$  (*space-mean speed*, SMS, *average travel speed*) se používají dva přístupy [2]:

- Harmonický průměr rychlostí vozidel vyskytující se na úseku vozovky určité délky.

$$\bar{v}_s = \frac{N}{\sum_{i=1}^N \frac{1}{v_i}} \quad (2.3)$$

- Výpočet založen na průměrném čase, který vozidlům zabere projet určitý úsek cesty. Délka úseku je vydělena průměrnou dobou průjezdu tohoto úseku, tento čas zahrnuje i dobu strávenou v kongescích nebo čekáním na semaforu. Označme  $t_i$  časy průjezdu naměřené pro  $N$  vozidel projíždějící úsek o délce  $L$ , výpočet pak vypadá následovně:

$$\bar{v}_s = \frac{L}{\frac{1}{N} \sum_{i=1}^N t_i} \quad (2.4)$$

### Hustota dopravy (*density*, $\rho$ )

Hustota je počet vozidel, které se vyskytují v jednom okamžiku na sledovaném úseku cesty. Je vyjádřen v jednotkách počet vozidel na kilometr nebo na délku celé sledované vozovky. Tento údaj může být složité získat pozorováním, lze však jednoduše vypočítat na základě následujícího vztahu, pokud je známá průměrná doba průjezdu a dopravní tok.

$$\rho = \frac{q}{\bar{v}_s} \quad (2.5)$$

V řídicích systémech se často používá obsazenost vozovky (*occupancy*) jako náhrada za hustotu, protože se jedná o jednodušěji měřitelný parametr. Obsazenost v prostoru je procento vozovky obsazené vozidly. Obsazenost v čase reprezentuje procento času, kdy je sledované místo na vozovce obsazeno vozidly. [2]

## 2.2 Druhy modelů

Dopravní modely je možné kategorizovat na základě různých kritérií. Následuje výčet nejčastěji používaných kategorií a následně detailnější popis rozdělení na základě úrovně detailu.

### Dělení na základě spojitosti

Základní rozdělení je možné provést na základě toho, zda pracují s diskrétním nebo spojitým časem. Počítače jsou z podstaty však diskrétní, proto se spojitě modely implementují za pomoci reálných čísel s různou přesností. Čas v diskrétních modelech je možné implementovat pomocí celých čísel. Stav se tedy mění v určitých časových okamžicích po intervalech o stejné délce.

### Dělení na základě determinističnosti

Některé modely pracují s náhodnou složkou, jedná se o modely stochastické. V systému hraje roli pravděpodobnost, výsledek simulace se může tedy při každém spuštění mírně lišit. Možností, kde se náhodný prvek může vyskytnout, je mnoho. Na pravděpodobnosti může být postavena distribuce vozidel, řízení dopravy, zpomalování vozidel, změna pruhu atp. Deterministické modely naopak neobsahují žádný náhodný prvek a výsledek simulace je předem daný přesnými vztahy modelovaných entit.

### Dělení na základě modelovaného úrovně detailu

Jiným rozdělením je podle úrovně detailu s jakou je systém modelován. Nejčastěji se modely rozdělují do dvou kategorií, na makroskopické a mikroskopické. V makroskopických modelech je doprava modelována ze souhrnného pohledu založeném na hydrodynamické analogii kdy se na dopravní tok dívá jako na fyzikální proces podobný toku nějaké kapaliny a lze charakterizovat souhrnnými makroskopickými veličinami: hustota, objem, rychlost. Mikroskopické modely obsahují popis systému na úrovni jednotlivých částic, ze kterých se systém skládá (vozidla, soustavy řídicí dopravy). V některé literatuře uvažují třetí kategorii, mezoskopické modely, jedná se o modely založené na zjednodušeném popisu dynamiky vozidel. [7]

#### 2.2.1 Makroskopické modely

Na dopravu je nahlíženo jako na spojitý proud kapaliny, z toho je odvozen i alternativní název *Fluid-dynamical theories* [16]. Nejsou modelována jednotlivá vozidla, ale zkoumají se souhrnná, agregovaná data jako jsou hustota a intenzita provozu, průměrná rychlost a doba dojezdu. Dopravu popisují na vysoké úrovni pomocí soustavy rovnic.

Nevýhodou může být až přílišné zobecnění, protože je velice náročné nalézt matematické vztahy, které by realisticky popisovaly dopravní tok. Tyto modely mají matematický základ postavený na parciálních diferenciálních rovnicích.

Výhodou je však vyšší výpočetní rychlost než jaké dosahují jiné druhy modelů a je tak možné pomocí nich modelovat rozsáhlé systémy. Výpočetní technika se však rychle vyvíjí a s rostoucím výkonem počítačů začínají převažovat výhody mikroskopických modelů, které dokáží modelovat skutečnou dopravu s větší přesností.

## 2.2.2 Mikroskopické modely

V průběhu 90. let vzrostl enormně zájem o mikroskopické modely [1]. V mikroskopických modelech mají svou reprezentaci základní prvky dopravního systému - vozidla, silnice, křižovatky, prvky řídicí dopravu atp. Chování celého systému je reprezentováno interakcí mezi jeho entitami.

Nevýhodou těchto modelů je delší výpočetní doba simulace, ale právě tato nevýhoda je odbourávána s rostoucím výkonem osobních počítačů. S využitím nových technologií a přístupů dochází k neustálému urychlování i v oblasti mikroskopických modelů dopravy. Jedním z typů mikroskopických modelů jsou modely založené na celulárním automatu u kterých je navíc možné efektivně využít paralelního zpracování [12].

Výhoda mikroskopických modelů je v tom, že dokáží simulovat skutečný dopravní systém s větší přesností a dokáží zachytit více detailů než modely makroskopické.

### Car-Following modely

Teorie sledování předcházejícího vozidla (*Car-Following*, CF) popisují chování vozidel na základě klasické newtonovské mechaniky. Čas a prostor jsou spojité veličiny a chování vozidel se řídí diferenciálními rovnicemi. Vozidlo mění rychlost na základě okolních dopravních podmínek, tzv. stimulů. Mezi stimuly může patřit například informace o rychlosti daného vozidla, rychlost předchozího vozidla nebo vzdálenost od předchozího vozidla. Rovnice, podle které se řídí pohyb jednotlivých vozidel, je charakterizována funkcí zrychlení závislou na vstupních stimulech. [7]

Základním předpokladem modelů založených na těchto teoriích je, že se řidič snaží jet stejnou rychlostí, jakou jede vedoucí vozidlo nacházející se před ním, a snaží se dodržovat bezpečnou vzdálenost. Pokud vedoucí vozidlo změní rychlost, změní se vzdálenost mezi těmito vozidly. Řidič druhého vozidla na to zareaguje tak, že také změní rychlost, přidá nebo zpomalí. Existují různé varianty a rozšíření, například že řidič nesleduje pouze jedno vozidlo před sebou, ale sleduje jich několik.

CF modely využívají realistického popisu chování řidiče a detailních charakteristik vozidel, což vyžaduje vysoký výpočetní výkon. Mnohé z CF modelů používají detailní modely zrychlení vozidel. Kvůli vysoké náročnosti těchto modelů může simulace velké dopravní sítě obsahující mnoho vozidel trvat déle než skutečný čas, který byl odsimulován [9]. Častou vlastností těchto modelů je, že vyžadují několik parametrů, které jsou stanoveny ve fázi kalibrace a váží se tak k příslušným empirickým datům, chování vozidel je deterministické.

### Particle hopping modely

Jedná se o diskrétní modely, které popisují dopravu na základě stochastické dynamiky individuálních vozidel. Particle hopping modely lze formulovat pomocí celulárního automatu (*cellular automaton*, CA). Čas a prostor jsou zde diskrétní veličiny, vozovka je rozdělena na buňky, po nichž se vozidla pohybují po skocích (odtud název *particle hopping*, což je možné přeložit jako přeskakující částice). Výhody těchto modelů tkví ve větší rychlosti a možnosti modelovat stochastické chování systému [16]. Particle hopping modelům založeným na CA se věnuje část 3.4, použití CA v dopravě.

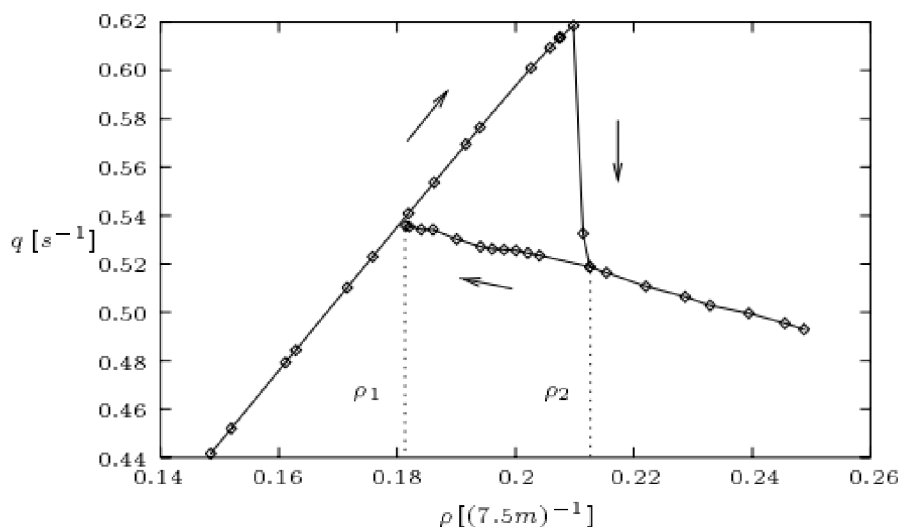
## 2.3 Fundamentální diagram dopravy

Fundamentální diagram znázorňuje závislost dopravního toku na hustotě dopravy, tedy závislost mezi dvěma základními makroskopickými veličinami [13]. Třetí veličina může být dopočítána pomocí fundamentálního vztahu teorie dopravního toku

$$q = \rho \cdot v, \quad (2.6)$$

kde  $q$  je dopravní tok,  $\rho$  průměrná hustota a  $v$  průměrná rychlost.

Na obr. 2.1 lze vidět tři rozdílné úseky, které odrážejí tři možné stavy, ve kterých se dopravní situace může nacházet.



Obrázek 2.1: Fundamentální diagram dopravního toku. Převzato z [13].

### Fáze plynulé dopravy

Doprava je plynulá, hustota dopravy je nízká, na grafu se jedná o oblast  $0 < \rho < \rho_1$ .

### Fáze metastabilních stavů

Doprava závisí kromě hustoty provozu také na počáteční konfiguraci vozidel. V této fázi se rozlišují dva možné přechody. Přechod zleva doprava, tzn. z plynulé dopravy do zácpy, vypadá tak, že hustota dopravy houstne do té doby, než některé z vozidel není schopno pokračovat stejnou rychlostí a musí přibrzdit. Vozidla za ním začnou také brzdit a začne se tvořit dopravní zácpa (kongesce). Při přechodu opačným směrem, kdy hustota dopravy řídne, dochází k lineárnímu růstu toku dopravy až do bodu  $\rho_1$ , kde se zácpa rozvolní systém přejde do stavu plynulé dopravy. Jedná se o úsek  $\rho_1 < \rho < \rho_2$ .

### Fáze dopravní zácpy

Doprava není plynulá, tvoří se kolony pomalu jedoucích až stojících vozidel, hustota dopravy je vysoká, na grafu se jedná o úsek  $\rho > \rho_2$ .

## Kapitola 3

# Celulární automaty

Celulární automat (*Cellular Automaton*, CA) je diskrétní dynamický systém s lokální interakcí [27]. Ve 40. letech 20. století vynalezl John von Neumann abstraktní modely biologické evoluce založené na principech celulárních automatů, tyto modely byly schopné reprodukce a vykazovaly velmi zajímavé chování. Od té doby našly celulární automaty uplatnění v řadě odvětví při řešení různorodých problémů. Osvědčily se jako univerzální modelovací prostředek evoluce v biologii, chemických jevů v chemii nebo při simulování pohybu tekutin ve fyzice. Jejich popularizaci a rozšíření přispěl mimo jiné i S. Wolfram, který se CA věnuje od 80. let 20. století [28]. Celulární automaty vykazují známky emergentního chování a samo-organizace. Uplatňují se při simulacích prostorových dynamických systémů v oblasti dopravy, šíření epidemií, chemických reakcí, růstu krystalů a mnoho dalších. Na bázi CA jsou postaveny některé modely umělého života a evoluce. Používají se na generování textur a fraktálů. Některé CA mají stejnou výpočetní sílu jako Turingův stroj, je možné je využít i pro obecné výpočty [19].

### 3.1 Definice CA

Základními stavebními kameny jsou buňky (*Cell*), které jsou uspořádány v  $n$ -rozměrném prostoru v pravidelné struktuře (*Lattice*, *Grid*), nejčastěji dvourozměrné mřížce. Každá buňka může nabývat několika definovaných stavů. Hodnoty stavů buněk v následujícím časovém kroku se vypočítávají pro každou buňku zvlášť na základě stanovených pravidel (*Rules*) uplatňovaných pomocí lokální přechodové funkce, která je pro všechny buňky stejná. Výpočet nového stavu probíhá pro všechny buňky v jednom okamžiku paralelně, stavy jsou tedy aktualizovány v diskrétních časových krocích pro všechny buňky najednou. Argumenty této funkce jsou hodnoty stavů dané buňky a buněk v definovaném okolí (*Neighbourhood*).

CA je tedy složen z následujících komponent [19]:

### Pole buněk

Obecně se jedná o  $n$ -rozměrnou strukturu, obvykle však 1D nebo 2D. Může být konečné nebo nekonečné, čehož může být docíleno rozdílnými hraničními vlastnostmi. Prostor je rozdělen rovnoměrně, všechny buňky mají stejnou velikost.

### Konečná množina $S$ stavů buňky

Každá množina je vždy v jednom ze stavů z této množiny. Příkladem takové množiny je  $S = \{0, 1\}$ .

### Okolí $N$

Okolí definuje počet a pozici sousedních buněk, které jsou použity jako vstup pro pravidla, pro výpočet nového stavu.

### Pravidla

Chování buňky je popsáno pravidly, které jsou reprezentovány tzv. lokální přechodovou funkcí (zkráceně LPF, anglicky *Local Transition Functions*, LTF). Všechny buňky dodržují stejná pravidla. Pravidla jsou funkce stavu buňky a jejího okolí definující nový stav buňky v následujícím časovém kroku:

$$s(t+1) = f(s(t), N_s(t)), \quad (3.1)$$

kde  $s \in S$  a  $N_s$  je stav všech buněk v definovaném okolí.

### Typy Okolí

Uvažují se různá okolí, v jednorozměrném CA je okolí dáno počtem sousedů po obou stranách buňky. Ve dvourozměrném CA patří mezi nejpoužívanější [19] (viz. 3.1):

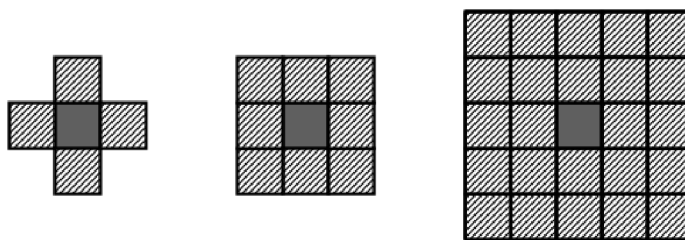
- Von-Neumann
- Moore
- Extended Moore

### Typické vlastnosti

Pro CA jsou charakteristické tyto vlastnosti:

- Paralelismus - výpočet nových hodnot stavů probíhá současně.
- Lokalita - nový stav buňky závisí jen na jejím předchozím stavu a stavu buněk v okolí.
- Homogenita - pro všechny prvky platí stejná lokální přechodová funkce.
- Konfigurace CA je definována jako stav všech buněk a popisuje stav systému.
- Čas i prostor jsou diskretizovány.



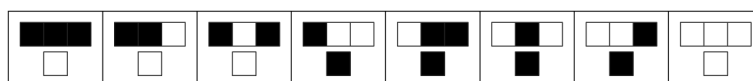


Obrázek 3.1: 2D okolí CA, zleva typy Von-Neumann, Moore, Extended Moore. [19]

### 3.2 Wolframovy 1D celulární automaty

Celulární automaty byly dlouho chápány jako výpočetní prostředek. Britský fyzik a matematik Stephen Wolfram je popularizuje a zkoumá již od 80. let. Věnuje se zkoumání různých pravidel CA a programů na nich založených. Místo snahy popsat složité jevy a systémy matematickými rovnicemi se snaží používat pro popis jednoduché programy založené na 1D celulárních automatech. [27] V roce 2002 vydal knihu *A new kind of science* [28], jejímž cílem je představit nový druh vědy, který je založen na obecných pravidlech, které je možné použít v jednoduchých počítačových programech. Zjistil, že program řídicí se jednoduchými pravidly může vykazovat velice komplexní chování.

Wolframovy 1D automaty používají jednoduché buňky, které mohou být pouze v jednom ze dvou stavů (1 nebo 0, pro vizualizaci pak černá nebo bílá). Pro tyto jednoduché automaty Wolfram zkoumal všechny možné přechodové funkce, čili pravidla. Následující stav každé buňky závisí pouze na aktuálním stavu této buňky a na stavu dvou buněk, které s ní sousedí, což je dohromady 8 možných kombinací. Pro každou z těchto 8 kombinací musí být definován následující stav prostřední buňky, to znamená, že pro CA tohoto typu existuje 255 různých pravidel. Jedno pravidlo lze reprezentovat 8-bitovým řetězcem. Tyto pravidla se nazývají Wolframovy pravidla a označují se číslem v desítkové soustavě o takové hodnotě, jakou má daný binární řetězec. Na obr. 3.2 je zobrazeno pravidlo 30.



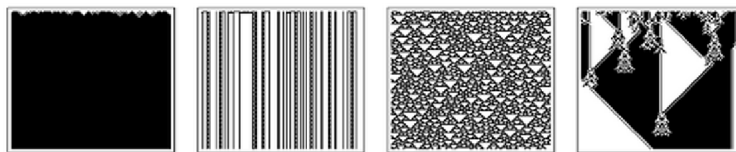
Obrázek 3.2: Wolframovo pravidlo 30, v binární podobě 00011110, ve čtyřech zobrazených případech bude hodnota buňky v následujícím kroku 1. Obr. převzat z [28].

### 3.3 Kategorie CA

Wolfram definoval 4 základní kategorie CA na základě toho, v jakém stavu se ustálí [28].

**Třída 1** Všechny buňky CA se ustálí v jednom stejném stavu po konečném počtu kroků.

**Třída 2** Chování se ustálí na periodicky se opakující posloupnosti stavů nebo automat zůstane stabilně v některém ze stavů.



Obrázek 3.3: Ukázky CA z jednotlivých kvalitativních tříd, zleva první až čtvrtá třída. Obr. převzat z [28].

**Třída 3** Automat se chová chaoticky, není možné pozorovat žádné pravidelně se opakující vzory v chování.

**Třída 4** Kombinace chaotického chování s periodicky se opakujícími vzory. Výsledkem je vznik složitých struktur. CA z této třídy vykazují nejzajímavější chování.

### 3.4 Použití CA v modelování dopravy

Němečtí fyzikové Kai Nagel a Michael Schreckenberg vyvinuli na počátku 90. let teoretický model pro simulaci dálniční dopravy. V původní verzi se jedná o jednoduchý celulární automat, který dokáže reprodukovat dopravní zácpy. Popis algoritmu je převzat z původního článku [17], kterým model představili. Jedná se o model mikrosimulační, stochastický a diskretní.

V tomto modelu je cesta rozdělena na buňky. V původním návrhu jsou tyto buňky zarovnané do jedné řady jejíž konce jsou spojeny (tzv. periodické hraniční podmínky), takže tvoří kruh. Každá buňka je buď prázdná, nebo obsazená vozidlem. Jednu buňku může okupovat vždy maximálně jedno vozidlo. Každé vozidlo má přidělenou rychlost, která je diskretní ve formě celého čísla - 0 až stanovená maximální rychlost, autoři v článku používají maximální rychlost 5. Rychlost je v jednotkách počet buněk za jednotku času. Čas je v tomto modelu také diskretní veličinou. Model je určen pro simulaci jednoproudé silnice a není umožněno předjíždění ani kolize vozidel. Model dodržuje principy particle hopping modelů s jejich výhodami i nevýhodami, které jsou zmíněny v kapitole rozdělení modelů a dále v této kapitole u popisu jednotlivých rozšíření tohoto modelu.

V každém kroku simulace je proveden výpočet nové rychlosti jednotlivých vozidel a případně jejich posunutí na novou pozici. Rychlost se počítá pro všechna vozidla paralelně. V základní podobě algoritmu je tento výpočet rozdělen na následující 4 kroky, které jsou provedeny v uvedeném pořadí.

1. Zrychlení

Pokud vozidlo nejede maximální rychlostí, je jeho rychlost zvýšena o jednu jednotku.

2. Zpomalení

Provede se kontrola vzdálenosti od předchozího vozidla. Pokud je vzdálenost menší, než současná rychlost vozidla, je rychlost snížena na hodnotu počtu volných buněk před vozidlem. Například pokud je současná rychlost vozidla 5 buněk za jeden simulační krok, ale před vozidlem jsou pouze 2 volné buňky, upraví se rychlost vozidla na 2. Tento krok zabraňuje kolizím mezi jednotlivými vozidly.

### 3. Náhodné zpomalení

Každé vozidlo s nenulovou rychlostí je s pravděpodobností  $p$  zpomalen o jednu jednotku. Tento krok vnáší do simulace stochastické chování vozidel, což má napodobovat reálné chování řidičů, kteří také nedodrží stále stejnou rychlost. Tento krok je nutný pro simulaci realistické dopravy, bez tohoto kroku by byl model čistě deterministický.

### 4. Pohyb vozidla

Vozidlo je posunuto o počet buněk shodný s hodnotou své současné rychlosti.

V popsáném systému se tedy s časem nemění počet vozidel a je možné definovat konstantní hustotu dopravy na silnici jako  $\rho = \frac{N}{L}$ , kde  $L$  je počet buněk silnice a  $N$  počet vozidel, které v simulaci figurují. Toto ve skutečném provozu většinou není možné, proto v zájmu napodobení reality se **hustota dopravy**  $\bar{\rho}^T$  v tomto modelu počítá na jedné vybrané buňce  $i$  jako průměr za časovou periodu  $T$ :

$$\bar{\rho}^T = \frac{1}{T} \cdot \sum_{t=t_0+1}^{t_0+T} n_i(t) \quad (3.2)$$

kde  $n_i(t) = 0(1)$  pokud je buňka prázdná (obsazená) v simulačním kroku  $t$ . S prodlužující se periodu  $T$  se hodnota  $\bar{\rho}^T$  přibližuje  $\rho$ :

$$\lim_{T \rightarrow \infty} \bar{\rho}^T = \rho. \quad (3.3)$$

**Průměrný tok** dopravy  $\bar{q}$  mezi buňkami  $i$  a  $i + 1$  za periodu  $T$  je definován podobně vztahem

$$\bar{q}^T = \frac{1}{T} \sum_{t=t_0+1}^{t_0+T} n_{i,i+1}(t) \quad (3.4)$$

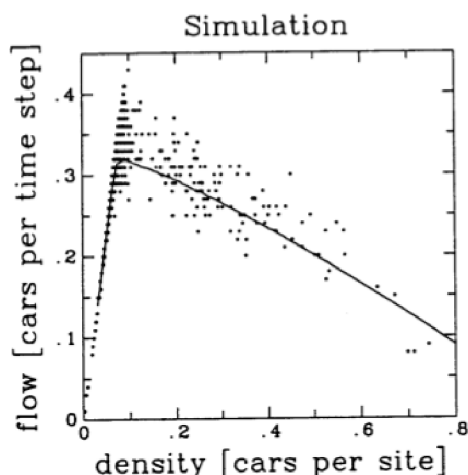
kde  $n_{i,i+1}(t) = 1$  pokud byl zaznamenán pohyb mezi buňkami  $i$  a  $i + 1$ .

Na základě experimentů s různou hustotou dopravy bylo zjištěno podobné chování, jaké lze pozorovat ve skutečnosti. Při nízké hustotě dopravního toku je provoz plynulý, kdežto při vyšší hustotě se náhodně objevují lokální kongesce, což je dáno kolísavou rychlostí vozidel. Shluky vozidel představují typické start-stop vlny, které lze pozorovat v dálniční dopravě.

Na obr. 3.4 je zobrazen fundamentální diagram modelu. Čára představuje zprůměrované výsledky  $10^6$  kroků simulace, tečky reprezentují průměr za 100 kroků simulace a jsou vhodné ke srovnání s výsledky z reálných dat na obr. 3.5. Z diagramu lze vyčíst, že přechod mezi plynulou dopravou a počátkem tvoření kongescí je přibližně při  $\rho = 0.008$ . Z dalších experimentů bylo zjištěno, že pozice tohoto přechodu a hodnota maximálního toku  $q(\rho)$  závisí na délce trati. Pokud se vypustí třetí krok randomizace, simulace nevykazují takovéto chování.

Model lze přirovnat k realitě různými způsoby. Ten nejvíce přímočarý způsob je stanovit, že jedna buňka představuje 7.5 metrů, což by dle pozorování v reálu měla být délka jednoho vozidla včetně volného místa, které si udržuje řidič před sebou, pokud vozidlo stojí v koloně. Průměrná rychlost při provozu bez kongescí se ustálila při simulaci na 4.5, což by mělo odpovídat v provozu na dálnici rychlosti 120 km/h, jeden krok simulace by pak představoval 1 sekundu ve skutečnosti.

Druhým možným přístupem je přiřadit parametrům protějšek na základě dosažené maximální hodnoty toku  $q$ . V diagramu lze vyčíst, že maximální tok při simulaci se pohybuje



Obrázek 3.4: Zobrazení dopravního toku  $q$  (jednotka počet vozidel za 1 krok simulace) v závislosti na hustotě dopravy (počet vozidel, která projela přes buňku) získaného z výsledku simulace na vozovce o délce 10 000 buněk. Tečky jsou průměry přes 100 kroků simulace, čára přes  $10^6$  kroků. Převzato z [17].

kolem hodnoty 0.32 vozidel za jeden krok simulace. V reálných datech pak kolem 2000 vozidel za hodinu, což je 0.56 vozidel za sekundu. Jeden krok simulace by tedy odpovídal času 0.5 sekundy.

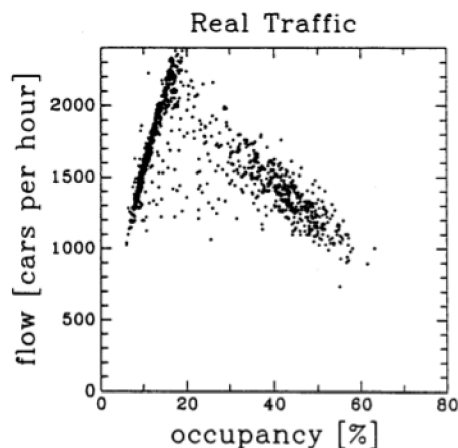
### Časoprostorový diagram

Časoprostorový diagram zobrazuje stav celulárního automatu v průběhu času. Jednotlivé řádky jsou zobrazením stavu vozovky v různém čase. Při textovém výstupu (ASCII reprezentaci) se používá následující notace: tečky na řádku představují prázdné buňky a obsazené buňky jsou reprezentovány číslem, které značí rychlost vozidla. Buňky jsou zobrazeny na ose  $x$  zleva doprava. Osa  $y$  představuje čas plynoucí shora dolů. Na obr. 3.6 je zobrazen příklad takového diagramu, jen se jedná o grafické zobrazení, bílá místa představují prázdné buňky, černá místa vozidla.

### Výhody a nedostatky

CA modely patří mezi jednoduché modely dopravy s přímočarou logikou, která si většinou vystačí s několika celočíselnými operacemi. Díky jejich jednoduchosti je možné provést i stovky milionů aktualizací stavu za sekundu [21], takže jsou teoreticky vhodné pro simulaci husté dopravy v rozsáhlých dopravních sítích.

Mají však i nedostatky, například v základní formě nemodelují realisticky chování řidiče a vozidel [9]. Vozidla jsou modelována jako částice s nerealistickými hodnotami zrychlení a zpomalení. Zrychlení vozidel vůbec nezávisí na jejich současné rychlosti. V původním modelu se mění rychlost vozidel skokově o hodnoty kolem 27 km/h během pouhé jedné sekundy. Dochází k náhlým změnám rychlosti, v modelu je možné, aby vozidlo jedoucí maximální rychlostí zpomalilo až do úplného zastavení během jedné sekundy. Zrychlování a zpomalování vozidel tedy neodpovídá skutečnosti. Jak už bylo zmíněno výše, vozovka je rozdělena na buňky o pevné délce, v původním modelu jedna buňka představuje úsek cesty



Obrázek 3.5: Zobrazení dopravního toku  $q$  (počet vozidel za hodinu) v závislosti na obsazenosti vozovky (procento obsazení vozovky) z dat nasbíraných v reálné dopravě. Převzato z [17].

o délce 7.5 metrů. To může být dostačující při simulaci dálniční dopravy, ale při simulaci dopravy ve městě už to může představovat velké odchylení od reality. Znamená to, že vozidla se mohou pohybovat pouze o násobky této délky a to může být pro některé účely až příliš velké zjednodušení. To jsou dle kritiků nejzávažnější problémy CA modelů, kvůli kterých nejsou vhodné pro provádění studií dopravy, kde je potřeba úroveň vysoké přesnosti [9]. Pro svou jednoduchost zůstávají však oblíbené a vzniká spousta variant jejichž cílem je přesnější napodobení skutečnosti [13] [10].

### 3.4.1 Možná vylepšení modelu

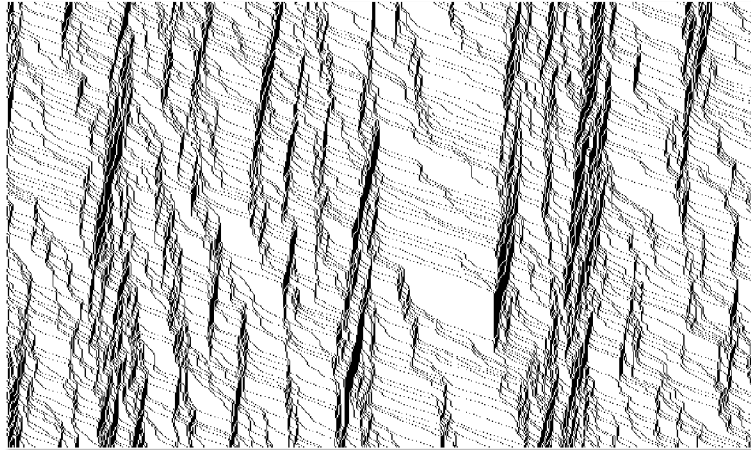
Představený model vykazuje už v základní podobě netriviální chování a je možné na něm zkoumat vlastnosti dopravy. Existuje však spousta možných způsobů, jak by se dal tento model rozšířit a využíval více poznatků o dopravě. [9]

#### Velikost buněk

Základní předpoklad, že jednu buňku obsazuje maximálně jedno vozidlo, s sebou přináší nevýhodu v tom, že všechna vozidla v simulaci zabírají na vozovce stejně místa. To neodpovídá příliš realitě, protože autobus nebo kamión je rozhodně delší než osobní automobil. Bylo by tedy vhodné rozdělit vozovku na kratší buňky a stanovit různé typy vozidel, které by zabíraly různý počet buněk na vozovce. Zde by byl prostor i pro experimentování s délkou jedné buňky, čím menší buňka by se brala v úvahu, tím přesnější simulaci by mělo být možné provést [25].

#### Maximální rychlost vozidla

Původní model stanovuje stejnou maximální rychlost pro všechna vozidla. To by mohlo reflektovat z reálné dopravy maximální povolenou rychlost na vozovce. Rychlost vozidla je ale ovlivňována i jinými vlivy jako jsou schopnosti řidiče, typ vozidla, nebo stav vozovky.



Obrázek 3.6: Časoprostorový diagram, na němž lze pozorovat tvoření kongescí v dopravě. Jedná se o výstup simulátoru vytvořeného v rámci této práce.

Toto rozšíření tedy souvisí s možností definovat různé typy vozidel. Pro každý typ vozidla by mohlo být možné nastavit maximální rychlost, kterou daný typ může dosáhnout.

V tomto modelu však není možné předjíždění, protože modeluje pouze jednoproudou vozovku. Rychlejší vozidla by tak byla zpomalována pomalejšími vozidly a tvořily by se kolony. Tento jev lze pozorovat i v běžném provozu, na silnicích jsou však úseky, na kterých je možné předjíždět. Tento nedostatek řeší až modely víceproudé vozovky [24].

### Změna simulačního kroku

Jeden krok simulace by mohl představovat časový úsek jiné délky. V původních experimentech představuje jeden krok simulace jednu sekundu. Bylo by možné experimentovat s různou délkou kroku simulace. S větší délkou simulačního kroku by se zmenšila hodnota nejmenší možné změny rychlosti, která je v původním modelu 27 km/h. Délka kroku simulace bývá v některých experimentech nastavována na reakční dobu řidiče. Je to doba, kterou neinformovaný řidič potřebuje k tomu, aby zareagoval na nepředvídatelnou událost. Tato doba je ovlivněna mnoha faktory, ale existují experimenty, z jejichž výsledků by se dalo vycházet. Např. v rámci validace IDS (*Iowa Driving Simulator*) bylo provedeno měření reakční doby na skutečné trati a v průměru začali řidiči točit volantem po 1.28s, pustili plyn po 1.67s a sešlápli naplno brzdu po 2.3s [15].

### Změna okolí buňky

Vozidla mění rychlost na základě vzdálenosti předchozího vozidla. V tomto kroku se zmenšuje rychlost pouze pokud je vzdálenost předchozího vozidla menší než aktuální rychlost vozidla. Při větších rychlostech si však řidiči často udržují větší odstup než při stání v zácpě nebo při pomalém provozu. Vozidla by mohla měnit rychlost na základě výskytu vozidel před sebou, které jsou i ve větší vzdálenosti, než je současná rychlost daného vozidla.

## Pravděpodobnost zpomalení a zrychlení

Je možné zavést k pravděpodobnosti zpomalení také pravděpodobnost zrychlení. Původní pravidlo č. 1 zahrnuje zrychlení vozidla o jednu jednotku vždy, pokud vozidlo zatím nedosáhlo maximální rychlosti. Ve skutečnosti se však některá vozidla rozjíždí pomaleji než ostatní, je tedy vhodné zavést pravděpodobnost zrychlení. Tyto dvě pravděpodobnosti by mohly být také různé pro jednotlivé typy vozidel. Pokud by se modeloval úsek cesty o kterém by byly dostupné detailní informace, mohla by se pravděpodobnost zpomalení vázat k jednotlivým úsekům cesty. Pokud je stav vozovky v horším stavu, je větší pravděpodobnost, že některá vozidla zpomalí.

## Hranice minimální rychlosti

V původním modelu není žádná hranice minimální rychlosti. V třetím kroku algoritmu výpočtu rychlosti vozidla je náhodné zpomalení možné až na nulovou rychlost. V praxi však vozidla většinou bezdůvodně nezastavují, na dálnici pak už opravdu výjimečně a to v případech nějaké nehody. Na běžných silnicích jsou výjimkou prostředky hromadné dopravy. Dobrý vliv by mohlo mít zavedení hranice minimální rychlosti. Třetí krok by v tom případě byl vykonán pouze pro vozidla, které mají vyšší než minimální rychlost.

## Ukládání předchozích rychlostí

Vozidla by si mohla uchovávat rychlosti z předchozích několika kroků simulace, tuto informaci by následující vozidlo mohlo použít k vyhodnocení akcelerace předchozího vozidla a podle toho upravit svou rychlost.

## Zjemnění akcelerace a zpomalování vozidel

Ve skutečném provozu lze pozorovat tzv. metastabilní stavy [8]. Dopravní tok je závislý kromě hustoty také na stavu dopravního toku na místě, odkud vozidla vyjíždí na sledovaný úsek cesty. Pro zanesení těchto požadavků je třeba změnit v algoritmu výpočtu rychlosti některá pravidla, tzv. slow-to-start pravidla. Tyto pravidla přidávají různá omezení právě pro vozidla, která vyjíždí z dopravní zácpy, většinou jejich rozjezd zpomalují.

Mezi tato rozšíření patří model pánů M. Takayasu a H. Takayasu. Tato varianta upravuje první pravidlo základního modelu. Pokud vozidlo s nulovou rychlostí má před sebou jednu prázdnou buňku, zrychlí s pravděpodobností  $q_t = 1 - p_t$ , ostatní vozidla zrychlují deterministicky. Důsledkem je, že na výjezdu z velkých dopravních kongescí nevznikají doprovodné malé zácpy. Výpočet rychlosti je upraven na základě počtu buněk před vozidlem.

## Slow-to-Start pravidla

Jiná varianta byla představena trojicí Benjamin, Johnson a Hui, podle kterých je nazvaná, zkráceně BJH. Upravuje první dva kroky základního algoritmu. Pokud vozidlo zastaví v kroku  $t$  kvůli stojícímu předchozímu vozidlu, je nastaven příznak značící tuto skutečnost. Pokud v následujícím kroku  $t + 1$  je před vozidlem alespoň jedna prázdná buňka, rozjede se toto vozidlo s pravděpodobností  $1 - p_{slow}$ . Tento příznak má životnost vždy jen jeden simulační krok, v následujícím kroku se již rozjede s pravděpodobností  $p_{slow}$ . Tato modifikace reflektuje skutečnost, že řidič se často nerozjede okamžitě poté, co úplně zastaví. Může to být z toho důvodu, že si řidič nevšimne okamžitě, že vozidlo před ním se rozjelo, nebo to může reflektovat dobu nutnou pro nastartování motoru atp.

## Slow-to-Stop pravidla

Pánové Clarridge a Salomaa vyvinuli modifikaci BJH modelu, která obsahuje tzv. *slow-to-stop* pravidla [10]. V tomto modelu je snaha vyřešit nerealistické chování vozidel přibližujících se k dopravní zácpě, které je možné pozorovat v modelu BJH. Vozidlo se totiž může pohybovat maximální rychlostí a pak náhle, až se přiblíží k dopravní zácpě, zpomalí až do úplného zastavení v jednom kroku simulace. V tomto modelu je kontrolován před jedoucím vozidlem větší počet buněk, než je aktuální rychlost vozidla a v určitých situacích dojde ke zpomalení vozidla dříve. Následující rychlost se počítá pouze na základě rychlosti dvou za sebou jedoucích vozidel a počtem volných buněk mezi nimi, nepřináší tedy žádné velké zesložnění modelu. Lidé totiž většinou věnují pozornost rychlosti vozidla, které následují, tato informace je tedy použita pro rozhodování, jak moc a kdy má vozidlo zpomalit.

Upravená pravidla jsou formulována následovně [10]:

### 1. Slow-to-Start

Pokud  $v = 0$  a  $d > 1$ , pak s pravděpodobností  $1 - p_{slow}$  zrychlí vozidlo normálně (tento krok je ignorován) a s pravděpodobností  $p_{slow}$  zůstane rychlost vozidla v tomto kroku nulová a vozidlo zrychlí na  $v = 1$  v následujícím kroku simulace.

### 2. Zpomalení, pokud je předchozí vozidlo blízko

Pokud  $d \leq v$  a buď  $v < v_{next}$  nebo  $v \leq 2$ , tak to znamená, že předchozí auto je buď velice blízko nebo jede rychleji. Je třeba zabránit kolizi nastavením rychlosti na  $v = d - 1$ . V opačném případě, pokud  $d \leq v$ ,  $v \geq v_{next}$  a  $v > 2$ , nastaví se rychlost vozidla na  $v = \min(d - 1, v - 2)$  a dojde tak k většímu zpomalení vozidla, protože předchozí vozidlo je pomalejší nebo stejně rychlé a rychlost daného vozidla je docela vysoká. Ve vyšších rychlostech je totiž běžné, že se dodržují větší vzdálenosti než při nižších rychlostech.

### 3. Zpomalení, pokud je předchozí vozidlo daleko

Pokud  $v < d \leq 2v$ , pak pokud  $v \geq v_{next} + 4$ , zpomal o 2 jednotky  $v = v - 2$ . Jinak, pokud  $v_{next} + 2 \leq v \leq v_{next} + 3$ , zpomal o 1 jednotku  $v = v - 1$ .

### 4. Zrychlení

Pokud rychlost nebyla zatím změněna jedním z pravidel 1-3 a pokud  $v < v_{max}$  a  $d > v + 1$ , zrychlí na  $v = v + 1$ .

### 5. Náhodné zpomalení

Pokud  $v > 0$ , zpomal o jednu jednotku  $v = v - 1$  s pravděpodobností  $p_{fault}$ .

### 6. Pohyb vozidla

Vozidlo postoupí o  $v$  buněk kupředu.

kde:

$d$  = vzdálenost od předchozího vozidla,

$v$  = rychlost daného vozidla,

$v_{next}$  = rychlost předchozího vozidla,

$p_{slow}$  = pravděpodobnost, že je aplikováno slow-to-start pravidlo,

$p_{fault}$  = pravděpodobnost, že vozidlo náhodně zpomalí,

$v_{max}$  = nejvyšší možná rychlost vozidla, model počítá s maximem 5.



## Víceproudé varianty

Protože převážná většina dálnic má dva a více pruhů v jednom směru, byly vyvinuty víceproudé modely dopravy. Jako první se této oblasti věnoval Rickert [20], další výzkumy následovaly [24]. Článek *Two-lane traffic rules for cellular automata* [18] pak přináší přehled existujících víceproudých modelů a navrhuje obecné schéma pro vytvoření realistických pravidel pro změnu pruhu. Jedná se o víceproudé modely založené na základech původního Nagel-Schreckenberg modelu. Tyto modely obsahují navíc různá pravidla pro změnu pruhu. Jedno z nutných pravidel je například to, že řidič měnící pruh musí kontrolovat pohyb vozidel v novém pruhu před sebou i za sebou. Bez tohoto pravidla nebyly pozorovány jevy typické ve skutečné dopravě. Některé varianty modelují i zvyklost, že v levém pruhu se pohybují rychleji jedoucí vozidla a pomalejší vozidla ji využívají k předjíždění.

## Kapitola 4

# Evoluční algoritmy

Evoluční algoritmy jsou optimalizační stochastické algoritmy založené na modelech evolučních procesů v přírodě a jsou charakterizovány několika společnými vlastnostmi. Pracují s populací možných řešení nějakého problému a řešení postupně vylepšují tak, že upřednostňují lepší řešení, která vznikla kombinací a mutací řešení z původní populace a horší řešení eliminují. Jsou založeny na Darwinově teorii přírodního výběru, podle které je biologická evoluce progresivní změna genetického obsahu populace v průběhu mnoha generací [22]. Obsahuje tři komponenty:

- **Přirozený výběr**  
Výběr jedinců, kteří vstupují do procesu reprodukce. Silnější jedinci mají větší šanci, že budou vybráni a jejich genetický kód bude obsažen i v následující populaci.
- **Náhodná změna genetické informace**  
V průběhu života může dojít k náhodné mutaci genetické informace jedince nebo může dojít k úhynu jedinců s vysokou fitness, takové změny se pak projeví na genetickém obsahu celé populace.
- **Reprodukční proces**  
Zajišťuje vznik nových jedinců. Genetická informace potomků je kombinací genetické výbavy jejich rodičů.

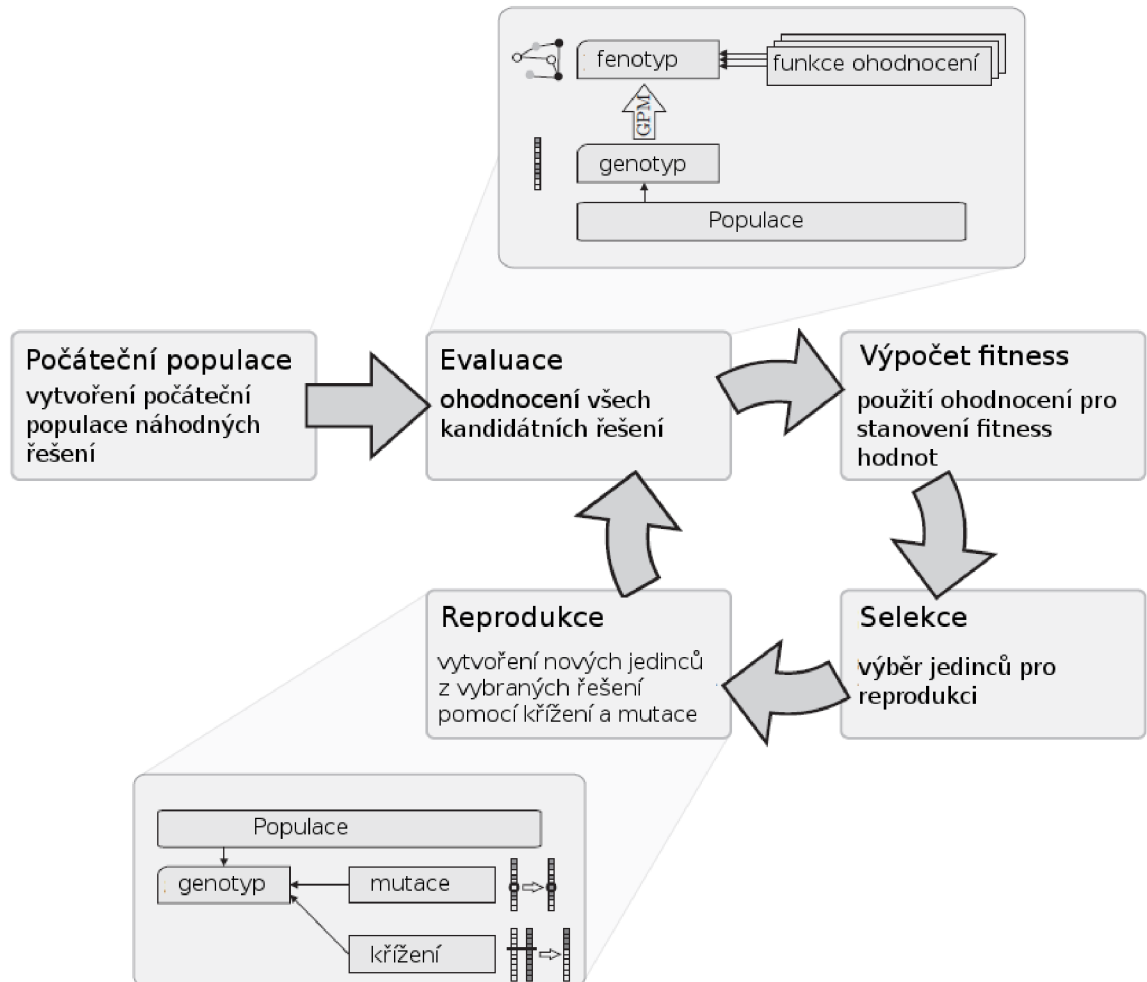
Evoluční algoritmy jsou založeny na stejném principu jako biologická evoluce. Nalézají uplatnění v oblastech, ve kterých nedostačují klasické analytické či heuristické metody, zejména při řešení NP úplných úloh [22]. Používají se v oblasti numerické optimalizace, plánování a řízení, dolování dat, strojové učení nebo umělé inteligence. Jsou vhodné pro kalibraci modelů strojového učení nebo simulačních modelů. V rámci této práce budou využity genetické algoritmy právě pro kalibraci simulačního modelu dopravy.

Evoluční algoritmy obecně mají i jisté nevýhody. Pro složitější úlohy je typická časová náročnost a není zaručeno, že nalezené řešení je to nejlepší. Zda je nalezené řešení globálním maximem nelze otestovat, protože hledání je obvykle zastaveno při dosažení postačující kvality řešení nebo po předem stanovené době. Kvalitu řešení lze tedy většinou ohodnotit pouze relativně k ostatním řešením.

Mezi evoluční algoritmy se dále řadí např. evoluční programování, evoluční strategie, genetické programování a genetické algoritmy.

## 4.1 Genetické algoritmy

Genetické algoritmy jsou podtřídou evolučních algoritmů, která je specifická tím, že jsou řešení reprezentována binárními řetězci nebo poli elementárních typů [26]. Základní proces je zobrazen na obr. 4.1, lze z něj vyčíst, že genotyp je použit při reprodukci a fenotyp se používá při ohodnocení kandidátního řešení.



Obrázek 4.1: Hlavní procesy genetického algoritmu, GPM je zkratka pro mapování genotypu na fenotyp (*genotype-phenotype mapping*). Převzato z [26] a přeloženo.

### Evaluace individuí

Ohodnocení jedince je realizováno účelovou funkcí  $f$  (*objective function*), která ohodnotí každé kandidátní řešení  $x$  reálným číslem. Cílem GA je nalézt globální extrém funkce  $f$ . Pokud se jedná o minimalizační úlohu a jedinec je reprezentován binárním vektorem, optimální řešení je lze vyjádřit takto:

$$x_{opt} = \arg \min_{x \in \{0,1\}^n} f(x) \quad (4.1)$$

Účelová funkce  $f$  se většinou transformuje na funkci výhodnosti  $F$ , tzv. **fitness funkci**, aby původní optimalizační úloha byla převedena na úlohu maximalizační a bylo dosaženo vhodného měřítka fitness funkce. Touto transformací je možné měnit selekční tlak, čímž dochází k ovlivnění konvergence evolučního procesu. [22]

Standardní GA lze popsat následujícím pseudokódem [22]:

1. Náhodně vygeneruj počáteční populaci  $D(0)$  s mohutností  $N$  a nastav čas  $t = 0$ .
2. Ohodnoť všechny jedince populace  $D(t)$  fitness funkcí  $F(X)$ .
3. Vygeneruj populaci potomků  $O(t)$  s mohutností  $M \leq N$  pomocí operátorů křížení a mutace.
4. Vytvoř novou populaci  $D(t+1)$  nahrazením části nebo úplným nahrazením populace  $D(t)$  jedinci z  $O(t)$ .
5. Nastav čas  $t = t + 1$ .
6. Pokud není splněna podmínka ukončení algoritmu, vrať se na krok 2.

## Reprezentace problému

Volba zakódování problému je klíčovým prvkem genetických algoritmů. Způsob kódování řešení se označuje jako genotyp, genom, nebo chromozom. Pro jedince, který z takto zakódovaného řešení vznikne, se používá označení fenotyp.

Genomy v GA jsou řetězce, lineární sekvence jistého datového typu, většinou homogenní, například řetězce bitů, celých nebo reálných čísel. Genetické algoritmy jsou původní prototyp evolučních algoritmů, platí pro ně tedy vše výše zmíněné. Poskytují genetické operátory, které napodobují sexuální a asexuální reprodukci z přírody. Křížení je představitelem sexuální reprodukce, v jeho průběhu dochází k rekombinaci dvou rodičovských genotypů. V asexuální reprodukci dochází pouze k náhodné mutaci jedinců. Často se aplikují oba dva principy, nejdříve se křížením vytvoří sada nových jedinců, kteří jsou následně předmětem mutace. Následující popis se týká chromozomu s pevnou délkou.

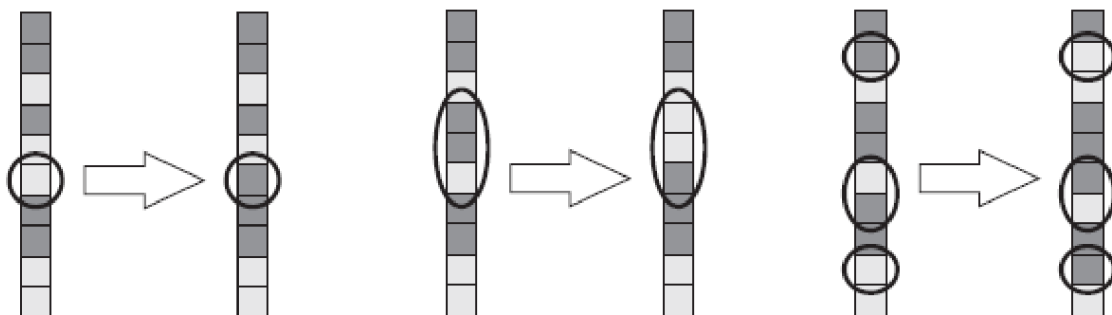
## Vytvoření počáteční populace

Počáteční populace se získá tak, že se vytvoří sada genomů a inicializují se na náhodnými hodnotami. Někdy může být výhodné využít znalosti problému a nenechat inicializaci úplně náhodnou.

## Mutace

Mutace způsobuje malé náhodné změny a udržuje tak určitou diverzitu kandidátních řešení v populaci. Nejjednodušší způsob mutace v chromozomu tvořeném řetězcem elementárních typů je náhodná změna hodnoty jednoho genu. Dají se vymyslet různé varianty mutace a je třeba při určování vhodnosti dané varianty vždy brát v potaz konkrétní řešený problém a použité kódování jedince. Mezi další možnosti patří změna delší sekvence genů, náhodný výběr určitého počtu genů nebo změna každého genu s určitou pravděpodobností. Na obr. 4.2 jsou zobrazeny některé varianty pro chromozom tvořený binárním řetězcem. Alternativní metodou je *permutace*, kde dojde k výměně genů tak, aby bylo dodrženo, že se žádný gen v jedinci nebude opakovat. Permutace je vhodná například u problémů hledání optimálního

pořadí nějakých elementů, například při řešení problému obchodního cestujícího, což je úloha nalezení nejkratší možné cesty procházející všemi zadanými body na mapě.

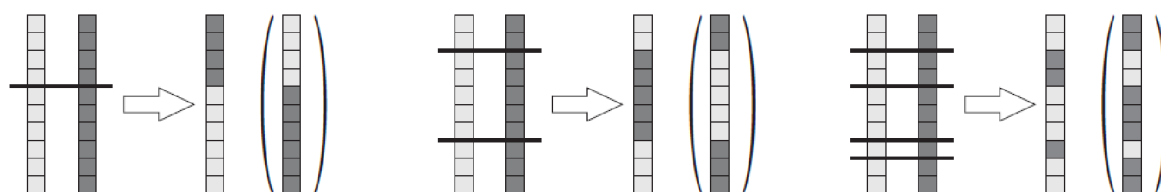


Obrázek 4.2: Ukázka možného řešení mutace. Vlevo mutace jednoho náhodně vybraného genu, uprostřed mutace souvislé sekvence genů a napravo mutace náhodně vybraných genů. Obr. převzat z [26].

### Křížení

Všechny evoluční algoritmy využívají nějaké operátory rekombinace, pro genetické algoritmy je typické použití křížení (*crossover*), které je velice podobné křížení v přírodě. Na obr. 4.3 jsou zobrazeny některé varianty křížení dvou chromozomů tvořených binárními řetězci. Křížení se provádí prohozením některých částí dvou genotypů.

Při jednobodovém křížení jsou chromozomy rozděleny v náhodně vybraném bodě a noví jedinci jsou vytvořeni ze dvou částí, každé z jiného rodičovského chromozomu. Při dvoubodovém křížení jsou rodičovské chromozomy rozděleny ve dvou bodech, první potokem je pak složen z první a třetí části z prvního rodiče a prostřední části z druhého rodiče, jak je zobrazeno na obr. 4.3. Obecně může být křížení v libovolném počtu. Při použití chromozomu o pevné délce jsou body křížení pro oba rodiče vždy totožné.



Obrázek 4.3: Ukázka možného řešení křížení. Vlevo jednobodové (*single-point*), uprostřed dvoubodové (*two-point*) a napravo obecné vícebodové křížení (*multi-point*). Obr. převzat z [26].

### Selekce

Pomocí operátoru selekce jsou vybíráni jedinci do reprodukčního procesu. Jedná se o důležitou část GA, musí do určité míry upřednostňovat jedince s vyšší hodnotou fitness, ale musí

zachovávat určitou diverzitu populace. Pokud by nové populace byly sestavovány pouze z nejlepších řešení ze staré populace, docházelo by k předčasné konvergenci, to znamená, že algoritmus mohl uváznout v lokálním optimu a nenalezl by optimum globální. Následuje popis některých běžně používaných operátorů selekce. [22]

### Ruleta

Jedná se o náhodný výběr pomocí rulety (*roulette wheel selection*), při němž je pravděpodobnost výběru jedince úměrná jeho fitness  $f$ :

$$p_i = \frac{f_i}{\sum_{j=0}^N f_j} \quad (4.2)$$

Nevýhoda tohoto přístupu se projeví, pokud je v populaci jeden opravdu silný jedinec, s postupem času pak tento jedinec nahradí všechny ostatní. To je možné řešit škálováním původní fitness tak, aby se zmenšil rozdíl v hodnocení jednotlivých kandidátů.

### Turnajová selekce

Turnajová selekce patří mezi jednoduché metody s dobrými výsledky. Z kandidátních řešení se vybere  $n$  jedinců a do další populace postupuje pouze nejlepší z nich. Pokud je třeba vytvořit populaci o velikosti  $N$ , opakuje se tento postup  $N$ -krát.

### Lineární a exponenciální uspořádání

Tyto metody (*ranking*) vyžadují nejdříve populaci setřídit podle fitness hodnoty jednotlivých řešení, pravděpodobnost výběru jedince je pak určena vztahem závislým na zvoleném selekčním tlaku, počtu kandidátních řešení a pozici konkrétního řešení v seřazené posloupnosti. Pravděpodobnost výběru je pak rozložena v populaci lineárně nebo s exponenciální závislostí. Exponenciální uspořádání pracuje s normalizační konstantou  $c$ , pomocí které je možné docílit vhodného tlaku při nízké ztrátě variability populace.

## Kapitola 5

# Analýza a návrh dopravního simulátoru

Cílem této práce je vytvořit mikroskopický simulátor dopravy s podporou kalibrace modelu. Simulátor bude založen na celulárním automatu. Nejdříve bude implementován standardní model, který byl představen autory Nagel a Schreckenberg v roce 1992. Poté bude implementováno rozšíření *Slow-to-Start* a *Slow-to-Stop* pravidel. U obou těchto variant bude provedena kalibrace modelu pomocí genetického algoritmu na empirických datech získaných ze senzorů na ulici Evropská v Praze. Jedná se o simulaci na otevřené trase, vozidla budou na jednom konci vozovky vjíždět a na druhém konci vyjíždět.

Simulátor bude implementován v jazyce *C++*, pro vizualizaci bude použita knihovna *Allegro*, genetický algoritmus pro kalibraci bude využívat knihovnu *GAlib*. Simulátor bude navržen objektově a při implementaci budou využity objektově orientované vlastnosti jazyka *C++*. Prioritou této práce není vytvořit co nejrychlejší implementaci CA, proto budou entity modelovány přímočaře a srozumitelně nezávisle na tom, zda se jedná o nejefektivnější řešení. Tím by mělo být také dosaženo jednoduché rozšiřitelnosti modelu.

### 5.1 Požadavky na systém

Systém by měl být schopný simulovat dopravu na úseku jednoproudé silnice. Systém by měl počítat s různými typy vozidel, protože součástí použitých dat je informace o typu vozidla, která může být využita ke zpřesnění modelu. Předjíždění, křižovatky a semaforey nebudou brány v úvahu. Pro účely experimentování by měl systém pracovat s konfiguračním souborem, který by obsahoval parametry tratě a jednotlivých typů vozidel.

Program by měl být schopen načíst dodaná data a využít je ke generování dopravy. Data obsahují přesné časy příjezdu i odjezdu konkrétních vozidel, dále pak čas průjezdu v sekundách a typ vozidla. Simulátor by měl na vozovku umisťovat vozidla v přesných časech na základě těchto dat. Druhou možností je data předzpracovat dle popisu v následující sekci a použít tato data pro náhodné generování dopravy dle nějakého pravděpodobnostního rozložení.

Výstupem programu by mělo být porovnání výsledků simulace s empirickými daty. Pokud budou použita data obsahující dojezdy konkrétních vozidel, bude se vhodnost nastavení určovat na základě průměrné hodnoty absolutní chyby (*Mean Absolute Error*, MAE) v čase dojezdu vozidel.

Simulátor by měl být schopen vypočítat a prezentovat statistické údaje jako je průměrná

dojezdová doba pro časové úseky nastavitelné délky a průměrná odchylka od očekávaných hodnot. Dále by mohl počítat i makrosimulační veličiny - intenzitu toku, hustotu a průměrnou rychlost dopravy.

Generování provozu by mělo být možné i pomocí náhodných generátorů s nastavitelnou intenzitou. V tomto režimu bude možné otestovat maximální kapacitu vozovky při daném nastavení.

Pro účely prezentace by měl být simulátor schopen vizualizovat stav silnice. Toho může být dosaženo několika způsoby:

- výstupem dat ve formátu vhodném k vykreslení časového diagramu v nějakém programu třetí strany,
- přímým vykreslením časového diagramu na obrazovku pomocí některé z dostupných GUI knihoven, nebo
- ASCII vizualizací na výstupu programu.

## 5.2 Předzpracování dat

Data není nutno pro základní použití předzpracovávat. Před spuštěním simulace bude nutné seřadit jednotlivé údaje podle časů příjezdu vozidel na vozovku, to může být vyřešeno přímo v aplikaci po načtení dat. Pro účely analýzy dat budou z dat získány souhrnné údaje pro časové úseky o různé délce (například 15, 30, 60 minut). K těmto údajům patří počet vozidel, která vjedou na vozovku v tomto čase a průměrná dojezdová doba. Tyto údaje budou vykresleny do grafu bude je možné porovnat s výsledky simulace. Pro účely kalibrace pomocí GA bude nutné data rozdělit na trénovací a testovací sadu. Trénovací sada bude tvořena daty z jednoho dne a nejlepší nalezené řešení bude poté otestováno na datech z jiného dne.

## 5.3 Kalibrace modelu

Simulátor má podporovat kalibraci modelu. Ta bude založena na hledání nejvhodnějších parametrů simulace pomocí genetického algoritmu. Pro tyto účely bude využita knihovna GAlib. Bude nutné vhodně zakódovat hledané parametry, zvolit vhodnou fitness funkci a vhodné genetické operátory.

Kalibrace modelu bude provedena hledáním nejvhodnějších parametrů simulátoru pomocí genetického algoritmu. Při návrhu genetického algoritmu je třeba řešit etapy zmíněné v kapitole 4. Některé však budou vybrány na základě výsledků experimentů nebo budou použity výchozí možnosti, které nabídne použitá knihovna. V této fázi je vhodné navrhnout vhodnou reprezentaci jednoho genomu, způsob jakým budou jednotlivá řešení ohodnocena a vybrat vhodné varianty rekombinačních operátorů.

### Reprezentace problému

Pro reprezentaci jedince bude použit binární řetězec konstantní délky. Ten musí obsahovat parametry vozovky i nastavení všech typů vozidel. Počet typů vozidel bude možné nastavit před spuštěním GA, v průběhu prohledávání stavového prostoru se však už měnit nebude.

Je vhodné definovat pro jednotlivé parametry omezující podmínky. Ne každá hodnota může mít totiž smysl, například není žádoucí, aby maximální rychlost vozidla byla nulová



atp. Možným řešením je použít pokutovou funkci a znevýhodňovat řešení obsahující nepřipustné hodnoty parametrů. Druhým přístupem je překódování nepřipustných řešení na přípustná, což znamená, že před každým vyhodnocením tohoto jedince se nahradí hodnoty nejbližšími použitelnými hodnotami. Oba dva přístupy s sebou přinášejí nutnost vykonávat při simulaci nějaké úkony navíc, proto bude preferován třetí způsob a tím je použití *problémově specifické reprezentace problému*. Tento způsob je nejefektivnější, protože zabraňuje vzniku nepřipustných řešení už při kódování jedince. Počet bitů a odpovídající rozsahy jednotlivých parametrů jsou rozepsány v tabulce 5.1.

Parametr	Počet bitů	Rozsah
Délka buňky	4	1 - 10
Délka vozidla	4	5 - 20
Maximální rychlost	5	10 - 120
Minimální rychlost	4	0 - 10
Pravděpodobnost zpomalení	7	0 - 0.8
Pravděpodobnost zrychlení	7	0.3 - 1.0

Tabulka 5.1: Počty bitů a návrh rozsahů pro zakódování parametrů do binárního řetězce.

## Evaluace individuů

Fitness funkce má vyhodnotit vhodnost daného řešení. V našem případě je řešením sada parametrů pro nastavení simulátoru dopravy. Aby bylo možné určit míru vhodnosti nastavení, je nutné provést simulaci s trénovací sadou dat.

Pro režim simulace s daty obsahující časy jednotlivých vozidel je možné jako hodnotu fitness funkce použít průměrnou absolutní odchylku dojezdových časů jednotlivých vozidel. Hodnotnou informací by mohla být průměrná hodnota této odchylky pro delší časové úseky, například 15 minut. Tak by bylo viditelné, jak se chyba mění v čase a které části dat dané nastavení simulátoru nejméně vyhovuje a naopak. Pokud jsou k dispozici takto detailní data, chyba bude vypočítána podle následujícího vztahu [23]:

$$MAE_t = \frac{1}{N_t} \sum_{i=1}^{N_t} |O_{it} - E_{it}| \quad t = 1, \dots, T \quad (5.1)$$

$$MAE = \frac{\sum_{t=1}^T MAE_t}{T}$$

kde:

- $t$  = časový úsek  $t$ , všechny periody mají stejnou délku
- $T$  = počet časových úseků od začátku do konce simulace
- $O_{it}$  = průměrná doba dojezdu v trénovacích datech během časového úseku  $t$
- $E_{it}$  = průměrná doba dojezdu při simulaci během časového úseku  $t$
- $N_t$  = celkový počet vozidel, která vyjela ze silnice během časového úseku  $t$

Při použití předzpracovaných dat, které obsahují průměrnou dojezdovou dobu v časových úsecích stejné délky se pak výpočet zjednoduší, protože nebude nutné tuto hodnoty z referenčních dat počítat za běhu simulace.

## Operátory rekombinace

Rekombinační operátory vytvářejí nové jedince, pro genetické algoritmy je charakteristický operátor *křížení*. Existuje mnoho variant tohoto operátoru, pro navrženou reprezentaci se jeví vhodné využít vícebodového křížení. Místa křížení by mohla být umístěná mezi stavebními bloky. Jako stavební bloky lze považovat parametry vozovky a pak každou sadu parametrů pro jeden typ vozidla. Výsledkem křížení tedy budou dva jedinci, každý s jiným nastavením vozovky i typů vozidel. Pokud bude genetický algoritmus nastaven tak, aby uvažoval pouze jeden typ vozidla, stavební bloky budou rozděleny podle tabulky 5.2.

délka buňky, vozidla	max., min. rychlost	pravděpodobnost zrychlení, zpomalení
----------------------	---------------------	--------------------------------------

Tabulka 5.2: Rozdělení stavebních bloků genomu pro genom s nastavením jednoho typu vozidla, křížení bude probíhat v bodech mezi stavebními bloky.

Druhým operátorem, který bude použit, je *mutace*. Často se používá s velice malou pravděpodobností výskytu, ale jedná se o velice významný operátor, protože je zdrojem naprosto nových řešení. Pokud by se používal s příliš velikou pravděpodobností, způsoboval by nestabilitu vývoje populace. Častým operátorem mutace při použití genomu zakódovaného do binárního řetězce je uniformní bitová negace každého genu s určitou pravděpodobností (většinou 0.0005 až 0.01) [22]. Bude použita tato varianta mutace s nastavitelnou pravděpodobností. Další možnosti mutace jsou zmíněny v části 4.1.

## 5.4 Reprezentace entit

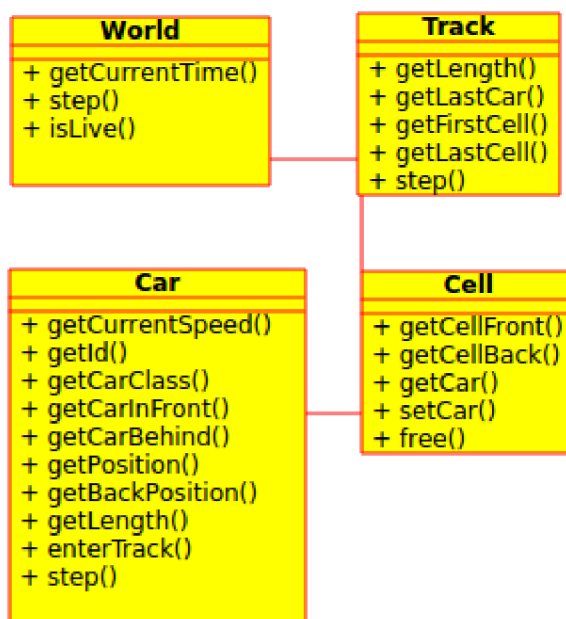
Všechny entity budou reprezentovány příslušným typem objektu. Na obr. 5.1 jsou zobrazeny 4 třídy s nejdůležitějšími metodami.

### Svět

Svět bude reprezentován objektem třídy *World*, který bude nositelem aktuálního času simulace a bude poskytovat informaci, zda simulace ještě běží. Simulace může být ukončena buď na základě nastavení maximálního času simulace nebo v okamžiku, kdy už je silnice prázdná a není naplánován vjezd žádného dalšího vozidla. Tento objekt bude zodpovědný také za likvidaci objektů vozidel, která už vyjela z tratě. Simulátor bude modelovat pouze jednu souvislou vozovku, objekt *World* bude tedy obsahovat referenci na jeden objekt třídy *Track*.

### Vozovka

Objekt třídy *Track* představuje jeden souvislý úsek vozovky o definované délce. Počet buněk, které daný úsek bude obsahovat, bude vypočítán na základě nastavení délky trasy a délky jedné buňky. Seznam aut na trati i seznam buněk, ze kterých se trať skládá, bude implementován formou obousměrně zřetěženého seznamu odpovídajících objektů.



Obrázek 5.1: Zjednodušený objektový návrh simulátoru.

## Buňka

Objekty třídy *Cell* představují úseky o stejné délce, po kterých se pohybují vozidla. Každá buňka představuje úsek o délce několika metrů, který je minimálním možným úsekem, který může vozidlo ujet za jeden krok simulace. Každá buňka je buď ve stavu "obsazeno", což znamená, že je na ní vozidlo a obsahuje referenci na objekt tohoto vozidla, nebo je ve stavu "prázdné".

## Vozidlo

Entity vozidel budou modelovány pomocí objektů třídy *Car*. Každé vozidlo si nese informaci o své aktuální rychlosti, svém typu, délce v počtu buněk, minimální a maximální rychlosti a pokud je k dispozici, tak i o skutečném dojezdovém čase.

## Konfigurace

Konfigurace simulátoru bude reprezentována objektem třídy *Config*. Ten musí obsahovat všechny nastavitelné parametry modelu. Parametry musí být schopni načíst z konfiguračního souboru a také by měl být schopen rozkódovat parametry zakódované pro účely evoluce, čehož bude využito při kalibraci modelu. Objekt bude obsahovat možnost vytisknout nastavení, aby mohl uživatel jednoduše použít parametry nalezené evolucí pro další experimenty. Konfigurační soubor bude obsahovat nastavení v běžně používaných jednotkách (délky v metrech, rychlosti v km/h), tento objekt bude zodpovědný za správné přepočty hodnot z konfiguračního souboru na jednotky používané interně (délky v počtu buněk, rychlosti v buňky/simulační krok). Pro nastavení evoluce bude použit objekt třídy *GACConfig*,

který bude obsahovat různá nastavení evoluce a simulátoru, který bude při evoluci použit.

## 5.5 Generování vozidel

Na začátek trasy bude vždy připojen objekt třídy implementující rozhraní *CarFactory*. Tento objekt bude zajišťovat přísun vozidel a může být implementován různými způsoby. Nejtriviálnější varianta bude vytvářet v každém kroku simulace nové vozidlo. Pro základní model zpracuje tento objekt CSV soubor s dodanými daty a bude generovat vozidla v předepsaných časech s danými parametry. Poslední variantou bude náhodný generátor, který bude generovat určitý počet vozidel v definovaném časovém úseku na základě normálního Gaussova pravděpodobnostního rozložení.

## 5.6 Vizualizace

Dostupné budou dvě možnosti vizualizace, textová a grafická. Při textové vizualizaci se bude na výstup programu vypisovat stav vozovky stejně jako je zvyklostí v různé literatuře. Každá buňka bude reprezentována jedním znakem, tečka představuje prázdnou buňku, číslo představuje rychlost vozidla na obsazené buňce. Pokud vozidlo zabírá více než jednu buňku, zobrazí se jako posloupnost stejných čísel. Na jednom řádku bude zobrazen stav vozovky v jednom kroku simulace.

V grafické vizualizaci bude zobrazena celá šířka vozovky, to znamená, že bude nutné obraz škálovat na rozumnou hodnotu, pokud bude počet buněk vozovky větší než počet pixelů použitého monitoru. V tomto režimu bude možné simulaci zastavit, aby bylo možné vizualizaci vyfotit nebo detailně prozkoumat.

## Kapitola 6

# Popis implementace

Pro vývoj simulátoru byly použity některé knihovny a programy třetích stran. Program *CMake* je multiplatformní systém pro kontrolu procesu kompilace, který generuje nativní *makefile* soubory pro program *make*. Je pomocí něj vyřešené vyhledání cest ke knihovnám, na kterých je program závislý. Pomocí grafické knihovny *Allegro* je vytvořena grafická vizualizace průběhu simulace. Automatizované testy využívají knihovnu *Google Test*. Otestováno je načítání konfigurace z konfiguračního souboru i z binárního řetězce, který se využívá při evoluci. Je otestován také správný převod na interně používané jednotky.

Existuje mnoho volně dostupných nástrojů usnadňujících implementaci genetického algoritmu (např. Beagle, Evolving Objects, GPC++ a GALib). V rámci analýzy byl proveden průzkum možností jednotlivých knihoven a pro použití byla vybrána knihovna GALib. Knihovna je relativně jednoduchá a mé předchozí zkušenosti s touto knihovnou jsou pozitivní. Galib dovoluje použití generické reprezentace řešení, není vázán na žádnou předem definovanou strukturu, ale jinak je knihovna relativně striktní. Nevýhody této knihovny dle [11]:

- Hodnota fitness musí být skalární hodnotou.
- Existuje pouze šest typů genetických operátorů: inicializace populace, vyhodnocení fitness, výběr jedinců, kritérium ukončení, křížení ze dvou rodičů a mutace. Pro existující evoluční modely není možné definovat jiný operátor než tyto.
- Pro každý evoluční model musí být vybrán právě jeden evoluční operátor od každého typu, nelze jich použít více stejného typu.
- Evoluční model není generický, pro každý dostupný genetický algoritmus je implementován zvlášť.
- Formát výstupu evoluce není možné změnit.

I přes tyto nedostatky je knihovna pro účely kalibrace vytvořeného modelu vhodná. Obsahuje několik předdefinovaných algoritmů a genetických operátorů, které jsou pro tento účel dostačující.

Při implementaci byl dodržen objektový návrh z předchozí kapitoly. Jednotlivé části simulátoru jsou lehce nahraditelné a rozšiřitelné. Stejně tak je možné lehce přidat další nastavení do konfiguračního souboru nebo například změnit podobu genomu pro evoluci.

## 6.1 Implementovaná vylepšení

Standardní podoba algoritmu je vylepšena vysokým počtem možných nastavení, se kterými je možné experimentovat. V původním modelu se počítá s některými pevně danými hodnotami, které nemusí být vhodné pro všechny úseky cest. Popis všech možností konfiguračního souboru je uveden v příloze C. Následuje seznam nejdůležitějších možností nastavení.

### Délka jedné buňky a délka vozovky v metrech

Na základě těchto údajů simulátor určí počet buněk vozovky a také přepočítává nastavené rychlosti na interní jednotky.

### Pravděpodobnost náhodného zpomalení vozidla

Tato vlastnost přináší do simulace náhodný prvek, ovlivňuje stálost rychlosti jednotlivých vozidel.

### Maximální rychlost v kilometrech za hodinu

Původní model počítá s maximální rychlostí 4 buňky za simulační krok. V našem případě je tento údaj možné měnit. Nastavená hodnota je automaticky převedena na jednotky buňky za simulační krok, který co nejvíce odpovídá zadané rychlosti.

Simulátor obsahuje několik rozšíření původního modelu. Všechna tato rozšíření je možné v konfiguračním souboru nastavit na stejné hodnoty s jakými se počítá ve standardním modelu a tím je deaktivovat. Následující seznam obsahuje popisy všech vylepšení a jejich vliv na simulaci.

### Nastavitelná pravděpodobnost zrychlení

K pravděpodobnosti zpomalení je možné navíc nastavit i pravděpodobnost zrychlení. Každé vozidlo má totiž jinou akceleraci a způsob jízdy záleží také na řidiči.

### Nastavitelná minimální rychlost

Náhodné zpomalení může být omezeno nastavitelnou minimální rychlostí. Na většině úsecích cesty totiž nedochází k tomu, že vozidla zpomalují samovolně až do úplného zastavení.

### Nastavitelná délka vozidla

V původním modelu představuje jedna buňka vždy maximálně jedno vozidlo a experimenty byly prováděny s délkou buňky 7.5 metrů. V tomto simulátoru je však možné, aby jedno vozidlo zabíralo více buněk. Délka vozidla je nastavitelný parametr, uvádí se v metrech a je automaticky přepočítána na počet buněk.

## Více typů vozidel

Je možné nastavit libovolný počet typů vozidel. Každé vozidlo zvlášť je charakterizováno 4 parametry: délka vozidla, maximální a minimální rychlost, pravděpodobnost zpomalení a zrychlení. Použitá data nemusí být kompletní a některá vozidla mohou být neznámého typu. Proto je nutné nastavit také výchozí typ vozidla, který je použit v případě nerozpoznaného typu.

## Slow-to-Stop pravidla

Simulátor obsahuje sadu pravidel, která jsou popsána v části 3.4.1. Pokud jsou tato pravidla aktivována, vozidla kontrolují větší úsek před sebou a zpomalují dříve v případě, že předchozí vozidlo jede pomaleji. Při rozjíždění se pak vozidla rozjíždějí pomaleji kvůli použitému *Slow-to-Start* pravidlu. Toto pravidlo se aplikuje s určitou pravděpodobností, která je také nastavitelná.

## Periodické hraniční podmínky

Je možné aktivovat periodické hraniční podmínky. Pokud jsou aktivní, generátor vozidel přidává na vozovku stále nová vozidla a cesta se časem úplně zaplní. Tohoto nastavení lze využít pro analýzu kapacity silnice při konkrétních parametrech simulace a k získání dat k vytvoření fundamentálního grafu.

## Modifikace třetího pravidla

Třetí pravidlo v modelu Nagel Schreckenberg má významný vliv pouze u vozidel, které jedou maximální rychlostí. Pokud se uplatní, zpomalí tato vozidla o jednu jednotku a v následujícím kroku opět zrychlí na maximální rychlost v důsledku uplatnění prvního pravidla. Pokud opět dojde k uplatnění třetího pravidla, zůstane rychlost stejná, už však nedojde k dalšímu snížení rychlosti. Situace se mění až v situaci, kdy je v dohledu další vozidlo. Toto chování je pozměněno rozšířením, které modifikuje třetí pravidlo. Nazval jsem jej opravdové zpomalení a v konfiguraci je možné ho nalézt pod klíčem *true\_slowdown*. Úprava spočívá v tom, že pokud dojde k náhodnému zpomalení vozidla, tzn. uplatní se třetí pravidlo, dojde opravdu ke snížení rychlosti vozidla. Pokud se ve stejném kroku tedy uplatnilo první pravidlo, je současná rychlost snížena o dvě jednotky.

# Kapitola 7

## Popis experimentů

### 7.1 Analýza použitých dat

Data o dopravě na ulici Evropská v Praze jsou k dispozici ze středy 20., čtvrtka 21. a pátku 22. června 2012. V datovém souboru jsou pro každé vozidlo k dispozici tyto hodnoty:

- Identifikace úseku.
- Čas opuštění úseku ve formátu den-měsíc-rok hodina-minuta-sekunda.
- Čas příjezdu na úsek.
- Celková doba strávená na úseku v sekundách.
- Typ vozidla, který byl určen při příjezdu vozidla.
- Typ vozidla, který byl určen při odjezdu vozidla.

Příčemž jsou definovány následující typy vozidel:

- Neznámé vozidlo
- Motocykl
- Osobní automobil do 3.5 tuny
- Dodávka do 3.5 tuny (např. Ford Transit)
- Autobus
- Lehké nákladní vozidlo nad 3.5 tuny do 6 tun (např. Avia)
- Těžké nákladní vozidlo nad 6 tun (např. Tatra 815)

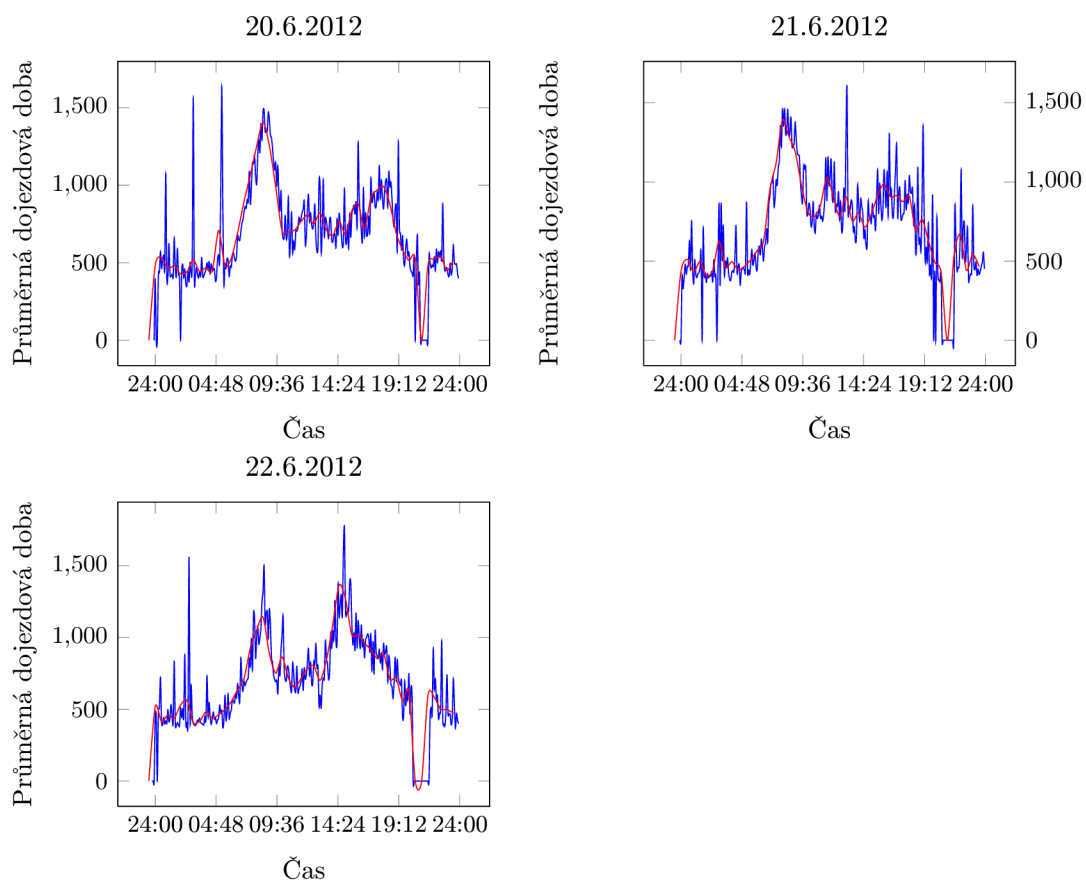
Vybraný úsek začíná na ulici Evropská, směr centrum, odjezd a končí na Evropská, Vítězná náměstí. Celková délka úseku je 5349 metrů a volná dojezdová doba je přibližně 400 sekund. Na úseku je stanovena maximální rychlost 50 km/h.

V tabulce na obr. 7.1 je uvedena průměrná doba průjezdu a průměrná rychlost vozidel z dat z 20.6., v ostatních dnech se hodnoty příliš neliší. V grafech 7.2 je zobrazena průměrná dojezdová doba v jednotlivých dnech. Modře je zobrazen průměr vypočítán pro 5 minut, červeně pro 30 minut.



Typ	Počet	T[s]	Rychlost[m/s]	Rychlost[km/h]
-1	24	527	10.15	36.54
1	3207	752	7.113	25.61
2	372	719	7.439	26.78
3	349	1072	4.99	17.96
4	37	938	5.703	20.53
5	32	886	6.037	21.73
	4021	816	6.9053	24.8592

Obrázek 7.1: Průměrné dojezdové doby a rychlosti pro jednotlivé typy vozidel ve středu 20.6.2012.

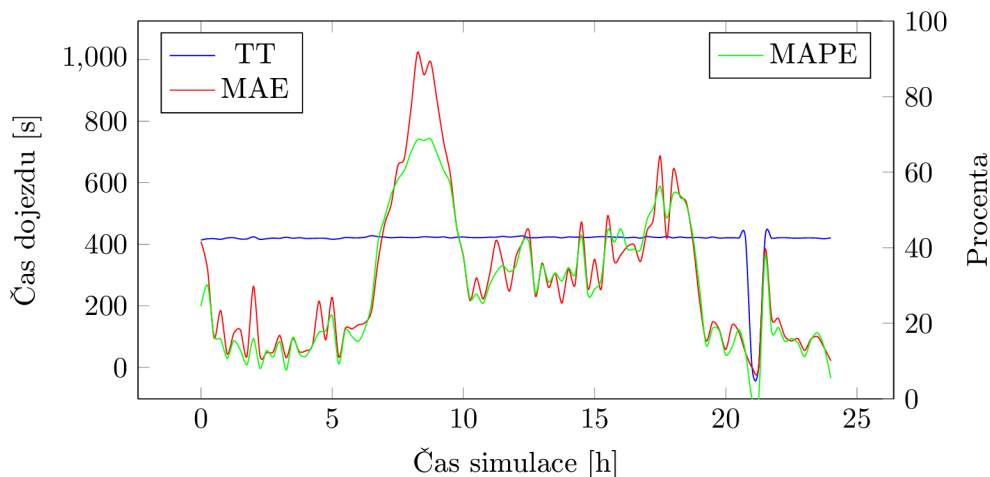


Obrázek 7.2: Dojezdová doba pro jednotlivé dny získaná ze zdrojových dat. Modře jsou vykresleny průměrné hodnoty pro intervaly o délce 5 minut, červeně pro 30 minut.

Úsek, ze kterého jsou data získána, jsem prozkoumal na detailní mapě Prahy. Skoro celý úsek má silnice v jednom směru dva proudy, na cestě je spousta přechodů pro chodce a asi 12 semaforů. Žádná z těchto skutečností není v použitém modelu zohledněna.

## 7.2 Průběh simulace

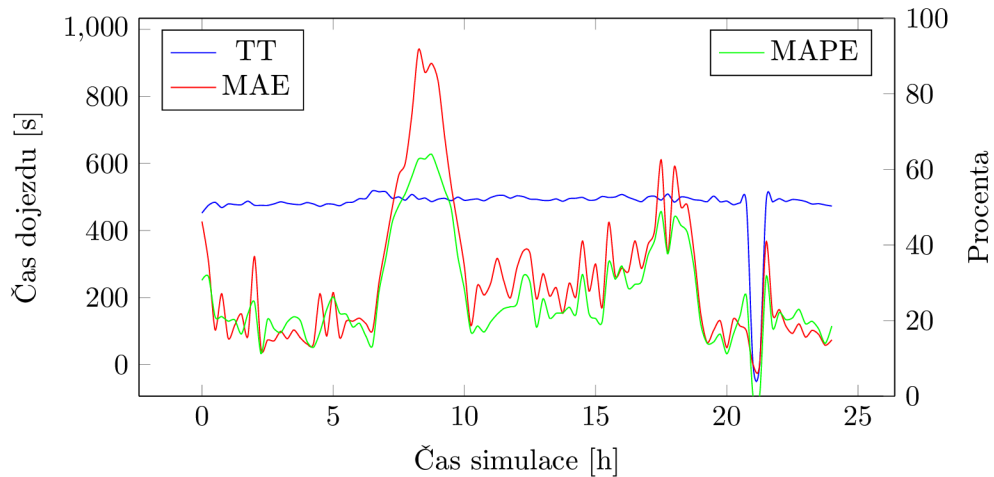
Nejdříve byla provedena simulace za pomoci modelu s nastavením takovým, aby co nejvíce odpovídal původnímu modelu. Tedy jedno auto na jednu buňku, jeden typ vozidla s pravděpodobností zpomalení 0.3, délka buňky 7.5 metrů (tedy 713 buněk), maximální rychlost 50 km/h. Průměrná chyba v dojezdové době se pohybovala kolem 300 sekund, což je 30%. V grafu 7.3 jsou zobrazeny údaje z jedné takové simulace z dat z jednoho dne (středy).



Obrázek 7.3: Průměrná doba dojezdu při simulaci (modře), odchylka v sekundách od empirických dat (červeně) a procentuální chyba (zeleně). Jedná se o simulaci s daty z 20.6. s nastavením podobným původnímu modelu.

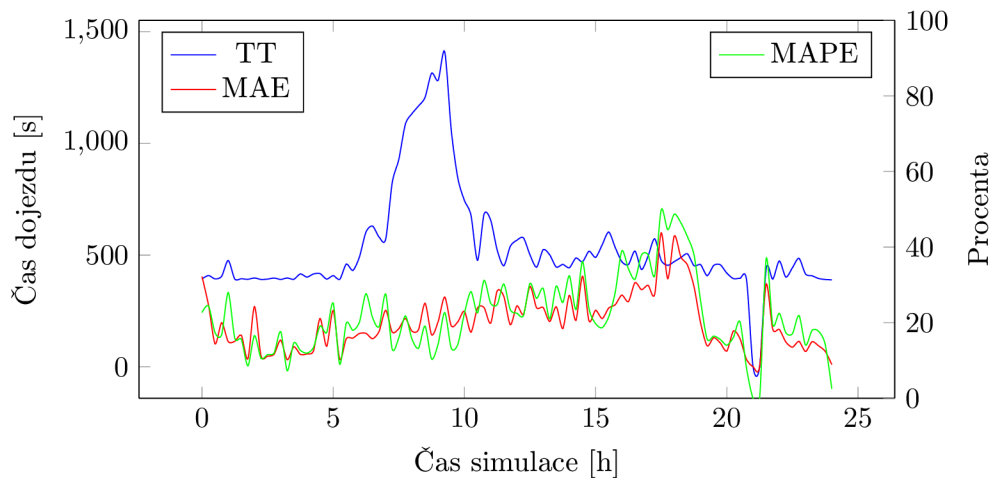
Na grafu 7.3 lze vidět, že dojezdová doba (*Travel Time*, TT) se v průběhu simulace skoro vůbec neměnila. Po detailním prozkoumání původních pravidel jsem si uvědomil, že pravidlo 3 neovlivňuje simulaci tak moc, jak jsem se domníval. V tomto kroku dojde sice k náhodnému snížení rychlosti, ale pokud vozidlo nejelo plnou rychlostí, byla jeho rychlost v prvním kroku algoritmu zvýšena. Jeho rychlost se tedy sníží pouze v případě, že jel předtím plnou rychlostí. To může být vhodné pro simulaci dopravy na dálnici, k čemuž byl model vytvořen, ale pro simulaci dopravy ve městě už je to vhodné méně. Tento třetí krok byl upraven tak, že pokud došlo v prvním kroku k akceleraci, sníží se rychlost o dvě jednotky. Zpomalení má tedy přednost před zrychlením. S touto úpravou vykazuje model trochu lepší výsledky. V grafu 7.4 jsou zobrazeny výsledky ze simulace za použití upraveného modelu. Na grafu to není příliš vidět, ale v době dopravní špičky vzrostla trochu dojezdová doba (nad 500 sekund). Další test tedy proběhl s tímto upraveným modelem. Tato změna lze aktivovat nebo deaktivovat v konfiguračním souboru.

Další test proběhl s ručně nastavenými hodnotami pro všechny implementovaná rozšíření. Parametry byly určeny co nejvhodněji na základě analýzy vstupních dat. Použité nastavení je k nalezení v příloze E. Protože simulace obsahuje náhodné jevy, výsledky jsou při každém spuštění trochu odlišné. Zobrazené grafy jsou však z důvodu zjednodušení z jednoho



Obrázek 7.4: Výsledky simulace s upraveným třetím pravidlem původního modelu.

spuštění simulace. Byla vybrána vždy simulace, která co nejvíce reprezentuje výsledky dosažené opakovaným spouštěním simulace. Chyba se pohybovala v rozmezí 21% až 31%. Na grafu 7.5 lze vidět, že v době dopravní špičky výrazně vzrostla dojezdová doba vozidel. Odchylka od skutečných časů je sice stále vysoká, ale model s tímto nastavením se podobá více realitě.



Obrázek 7.5: Výsledky simulace s daty z 20.6. s ručně nastavenými parametry modelu a aktivovanými všemi implementovanými rozšířeními.

### 7.3 Kalibrace modelu

Nastavovat ručně parametry modelu není zrovna efektivní způsob hledání nejvhodnějších parametrů pro sledovaný úsek silnice. Pro tento účel byl vytvořen program, který pomocí genetického algoritmu vyhledává nejlepší parametry simulátoru. Každá kombinace paramete-

trů představuje kandidátní řešení, které musí být nějakým způsobem ohodnoceno. Model byl kalibrován podle dojezdové doby jednotlivých vozidel. Hodnocení kandidátního řešení je dáno průměrnou chybou v dojezdové době v sekundách. Při evoluci se hledalo řešení s nejmenším ohodnocením.

Jako trénovací data byla použita data o dopravě ze dne 20.6. (středa) a nejlepší řešení pak bylo otestováno na testovací sadě dat z 21.6. (čtvrtek) a 22.6. (pátek).

Byla provedena sada různých testů, použitá nastavení a výsledky evoluce jsou přiloženy v souborech na CD v adresáři *results*.

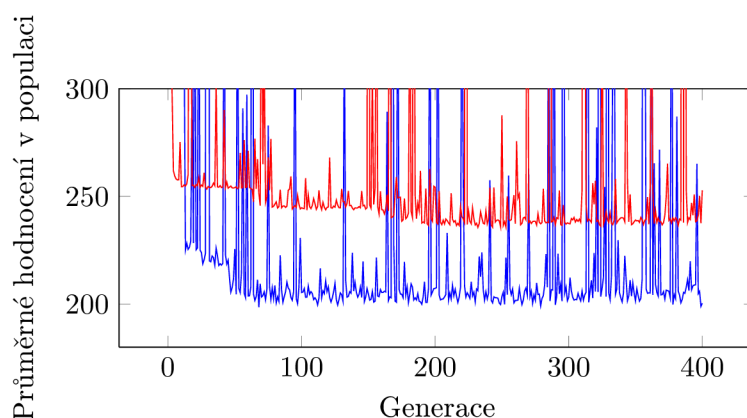
## Test 1 a 2

Pro první 4 testy byla evoluce nastavena následovně:

- 400 populací
- velikost populace 30
- pravděpodobnost mutace 0.001
- pravděpodobnost křížení 0.9
- původní třetí krok algoritmu

V testu 1 se vyhledávalo nastavení pro jeden typ vozidla, v testu 2 pro 5 typů vozidel a byl použit *SimpleGA*. Tento GA používá nepřekrývající se populace, každou generaci je vytvořena populace nových jedinců křížením a mutací jedinců z předchozí generace.

V grafu na obr. 7.6 je zobrazena konvergence evoluce při použití výše uvedených parametrů pro standardní model. V grafu je zobrazen pouze výřez s omezením hodnocení od 180 do 300, aby byly viditelné nejlepší dosažené hodnoty. Při použití jednoduchého GA nebylo hodnocení generací příliš stálé, i ke konci evoluce bylo průměrné hodnocení některých populací příliš vysoké. Modře je evoluce hledající nastavení pro 1 typ vozidla, modře hledající 5 typů vozidel. Je zajímavé, že při použití jednoho typu vozidla byla evoluce schopna nalézt lepší řešení. Průměrná doba dojezdu je přibližně 800 sekund, průměrná chyba nad 200 sekund je tedy dost vysoká.

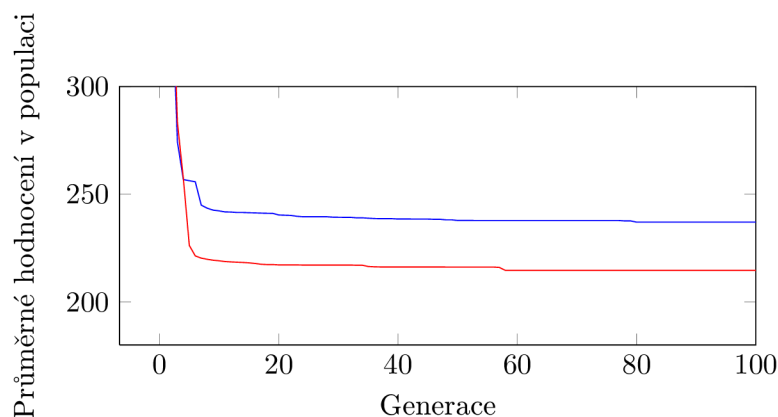


Obrázek 7.6: Průběh evoluce pro testy 1 (modře) a 2 (červeně).

V testu 1 měl na trénovací sadě (z 20.6.) nejlepší jedinec hodnocení 176, při opakovaném spuštění byla průměrná chyba z 15 simulací 194, což je chyba 22.7%. Na testovací sadě z 21.6. se chyba pohybuje kolem 260s (25%), na datech z 22.6. kolem 320s (30%). Druhý test dosáhl horších výsledků, jak je zřejmé z grafu 7.6.

### Test 3 a 4

V testu 3 a 4 byl použit *SteadyStateGA*, ve kterém se jednotlivé generace řešení překrývají. Nová populace je vytvořena tak, že k současné populaci jsou přidáni nově vytvoření jedinci získaní křížením a mutací a jsou odstraněni nejhůře ohodnocení jedinci, aby byla zachována velikost populace. Jaká část se nahradí, je nastavitelný parametr, v těchto testech je použito nahrazení 0.8% populace. V testu 3 se hledá nastavení opět pro jeden typ vozidla, v testu 4 pro 5 typů vozidla. Výsledky jsou podobné jako při použití GA s nepřekrývajícimi se populacemi. Chyba neklesla v ani jednom případě pod 20%. Na grafu jsou zobrazeny průměrné hodnocení populací v průběhu evolucí. Evoluce byla ukončena až po 400 generacích, po 100 generacích se už však průměrné hodnocení populace skoro neměnilo, graf je tedy zobrazen pouze pro 100 generací. Průběh evoluce je zobrazen v grafu 7.7.



Obrázek 7.7: Průběh evoluce pro testy 3 (modře) a 4 (červeně).

### Test 7 a 8

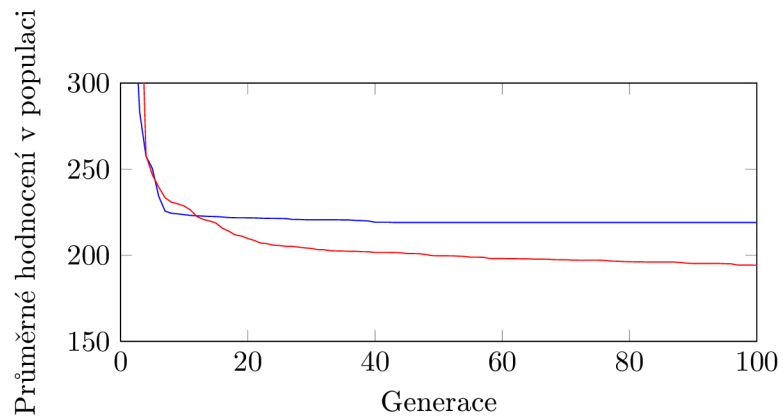
Výsledky testu 7 a 8 se také příliš neliší. Pro tyto testy byly upraveny rozsahy genomu, které jsou uvedeny v tabulce 5.1. Tyto rozsahy je možné měnit v hlavičkovém souboru *config.h*, kde jsou definovaná jako makra preprocesoru. Nové hodnoty jsou v tabulce 7.1. Dále pak byla v obou testech aktivována modifikace třetího pravidla algoritmu, která je popsána v části 6.1. Vyhledávalo se nastavení pro 6 typů vozidel (v původních datech je jich 5 a jeden typ zastupuje neznámý typ vozidla). V testu 7 nebylo aktivní rozšíření Slow-to-Stop, v testu 8 bylo aktivní a pravděpodobnost Slow-to-Start byla nastavena na 0.5. V testu 8 byly pozměněny parametry evoluce - pravděpodobnost mutace byla zvýšena na 0.01 a velikost populace zvýšena na 50 jedinců.

V grafu 7.8 je zobrazen průběh evoluce. V grafu 7.9 pak hodnocení simulace s nastavením nalezeným testem 7 a v grafu 7.10 je simulátor nastaven hodnotami nalezenými při testu 8.

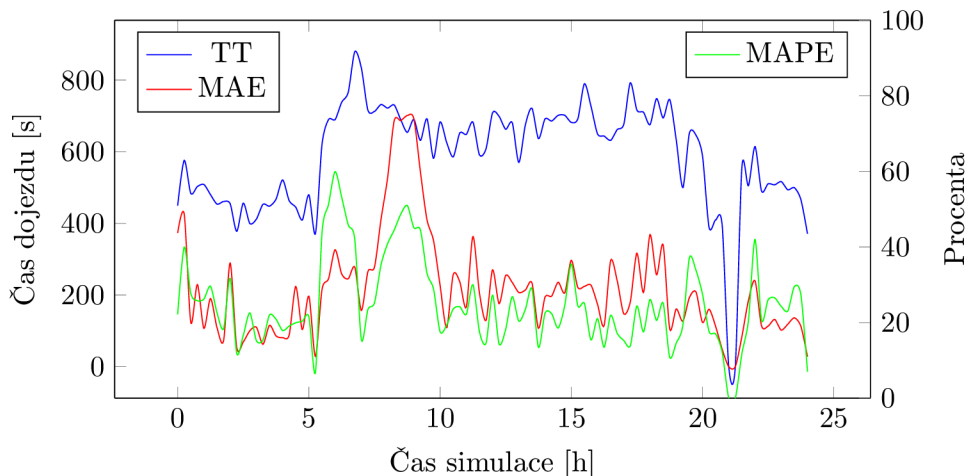
Při použití nastavení z testu 7 se pohybuje odchylka dojezdové doby na datech ze středy kolem 220 sekund, jedná se přibližně o chybu 26%. Na testovací sadě ze čtvrtka kolem 260

Parametr	Počet bitů	Rozsah
Délka buňky	6	0.3 - 16
Délka vozidla	6	5 - 20
Maximální rychlost	6	20 - 70
Minimální rychlost	4	0 - 10
Pravděpodobnost zpomalení	5	0 - 0.6
Pravděpodobnost zrychlení	5	0.3 - 1.0

Tabulka 7.1: Počty bitů a rozsahy pro zakódování parametrů do binárního řetězce.



Obrázek 7.8: Průběh evoluce pro testy 7 (modře) a 8 (červeně).



Obrázek 7.9: Průměrná doba dojezdu při simulaci (modře), odchylka v sekundách od empirických dat (červeně) a procentuální chyba (zeleně). Jedná se o simulaci s daty z 20.6. s parametry nalezenými v testu 7.

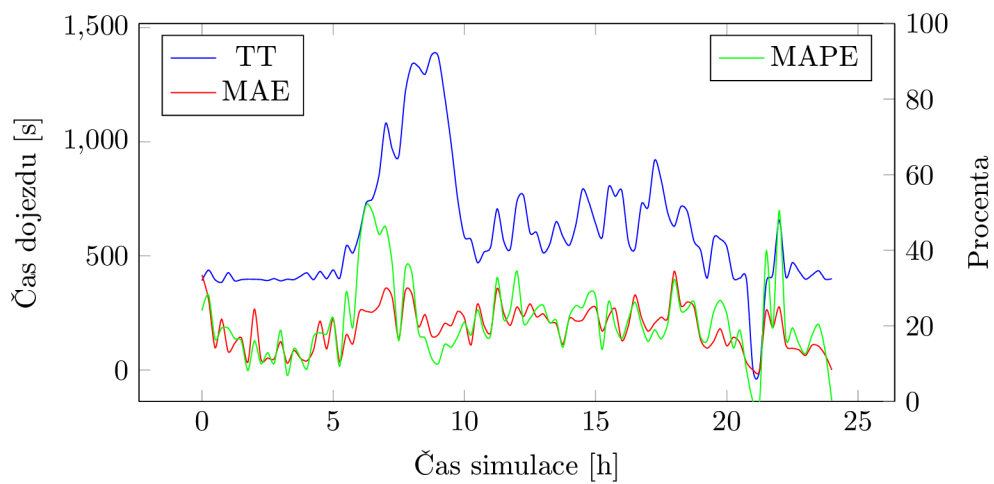
sekund (chyba 28%). Na datech z pátku je chyba největší, pohybuje se kolem 300 sekund (31%).

Při evoluci v testu 8 kleslo průměrné hodnocení populace kandidátních řešení na 194 sekund. S použitým nastavením se chyby u provedených simulací pak pohybovaly kolem těchto hodnot: středa 200s (23%), čtvrtek 270s (27%), 315s (28%-35%). Řešení nalezené evolucí se příliš neliší od parametrů, které byly ručně vybrány jako nejvhodnější.

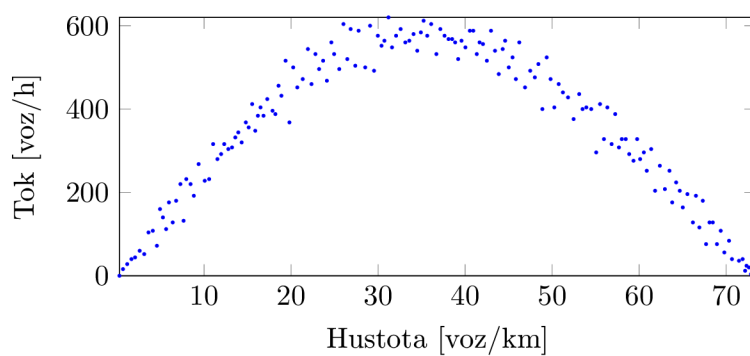
Vysoká odchylka simulací od reálných dat je zřejmě způsobena tím, že na úseku jsou prvky ovlivňující dopravu, které nejsou obsaženy v modelu. Hlavní vliv mají zřejmě semaforey, přechody a druhý pruh vozovky. V použitých datech jsou u některých vozidel příliš vysoké doby dojezdu a to i mimo dopravní špičku. Čím jsou tyto pomalé průjezdy způsobeny není možno z dat určit a není tak možno model více zpřesnit.

## Porovnání nalezených řešení

Porovnat různá nastavení modelu je možné například pomocí makrosimulačních veličin. Pro nalezená nastavení byly vytvořeny grafy závislosti toku na hustotě dopravy. Z grafů lze vyčíst při jaké hustotě dopravy je přetížena kapacita vozovky a začínají se tvořit kongesce. Pro různá nastavení se tato hodnota liší. V grafu 7.11 jsou výsledky simulace s parametry nalezenými v testu 7, v grafu 7.12 z testu 8 a v grafu 7.13 s nastavením odpovídajícím původnímu Nagel-Schreckenberg modelu. Data byla získána simulací na uzavřeném úseku silnice s použitím periodických okrajových podmínek. Délka vozovky byla nastavena na 10 km a pro generování vozidel byl použit generátor vytvářející vozidla v časových intervalech s normálním Gaussovo rozložením. Postupem času došlo k úplnému zaplnění vozovky, data tak obsahují hodnotu toku při různé hustotě provozu.

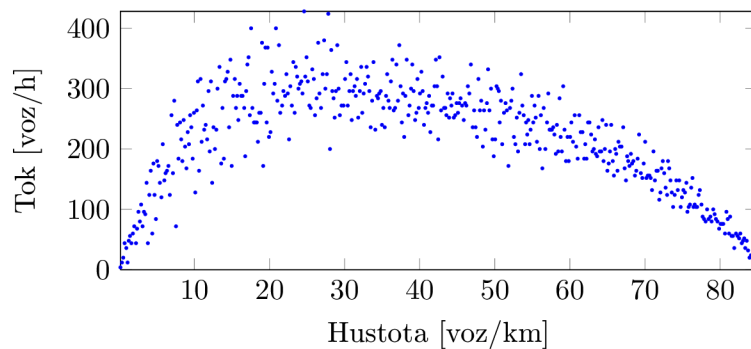


Obrázek 7.10: Průběh jedné simulace s parametry nalezenými evolucí v testu 8.

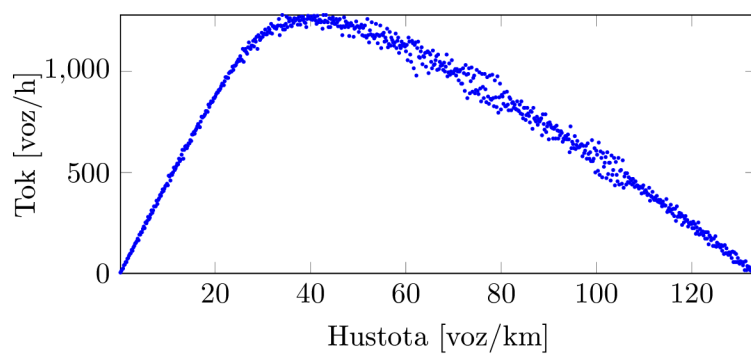


Obrázek 7.11: Zobrazení závislosti dopravního toku na hustotě dopravy při použití nastavení nalezeného v testu 7.





Obrázek 7.12: Zobrazení závislosti dopravního toku na hustotě dopravy při použití nastavení nalezeného v testu 8.



Obrázek 7.13: Zobrazení závislosti dopravního toku na hustotě dopravy s parametry odpovídající standardnímu modelu.

# Kapitola 8

## Závěr

V průběhu řešení této práce jsem se seznámil s problematikou mikroskopické simulace dopravy. Kapitola 2 se zabývá možnostmi popisu dopravy a používanými teoretickými modely dopravy. Mikroskopické simulátory na počítači mohou být implementovány například pomocí celulárních automatů. Kapitola 3 popisuje principy celulárních automatů a dále se zabývá jejich použitím v modelování dopravy. Pro kalibraci modelu je možné použít evoluční algoritmy, které dokáží efektivně prohledávat stavový prostor řešení. Základní principy evolučních algoritmů jsou popsány v kapitole 4, která se detailněji věnuje genetickým algoritmům. V rámci praktické části byl implementován simulátor dopravy postaven na celulárním automatu. V kapitole 5 je uveden návrh simulátoru, který byl vytvořen. V kapitole 6 jsou popsány implementované rozšíření základního modelu. Implementovaný model byl kalibrován za pomoci reálných dat, dosažené výsledky jsou popsány v kapitole 7.

Byl vytvořen komplexní nástroj pro simulaci dopravy založený na celulárním automatu. Tento nástroj má bohaté možnosti nastavení a je vhodný pro experimentování s reálnými daty. V provedených experimentech se výsledné dojezdové doby vozidel při simulaci lišily od reálných dat většinou o více než 20%. To je zřejmě způsobeno nedostatečným modelováním skutečnosti. V čem se skutečná trasa liší od vytvořeného modelu, je popsáno v části 7.1. Nejvíce je model znevýhodněn zřejmě tím, že je na trase 12 semaforů, které nejsou modelem brány v potaz. Model Nagel-Schreckenberg byl původně použit pro simulaci dopravy na dálnici, tomu také odpovídají původní parametry. Změnou nastavení parametrů na základě analyzovaných dat bylo však dosaženo toho, že model více odpovídal realitě. Doba dojezdu výrazně vzrostla v období dopravní špičky a profil se podobal skutečným časům. Pomocí evoluce bylo pak nalezeno podobné nastavení. Nepodařilo se však dostat pod průměrnou chybu 20%. V dojezdových časech některých vozidel v datech jsou příliš velké odchylky, jejichž původ není přesně znám. Simulátor dokáže napodobit reálné chování vozidel, ale pro použitá data nemodeluje úsek zřejmě dostatečně detailně. Pro ověření správnosti modelu by měly být provedeny experimenty i na jiných datech.

Simulátor by bylo možné vylepšit o další rozšíření modelu. Např. možnost změny simulačního kroku, který je v simulátoru pevně nastaven na 1 sekundu. Pravidla pro výpočet nové rychlosti by mohla být rozšířena o rozhodování na základě akcelerace předchozího vozidla. Simulátor by mohl být také rozšířen o možnost modelování více pruhů vozovky, předjíždění vozidel, semaforů, přechodů pro chodce, nebo křižovatky. Možností je spousta a zvolená úroveň záleží vždy na účelu použití simulátoru. Simulátor, který byl vytvořen v rámci této práce, je určen pro experimentování s nastavením parametrů modelu a jeho různými rozšířeními. K tomuto účelu jsou jeho možnosti dostačující.



# Literatura

- [1] Simulation Modelling Applied to Road Transport European Scheme Tests (SMARTTEST) - Review of MicroSimulation Models. 1998.  
URL <http://www.its.leeds.ac.uk/projects/smartest/Deliv3.PDF>
- [2] *Highway Capacity Manual 2000*. Transportation Research Board, National Research Council, 2000, ISBN 0-309-06681-6.
- [3] Impact Assessment accompanying the White Paper - Roadmap to a Single European Transport Area - Towards a competitive and resource efficient transport system. 2011.  
URL [http://ec.europa.eu/transport/themes/strategies/doc/2011\\_white\\_paper/white\\_paper\\_2011\\_ia\\_full\\_en.pdf](http://ec.europa.eu/transport/themes/strategies/doc/2011_white_paper/white_paper_2011_ia_full_en.pdf)
- [4] MEMO Doprava 2050: hlavní úkoly a klíčová opatření. 2011.  
URL [http://europa.eu/rapid/press-release\\_MEMO-11-197\\_cs.htm](http://europa.eu/rapid/press-release_MEMO-11-197_cs.htm)
- [5] International Organization of Motor Vehicle Manufacturers Production Statistics. 2012.  
URL <http://oica.net/category/production-statistics>
- [6] Travel in London, Key trends and developments Report number 5. 2012.  
URL <http://www.tfl.gov.uk/assets/downloads/corporate/travel-in-london-report-5.pdf>
- [7] Barceló, J.: *Fundamentals of Traffic Simulation*. International series in operations research & management science, Springer New York, 2010, ISBN 9781441961426.
- [8] Barlovic, R.: Metastable states in cellular automata for traffic flow. 1998.
- [9] Bham, G.; Benekohal, R.: A high fidelity traffic simulation model based on cellular automata and car-following concepts. *Transportation Research Part C: Emerging Technologies*, ročník 12, č. 1, 2004: s. 1 – 32, ISSN 0968-090X, doi:10.1016/j.trc.2002.05.001.  
URL [http://transportation.mst.edu/media/research/transportation/documents/Bham\\_2004\\_Transportation-Research-Part-C-Emerging-Technologies.pdf](http://transportation.mst.edu/media/research/transportation/documents/Bham_2004_Transportation-Research-Part-C-Emerging-Technologies.pdf)
- [10] Clarridge, A.; Salomaa, K.: Analysis of a cellular automaton model for car traffic with a slow-to-stop rule. *Theoretical Computer Science*, ročník 411, č. 38-39, 2010: s. 3507–3515.  
URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.4979&rep=rep1&type=pdf>

- [11] Gagné, C.; Parizeau, M.: Genericity in Evolutionary Computation Software Tools: Principles and Case Study. *International Journal on Artificial Intelligence Tools*, ročník 15, č. 2, April 2006: s. 173–194.
- [12] Korček, P.; Sekanina, L.; Fučík, O.: Cellular automata based traffic simulation accelerated on GPU. In *Proceedings of the 17th International Conference on Soft Computing (MENDEL2011)*, Institute of Automation and Computer Science FME BUT, 2011, ISBN 978-80-214-4302-0, s. 395–402.  
URL [http://www.fit.vutbr.cz/research/view\\_pub.php?id=9555](http://www.fit.vutbr.cz/research/view_pub.php?id=9555)
- [13] Krauss, S.; Wagner, P.; Gawron, C.: Metastable states in a microscopic model of traffic flow. *Phys. Rev. E*, ročník 55, May 1997: s. 5597–5602, doi:10.1103/PhysRevE.55.5597.
- [14] Maerivoet, S.; De Moor, B.: Traffic Flow Theory. *ArXiv Physics e-prints*, Červenec 2005.  
URL <http://arxiv.org/abs/physics/0507126>
- [15] McGehee, D. V.; Mazzae, E. N.; Baldwin, S.: Driver Reaction Time in Crash Avoidance Research: Validation of a Driving Simulator Study on a Test Track. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, ročník 44, č. 20, 2000: s. 3–320–3–323, doi:10.1177/154193120004402026,  
<http://pro.sagepub.com/content/44/20/3-320.full.pdf+html>.  
URL <http://pro.sagepub.com/content/44/20/3-320.abstract>
- [16] Nagel, K.: Particle hopping models and traffic flow theory. *Physical Review E*, ročník 53, č. 5, 1996: str. 4655.
- [17] Nagel, K.; Schreckenberg, M.: A cellular automaton model for freeway traffic. *Journal de Physique I*, ročník 2, č. 12, Prosinec 1992: s. 2221–2229, ISSN 1155-4304, doi:10.1051/jp1:1992277.  
URL [http://www.uni-due.de/imperia/md/content/ptt/paper/1992\\_origca.pdf](http://www.uni-due.de/imperia/md/content/ptt/paper/1992_origca.pdf)
- [18] Nagel, K.; Wolf, D.; Wagner, P.; aj.: Two-lane traffic rules for cellular automata: A systematic approach. *Phys. Rev. E*, ročník 58, Aug 1998: s. 1425–1437, doi:10.1103/PhysRevE.58.1425.  
URL <http://arxiv.org/pdf/cond-mat/9712196.pdf>
- [19] Peringer, P.: Studijní opora k předmětu Modelování a simulace. 2012: s. 88 – 91.
- [20] Rickert, M.; Nagel, K.; Schreckenberg, M.; aj.: Two lane traffic simulations using cellular automata. *Physica A: Statistical Mechanics and its Applications*, ročník 231, č. 4, 1996: s. 534 – 550, ISSN 0378-4371.
- [21] Schreckenberg, M.; Schadschneider, A.; Nagel, K.; aj.: Discrete stochastic models for traffic flow. *Phys. Rev. E*, ročník 51, Apr 1995: s. 2939–2949, doi:10.1103/PhysRevE.51.2939.  
URL <http://arxiv.org/pdf/cond-mat/9412045.pdf>
- [22] Schwarz, J.; Sekanina, L.: Studijní opora k předmětu Aplikované evoluční algoritmy. 2006.

- [23] Spiegelman, C.; Park, E. S.; Rilett, L.: *Transportation Statistics and Microsimulation*. Chapman & Hall/CRC, 2010, ISBN 9781439800232.
- [24] Wagner, P.; Nagel, K.; Wolf, D. E.: Realistic Multi-Lane Traffic Rules for Cellular Automata. *PHYSICA A*, ročník 234, 1996: s. 96–2586.
- [25] Weber, D.; Chrobok, R.; Hafstein, S.; aj.: OLSIM: Inter-urban Traffic Information. In *Innovative Internet Community Systems, Lecture Notes in Computer Science*, ročník 3473, Springer Berlin Heidelberg, 2006, ISBN 978-3-540-28880-0, s. 296–306, doi:10.1007/11553762\_27.  
URL [http://dx.doi.org/10.1007/11553762\\_27](http://dx.doi.org/10.1007/11553762_27)
- [26] Weise, T.: *Global Optimization Algorithms – Theory and Application*. it-weise.de (self-published): Germany, 2009.  
URL <http://www.it-weise.de/projects/book.pdf>
- [27] Wolfram, S.: Cellular automata. *Los Alamos Science*, ročník 9, 1983: s. 2–27.  
URL <http://www.stephenwolfram.com/pdf/Cellular-Automata-Stephen-Wolfram-Article.pdf>
- [28] Wolfram, S.: *A new kind of science*. General science, Wolfram Media, 2002.  
URL <http://www.wolframscience.com/nksonline/toc.html>

# Příloha A

## Obsah CD

Na přiloženém CD jsou zdrojové soubory práce ve formátu L<sup>A</sup>T<sub>E</sub>X včetně všech použitých obrázků a dat použitých k vytvoření grafů. Soubory se nachází ve složce **text**.

Ve složce **kod** jsou umístěny zdrojové soubory všech vytvořených programů. Konkrétně pak složka **cmake** obsahuje rozšíření pro program CMake, která jsou použita pro vyhledání knihoven Allegro, GALib a Google Test. Složka **configs** obsahuje ukázkové nastavení simulátoru (**default.config**) a evoluce (**default.ga.config**). Složka **data** obsahuje původní data z Prahy (**prague**), statistiky pro jednotlivé dny a průměrné údaje pro intervaly o délce 5, 10, 15, 30 a 60 minut (**prague\_summary**). Ve složce **doc** je umístěn návod k použití programu, popis konfiguračních souborů a dokumentace ke knihovně GALib, která obsahuje popis všech parametrů GA. Složka **results** obsahuje konfigurace použité při testování spolu s výsledky evoluce. Ve složce **src** jsou samotné zdrojové soubory vytvořených programů. Složka **tests** obsahuje jednoduché testy některých důležitých vlastností tříd *Config* a *Track*.

# Příloha B

## Návod k použití

Tato příloha obsahuje stručný návod k použití vytvořených programů. Pro vytvoření souboru Makefile je nutné mít nainstalován program `cmake`. Příkaz `cmake CMakeLists.txt` vyhledá potřebné knihovny a vytvoří soubor Makefile. Příkazem `make` je pak možné zkompileovat všechny programy v projektu. Spustitelné soubory se umístí do složky `bin`.

### B.1 Simulátor

Simulace se spouští příkazem

```
./bin/nasch [config_file] [samples_file]
```

Program přijímá dva parametry. Prvním parametrem je cesta ke konfiguračnímu souboru, jehož popis je v příloze C. Pokud je program spuštěn bez parametrů, hledá konfigurační soubor `configs/default.config`. Druhým možným parametrem je cesta k datovému souboru. Pokud je program spuštěn bez tohoto parametru, použije se cesta uvedená v konfiguračním souboru pod klíčem `samples_file`. Pokud není ani v konfiguračním souboru, použije se výchozí cesta `data/samples.csv`.

Po spuštění vypíše program použité nastavení a dále vypisuje statistické údaje nasbírané v průběhu simulace. Vypisuje průměrné hodnoty pro intervaly o nastavené délce `stats_frequency`. Pro každý interval následující hodnoty:

- *t<sub>from</sub>* a *t<sub>to</sub>* Časy od a do pro daný interval, v sekundách.
- *MAE* a *MAPE* Průměrná absolutní odchylka časů dojezdů v daném intervalu a průměrná procentuální chyba.
- *flow*, *density* Tok (voz/h) a hustota (voz/vozovce) vypočítány podle popisu v kapitole 2
- *cars* Počet aut, která v daném intervalu opustila vozovku.
- *speed* Průměrná rychlost vozidel (km/h) vypočítaná podle vztahu uvedeného v kapitole 2. Jde o podíl délky trasy k průměrné době průjezdu.
- *time* Průměrná doba průjezdu jednoho vozidla v sekundách. Za kolik sekund průměrně v daném intervalu dojele vozidlo ze začátku trasy na konec.

Poslední řádek obsahuje aritmetické průměry hodnot všech simulovaných intervalů.



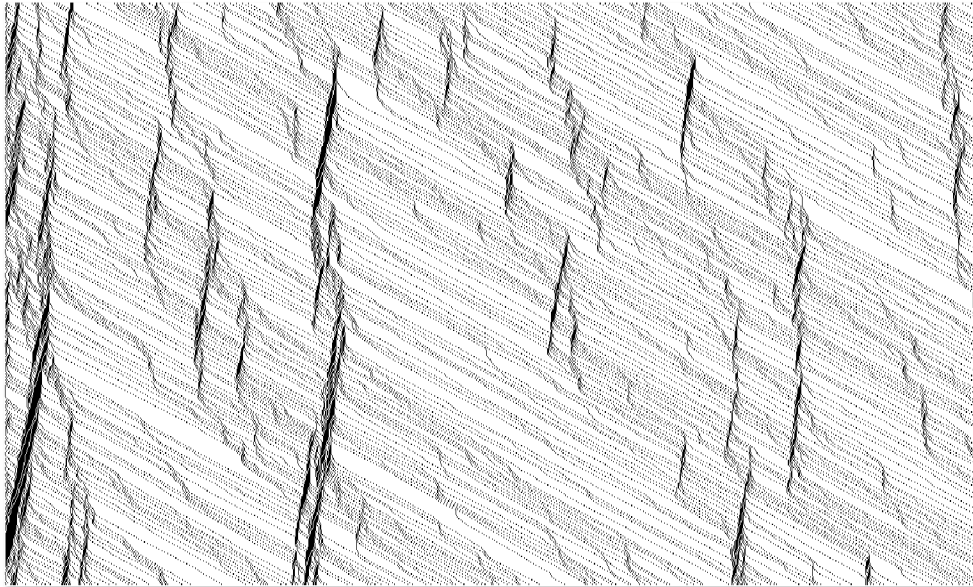
### B.1.1 Vizualizace

Jsou k dispozici dvě možnosti, jak vizualizovat průběh simulace. Program `./bin/ascii-nasch` vypisuje na výstup stav vozovky v textové podobě. Každá buňka je v tomto režimu reprezentována jedním znakem. Prázdné buňky jsou reprezentovány tečkami a obsazené buňky jsou reprezentovány číslem, které představuje rychlost vozidla, které se na buňce nachází. Výstup pak může vypadat např. takto:

```
| .1.2..3.....4....4.....1.00.01.0.1..001.1.....3..3..... |
| ..1..3..4.....3....3....000.0.01..1.00.1.1.....3..4..... |
| .1.1...4....4.....4....1..001.1.1.2..000..1.2.....4....4.. |
| ..1.1.....3....4.....1.1.01.1.0.2..0000...1..3.....4.... |
| .0.1.2.....3....4....1.01.0.01..00000...2....4..... |
| .1..1..3.....4.....2..01.00.0.2..00001.....2.....3..... |
| ..1..2...3.....3....00.001.0...00000.2.....3.....3..... |
| .0.2...1....4.....1.01.00.00...00000...3.....4.....4.. |
| .0..2..2.....4.....00.000.00...00000.....3.....3..... |
| .0....2..3.....4....01.001.00...00001.....3.....3... |
| .1.....3..3.....01.001.000...0000.1.....4.....4 |
| ..2.....3..4.....0.000.0000...0001..1.....4..... |
| .1..2.....4....4....0.000.0001...001.2..2.....4.. |
| ..0..2.....3....01.000.001.2..01.1..2..1..... |
| ..1.....3.....1.1.0000.00.1..01.1.2..1.2..... |
| .1.1.....4.....1.00001.01..1.0.1.2..2..2..... |
| ..1.2.....4.....00000.00.1..01..1..2..2..1..3..... |
| .1.1..2.....4....00000.01..0.0.1..1..1..1.2...3..... |
| ..1.0..2.....000001.0.1.1.0..1..2..1..2..2....4..... |
| .1.01....3.....00001.01..0.01...1..1..2..2..3.....4... |
| ..00.1.....3.....0000.00.1.1.1.1..2..2..3..3..4..... |
| ..01..2.....3.....0000.00..1.1.1.1...3..3...3..4....4... |
| ..0.2..3.....3...0001.00...1.0.0.2.....2..3...2...4....4 |
| ..0..3...3.....0000.001...00.1..2.....3...2..3.....4. |
| ..0.....4....4.....0000.00.2..01..2...3.....2..3...3..... |
| ..0.....3...3...0001.00...0.0.1..3....4.....3...4....4.. |
| ..0.....3...0001.001...0.0..2....3.....4....3....4... |
| ..1.....1.000.000.2..1.1...3.....3.....3...4..... |
```

Na každém řádku je zobrazen stav vozovky po tom, co jsou vypočítány nové rychlosti vozidel pro další krok, ale vozidla ještě nejsou posunuta na nové pozice.

Druhá možnost jak vizualizovat průběh simulace je pomocí grafické nadstavby využívající knihovny Allegro. Program `./bin/gui-nasch` přijímá stejné parametry jako základní varianta. Po spuštění otevře plátno na které vykresluje obsazené buňky jako černé pixely a prázdné buňky bílé. Pomocí tohoto programu je možné zobrazit o mnoho delší úsek cesty, než jaký je vhodné zobrazovat v textovém režimu. Na obr. B.1.1 je zobrazen 10km úsek vozovky. V tomto režimu je možné simulaci pozastavit stiskem klávesy *mezerník* a ukončit klávesou *escape*.



Obrázek B.1: Příklad výstupu grafické vizualizace.

## B.2 Genetický algoritmus

Evoluce je implementovaná s pomocí knihovny GALib. Spouští se příkazem

```
./bin/search_params [config_file] [galib_params...]
```

Pokud není zadaná cesta ke konfiguračnímu souboru, hledá se v `configs/default.ga_config`. Konfigurační soubor obsahuje nastavení genetického algoritmu i některé parametry simulace, více v [D](#). Parametry GA je možné nastavit přes parametry programu (více v dokumentaci knihovny GALib).

Po spuštění vypíše program použité nastavení a dále vypisuje výsledky každé evaluace kandidátního řešení ve formě tabulky. Po skončení evoluce jsou parametry simulátoru nastaveny podle nejlepšího nalezeného řešení a je spuštěna simulace na testovací sadě. Nejlepší nalezené řešení je otestováno na testovací sadě dat i v případě, že je evoluce manuálně ukončena dříve signálem SIGKILL.

## B.3 Nástroje pro analýzu dat z Prahy

Jedná se o malý program, jehož úkolem je analyzovat dodaná data. V módu *analyze* zjistí průměrné hodnoty dojezdové doby a rychlosti pro jednotlivé typy vozidel. V režimu *extract* zjistí údaje o dopravě pro intervaly nastavené délky. Na výstup vypíše počáteční čas intervalu, průměrnou dobu dojezdu v daném intervalu a počty vozidel, které přijela/odjela ze sledovaného úseku.

```
./bin/tools analyze [data_file]
./bin/tools extract [data_file] [interval_size]
```

## B.4 Automatizované testy

Příkazem `./bin/run_tests` se použije sada automatických testů pro třídu starající se o načítání konfigurace ze souboru nebo z binárního řetězce používaném jako genom v genetickém algoritmu.

## Příloha C

# Konfigurace simulace

Následuje stručný popis možných nastavení simulátoru. Všechny tyto možnosti je možné měnit v konfiguračním souboru bez nutnosti kompilace programu.

**car\_factory** Výběr použitého generátoru dat. Jsou dostupné tři možnosti. Jednoduchý deterministický generátor s nastavením pevného intervalu příjezdu vozidel. Generátor s normálním Gaussovo rozložením pravděpodobnosti, pro který je možné nastavit střední hodnotu a odchylku. Poslední generátor je určen pro načítání dat ze souboru csv ve formátu, v jakém byla použita data.

**max\_time** Maximální délka simulace v hodinách simulačního času. Vhodné nastavit při použití náhodných generátorů vozidel.

**periodic\_boundary** Zapnutí periodických hraničních podmínek.

**site\_length** Délka jedné buňky vozovky v metrech.

**track\_length** Délka celého úseku vozovky v metrech.

**default\_car** Výchozí typ vozidla, který se použije, pokud v datech není typ vozidla uveden.

**slowdown\_probability** Pravděpodobnost náhodného zpomalení vozidla v každém kroku.

**acceleration\_probability** Pravděpodobnost zrychlení vozidla v každém kroku.

**max\_speed** Maximální rychlost vozidla v km/h.

**min\_speed** Minimální rychlost vozidla v km/h, pod kterou vozidlo nezpomalí v důsledku náhodného zpomalení.

**length** Délka vozidla v metrech.

**table\_format** Zapnutí úsporného tabulkového výpisu.

**stats\_frequency** Délka časového úseku, po kterém se v průběhu simulace vyhodnocují data a vypisují se průměrné hodnoty.

**slow\_to\_stop** Aktivování Slow-to-Start a Slow-to-Stop pravidel.

**slow\_to\_start\_probability** Pravděpodobnost uplatnění Slow-to-Start pravidla.

**true\_slowdown** Aktivuje modifikaci kroku náhodného zpomalení, pokud ve stejném kroku došlo k akceleraci vozidla, je snížena rychlost o dvě jednotky.

## Příloha D

# Konfigurace evoluce

Nastavení evoluce je možné měnit bez nutnosti zásahu do zdrojových souborů a rekompilace. Parametry knihovny GALib, které není možné nastavit přes tento konfigurační soubor, je možné předat jako argumenty příkazové řádky při spuštění.

**objective\_value** Hodnota, kterou má genetický algoritmus brát jako hodnocení jedince. Dostupné možnosti jsou průměrná absolutní chyba v dojezdové době, procentuální odchylka dojezdové doby, kvadratický průměr odchylky, míra toku, průměrná rychlost, počet vozidel, průměrná dojezdová doba.

**minimize** Nastavení, zda se jedná o minimalizační nebo maximalizační úlohu.

**car\_types** Počet typů vozidel, která mají být obsažena v genomu.

**train\_file a test\_file** Trénovací a testovací soubor s daty.

**Nastavení simulace** Součástí nastavení jsou i parametry simulace, která je spouštěna za účelem získání ohodnocení kandidátních řešení. Jsou zde parametry `default_car`, `track_length`, `slow_to_stop`, `slow_to_start_probability`, `stats_frequency`, `car_factory`, `periodic_boundary`, `max_time` a `true_slowdown`.

**ga** Výběr genetického algoritmu, který má být použit pro evoluci. Dostupné možnosti s odpovídajícím nastavením: *GASimpleGA*, *GASteadyStateGA* (`replacement_percentage`), *GADemeGA* (`replacement_number`, `number_of_populations`, `migration_number`), *GA-IncrementalGA* (`number_of_offspring`).

**population\_size** Počet jedinců v jedné populaci.

**mutation\_probability** Pravděpodobnost mutace.

**crossover\_probability** Pravděpodobnost křížení.

**terminator** Podmínky ukončení evoluce, možnost nastavit ukončení po určitém kroku simulace. Nebo může být evoluce ukončena v případě, že se skóre v průběhu několika generací nemění nebo když už je diverzita populace velice nízká.

**scaling** Škálování ohodnocení jedince na fitness hodnotu. Při experimentech nebylo nutné, jako fitness hodnota bylo použito přímo hodnocení jedince. Možnosti škálování jsou vypnuto, lineární, Sigma Truncation, nebo Power Law.

## Příloha E

# Ukázka nastavení simulátoru

```
# Nastaveni simulace

#== generator vozidel =====

car_factory = 3
# Dostupne generatory:
# 1 - jednoduchy deterministicky generator
scf_interval = 20
# 2 - normalni gaussovo rozlozeni
ncf_mean = 3
ncf_deviation = 2
# 3 - nacita casy vozidel z csv dat z Prahy

# soubor ze ktereho se maji nacistat data, pokud je aktivovan odpovidajici generator
samples_file = data/prague/evropska_all.csv

#== nastaveni simulace =====

# delka simulace v hodinach, pokud se nastavi 0, neni simulace
# omezena casem (pri pouziti csv je ukoncena po projeti vseh aut)
max_time = 0 [h]

#== parametry vozovky =====

# moznost zapnout periodicke hranicni podminky
periodic_boundary = 0

# site_length - delka jedne bunky [m]
site_length = 0.3

# track_length - delka trasy [m]
track_length = 5350

#== parametry vozidel =====
```

```

# default_car - vychozi typ vozidla, pokud neni znam typ
default_car = 0

# slowdown_probability - pravdepodobnost zpomalení
# acceleration_probability - pravdepodobnost zrychlení
# max_speed - maximalni rychlost [km/h]
# min_speed - hranice minimalni rychlosti [km/h]
# length - delka vozidla v metrech [m]

# Typ vozidel v datech z ulice Evropska, Praha
# vozidlo 0 bude predstavovat vozidlo neznameho typu
car 0
  slowdown_probability = 0.1
  acceleration_probability = 0.9
  max_speed = 50 [km/h]
  min_speed = 0 [km/h]
  length = 10 [m]
endcar

# 1 - osobni automobil do 3.5 tuny
car 1
  slowdown_probability = 0.2
  acceleration_probability = 0.7
  max_speed = 50 [km/h]
  min_speed = 0 [km/h]
  length = 10 [m]
endcar

# 2 - dodavka do 3.5 tuny
car 2
  slowdown_probability = 0.3
  acceleration_probability = 0.8
  max_speed = 50 [km/h]
  min_speed = 0 [km/h]
  length = 14 [m]
endcar

# 3 - autobus / karosa 11.5m
car 3
  slowdown_probability = 0.35
  acceleration_probability = 0.8
  max_speed = 40 [km/h]
  min_speed = 0 [km/h]
  length = 15 [m]
endcar

# 4 - nakladni vozidlo do 6 tun
car 4

```

```

slowdown_probability = 0.3
acceleration_probability = 0.8
max_speed = 40 [km/h]
min_speed = 0 [km/h]
length = 15 [m]
endcar

# 5 - nakladni vozidlo nad 6 tun
car 5
slowdown_probability = 0.4
acceleration_probability = 0.8
max_speed = 40 [km/h]
min_speed = 0 [km/h]
length = 15 [m]
endcar

#== statistiky =====

# table_format - 1/0 zapne/vypne tabulkovy vypis statistik, ktery je vhodny
# pro dalsi zpracovani, napr. vykresleni grafu
table_format = 1

# stats_frequency - delka casoveho intervalu po kterem se maji vypocitat
# a vytisknout statisticke udaje, v minutach
stats_frequency = 15 [min]

#== rozsireni modelu =====

# aktivuje rozsireni modelu o Slow-to-Start a Slow-to-Stop pravidla
slow_to_stop = 0

# pokud je aktivovan rozsireny rezim, je treba nastavit pravdepodobnost
# s jakou se maji aplikovat slow to start pravidla (p_slow)
slow_to_start_probability = 0.5

true_slowdown = 1

```