

Modeliranje agentim

Grahović, Saša

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Civil Engineering / Sveučilište u Rijeci, Građevinski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:157:666433>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-03-28**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Civil Engineering - FCERI Repository](#)



image not found or type unknown

**SVEUČILIŠTE U RIJECI
GRAĐEVINSKI FAKULTET**

Saša Grahović

Modeliranje agentima: jednostavno čekanje na semaforu

Diplomski rad

Rijeka, 2020.

**SVEUČILIŠTE U RIJECI
GRAĐEVINSKI FAKULTET**

**Sveučilišni diplomski studij Građevinarstvo
Geotehnika
Računalno modeliranje**

**Saša Grahović
JMBAG: 0195029100**

Modeliranje agentima: jednostavno čekanje na semaforu

Diplomski rad

Rijeka, studeni 2020.

Naziv studija: **Sveučilišni diplomski studij Građevinarstvo**

Znanstveno područje: Tehničke znanosti

Znanstveno polje: Građevinarstvo, Interdisciplinarne tehničke znanosti

Znanstvena grana: Inženjerstvo okoliša

Tema diplomskog rada

**MODERIRANJE AGENTIMA: JEDNOSTAVNO ČEKANJE NA SEMAFORU
AGENT BASED MODELING: SIMPLE TRAFFIC-LIGHT PROBLEM**

Kandidat: **SAŠA GRAHOVIĆ**

Kolegij: **RAČUNALNO MODELIRANJE**

Diplomski rad broj: **G-2020-53**

Zadatak:

Potrebno je načiniti računalni model baziran na "agentima" za modeliranje prolaska vozila kroz semafor. Model se može načiniti u slobodno dostupnom programu za modeliranje agentima NetLogo. Rad treba sačinjavati uvodni dio o kompleksnosti, pojmu "emergencije" i modeliranju agentima. Potrebno je definirati konkretni model s cestom, semaforima i agentima ("turtles"), prikazati dimenzionalnu analizu, razraditi učinkovitost veličina pojedinih parametara na vrijeme čekanja vozila i naći optimalne veličine relevantnih parametara.

Tema rada je uručena: 25. veljače 2020.

Komentorica:

doc. dr. sc. Neira Torić Malić,
dipl. ing. građ.

Mentor:

prof. dr. sc. Ivica Kožar,
dipl. ing. građ.

IZJAVA

Diplomski rad sam izradio samostalno, u suradnji s mentorom i komentoricom i uz poštivanje pozitivnih građevinskih propisa i znanstvenih dostignuća iz područja građevinarstva. Građevinski fakultet u Rijeci je nositelj prava intelektualnog vlasništva u odnosu na ovaj rad.

Saša Grahović

U Rijeci, 30. listopada 2020.

Sadržaj

1.	UVOD	1
1.1.	Kompleksni sistemi i pojave	1
1.1.1.	<i>Pojave novih feneomena</i>	1
1.1.2.	<i>Promet vozila kao kompleksan sustav</i>	2
2.	MODELIRANJE POMOĆU AGENATA	3
2.1.	NetLogo	4
2.1.1.	<i>Sučelje</i>	5
2.1.2.	<i>Programski kod</i>	5
2.1.3.	<i>Simulacijsko okruženje</i>	5
3.	NETLOGO MODEL ZASTOJA U PROMETU	7
3.1.	Model	7
3.1.1.	<i>Globalne varijable</i>	7
3.1.2.	<i>Postav sučelja</i>	8
3.1.3.	<i>Definiranje kretanja vozila u modelu</i>	9
3.1.4.	<i>Ponašanje agenata</i>	12
3.1.5.	<i>Nesreće u prometu</i>	15
3.1.6.	<i>Semafori u modelu</i>	16
3.2.	Sučelje modela	16
4.	DIMENZIONALNA ANALIZA	18
4.1.	Brzina	18
5.	SIMULACIJA MODELA	23
5.1.	Simulacija 1	23
5.1.1.	<i>Simulacija 1.1</i>	24
5.1.2.	<i>Simulacija 1.2.</i>	27
5.1.3.	<i>Similacija 1.3.</i>	30
5.2.	Simulacija 2	33
5.2.1.	<i>Simulacija 2.1.</i>	33
5.2.2.	<i>Simulacija 2.2.</i>	36
5.2.3.	<i>Simulacija 2.3.</i>	38
5.3.	Simulacija 3	40
5.3.1.	<i>Simulacija 3.1.</i>	41

5.3.2.	<i>Simulacija 3.2.</i>	44
5.3.3.	<i>Simulacija 3.3.</i>	47
6.	ANALIZA REZULTATA I OPTIMALNIH PARAMETARA	50
7.	ZAKLJUČAK	53
8.	LITERATURA	54

Popis slika

Slika 1: Prikaz globalnih varijabli	7
Slika 2: Postav sučelja u NetLogo kodu	8
Slika 3: Postav semafora u modelu	8
Slika 4: Naredba za kretanje vozila u modelu	9
Slika 5: Naredba za kreiranje vozila u modelu	10
Slika 6: Definiranje kretanja vozila u modelu	11
Slika 7: Podešavanje brzine vozila u modelu	12
Slika 8: Varijable za kretanje vozila u modelu	14
Slika 9: Prepoznavanje prometnih nesreća u modelu	15
Slika 10: Naredbe za promjenu boje semafora u modelu	16
Slika 11: Sučelje modela	17

Popis tablica

Tablica 1: Prikaz parametara prve simulacije	24
Tablica 2: Prikaz rezultata simulacije 1.1	26
Tablica 3: Prikaz rezultata simulacije 1.2	28
Tablica 4: Prikaz rezultata simulacije 1.3	31
Tablica 5: Prikaz parametara simulacije 2	33
Tablica 6: Prikaz rezultata simulacije 2.1	34
Tablica 7: Prikaz rezultata simulacije 2.2	37
Tablica 8: Prikaz rezultata simulacije 2.3	39
Tablica 9: Prikaz parametara simulacije 3	41
Tablica 10: Prikaz rezultata simulacije 3.1	43
Tablica 11: Prikaz rezultata simulacije 3.2	46
Tablica 12: Rezultati simulacije 3.3.	48
Tablica 13: Prikaz najpovoljnijih rezultata	50
Tablica 14: Redni broj simulacija sa najpovoljnijim parametrima	51
Tablica 15: Vrijednosti parametara za najpovoljnije dobivene rezultate	51

Popis dijagrama

Dijagram 1: Simulacija br. 1	25
Dijagram 2: Broj generiranih vozila u vremenu	26
Dijagram 3: Normalizirana krivulja	27
Dijagram 4 Simulacija 1.2.	28
Dijagram 5: Broj generiranih automobila - 1.2.	29
Dijagram 6: Normalizirani prikaz čekanja vozila - 1.2.	29
Dijagram 7: Simulacija 1.3. - Broj vozila u čekanju	30
Dijagram 8: Broj generiranih vozila - 1.3.	31
Dijagram 9: Normalizirani prikaz čekanja vozila - 1.3.	32
Dijagram 10: Rezultati simulacije 2.1.	34
Dijagram 11: Generirani broj vozila - 2.1.	35
Dijagram 12: Normalizirani prikaz čekanja vozila - 2.1.	35
Dijagram 13: Simulacija 2.2.	36
Dijagram 14: Broj generiranih vozila - 2.2.	37
Dijagram 15: Normalizirani prikaz čekanja vozila - 2.2.	38
Dijagram 16: Simulacija 2.3.	39
Dijagram 17: Generirani broj vozila - 2.3.	40
Dijagram 18: Normalizirani prikaz čekanja - 2.3.	40
Dijagram 19: Simulacija 3.1.	42
Dijagram 20: Broj generiranih vozila - 3.1.	43
Dijagram 21: Normalizirani prikaz čekanja vozila - 3.1.	44
Dijagram: 22 Simulacija 3.2.	45
Dijagram 23: Broj generiranih vozila - 3.2.	46
Dijagram 24: Normalizirani prikaz čekanja vozila - 3.2.	47
Dijagram 25: Simulacija 3.3.	48
Dijagram 26: Broj generiranih vozila - 3.3.	49
Dijagram 27: Normalizirani prikaz čekanja vozila - 3.3.	49
Dijagram 28: Broj vozila u zastoju u ovisnosti o trajanju zelenog svjetla	51
Dijagram 29: Protočnost promatranog segmenta u ovisnosti o trajanju zelenog svjetla	52

1. UVOD

1.1. Kompleksni sistemi i pojave

Nauka o kompleksnim sustavima se počela ubrzano razvijati u 20. stoljeću paralelno sa ubrzanim razvojem znanosti i tehnologije. Kao i ostali sustavi očituju se kao međusobno vezani elementi koji djeluju u nekoj okolini i prisutni su kroz cijelu povijest čovječanstva, međutim tek u suvremenom dobu pojavila se značajnija potreba da izučavanjem ovakvih sistema. Kompleksnost u njihovom slučaju znači da odnosi između dijelova sistema nisu linearni i na razini sustava pokazuju značajke koje su drugačije od značajki njihovih sastavnih dijelova. Teorija kompleksnih sistema razvija načela i alate pomoću kojih se objašnjavaju kompleksni sistemi i definira ih kao sustave koji u sebi sadrže više individualnih elemenata koji vrše interakciju među sobom ali osobine produkata te interakcije nisu predvidljive iz samih elemenata. Modeli koji predviđaju rast populacije, na primjer, pretpostavljaju da su svi članovi jednaki bez da istraže međusobnu interakciju između pojedinaca. Usvajanjem novih znanja razvili su se sofisticirani alati koji omogućuju preciznije proučavanje fenomena koji imaju kompleksno ponašanje. Novi kompjuterski alati pružaju mogućnost modeliranja, simulacije i istraživanja kompleksnih sistema. Kroz interakcije višestrukih elemenata dolazi do pojave novih fenomena.

1.1.1. *Pojave novih feneomena*

Pojam pojave novih fenomena kao produkt kompleksnih sustava mogao bi se opisati kao nastajanje novih i koherentnih struktura, obrazaca i svojstava kroz interakcije višestruko raspoređenih elemenata [1]. Osobine novih struktura ne mogu se predvidjeti samo na temelju svojstava elemenata od kojih nastaju, nego proizlaze iz međusobnih interakcija elemenata. Bitna značajka ovakvih pojava su spontano nastali rezultati koji nastaju bez jasno definirane koordinacije, odnosno sustav se „samoorganizira“.

1.1.2. Promet vozila kao kompleksan sustav

Promet vozila je primjer široko rasprostranjenog kompleksnog sustava. Potreba za većom mobilnosti dovela je do problema zagušenja prometa u cijelom svijetu. Uvriježeno je mišljenje da zastoj u prometu nastaje radi prometne nestreće ili prisilnog usporavanja vozila na trasi međutim većina zastoja u prometu nastaje radi nasumičnog ulaženja vozila u određeni pravac (npr. autocestu) i kao rezultanta statističke distribucije vozila i njihovih brzina. Pojedinac vidi promet kao sustav sačinjen od velikog broja vozila ali iz ptičije perspektive zastoj u prometu se prikazuje kao jedan objekt koji se kreće u suprotnom smjeru od kretanja prometnog toka [1].

2. MODELIRANJE POMOĆU AGENATA

Modeliranje zasnovano na agentima je metoda modeliranja koja se u često koristi za proučavanje kompleksnih sistema i izradu računalnih modela simulacije ponašanja sustava i razvoj modela kompleksnih mreža. Model rađen pomoću agenata predstavlja skup pojedinačnih entiteta odnosno „agenata” i njihovo ponašanje. Unutar svakog modela postoje agenti modela koji imaju svoja „stanja” odnosno pravila ponašanja, pravila po kojima dva entiteta komuniciraju odnosno vrše interakciju između sebe, i okolina unutar koje se agenti nalaze. Agent nikada neće samostalno izvršiti neku radnju, već će uzeti u obzir druge agente, i na temelju njih može izvršiti ili ne izvršiti neku radnju. Općenito, kada gradimo model koji simulira određeni fenomen, moramo prvo identificirati činitelje odnosno „agente”. Zatim moramo razmotriti pravila i procese koji odeđuju na koji način se odvija interakcija između agenata.

Da bi to izveli, trebalo bi obratiti pozornost na sljedeće:

- Agenti imaju svoja unutarnja stanja (podatke, atribute). Ovakva unutarnja stanja u modelu mogu biti opisana pomoću diskretnih ili kontinuiranih varijabli
- Razina komunikacije među agentima mora biti određena npr;
 - globalna interakcija (svaki agent komunicira sa svim ostalim agentima)
 - lokalna interakcija (svaki agent komunicira samo sa “susjednim” agentima)
 - lokalna interakcija sa dopuštenom globalnom interakcijom do nekog stupnja
- Ponašanja agenata su određena pravilima koja mogu varirati od logike prvog reda do jako opširnih algoritama
- Vrijeme i prostor mogu biti diskretni ili kontinuirani; zbog toga postoje:
 - kontinuiran prostor i kontinuirano vrijeme,
 - kontinuiran prostor i diskretno vrijeme,
 - diskretni prostor i kontinuirano vrijeme,
 - diskretni prostor i diskretno vrijeme.
- Kod mnogih aplikacija rađenih na bazi agenata, u model se uvodi određeni prostorni krajolik koji ograničava određene potencijalne interakcije. U nekim slučajevim ovakav

prostor je prikazan kao pasivna platforma na kojoj agenti komuniciraju. U drugim slučajevima, sam krajolik može biti agent sa svojim stanjima i pravilima ponašanja.

- Tijekom vremena model evolvirao. Interakcije između agenata se događaju po određenim pravilima koja ne bi trebala pratiti određeni slijed jer agenti se ponašaju individualno i paralelno jedni sa drugima. Interakcije treba definirati na način da se događaju nasumično.

Modele rađene pomoću agenata lakše je za razumjeti od matematičkih modela zato što su modeli od agenata sastavljeni od individualnih objekata i jednostavnih pravila koja određuju ponašanje tog objekta. Takav način prikaza je mnogo prirodniji čovjeku za objašnjenje nekoga fenomena nego što je prikaz preko matematičkih simbola. Kada razmišljamo o pojedinačnom agentu, možemo ih bolje pojasniti tako što svoja vlastita iskustva poistovjetimo sa agentom. Također, agenti u modelu obično imaju i svoju grafičku komponentu tako da ih jasno možemo vidjeti na ekranu. Programiranje ABM modela se može raditi u svakome programskom jeziku, ali objektno orijentirano programiranje se pokazalo kao najpodobnija vrsta programskog jezika jer koncept „objekta“ u objektno orijentiranom programiranju je vrlo sličan „agentu“ u modeliranju pomoću agenata. Postoji veliki broj alata i platformi koji su besplatni za korištenje, a najpoznatiji su „Repast“, „Netlogo“, „SWARM“ itd...

2.1. NetLogo

Za potrebe ovog diplomskog rada koristit će se program NetLogo. NetLogo je besplatan softver otvorenog koda razvijen na Sveučilištu Northwestern u SAD-u i najrašireniji je alat za primjenu modeliranja na bazi agenata. Sadrži veliku bazu osnovnih primjera modela na bazi agenata koji simuliraju pojave u stvarnom svijetu te programski kodovi tih osnovnih modela uvelike pomažu shvatiti način na koji funkcionira. Osnova NetLoga je programski jezik Logo. NetLogo koristi arhitekturu temeljenu na događajima gdje agenti kroz naizgled jednostavno reaktivno ponašanje mogu prikazati složene obrasce ponašanja. Cijelo programsko okruženje NetLoga možemo podijeliti na tri osnovna dijela: korisničko sučelje, programski kod i simulacijsko okruženje.

2.1.1. Sučelje

Sučelje je osnovni dio NetLoga koji vizualno prikazuje model kojega korisnik razvija. Osim vizualnog okvira, sučelje sadrži i različite tipove kontrola za upravljanje modelom i podešavanje parametara. Parametre je moguće podešavati putem klizača, jednostavnim prekidačem ili ih je moguće ručno unijeti u program. Osim kontrola koje korisnik sam implementira u model, sučelje sadrži i predefiniranu varijablu „ticks“ koja prikazuje broj izvršenih naredbi tokom simulacije.

2.1.2. Programski kod

Programski kod NetLogo modela nalazi se u odvojenom prozoru. Sve procedure koje se izvršavaju kada pokrenemo model su definirane u programskome kodu. Preko komanda i izvještača daju se zapovijedi kojima se upravlja ponašanjem pokretnih kornjača. Komanda je direktna zapovijed agentu koju zatim agent izvrši. Izvještači su instrukcije za izračun vrijednosti koju zatim agent izvijesti onome od koga je dobio zapovijed. NetLogo u sebi sadrži određene ugrađene zapovijedi i izvještaje koji se zajednički nazivaju primitivi. Osim već ugrađenih primitiva korisnik sam može programirati i definirati vlastite komande i izvještaje. Zapovijedi i komande koje korisnik sam definira se zovu procedure. Svaka procedura ima svoje ime i započinje prefiksom „to“ ili „to-report“. Prilikom definiranja procedure, istu je potrebno i završiti. To se radi unošenjem ključne riječi „end“ koja označava kraj procedure.

2.1.3. Simulacijsko okruženje

NetLogo predstavlja simulirano okruženje nekog zadanog svijeta u kojem se nalaze agenti koji izvode različite aktivnosti ovisno o naredbama koje su im zadane kroz kod. Veličina simulacijskog okruženja nije unaprijed zadana već je moguće podešavati dimenzije svijeta te njegove karakteristike kao što su odredište, veličina svakog polja te da li agent može prolaziti kroz rubove svijeta ili ne. Agenti su osnovna sastavnica modela i prate instrukcije koje su im zadane. U NetLogu razlikujemo četiri vrste agenata: pokretne

kornjače („turtles“), polja po kojima se agenti kreću („patches“), veze („links“) i promatrač („observer“). „Pokretne kornjače“ su agenti koji se kreću po svijetu. „Svijet“ je zapravo grafički dvodimenzionalni prikaz kretanja i interakcije agenata koji je sačinjen od polja. Jedno polje predstavlja kvadratni komadić površine „svijeta“ po kojima se kreću „pokretne kornjače“. Svako polje ima svoje koordinate te se broj polja kao i apsolutne koordinate svijeta na kojemu radimo model može proizvoljno zadati. Veze su agenti koji povezuju dvije kornjače. Promatrač se ne nalazi u samom modelu već izvan njega. Možemo ga se zamisliti kao promatrača koji ima mogućnost upravljanja nad drugim agentima. Varijable su mjesta na kojima se pohranjuje vrijednost u agentima. Varijable koje agent ima mogu biti globalne ili lokalne (pridružene samo jednom agentu).

3. NETLOGO MODEL ZASTOJA U PROMETU

U nastavku će biti prikazan NetLogo model koji simulira problem u prometu koji se javlja prilikom privremene regulacije prometa na nekoj prometnici gdje se zbog radova na cesti postavljaju semafori te vozila naizmjenično kreću. Cilj je, za zadane uvjete, odrediti koji bi bili optimalni parametri za koje će zastoje u prometu biti najmanji. Parametri koje je potrebno odrediti su trajanje crvenog i zelenog svjetla na semaforu i maksimalna brzina kojom se automobil smije kretati.

3.1. Model

Model je rađen kao modifikacija postojećeg NetLogo modela iz repozitorija pod imenom "Traffic Intersection".

3.1.1. Globalne varijable

```
globals [
  ticks-at-last-change
]

breed [ lights light ]

breed [ accidents accident ]
accidents-own [
  clear-in
]

breed [ cars car ]
cars-own [
  speed
]
```

Slika 1: Prikaz globalnih varijabli

Programski kod započinje definiranjem varijabli koje će se koristiti u modelu. Definirane su varijable „lights“ kojima su opisani semafori, „accidents“ koja će služiti za prepoznavanje nesreća koje se mogu dogoditi i automobili („cars“). Varijabli „cars“ je pridružena varijabla brzine („speed“). Definiranjem varijable automobila kao zasebne varijable i

pridodavanjem svojstva brzine na početku modela, omogućeno je da svaki automobil u nastavku koda ima u sebi brzinu kojom se može kretati.

3.1.2. Postav sučelja

```
to setup
  clear-all
  setup-semafori
  set-default-shape lights "square"
  set-default-shape accidents "fire"
  set-default-shape cars "car"
  ask patches [
    ifelse abs pycor <= 1
      [ set pcolor black ]
      [ set pcolor green - 1 ]
  ]

  reset-ticks
end
```

Slika 2: Postav sučelja u NetLogo kodu

Za inicijalni postav modela definirana je naredba „setup“. „Setup“ naredbom pridodali smo značenje gumbu kojega je potrebno uvesti na sučelje modela. Unutar naredbe je zadan je uvijet da polja sa apsolutnim koordinatama koje su manja ili jednaka 1, promijene boju u crno; u suprotnom postavi boju polja u zeleno. Dakle cesta na grafičkom viziru je crne boje i nalazi se na poljima koja imaju koordinate $y = 1$ i $y = -1$. Sva ostala polja na grafičkom viziru su zelene boje. Unutar naredbe još su definirani oblici koje će ranije definirani agenti imati. Semafori će imati oblik kvadrata, prometne nesreće (kolizije automobila) će biti prikazane oblikom vatre, a agenti koji predstavljaju automobile imati će oblik automobila koji postoji unutar NetLogo programa. Pokretanjem naredbe također će se postaviti i semafori na svoje mjesto unutar NetLogo grafičkog vizira.

```
to setup-semafori

  ask patch (- razmak2) -1 [ sprout-lights 1 [ set color green ] ]
  ask patch razmak2 1 [ sprout-lights 1 [ set color red ] ]
end
```

Slika 3: Postav semafora u modelu

Naredbe za postavljanje semafora u prostor je odrađeno idućom procedurom: „setup-semafori“. Unutar procedure dan je jednostavan uvjet da se na poljima sa koordinatama $y = 1$ i $y = -1$ postave semafori sa svojom bojom. Međusobna udaljenost semafora po x osi je definirana kao zasebna varijabla „razmak2“. Vrijednost varijable „razmak2“ nije zadana unutar NetLogo koda nego se zadaje na sučelju preko klizača. Na taj način omogućeno je lako podešavanje razmaka između dva semafora bez da se intervenira u programski kod. Način na koji semafori mijenjaju boju definiran je kasnije u kodu.

3.1.3. Definiranje kretanja vozila u modelu

```
to go
  ask cars [ move ]
  check-for-collisions
  make-new-car freq-west max-pxcor 1 270
  make-new-car freq-east min-pxcor -1 90
  if auto? and elapsed? green-length [
    change-to-yellow
  ]
  if any? lights with [ color = yellow ] and elapsed? yellow-length [
    change-to-red
  ]
  tick
end
```

Slika 4: Naredba za kretanje vozila u modelu

Nakon što je naredbom „setup“ i „setup semafori“ postavljen grafički vizir u NetLogu, potrebno je u njega smjestiti automobile i definirati kretanje automobila kao agenata u modelu. Naredbom „to go“ automobili dobivaju svoje mjesto u modelu. Automobili moraju biti smješteni na cestu, odnosno na polja unutar apsolutne koordinate $y \leq 1$. Model čine dvije prometne trake na koje su automobili smješteni na idući način: Za automobile koji se kreću iz smjera istoka prema zapadu – automobili se učitavaju:

- x koordinata na kojoj se učita vozilo: najveća x koordinata u modelu („max-pxcor“; moguće je mijenjati po volji u postavkama modela)
- y koordinata na kojoj se učita vozilo: $y = 1$ (sva polja sa koordinatom $y = 1$ predstavljaju lijevi trak ceste)

- smjer kretanja automobila: 270 – automobil će biti usmjeren po kutem od 270° u odnosu na sjever

Vozila koja se kreću iz smjera zapada prema istoku učitavaju se na slijedeći način:

- x koordinata na kojoj se učita vozilo: najmanja x koordinata u modelu („min-pxcor“)
- y koordinata na kojoj se učita vozilo: $y = -1$ (sva polja sa koordnatom $y = -1$ predstavljaju desni trak ceste)
- smjer kretanja vozila: 90 – vozilo će biti usmjereno pod kutem od 90° u odnosu na sjever

Procedurom „to-go“ definirali smo što će se dogoditi kada stisnemo gumb „go“ na sučelju, međutim procedura koristi neke varijable i naredbe koje do sada nisu definirane. U nastavku je potrebno pridodati začenje tim novim varijablama bez kojih programski kod neće izvršiti svoju zadaću. Procedurom „to-go“ postavili smo automobile na odgovarajući prometni trak i taj proces je nazvan “make-new-car freq”. Za sada programski kod ne prepoznaje tu naredbu pa je potrebno i nju definirati.

```

to make-new-car [ freq x y h ]
  if (random-float 100 < freq) and not any? turtles-on patch x y [
    create-cars 1 [
      setxy x y
      set heading h
      set color one-of base-colors
      adjust-speed
    ]
  ]
end

```

Slika 5: Naredba za kreiranje vozila u modelu

Funkcija „freq“ u sebi sadrži vrijednosti x, y i h kojima je u prošlom koraku određen položaj automobila. Brzina generiranja novih automobila ovisi o varijablama „freq-east“ i „freq-west“ na sučelju. Program će generirati novi automobil u vremenskom intervalu danom kao postotak otkucaja. Prilikom pozivanja agenta automobila u svijet naredbom „not

any?“ ispitujemo da se na ciljanom polju već nalazi agent. Ako je odgovor na postavljeno pitanje „ne“; na koordinate ciljanog polja će se postaviti jedan automobil pomoću naredbe „create-cars“. Naredba „create-cars“ uvijek postavlja samo jedan automobil na provjerenu koordinatu. Agent koji se stvori na svijetu odmah ima svoju brzinu koju prilagođava okolnostima na poljima ispred sebe.

```
to move
  adjust-speed
  repeat speed [
    fd 0.25
    if not can-move? 0.25 [ die ]
    if any? accidents-here [
      ask accidents-here [ set clear-in 5 ]
      die
    ]
  ]
end
```

Slika 6: Definiranje kretanja vozila u modelu

Funkcija „move“ je ranije dodana proceduri „to-go“ (Slika 4) kao varijabla o kojoj ovisi kretanje automobila. Ona u kodu do sada nije definirana te ju je potrebno odrediti. Za potrebe modela, dovoljno je da se automobili kreću samo u svojoj traci te ih nije potrebno preusmjeravati. Naime skretanje automobila na suprotni trak vrlo je malog trajanja u odnosu na ukupno trajanje prolaska kroz zonu radova te kao takvo neće mnogo utjecati na točnost modela. Dakle agenti se kreću samo pravocrtno i to za udaljenost koja ovisi o trenutnoj brzini koju agent ima. Ako agent ima brzinu koja iznosi 1 on će se pokrenuti za 0.25 jedinica u naprijed za vrijeme jednog otkucaja. Također ako agent prepozna da se na udaljenosti od 0.25 jedinica ispred njega ne nalazi polje on će nestati iz modela. Ovo služi da agent kada dođe do zadnjeg polja na grafičkom viziru nestane sa modela kako model ne bi sadržavao previše agenata. Kretanje u modelu je, osim rubom modela, ograničeno i potencijalnim sudaranjem dvaju automobila. Kako bi se prepoznali trenutci kada agenti dodiruju jedan drugog, model prepoznaje prometne nesreće. Prometne nesreće ili „accidents“ je globalna varijabla koja je zadana na početku modela (Slika 1) što znači da se u svakom trenutku može prepoznati u cijelom modelu. Prilikom izvršenja naredbe „move“ uvjetovano je da ako se dogodi kontakt između dvaju automobila, oba dva automobila

nestanu sa modela unutar 5 vremenskih jedinica. Tijekom simulacije problema parametrima će se nastojati izbjeći prometne nesreće jer one nisu predmet istraživanja. Za potrebe ovog modela prepoznavanje prometnih nesreća služi samo kako bi ih izbjegli manipulacijom parametara.

3.1.4. Ponašanje agenata

Prethodni dio koda sadrži osnovne procedure i funkcije kojima su definirane neke osnovne radnje unutar modela. Naredbe koje su do sada zadane mogu se jasno vidjeti na grafičkom sučelju modela. One iscrtavaju cestu i okolinu, smiještaju agente u prostor, pokreću automobile i rade sve ono što je vidljivo tijekom simulacije. Međutim mnogo varijabli i naredbi koje su spomenute u ranijim procedurama do sada nisu definirane. Model još nema zadana sva stanja koje agenti mogu imati, niti interakciju među agentima. Nejasno je kako će svaki pojedinačni agent koji ima ulogu automobila odrediti i prilagoditi svoju brzinu tako da ne udari u agent ispred sebe. Nejasno je i kako će svaki agent prepoznati na kojem se polju nalazi idući agent kao i što se treba dogoditi da semafor promijeni boju. Sve ovo je definirano u nastavku koda.

```
to adjust-speed
  let min-speed max (list (speed - max-brake) 0)
  let max-speed min (list (speed + max-accel) speed-limit)

  let target-speed max-speed
  let blocked-patch next-blocked-patch
  if blocked-patch != nobody [
    let space-ahead (distance blocked-patch - 0.25)
    while [
      breaking-distance-at target-speed > space-ahead and
      target-speed > min-speed
    ] [
      set target-speed (target-speed - 1)
    ]
  ]

  set speed target-speed
end
```

Slika 7: Podešavanje brzine vozila u modelu

Osnovni atribut svakog agenta koji predstavlja automobil u modelu je njegova brzina. Parametar brzine vrlo je značajan u ponašanju modela jer treba spriječiti sudaranje agenata, kao i postaviti kretanje agenata da bude što sličnije kretanju vozila u stvarnosti. Naredba „adjust-speed“ čini to da kretanje agenata u modelu bude što sličnije kretanju vozila u prometu. Pomoću dvije privremene varijable „min-speed“ i „max-speed“ određujemo koje su najmanja i najveća brzina kojima se agent smije kretati u modelu, a da ne prouzroči sudar sa agentom ispred sebe. Agent će se uvijek nastojati kretati, a što manje stajati (kada agent stoji imati će brzinu jednaku 0) i njegova najmanja dopuštena brzina u svakom trenutku biti će najveća vrijednost od :

- a) trenutna brzina – brzina usporavanja
- b) 0

Najveća brzina biti će veća vrijednost od:

- a) trenutna brzina + ubrzanje
- b) ograničenje brzine

Ograničenje brzine postavlja se na sučelju preko klizača i prikazana je u jedinicama modela koji ovise o mjerilu modela. Slijedeći set naredbi određuje brzinu vozila u ovisnosti o udaljenosti vozila ispred promatranog agenta. Ako je ciljana brzina najveća brzina koju automobil postiže („max-speed“), uvodimo privremenu varijablu („blocked-patch“) odnosno polje koje je zauzeto. Varijaba „blocked-patch“ poprima vrijednost idućeg blokiranog polje koje se nalazi ispred automobila („next-blocked-patch“). Program tada provjerava udaljenost idućeg zauzetog polja. Kada program odredi iduće zauzeto polje tada se uvodi varijabla „space-ahead“ u koju se sprema vrijednost udaljenosti do idućeg zauzetog polje umanjena za vrijednost jednog koraka agenta (0.25) . Varijabla „space-ahead“ predstavlja prostor po kojem se automobil može kretati prije nego što mora stati da bi izbjegao sudar sa vozilom ispred sebe.

```

to-report breaking-distance-at [ speed-at-this-tick ]
  let min-speed-at-next-tick max (list (speed-at-this-tick - max-brake) 0)
  report speed-at-this-tick + min-speed-at-next-tick
end

to-report next-blocked-patch
let patch-to-check patch-here
while [ patch-to-check != nobody and not is-blocked? patch-to-check ] [
  set patch-to-check patch-ahead ((distance patch-to-check) + 1)
]
report patch-to-check
end

to-report is-blocked? [ target-patch ]
report
  any? other cars-on target-patch or
  any? accidents-on target-patch or
  any? (lights-on target-patch) with [ color = red ] or
  (any? (lights-on target-patch) with [ color = yellow ] and
  target-patch != patch-here)
end

```

Slika 8: Varijable za kretanje vozila u modelu

Varijable „breaking-distance-at“, „next-blocked-patch“ i „is-blocked“ su izvještaji koje program prepoznaje i izvještava agenta o vrijednosti koju izračuna. Program izvještava vrijednost varijable „is-blocked“ tako što provjerava dali postoji smetnja na promatranom polju. Program prelazi po svakom polju i provjerava slijedeće:

- dali na promatranom polju već postoji automobil
- dali na promatranom polju postoji prometna nesreća
- dali se na promatranom polju nalazi semafor na kojem je crveno svjetlo
- dali se na promatranom polju nalazi semafor na kojem je žuto svjetlo

Ako program utvrdi da se na polju nalazi neka od navedenih smetnji to polje označava kao blokirano polje i u modelu je sada to polje opisano kao zauzeto. Nakon što je prepoznato zauzeto polje program treba prepoznati gdje se ono nalazi u odnosu na agenta. Varijablom „next-blocked-patch“ program prelazi preko polja ispred promatranog agenta i ispituje koje je iduće polje zauzeto. Ako program ne naiđe na zauzeto polje, proverava iduće polje na poziciji + 1 od polja koje je zadnje pregledao. Nakon što je pronašao zauzeto polje,

izvještava poziciju polja koje je zauzeto. Posljednja informacija koju agent mora dobiti je ona o brzini zaustavljanja. Izvještaj „breaking-distance-at“ provjerava brzinu automobila u promatranom trenutku i uvodi privremenu varijablu koja sadrži minimalnu brzinu automobila u sljedećem otkucaju („min-speed-at-next-tick“). Privremena varijabla minimalne brzine u sljedećem otkucaju će biti najveća vrijednost od:

- trenutna brzina – deceleracija
- 0

Ako je trenutna brzina manja od predviđene deceleracije vozila tada će vrijednost brzine biti manja od nule. U tome slučaju vozilo će poprimit brzinu jednaku nuli. Informacija koju program izvjesti je koliku udaljenost će agent prijeći ako započne kočenje u idućem otkucaju s obzirom na trenutnu brzinu koju ima.

3.1.5. Nesreće u prometu

Do sada agenti već mogu prepoznati ostale agente ispred sebe, međutim još ne mogu prepoznati prometne nesreće koje su se dogodile.

```
to check-for-collisions
  ask accidents [
    set clear-in clear-in - 1
    if clear-in = 0 [ die ]
  ]
  ask patches with [ count cars-here > 1 ] [
    sprout-accidents 1 [
      set size 1.5
      set color yellow
      set clear-in 5
    ]
    ask cars-here [ die ]
  ]
end
```

Slika 9: Prepoznavanje prometnih nesreća u modelu

Na Slika 9 prikazane su naredbe za proceduru koja se vrši kada se dogodi prometna nesreća. Kontakt dvaju automobila će program prepoznati tako što će se na istome polju nalaziti više od jednog automobila. Na grafičkom viziru prometne nestreće će biti

označene posebnom ikonom i trajati će 5 vremenskih jedinica. Ako nadolazeći automobil stigne do polja na kojem se dogodila nesreća za vrijeme trajanje nesreće od 5 vremenskih jedinica, on će se isto smatrati sudionikom te će nestati iz modela. Svaki automobil prepoznaje da je na poziciji nesreće pomoću naredbe „check-for-collision“.

3.1.6. Semafori u modelu

```
to change-to-yellow
  ask lights with [ color = green ] [
    set color yellow
    set ticks-at-last-change ticks
  ]
end

to change-to-red
  ask lights with [ color = yellow ] [
    set color red
    ask other lights [ set color green ]
    set ticks-at-last-change ticks
  ]
end

to-report elapsed? [ time-length ]
  report (ticks - ticks-at-last-change) > time-length
end
```

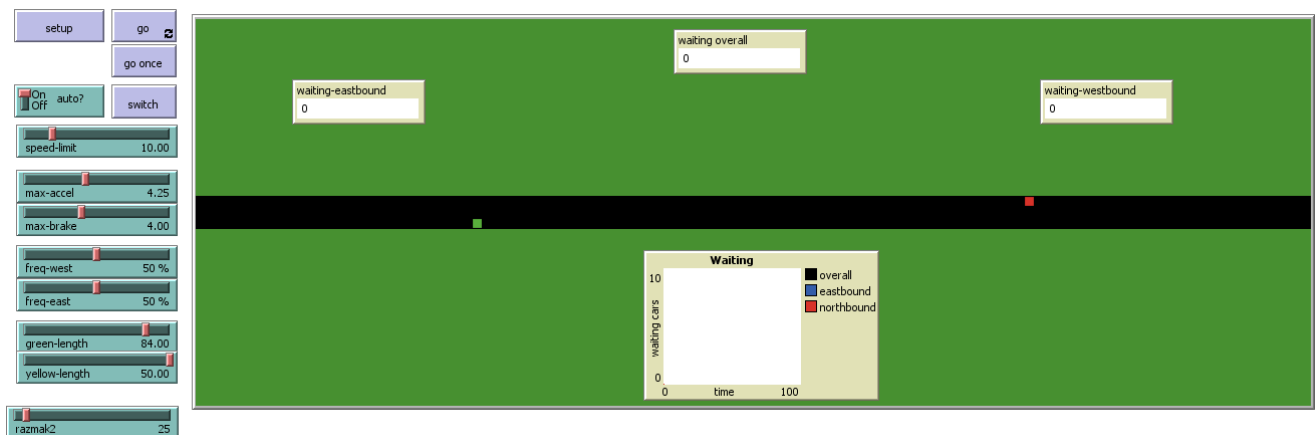
Slika 10: Naredbe za promjenu boje semafora u modelu

Promjena boje na semaforu je jednostavna naredba kojom se daje naredba agentima da promijene boju ukoliko je prošlo vrijeme trajanja svjetla na semaforu. Trajanje se mjeri u otkucajima i zadano je na sučelju preko klizača i mogu se jednostavno modificirati. Izvještaj „elapsed?“ provjerava dali je broj otkucaja veći od broja otkucaja trajanja svjetla na semaforu te ako je odgovor potvrđan, agenti dobivaju informaciju da se promijene svjetla. Svaki put kada se promjeni boja semafora, otkucaji se mjere iznova.

3.2. Sučelje modela

Sučelje modela se sastoji od podešivača kojima se manipulira varijablama u modelu. Preko sučelja mogu se podesiti:

- Ograničenje brzine: niti jedan automobil neće prekoračiti zadano ograničenje („speed-limit“)
- Najveće ubrzanje: Ubrzanje koje automobil ima prije nego što dosegne ograničenje brzine („max-accel“)
- Brzina usporavanja vozila: Brzina usporavanja automobila odnosno brzina kočenja („max-brake“)
- Dinamika pojave novih automobila u modelu u smjeru zapada („freq-west“)
- Dinamika pojave novih automobila u modelu u smjeru istoka („freq-east“)
- Trajanje zelenog svjetla („green-length“)
- Trajanja žutog svjetla („yellow-length“)
- Udaljenost između koordinate ishodišta i semafora („razmak2“ - ova varijabla daje duljinu zastoja na prometnici)



Slika 11: Sučelje modela

Lijeva strana sučelja sadrži ranije spomenute klizače i pokretače modela. Centralni dio sučelja zauzima grafički vizir na kojem će se simulacija odvijati. Zbog potrebe praćenja rezultata na sučelju se nalaze i mjerači koji prikazuju broj automobila koji čekaju te dijagram vozila koji čekaju u vremenu.

4. DIMENZIONALNA ANALIZA

Svrha modela je ispitati čekanje automobila tijekom privremene regulacije prometa i utvrđivanje optimalnih parametara brzine, trajanja zelenog i crvenog svjetla za određenu duljinu privremene regulacije. Simulacija se vršila za ograničenje brzine u iznosu od 30, 40 i 50 km/h

Da bi izvršili simulaciju modela potrebno je napraviti dimenzionalnu analizu modela. NetLogo program radi u jedinicama koje nemaju dimenziju stoga je potrebno otkucaje i jedinične vrijednosti modela pretvoriti u stvarne dimenzije.

4.1. Brzina

Prvi zastoje će se ispitati za predviđeno ograničenje brzine od 40 km / h. Prilikom pretvorbe jedinica potrebno je odrediti duljinu zastoja u modelu. Varijabla „razmak2“, koja određuje duljinu zastoja, iznosi 25. To znači da je ukupan razmak između dva semafora jednak 50 polja u modelu (semafori se nalaze na koordinatama (-50 -1) i (50, 1)). Iz toga slijedi:

$$\text{Veličina jednog polja } (P) = \frac{\text{broj polja}}{\text{razmak2} * 2}$$

$$P = \frac{100}{25 * 2} = \frac{100}{50} = 2 \text{ m}$$

Zadana je brzina od 40 km/h

$$V = 40 \frac{\text{km}}{\text{h}} = 11.111 \frac{\text{m}}{\text{s}}$$

Prvi korak dimenzionalne analize je izjednačavanje stvarne brzine vozila sa brzinom zadanom u modlelu. Brzina koja iznosi 8 jedinica pomaka tijekom jednog otkućaja u modelu izjednačiti će se sa brzinom od 40 km/h. To znači da će u jednom ticku agent prijeći ukupno 2 polja. Agent se za svaku jedinicu brzine u modelu pomakne prema naprijed za 0.25.

$$1 \text{ otkućaj} = \text{agent prijeđe 2 polja} = 2 * P = 4m$$

Iz izjednačavanja duljine koju automobil prijeđe u $t = 1$ s pri brzini od 11.111 m/s sa brojem polja koje agent mora prijeći da bi postigao istu udaljenost proizlazi:

$$\frac{11.111m}{P} = \frac{11.111m}{2 \frac{m}{polje}} = 5.555 \text{ polja}$$

Duljina od 11.111m je jednaka iznosu 5.555 polja u modelu. Ako znamo da vozilo prijeđe dva polja u jednom otkucaju slijedi:

$$\frac{5.555 \text{ polja}}{2 \frac{polja}{t_c}} = 2.777$$

Dakle, da bi vozilo prešlo duljinu od 5.555 polja potrebno mu je 2.777 otkucaja. Iz ovoga proizlazi da je jedna sekunda jednaka 2.777 otkucaja u modelu.

$$1 \text{ s} = 2.777 \text{ otkucaja}$$

$$1 \text{ otkucaj} = 0.36 \text{ s}$$

Preko ove jednakosti određivati će se svi ostali parametri u modelu.

a) Akceleracija vozila

Maksimalna akceleracija je vrijednost koja je izvučena sa stranice Hrvatkog Auto Kluba (HAK).

Prema dostupnim podacima prosječno vozilo ima ubzanje od [4]:

$$a = 2.14 \frac{m}{s^2}$$

Pretvaranjem stvarnih jedinica u jedinice modela dobiva se:

$$1 \text{ m} = 0.5 \text{ P}$$

$$1 \text{ s} = 2.777 \text{ otkucaja}$$

$$a_m \approx 2,00 \text{ jedinice u modelu}$$

b) Deceleracija vozila

Prosječna duljina kočenja automobila iz koje se može izračunati prosječna deceleracija iznosi

$$d = 1,50 \frac{m}{s^2}$$

Pretvaranjem stvarnih jedinica u jedinice modela dobiva se:

$$d_m = 0.25 \frac{\text{polja}}{1 \text{ otkucaj}} \approx 1,00 \text{ jedinice u modelu}$$

4.2. Vremena semafora

Definirane su tri različite duljine trajanja svjetala na semaforu za koje su vršeni eksperimenti pri brzini vozila od 30, 40 i 50 km/h. Brzine su odabrane na način da zadovolje uvjete koji će se ispitati modelom, a to su:

- Kratko trajanje svjetla: svi automobili koji čekaju na semaforu neće uspjeti proći u prvom idućem ciklusu zelenog svjetla
- Srednje trajanje svjetla: svi automobili koji čekaju će uspjeti proći kroz prvi idući ciklus zelenog svjetla
- Dugo trajanje svjetla: svi automobili koji stoje će proći kroz prvi idući ciklus zelenog svjetla, ali na semaforu će se još neko vrijeme zadržati zeleno svjetlo (neće biti novih vozila)

Ranije je ustanovljena veličina jedne sekunde u modelu u iznosu od 2.777 otkucaja. Sva vremena koja ispitana eksperimentom iznose:

1) Kratki ciklus svjetala u prvoj simulaciji:

$$t_{zeleno} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{zuto} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{crveno} = \frac{100 \text{ otkucaja}}{2.777} = 36 \text{ sekundi}$$

2) Srednji ciklus svjetala u prvoj simulaciji:

$$t_{zeleno} = \frac{84 \text{ otkucaja}}{2.777} = 30 \text{ sekundi}$$

$$t_{\text{žuto}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{134 \text{ otkucaja}}{2.777} = 48 \text{ sekundi}$$

3) Dugi ciklus semafora u prvoj simulaciji:

$$t_{\text{zeleno}} = \frac{100 \text{ otkucaja}}{2.777} = 36 \text{ sekundi}$$

$$t_{\text{žuto}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{150 \text{ otkucaja}}{2.777} = 54 \text{ sekundi}$$

4) Kratki ciklus semafora u drugoj simulaciji:

$$t_{\text{zeleno}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{žuto}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{100 \text{ otkucaja}}{2.777} = 36 \text{ sekundi}$$

5) Srednji ciklus u drugoj simulaciji:

$$t_{\text{zeleno}} = \frac{84 \text{ otkucaja}}{2.777} = 30 \text{ sekundi}$$

$$t_{\text{žuto}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{134 \text{ otkucaja}}{2.777} = 48 \text{ sekundi}$$

6) Dugi ciklus u drugoj simulaciji:

$$t_{\text{zeleno}} = \frac{95 \text{ otkucaja}}{2.777} = 34 \text{ sekunde}$$

$$t_{\text{žuto}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{145 \text{ otkucaja}}{2.777} = 52 \text{ sekunde}$$

7) Kratki ciklus u trećoj simulaciji:

$$t_{\text{zeleno}} = \frac{40 \text{ otkucaja}}{2.777} = 14 \text{ sekundi}$$

$$t_{\text{žuto}} = \frac{40 \text{ otkucaja}}{2.777} = 14 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{80 \text{ otkucaja}}{2.777} = 28 \text{ sekunde}$$

8) Srednji ciklus u trećoj simulaciji:

$$t_{\text{zeleno}} = \frac{84 \text{ otkucaja}}{2.777} = 30 \text{ sekundi}$$

$$t_{\text{žuto}} = \frac{50 \text{ otkucaja}}{2.777} = 18 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{134 \text{ otkucaja}}{2.777} = 48 \text{ sekunde}$$

9) Dugi ciklus u trećoj simulaciji:

$$t_{\text{zeleno}} = \frac{90 \text{ otkucaja}}{2.777} = 32 \text{ sekunde}$$

$$t_{\text{žuto}} = \frac{40 \text{ otkucaja}}{2.777} = 14 \text{ sekundi}$$

$$t_{\text{crveno}} = \frac{130 \text{ otkucaja}}{2.777} = 46 \text{ sekundi}$$

5. SIMULACIJA MODELA

Za odabrane parametre modela vršena je simulacija modela. Problem se pručavao u odabranom vremenskom periodu od 10 minuta što iznosi približno 1600 otkucaja u modelu. Za svako ograničenje brzine, napravljene su tri simulacije sa različitim vremenima trajanja svjetla na semaforu. Pretpostavljena vremena trajanja zelenog, žutog i crvenog svjetla su odabrana na način da pokrivaju različite slučajeve prolaska vozila kroz promatrani zastoje u prometu. Prvi slučaj je slučaj „kratkog“ trajanja u kojem se postiglo da automobili koji čekaju u koloni na crvenom svjetlu ne uspiju svi proći kroz zeleno svjetlo nakon što se ono upali. Slučaj „srednjeg“ trajanja zelenog svjetla je odabran na način da se zadovolji uvjet da su svi automobili koji su čekali, prošli kroz zeleno svjetlo u idućem ciklusu. Treći slučaj je slučaj „dugog“ trajanja zelenog svjetla kojim se postiglo to da svi automobili prođu kroz dionicu za vrijeme trajanja zelenog svjetla, a boja na semaforu se još nije promijenila. Tijekom simulacije u svakom trenutku bilježen je broj vozila koja čekaju, broj vozila koja su prošla kroz model u svakom smjeru i ukupan broj generiranih vozila. Iz direktnih rezultata modela prikazani su i normalizirani rezultati. Broj automobila u modelu će uvijek biti drugačiji i zbog toga je bilo potrebno normalizirati rezultate eksperimenta. Rezultati su normalizirani na način da je broj vozila u čekanju u modelu podijeljen sa ukupnim brojem generiranih vozila u tom trenutku. Rezultati su bilježeni i analizirani da bi se pronašli optimalni parametri.

5.1. Simulacija 1

Prva simulacija modela predstavlja slučaj ograničenja brzine od 30 km/h. Promatrati će se rezultati za različita vremena trajanja svjetala na semaforu.

Tablica 1: Prikaz parametara prve simulacije

Simulacija 1							
Parametar	Brzina [pomak/otkucaj]	Akceleracija [pomak/otkucaj g]	Deceleracija [pomak/otkucaj]	Trajanje zelenog svjetla [otkucaj]	Trajanje žutog svjetla [otkucaj]	Trajanje crvenog svjetla [otkucaj]	
Simulacija 1.1	6	2,00	1,00	50	50	100	Kratko
Simulacija 1.2	6	2,00	1,00	84	50	134	Srednje
Simulacija 1.3	6	2,00	1,00	100	50	150	Dugo

5.1.1. Simulacija 1.1

Prva simulacija za zastoje od 100 metara, u trajanju od 1600 otkucaja. Parametri zadani u modelu su:

$$v = 6,00 \text{ [pomak/otkucaj]}$$

$$a_m = 2,00 \text{ [pomak/otkucaj]}$$

$$d_m = 1,00 \text{ [pomak/otkucaj]}$$

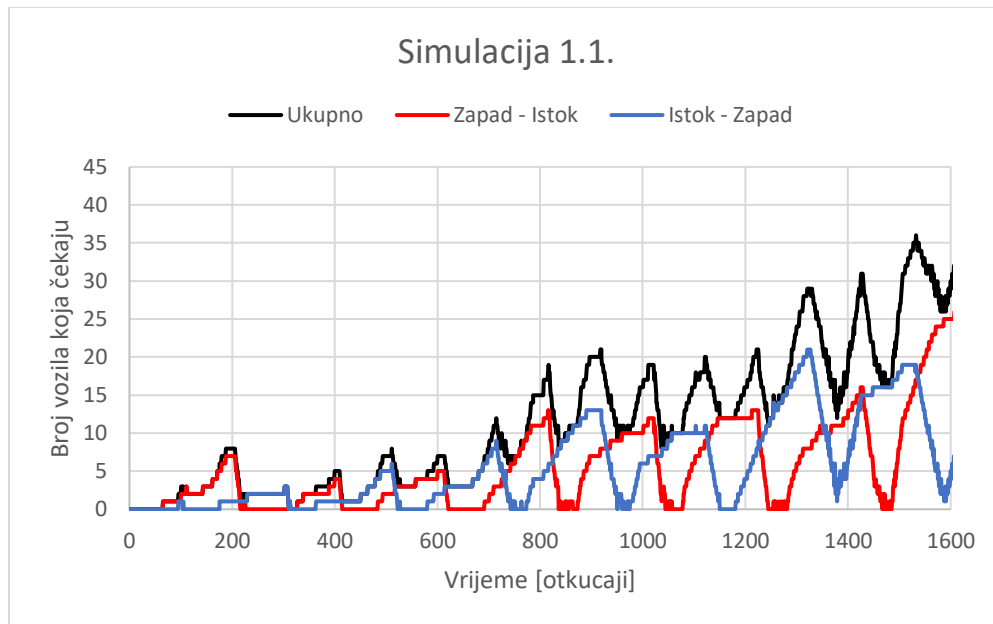
$$t_{zeleno} = 50 \text{ otkucaja}$$

$$t_{žuto} = 50 \text{ otkucaja}$$

$$t_{crveno} = 100 \text{ otkucaja}$$

Ovakvi parametri odgovaraju slučaju „kratkog“ trajanja zelenoga svjetla na semaforu.

Tijekom simulacije, primjećeno je da svi automobili koji čekaju na semaforu ne uspijevaju proći kroz prvi ciklus zelenog svjetla nakon što se promjeni boja.



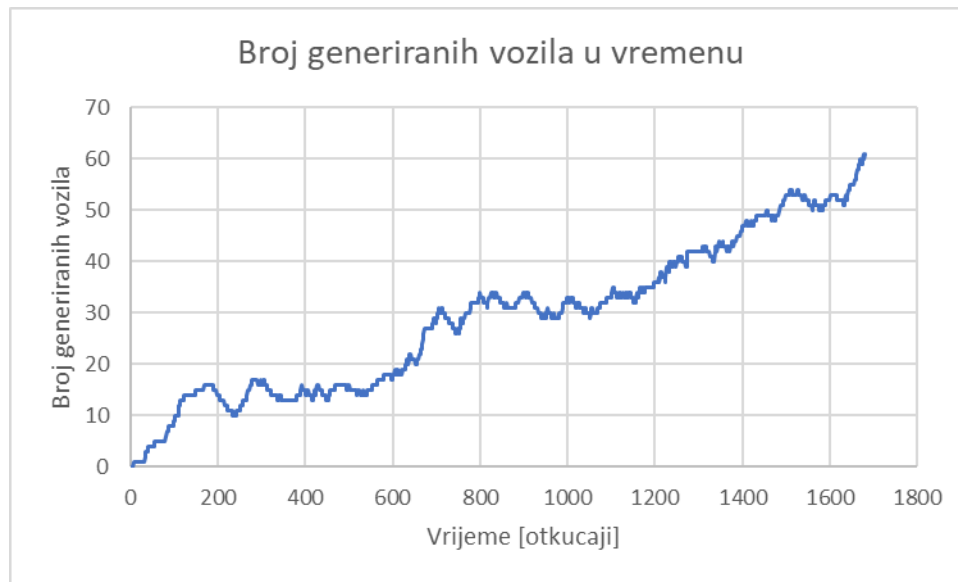
Dijagram 1: Simulacija br. 1

Na Dijagram 1 vidljivi su rezultati pokusa. Nakon određenog broja otkucaja kretanje vozila dobiva određeni uzorak po kojemu se promet nastavlja odvijati. U promatranom vremenskom periodu veći zastoje će nastati u pravcu istoka, gdje će najviše 26 vozila biti u koloni za vrijeme trajanja crvenog svjetla na semaforu. Pravac istok – zapad je nešto rasterećeniji i tamo se u maksimalno 21 vozilo nalazilo u koloni za vrijeme trajanja crvenog svjetla. Ne smije se zanemariti niti da u svakom trenutku postoji određeni broj vozila koji čekaju u oba smjera. U trenutku kada je promet bio najgušći sve ukupno 40 vozila se nije kretalo.

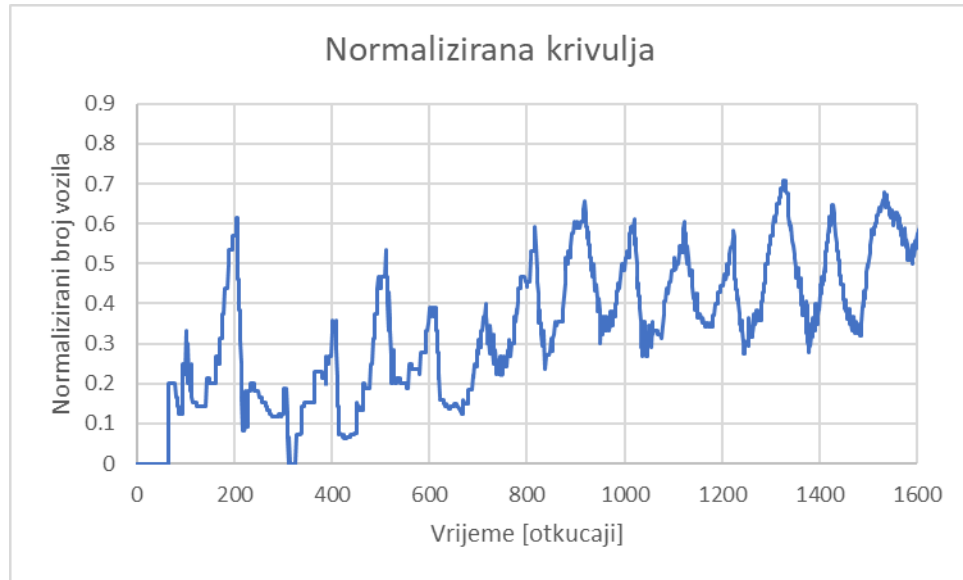
Za vrijeme trajanja pokusa bilježena je i ukupna protočnost cijelog promatranog segmenta. Kroz model je za vrijeme pokusa ukupno prošlo 104 vozila u smjeru istoka, a 93 vozila u smjeru zapada. Podaci o normaliziranom broju vozila u svakom praćenom trenutku dani su na Dijagram 1.

Tablica 2: Prikaz rezultata simulacije 1.1

Simulacija 1.1.			
Smjer	Broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	26	36	104
Istok - Zapad	21	26	93
Ukupno	40	-	197



Dijagram 2: Broj generiranih vozila u vremenu



Dijagram 3: Normalizirana krivulja

5.1.2. Simulacija 1.2.

$$v = 6 \text{ [pomak/otkucaj]}$$

$$a_m = 2.00 \text{ [pomak/otkucaj]}$$

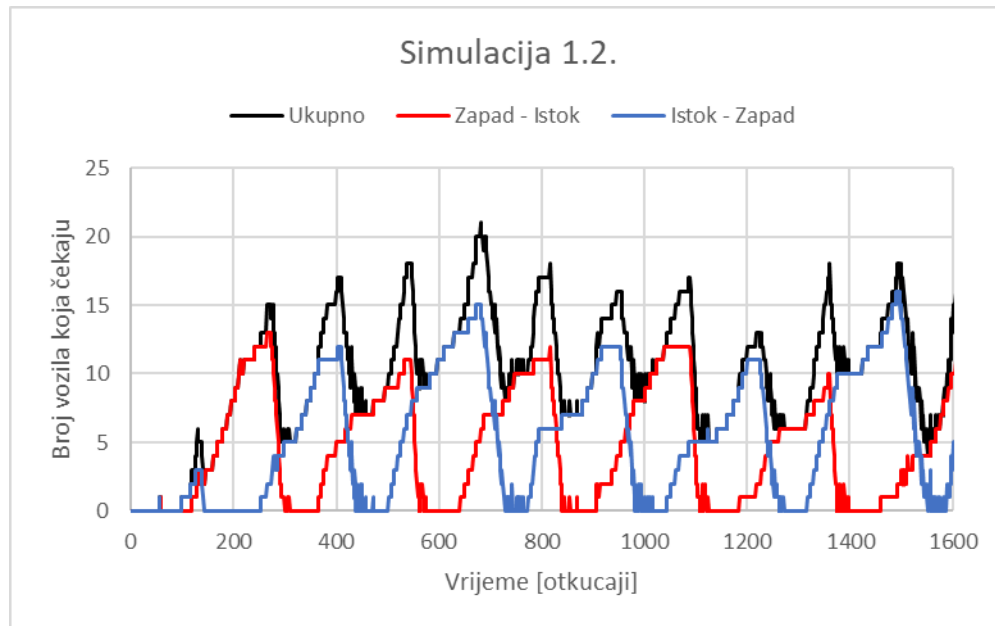
$$d_m = 1.00 \text{ [pomak/otkucaj]}$$

$$t_{zeleno} = 84 \text{ otkucaja}$$

$$t_{zuto} = 50 \text{ otkucaja}$$

$$t_{crveno} = 134 \text{ otkucaja}$$

Simulacija 1.2. odgovara srednjem vremenu trajanja zelenog svjetla. Svi automobili koji čekaju će uspjeti proći u prvom idućem ciklusu zelenog svjetla.

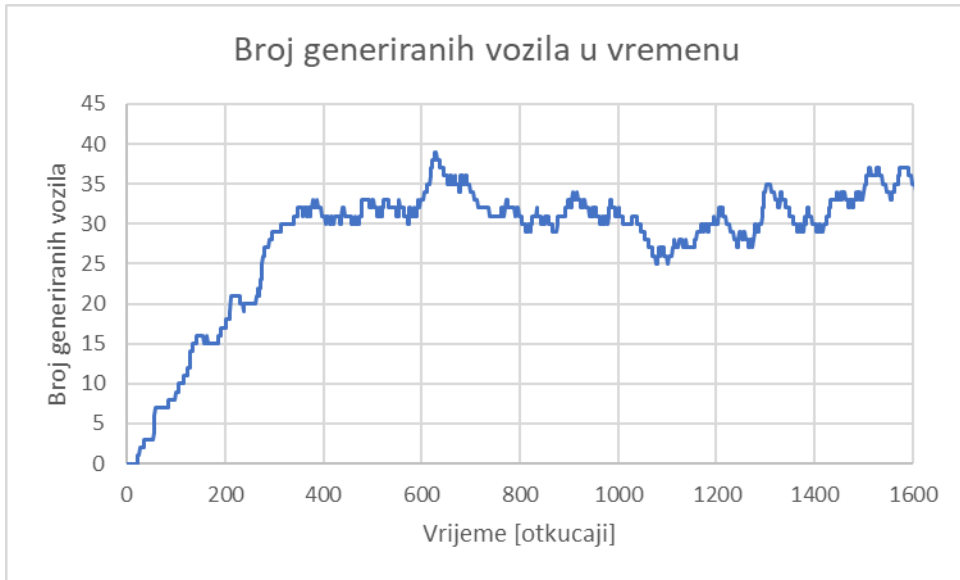


Dijagram 4 Simulacija 1.2.

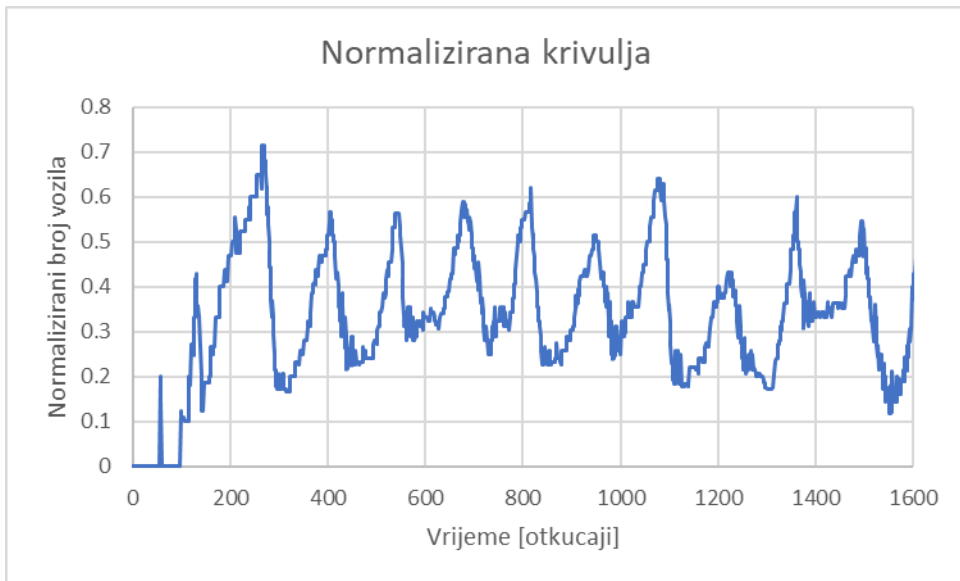
Na Dijagram 4 mogu se iščitati rezultati drugog pokusa za brzinu od 30 km/h. Najveći broj vozila stajao je u koloni u smjeru zapada i iznosi 16. Najveći broj vozila koja čekaju za vrijeme crvenog svjetla u smjeru istoka je iznosio 13. U trenutku najvećeg zagušenja prometa ukupno se 22 vozila nije kretalo.

Tablica 3: Prikaz rezultata simulacije 1.2

Simulacija 1.2.			
Smjer	Broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	13	22	107
Istok - Zapad	16	25	92
Ukupno	22	-	199



Dijagram 5: Broj generiranih automobila - 1.2.



Dijagram 6: Normalizirani prikaz čekanja vozila - 1.2.

5.1.3. Simulacija 1.3.

$$v = 6$$

$$a_m = 2.00$$

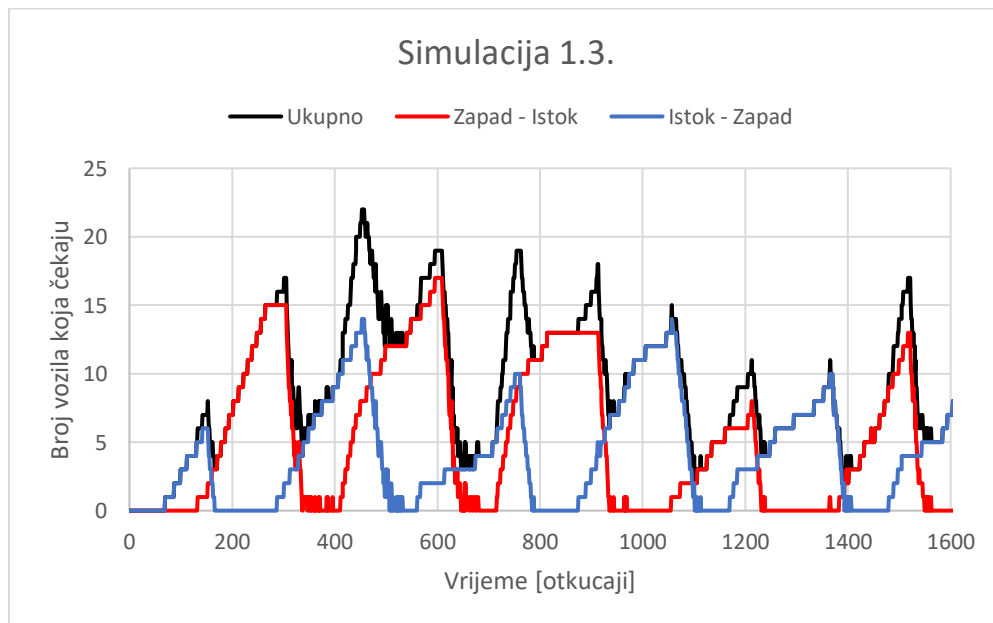
$$d_m = 1.00$$

$$t_{zeleno} = 100 \text{ otkucaja}$$

$$t_{zuto} = 50 \text{ otkucaja}$$

$$t_{crveno} = 150 \text{ otkucaja}$$

Simulacija 1.3. odgovara dugom trajanju zelenog vremena na semaforu. Osim što će svi automobili proći kroz prvi ciklus zelenog svjetla, dogoditi će se određeni vremenski period kada će zeleno svjetlo i dalje biti upaljeno, ali niti jedno vozilo u tom trenutku se neće nalaziti na semaforu.



Dijagram 7: Simulacija 1.3. - Broj vozila u čekanju

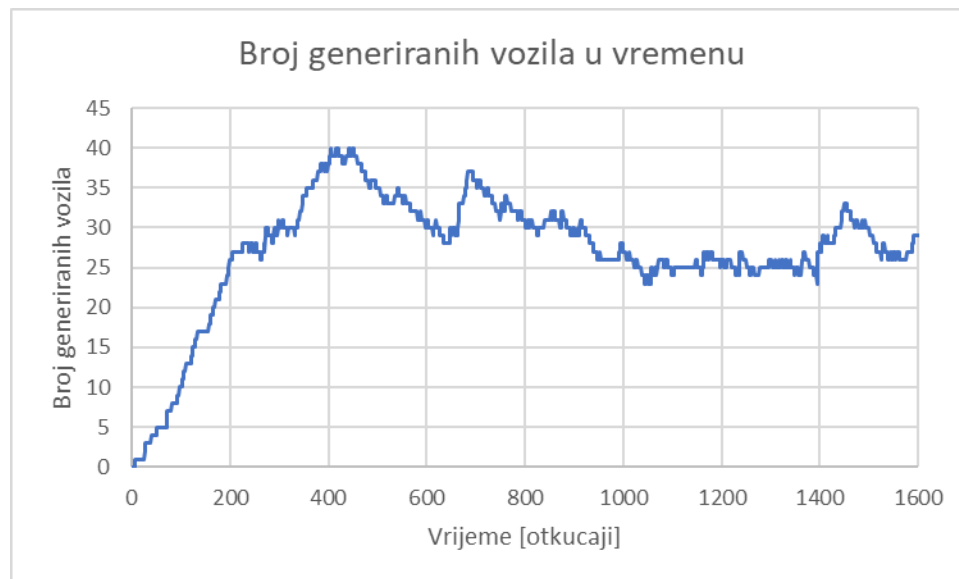
Za promatrani vremenski period u iznosu od 1600 otkucaja (cca 10 minuta), najveće čekanje će biti u smjeru istoka i tada će 17 vozila biti u zastoju, a u smjeru zapada će 14

vozila čekati. U periodu kada se najviše vozila u modelu nije kretalo, program je zabilježio 22 vozila koja su bila u zastoju.

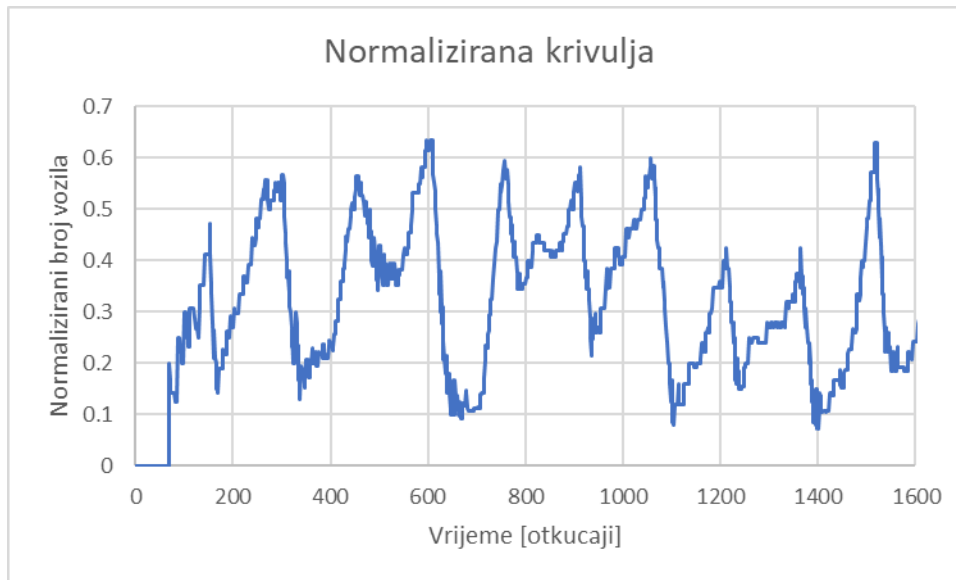
Tablica 4: Prikaz rezultata simulacije 1.3.

Simulacija 1.3.			
Smjer	Broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	17	27	90
Istok - Zapad	14	20	91
Ukupno	22	-	181

Za vrijeme simulacije ukupno je 181 vozilo prošlo kroz model.



Dijagram 8: Broj generiranih vozila - 1.3.



Dijagram 9: Normalizirani prikaz čekanja vozila - 1.3.

5.2. Simulacija 2

Druga simulacija modela predstavlja slučaj kretanja vozila brzinom od 40 km/h. Promatrati će se rezultati za različita vremena trajanja svjetala na semaforu. Vrijednosti parametara analiziranih u drugoj simulaciji mogu se iščitati iz Tablica 5.

Tablica 5: Prikaz parametara simulacije 2

Simulacija 2							
Parametar	Brzina [pomak/otkucaj]	Akceleracija [pomak/otkucaj g]	Deceleracija [pomak/otkucaj]	Trajanje zelenog svjetla [otkucaj]	Trajanje žutog svjetla [otkucaj]	Trajanje crvenog svjetla [otkucaj]	
Simulacija 2.1	8	2,00	1,00	50	50	100	Kratko
Simulacija 2.2	8	2,00	1,00	84	50	134	Srednje
Simulacija 2.3	8	2,00	1,00	95	50	145	Dugo

5.2.1. Simulacija 2.1.

Prva simulacija za zastoje od 100 metara, u trajanju od 1600 otkucaja. Parametri zadani u modelu su:

$$v = 8,00 \text{ [pomak/otkucaj]}$$

$$a_m = 2.00 \text{ [pomak/otkucaj]}$$

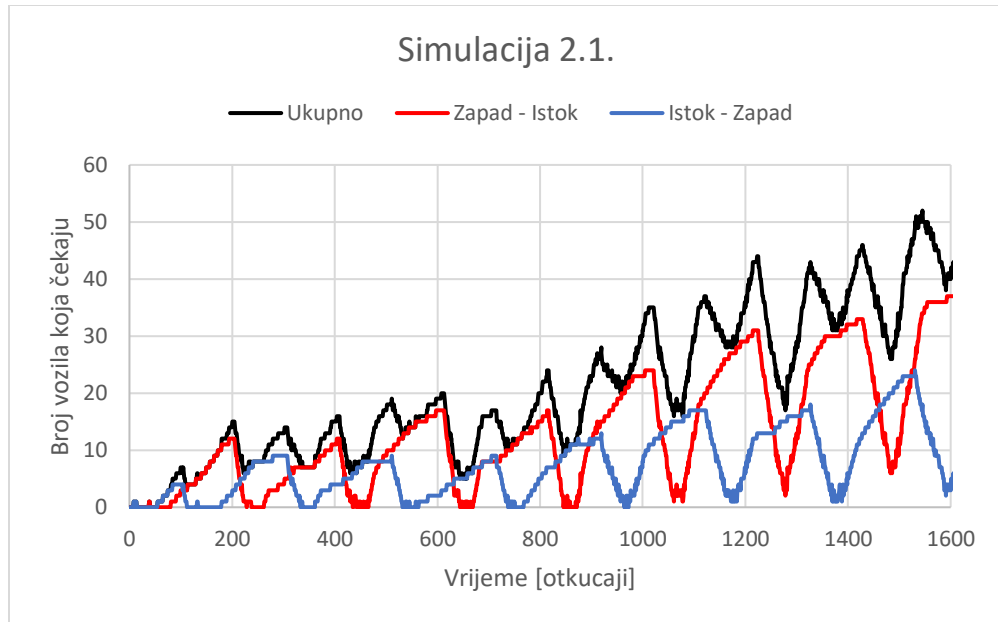
$$d_m = 1,00 \text{ [pomak/otkucaj]}$$

$$t_{zeleno} = 50 \text{ otkucaja}$$

$$t_{žuto} = 50 \text{ otkucaja}$$

$$t_{crveno} = 100 \text{ otkucaja}$$

Simulacija 2.1. odgovara kratkom vremenu trajanja zelenog svjetla.



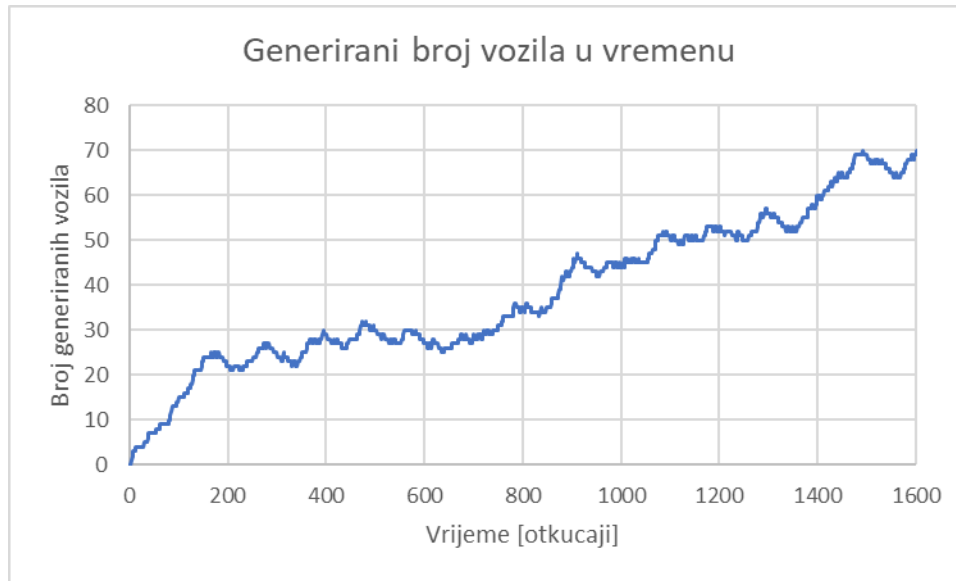
Dijagram 10: Rezultati simulacije 2.1.

Tablica 6: Prikaz rezultata simulacije 2.1.

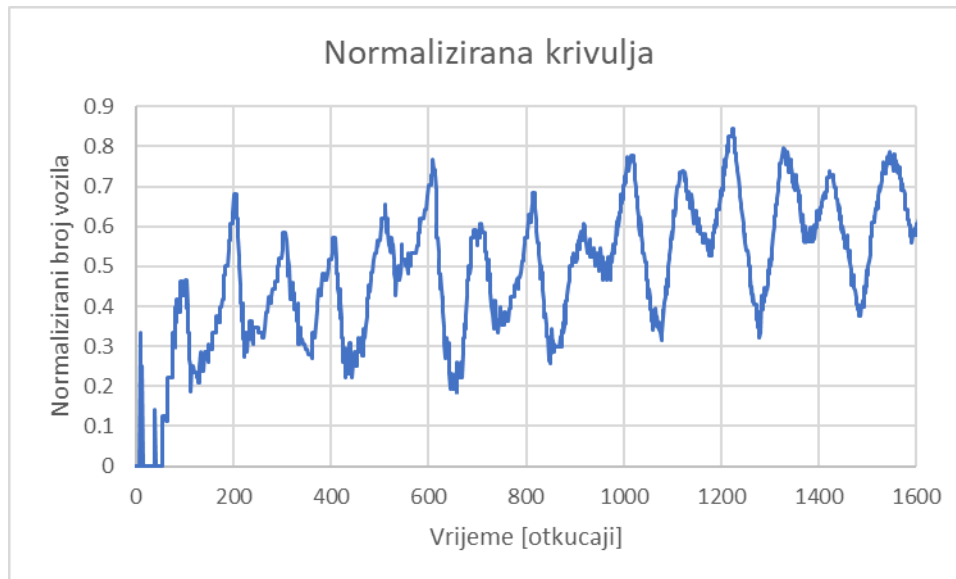
Simulacija 2.1.			
Smjer	Najveći broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	38	46	120
Istok - Zapad	24	31	103
Ukupno	52	-	223

Na Dijagram 10 vidljivi su rezultati pokusa. Može se uočiti da nakon određenog broja otkucaja kretanje vozila dobiva određeni uzorak po kojemu se promet nastavlja odvijati. U promatranom vremenskom periodu veći zastoje će nastati u pravcu istoka, gdje se najviše 38 vozila nalazi u koloni za vrijeme trajanja crvenog svjetla na semaforu. Pravac istok – zapad je nešto rasterećeniji i tamo se maksimalno 24 vozila nalaze u koloni za vrijeme trajanja crvenog svjetla. U trenutku kada je promet bio najgušći se ukupno 52 vozila nisu kretala.

Kroz model je ukupno prošlo 223 vozila za vrijeme trajanja simulacije. U smjeru istoka prošlo je 120 vozila, a u smjeru zapada 103 vozila.



Dijagram 11: Generirani broj vozila - 2.1.



Dijagram 12: Normalizirani prikaz čekanja vozila - 2.1.

5.2.2. Simulacija 2.2.

$$v = 8,00 \text{ [pomak/otkucaj]}$$

$$a_m = 2,00 \text{ [pomak/otkucaj]}$$

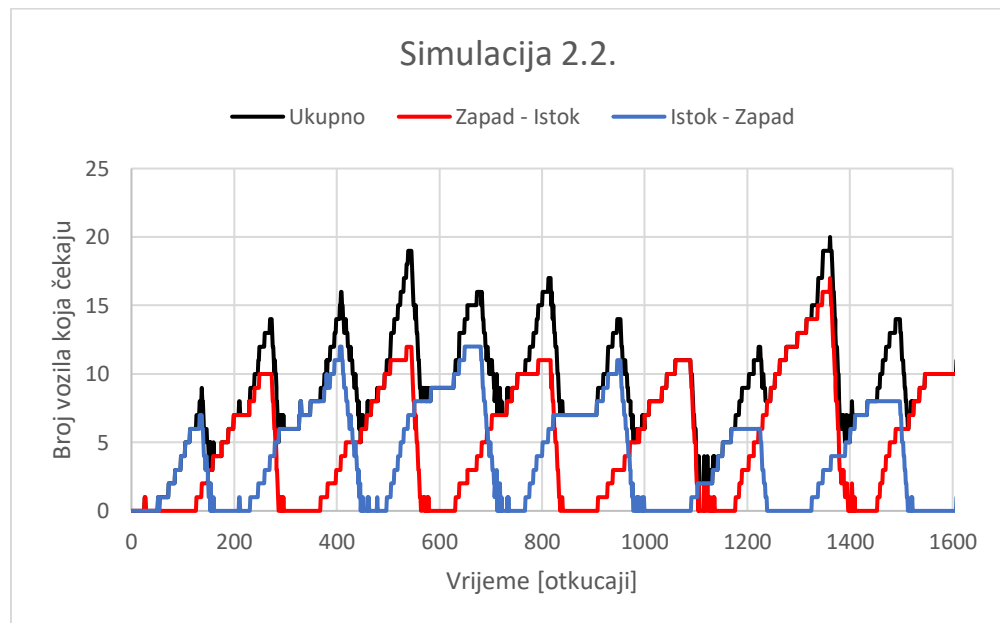
$$d_m = 1,00 \text{ [pomak/otkucaj]}$$

$$t_{zeleno} = 84 \text{ otkucaja}$$

$$t_{\text{žuto}} = 50 \text{ otkucaja}$$

$$t_{crveno} = 134 \text{ otkucaja}$$

Simulacija 2.2. odgovara slučaju srednjeg trajanja zelenog svjetla.

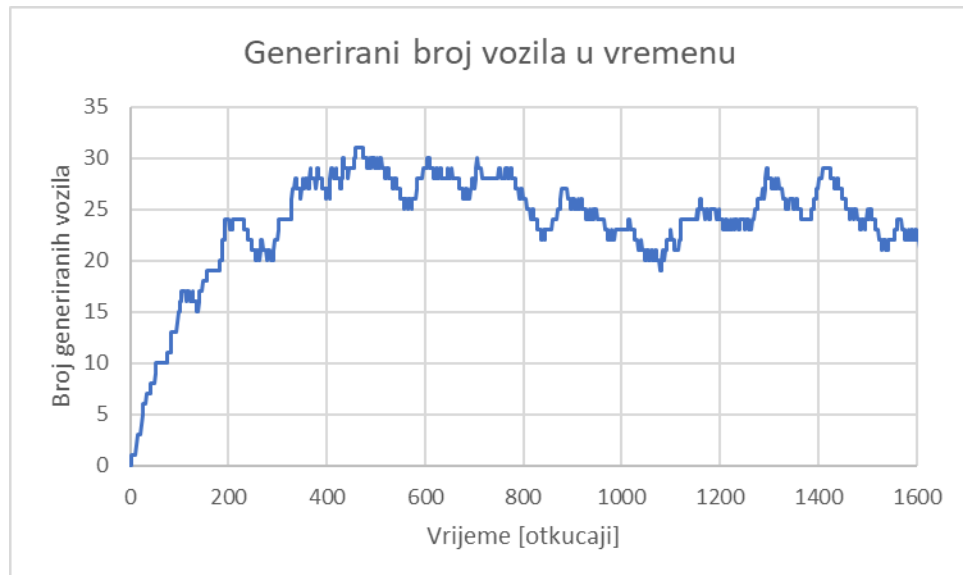


Dijagram 13: Simulacija 2.2.

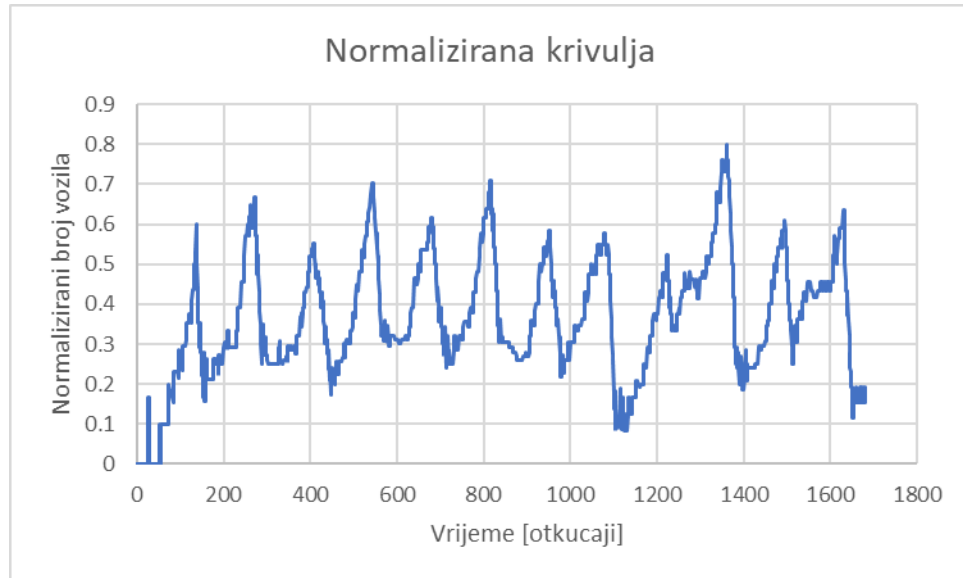
Iz Tablica 7 mogu se iščitati rezultati praćenja simulacije 2.2. Navije je vozila stajalo u smjeru istoka, 17 vozila, a u smjeru zapada najveći broj vozila koja su stajala je iznosio 12. Ukupno je kroz model za vrijeme simulacije prošlo 187 vozila, od čega 100 u pravcu istoka, a 87 u pravcu zapada.

Tablica 7: Prikaz rezultata simulacije 2.2

Simulacija 2.2.			
Smjer	Najveći broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	17	21	100
Istok - Zapad	12	20	87
Ukupno	20	-	187



Dijagram 14: Broj generiranih vozila - 2.2.



Dijagram 15: Normalizirani prikaz čekanja vozila - 2.2.

5.2.3. Simulacija 2.3.

$$v = 8,00$$

$$a_m = 2,00$$

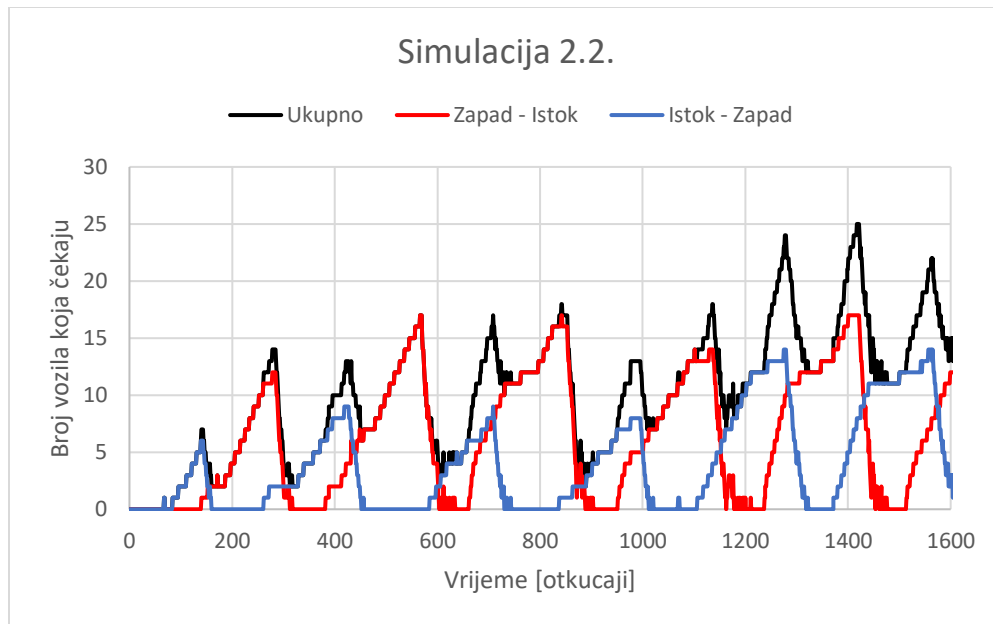
$$d_m = 1,00$$

$$t_{zeleno} = 95 \text{ otkucaja} = 95/2.777 = 35 \text{ s}$$

$$t_{\text{žuto}} = 50 \text{ otkucaja} = 18 \text{ s}$$

$$t_{crveno} = 145 \text{ otkucaja} = 52 \text{ s}$$

Simulacija 2.2. predstavlja slučaj dugog trajanja zelenog svjetla. Na Dijagram 1 može se iščitati tijek pokusa.

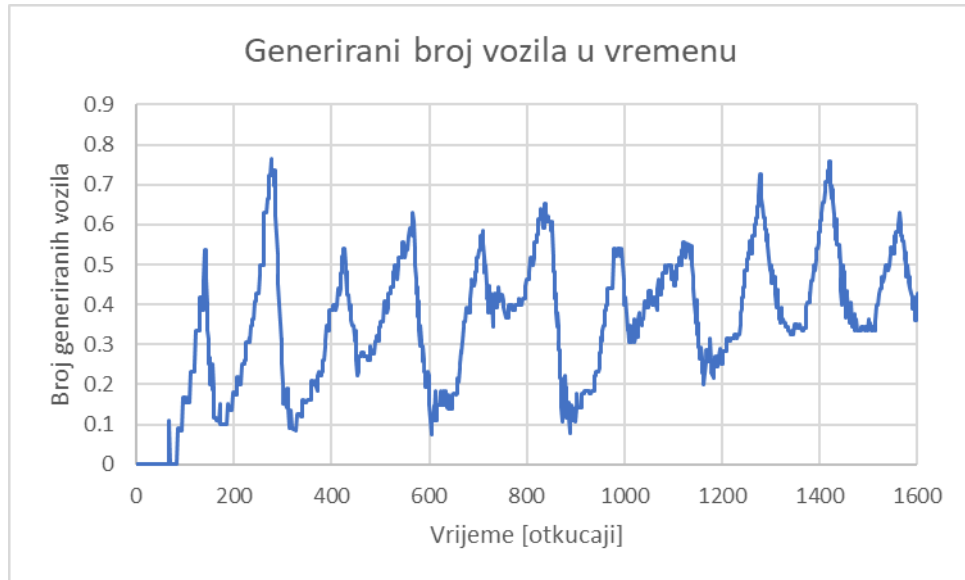


Dijagram 16: Simulacija 2.3.

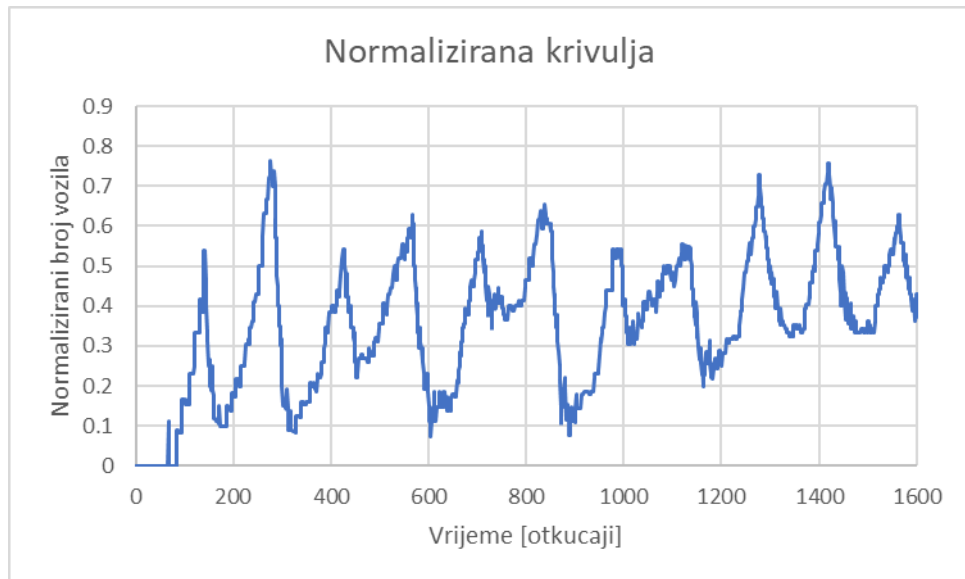
Smjer od zapada prema istoku je najviše bio opterećen sa 17 vozila u čekanju, dok je smjer od istoka prema zapadu bio najviše opterećen sa 14 vozila u čekanju. U istom trenutku je najviše stajalo 25 vozila u oba smjera. U Tablica 8 mogu se iščitati sva mjerenja.

Tablica 8: Prikaz rezultata simulacije 2.3.

Simulacija 2.3.			
Smjer	Najveći broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	25	27	107
Istok - Zapad	17	22	97
Ukupno	14	-	204



Dijagram 17: Generirani broj vozila - 2.3.



Dijagram 18: Normalizirani prikaz čekanja - 2.3.

5.3. Simulacija 3

Simulacija 3 modela predstavlja slučaj kretanja vozila brzinom od 50 km/h. Promatrati će se rezultati za različita vremena trajanja svjetala na semaforu.

Tablica 9: Prikaz parametara simulacije 3

Simulacija 3							
Parametar	Brzina [pomak/otkucaj]	Akceleracija [pomak/otkucaj g]	Deceleracija [pomak/otkucaj]	Trajanje zelenog svjetla [otkucaj]	Trajanje žutog svjetla [otkucaj]	Trajanje crvenog svjetla [otkucaj]	
Simulacija 3.1.	10	2,00	1,00	40	40	80	Kratko
Simulacija 3.2.	10	2,00	1,00	84	50	134	Srednje
Simulacija 3.3.	10	2,00	1,00	90	40	130	Dugo

5.3.1. Simulacija 3.1.

$$v = 10,00$$

$$a_m = 2,00$$

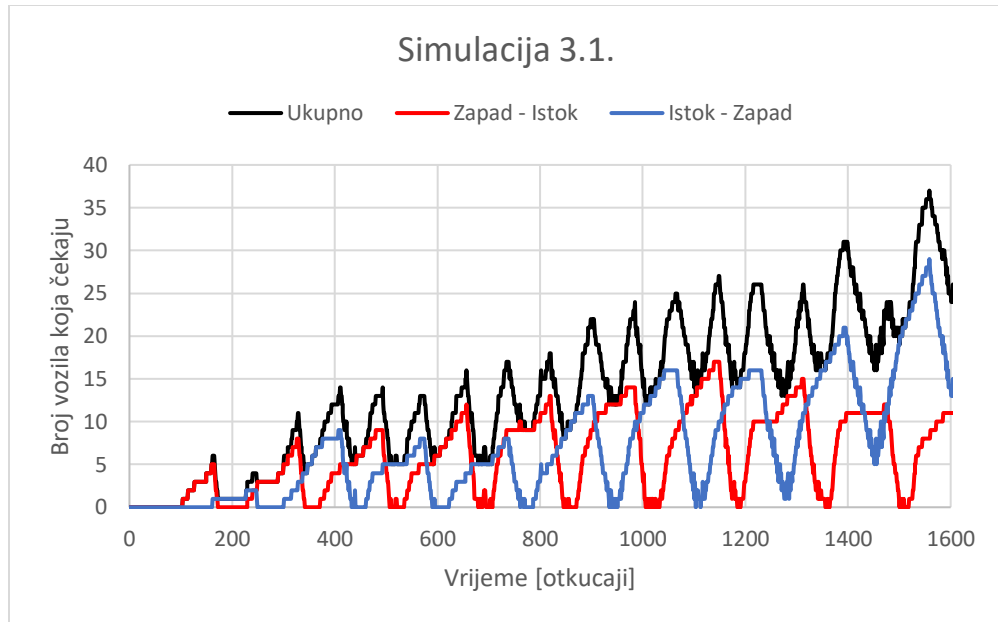
$$d_m = 1,00$$

$$t_{zeleno} = 40 \text{ otkucaja} = 40/2.777 = 15 \text{ s}$$

$$t_{žuto} = 40 \text{ otkucaja} = 15 \text{ s}$$

$$t_{crveno} = 80 \text{ otkucaja} = 20 \text{ s}$$

Simulacija 3.1. odgovara slučaju za kratko trajanje zelenog svjetla.

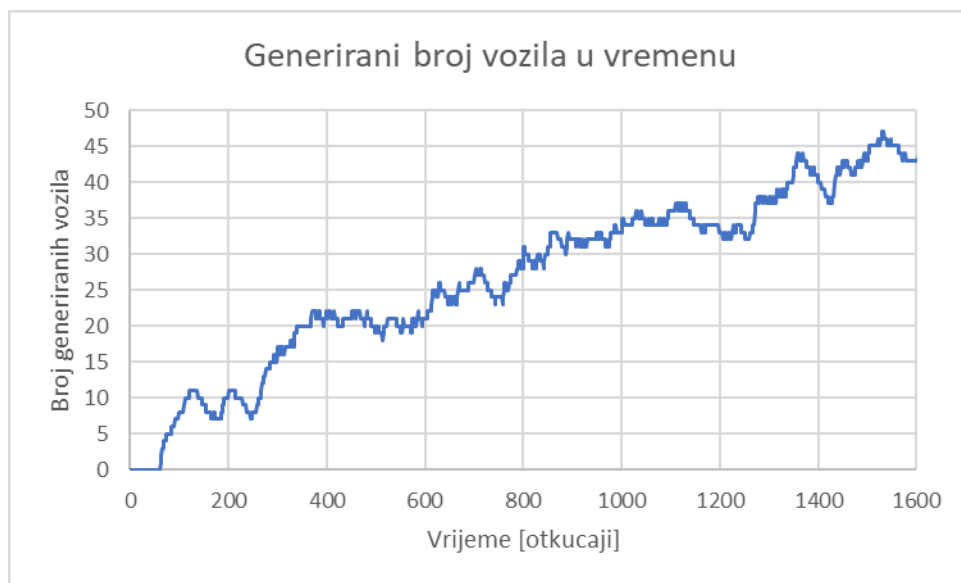


Dijagram 19: Simulacija 3.1.

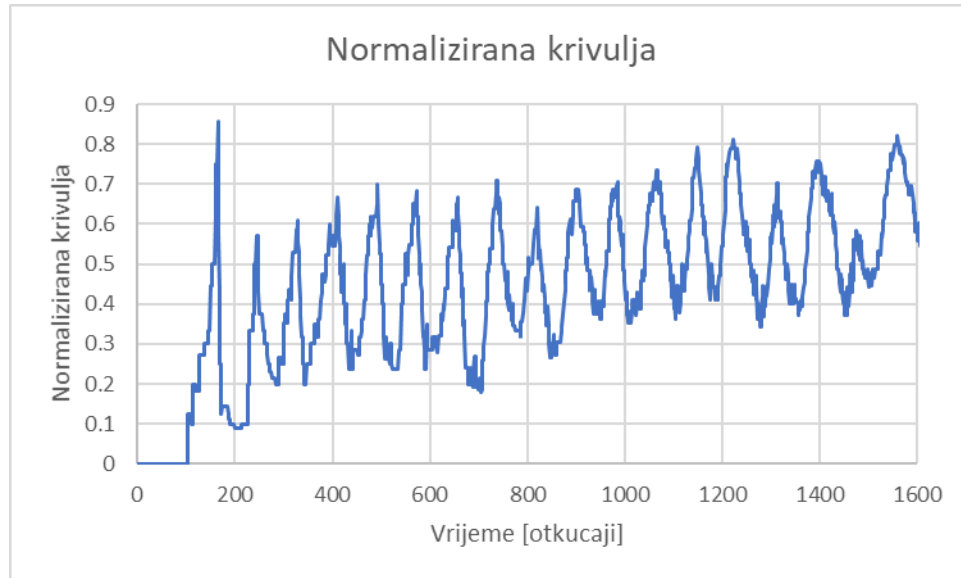
Za simulaciju 3.1. mogu se iščitati parametri na Tablica 10. Maksimalan broj vozila koja čekaju u smjeru istoka iznosi 17, a najveći broj vozila koja čekaju u smjeru zapada iznosi 29. Za vrijeme trajanja simulacije ukupno je 205 automobila prošlo kroz model.

Tablica 10: Prikaz rezultata simulacije 3.1.

Simulacija 3.1.			
Smjer	Najveći broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	17	20	105
Istok - Zapad	29	32	100
Ukupno	37	-	205



Dijagram 20: Broj generiranih vozila - 3.1.



Dijagram 21: Normalizirani prikaz čekanja vozila - 3.1.

5.3.2. Simulacija 3.2.

$$v = 10,00$$

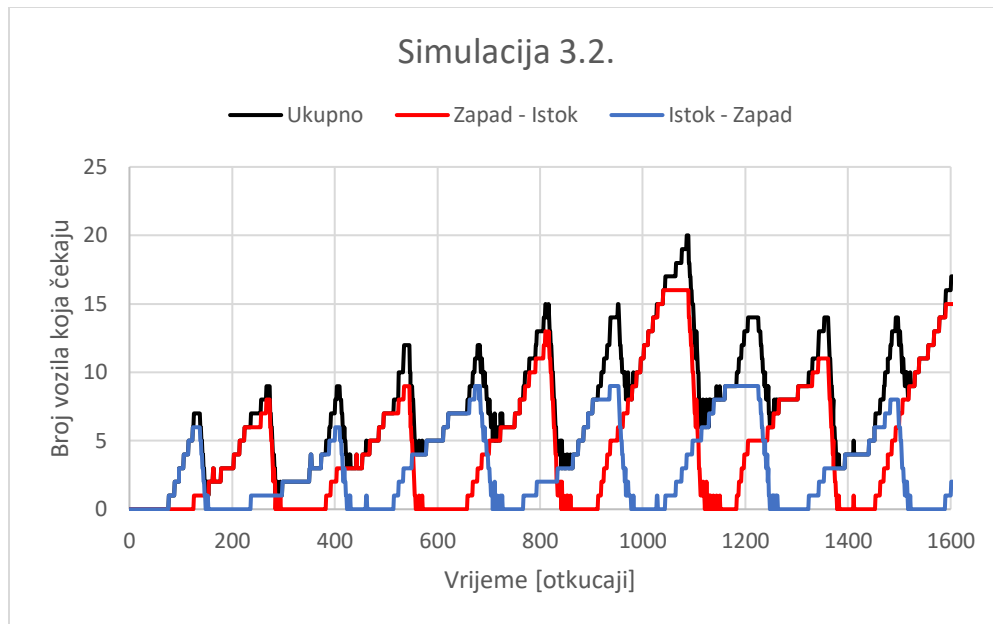
$$a_m = 2,00$$

$$d_m = 1,00$$

$$t_{zeleno} = 84 \text{ otkucaja}$$

$$t_{\text{žuto}} = 50 \text{ otkucaja}$$

$$t_{crveno} = 134 \text{ otkucaja}$$

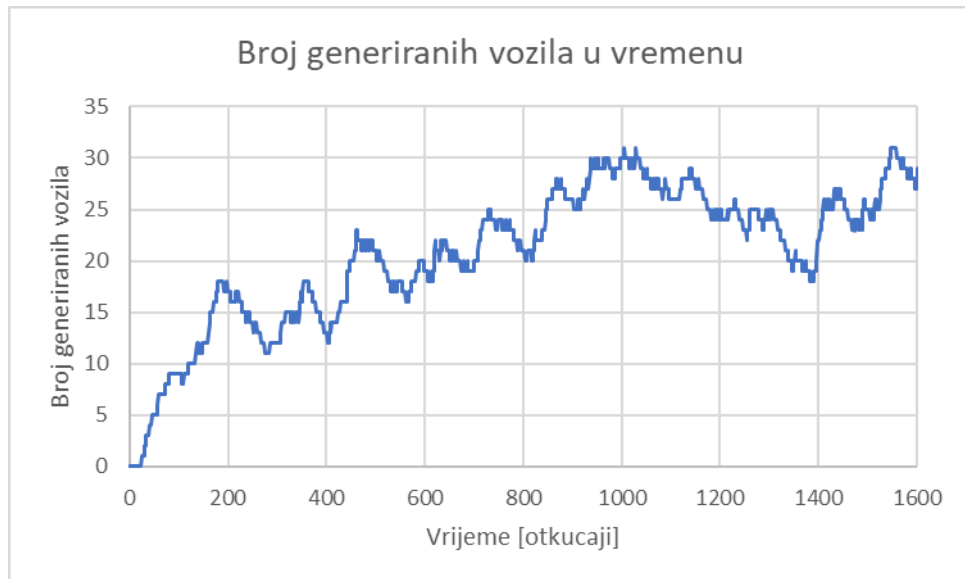


Dijagram: 22 Simulacija 3.2.

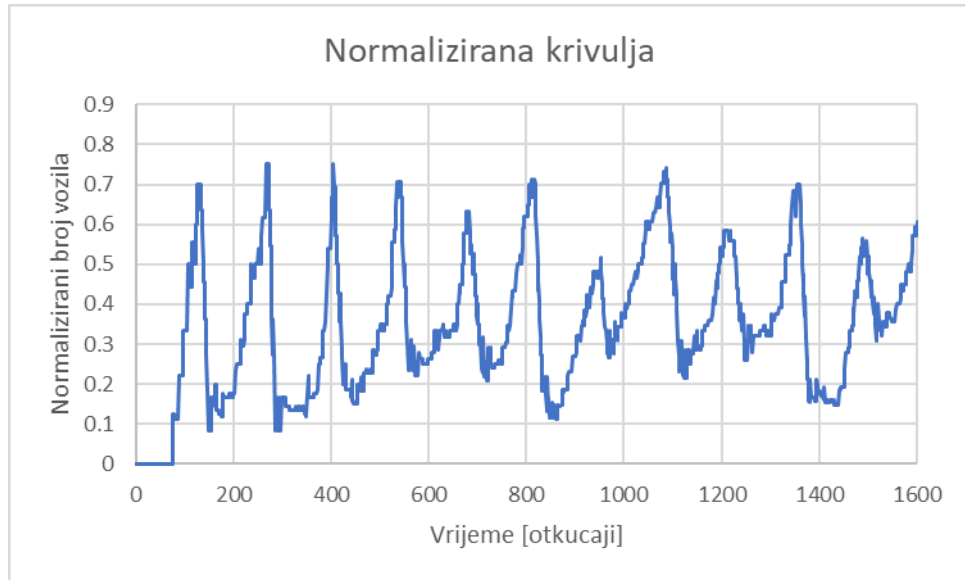
Za simulaciju 3.2., koja odgovara srednjem vremenu trajanja zelenog svjetla, najviše su 16 vozila stajala u smjeru istoka, a 9 u smjeru zapada. Za vrijeme simulacije najviše je 20 vozila stajalo u oba pravca. Na Tablica 11 mogu se iščitati bilježeni rezultati simulacije.

Tablica 11: Prikaz rezultata simulacije 3.2

Simulacija 3.2.			
Smjer	Najveći broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	16	20	98
Istok - Zapad	9	17	97
Ukupno	20	-	195



Dijagram 23: Broj generiranih vozila - 3.2.



Dijagram 24: Normalizirani prikaz čekanja vozila - 3.2.

5.3.3. Simulacija 3.3.

$$v = 10,00$$

$$a_m = 2,00$$

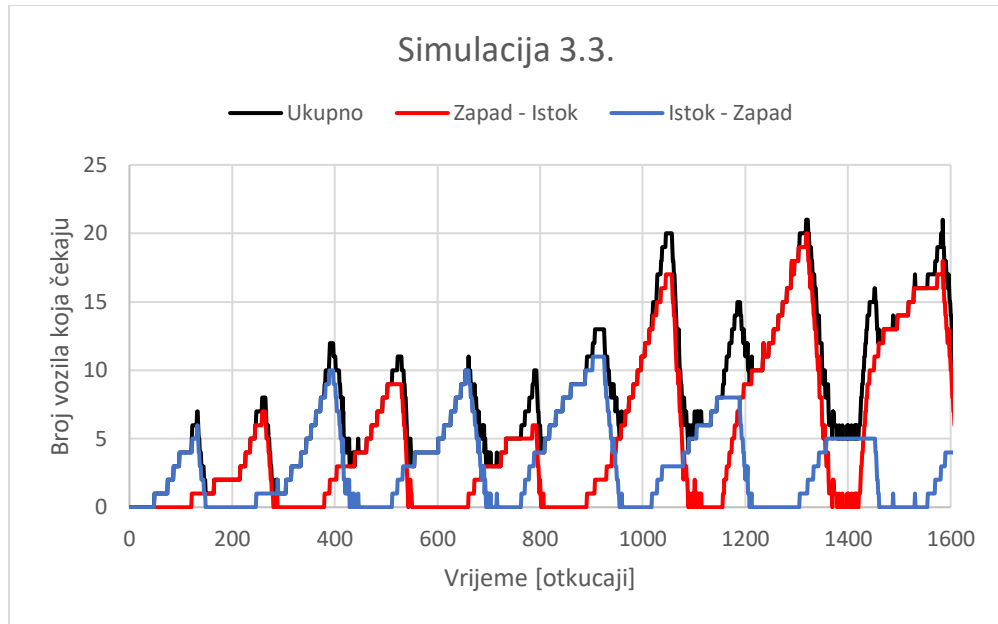
$$d_m = 1,00$$

$$t_{zeleno} = 90 \text{ otkucaja}$$

$$t_{\text{žuto}} = 40 \text{ otkucaja}$$

$$t_{crveno} = 130 \text{ otkucaja}$$

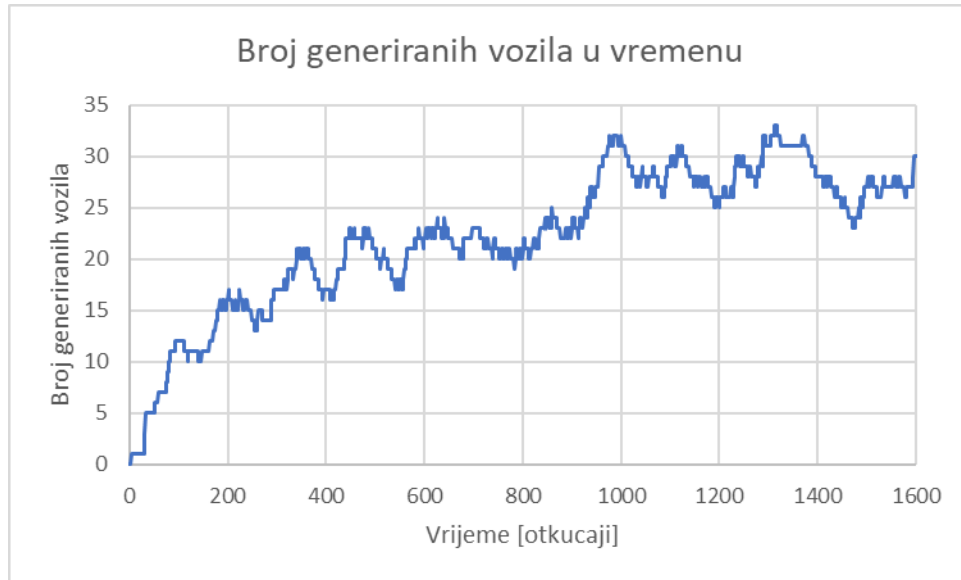
Simulacija 3.3. odgovara simulaciji dugog zelenog svjetla. Na Dijagram 25 može se vidjeti tijekom simulacije. U trenutku kada je model bio najzagušeniji, ukupno je 21 vozilo stajalo. U smjeru istoka zabilježen je najveći broj vozila koja stoje u iznosu od 20 vozila, dok je u smjeru zapada taj broj iznosio 11. Na Tablica 12 mogu se vidjeti svi podaci koji su izmjereni.



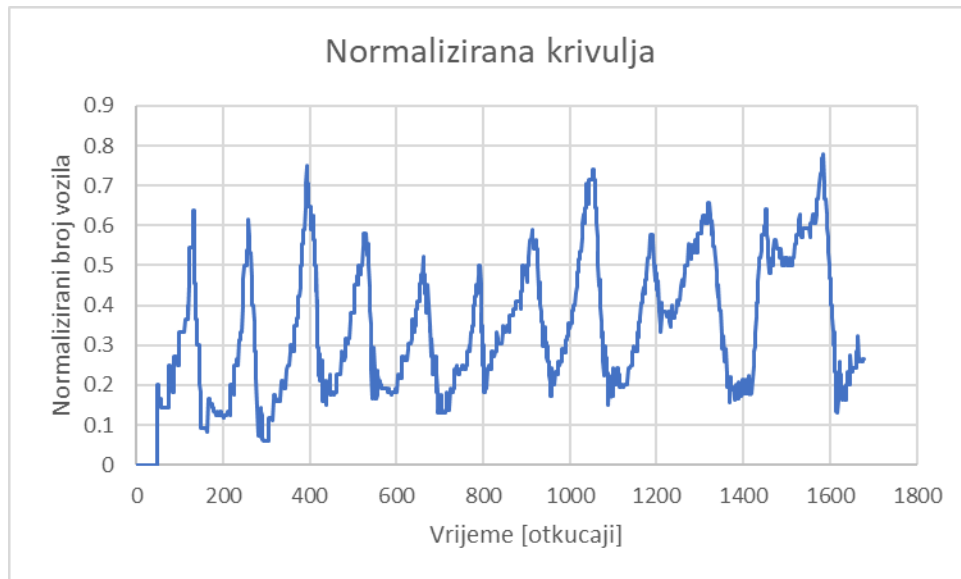
Dijagram 25: Simulacija 3.3.

Tablica 12: Rezultati simulacije 3.3.

Simulacija 3.3.			
Smjer	Najveći broj vozila koja čekaju	Najveći broj vozila u modelu	Protočnost cijelog segmenta
Zapad - Istok	20	27	107
Istok - Zapad	11	18	92
Ukupno	21	-	199



Dijagram 26: Broj generiranih vozila - 3.3.



Dijagram 27: Normalizirani prikaz čekanja vozila - 3.3.

6. ANALIZA REZULTATA I OPTIMALNIH PARAMETARA

Rezultati pokusa daju uvid u moguće ponašanje agenata u promatranom problemu. Iz analize modela i promatranja rezultata primjetno je postojanje određenih uzoraka u ponašanju modela za istu vrstu simulacije (simulacije sa kratkim, srednjim i dugim zelenim svjetlom). Za kratku duljinu trajanja zelenog svjetla količina automobila u zastoju može se aproksimirati pravcem i zaključiti da njihov broj linearno raste (Dijagram 1, Dijagram 10 i Dijagram 19). Razlog tomu je očit – generiranje automobila koji se pojavljuju u modelu je veći od propusnosti semafora. Svakim novim ciklusom broj automobila u koloni raste, dok kolona ne dođe do kraja modela, kada uzorak postane približno konstantan i nepromjenjiv. Sukladno tomu, najveći zastoji u modelu su uočeni upravo u simulacijama sa kratkim trajanjem zelenog svjetla. Bitno je naglasiti, da će rezultati pokusa varirati ovisno o parametrima na koje u simulacijama nije obraćana velika pozornost poput na pr. – na početku simulacije, na kojemu će se semaforu prije upaliti zeleno svjetlo? Za vršene simulacije u ovome radu zeleno svjetlo je uvijek prvo bilo upaljeno na lijevoj stani modela odnosno u pravcu istoka. Kada se promatra propusnost cijelog modela, može se jasno uočiti da će ona biti najveća za ekperiment 2.1. gdje su ukupno 223 vozila prošla kroz model. Najmanje vozila kroz cijeli model je prošlo prilikom ekperimenta po brojem 1.3., za dugo zeleno svjetlo i brzinu od 30 km/h. Dijagram 28 grafički prikazuje najpovoljnije rezultate čekanja vozila za sve ekperimente. Najpovoljniji rezultati u pogledu protočnosti cijeloga promatranog segmenta prikazani su grafički na Dijagram 29. Iz rezultata je vidljivo da najmanji zastoj na semaforu neće nužno značiti i veću propunost cijeloga modela.

Tablica 13: Prikaz najpovoljnijih rezultata

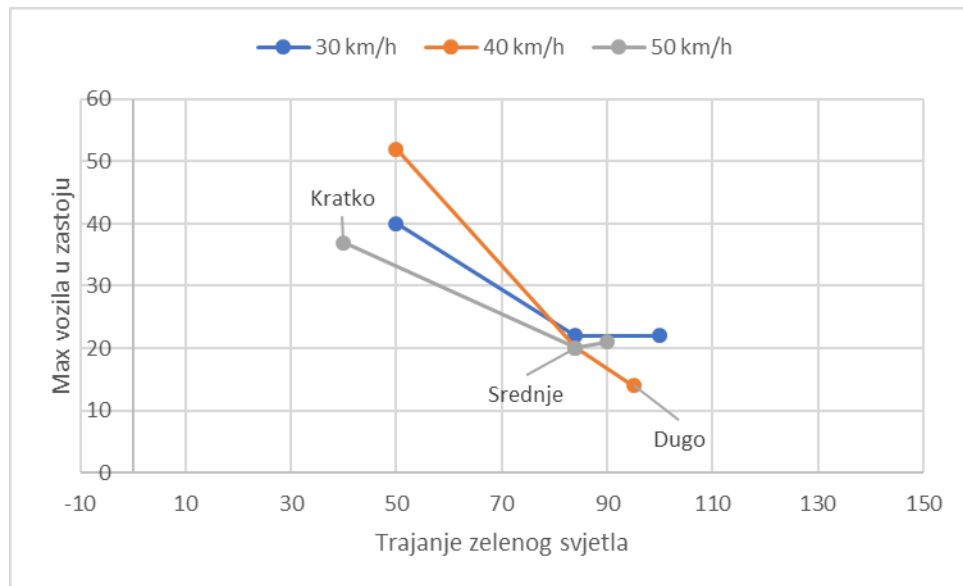
Smijer	Broj vozila u čekanju [min]	Ukupan prolaz vozila [max]
Zapad - Istok	13	120
Istok - Zapad	9	103
Ukupno	14	223

Tablica 14: Redni broj simulacija sa najpovoljnijim parametrima

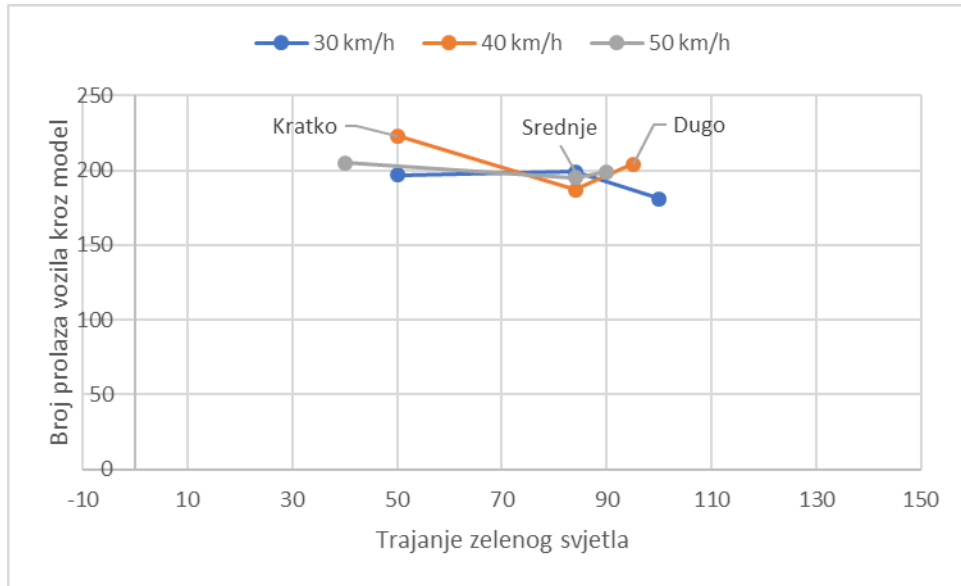
Smijer	Broj vozila u čekanju [redni br. simulacije]	Ukupan prolaz vozila [redni br. simulacije]
Zapad - Istok	1.2	2.1
Istok - Zapad	3.2	2.1
Ukupno	2.3	2.1

Tablica 15: Vrijednosti parametara za najpovoljnije dobivene rezultate

Smijer	Brzina	Trajanje zelenog svjetla	Trajanje žutog svjetla	Crvenog svjetla
Zapad - Istok	6.00	Srednje	50.00	134.00
Istok - Zapad	10.00	Srednje	50.00	150.00
Ukupno	8.00	Dugo	50.00	150.00



Dijagram 28: Broj vozila u zastoju u ovisnosti o trajanju zelenog svjetla



Dijagram 29: Protočnost promatranog segmenta u ovisnosti o trajanju zelenog svjetla

7. ZAKLJUČAK

Problem modeliranja prometa može se sagledati sa više aspekata. Iako se čini da je promet eksplicitan sustav koji se odvija na jasno ograničenom prostoru (cesta) i sa definiranim parametrima koji određuju odvijanje prometa (ograničenje brzine, brzine ubrzanja i kočenja vozila itd...) treba uzeti u obzir da je sastavni dio svakog prometnog sustava sam čovjek čije odluke diktiraju kako će se sustav ponašati. Brojnim pravilnicima pokušano je predvidjeti i ograničiti utjecaj odluke pojedinca u sustavu međutim vrlo je teško postići savršen prometni sustav koji će se ponašati po jasno definiranim pravilima. Fenomen koji je istražen u ovom radu prisutan je u svakodnevom životu i daje dobar uvid u nepredvidljivost takvog sustava. Analiza problema dala je rezultate koji možda nisu bili intuitivni; na pr. moglo bi se pretpostaviti da će se najduže vrijeme trajanja zelenog svjetla i za najveću brzinu vozila propusnost promatranog prometnog pravca biti najveća. Također bi se moglo pretpostaviti da će dugo trajanje zelenog svjetla samo po sebi značiti manji broj automobila u zastoju ali to ovdje nije slučaj. Računalni model izrađen u ovom radu ima mnogo mogućnosti za proširenje modela te bi se ovakva analogija mogla potencijalno koristiti za prikaz i analizu šireg problema prometa. Može se reći da su rezultati eksperimenta promatrani kroz dva pristupa – užeg i šireg. Uži pristup problemu bio bi problem čekanja na semaforu gdje je cilj postići što manje automobila koji stoje u koloni. Širi pristup obuhvaća cijeli model i promatra se kakva je propusnost vozila tijekom određenog vremenskog perioda u modelu. Dva pristupa promatranja imaju i različite rezultate stoga se za odabir optimalnih parametara za promatrani problem treba pristupiti sa jasnim ciljem.

8. LITERATURA

- [1] Uri Wilensky and William Rand An Introduction to Agent-Based Modeling; Modeling Natural, Social, and Engineered Complex Systems with NetLogo; The MIT Press, Cambridge, Massachusetts London, England (2015)
- [2] Castiglione F.: Agent based modeling, http://scholarpedia.org/article/Agent_based_modeling , pristup 20.09.2020.
- [3] Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL
- [4] B.T.: Posljedice prometnih nesreća, <https://revijahak.hr/2018/11/11/zastrasujuce-posljedice-prometnih-nesreca-na-ljudsko-tijelo/> , pristup 01.10.2020.