

JU UNIVERZITET U TUZLI  
PRIRODNO - MATEMATIČKI FAKULTET  
ODSJEK MATEMATIKA  
SMJER TEORIJSKA KOMPJUTERSKA NAUKA

PROJEKAT

MODELIRANJE SAOBRAĆAJA KORIŠTENJEM  
ĆELIJSKIH AUTOMATA

PREDMET: MATEMATIČKO MODELIRANJE I  
SIMULACIJA

Nadir Bašić

MENTOR:

Dr. sc. Samra Moranjkić

TUZLA

05.06.2023

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Ćelijski automati</b>	<b>3</b>
2.1	Formalna definicija . . . . .	7
2.2	Elementarni ćelijski automati . . . . .	8
2.3	Ćelijski automati i izračunljivost . . . . .	8
<b>3</b>	<b>Modeliranje saobraćaja</b>	<b>10</b>
3.1	Kvantitativne karakteristike saobraćaja . . . . .	11
3.2	Pravilo 184 . . . . .	12
3.3	Nagel Schreckenberg (NaSch) model . . . . .	14
<b>4</b>	<b>Zaključak</b>	<b>20</b>
<b>5</b>	<b>Literatura</b>	<b>21</b>

# 1 Uvod

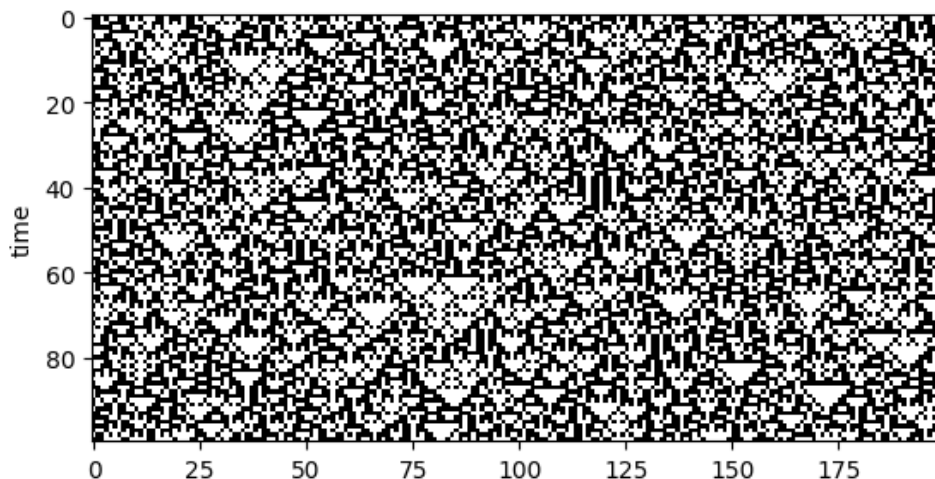
Matematičko modeliranje saobraćaja je vrlo bitno za razumijevanje i rješavanje problema zastoja, i povećavanje *protoka* saobraćaja. Prvi zabilježeni pokušaji modeliranja saobraćaja su se desili 1930-tih godina, i pomogli su pri razotkrivanju zanimljivih fenomena koji se dešavaju u saobraćaju. Matematičko modeliranje se inače koristi diferencijalnim ili diferentnim jednačinama odnosno dinamičkim sistemima za preslikavanje fenomena iz stvarnog svijeta u matematičke modele, te su ti pristupi rodili plodom i pri modeliranju saobraćaja. Međutim, ovakvi klasični pristupi modeliranja sistemima jednačina nisu u mogućnosti obuhvatiti svu kompleksnost koja se javlja iz interakcije mnogih vozila međusobno, i sve posljedice tih interakcija, nego saobraćaj shvataju pojednostavljeno, kao tok fluida (makroskopski kontinualni modeli), ili se fokusiraju na pojedinačna vozila (mikroskopski modeli), što je dobar pristup za modeliranje interakcije 2 vozila, ali teško odražava makroskopske saobraćajne fenomene. Stoga je 1992. godine predložen drugačiji pristup, koji koristi *ćelijske automate*, i uspostavlja diskretni mikroskopski model saobraćaja. Ovakvi modeli su jednostavni za shvatiti, vizualizovati, i računarski implementirati, te su u mogućnosti reproducirati kompleksne saobraćajne fenomene, čak i koristeći najjednostavnije modele, što je jedna od generalnih odlika ćelijskih automata. Ovaj rad će predstaviti jedan takav model, i demonstrirati popratnu biblioteku otvorenog koda za saobraćajne ćelijske automate napisanu u Pythonu<sup>1</sup>.

## 2 Ćelijski automati

Ćelijski automat je kolekcija "obojanih" ćelija na "rešetki" određenog oblika, koje evoluiraju kroz određen broj diskretnih vremenskih koraka, na osnovu susjednih (okolnih) ćelija. Ta pravila se primjenjuju iterativno, do volje.

---

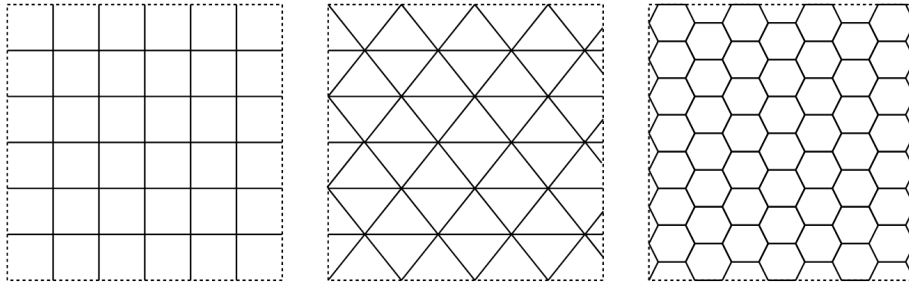
<sup>1</sup>Najnovija verzija biblioteke se može pronaći na <https://github.com/nake-b/traffic-ca>



Slika 1: Sierpinskijev čelijski automat, ili Pravilo 90

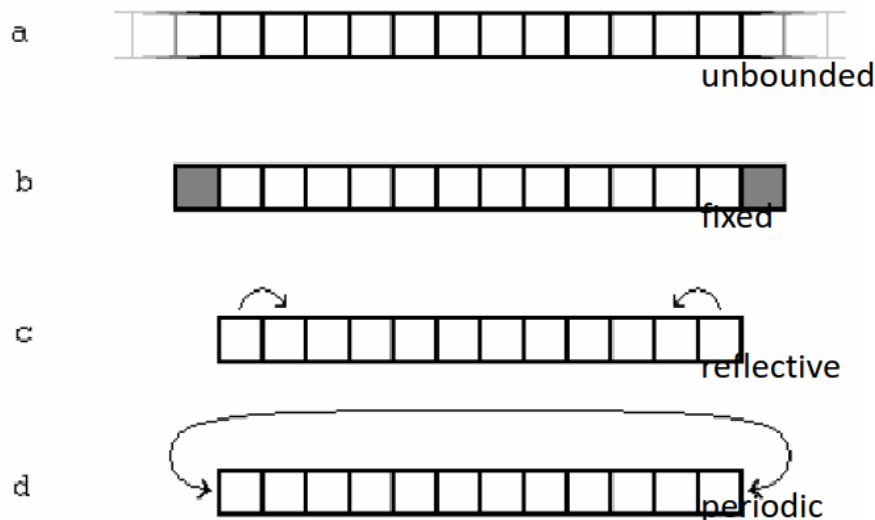
Prvi put se spominju 1948., kada ih je von Neumann iskoristio za proučavanje živih bioloških sistema. Ključno za njegov rad jeste bilo svojstvo samoreprodukcije, koje čelijski automati lako implementiraju. Budućih godina, ovi automati su i dalje proučavani, ali uglavnom čisto iz matematičke rasonode, što je rodilo zanimljivim plodovima. Čelijski automati su se međutim popularizovali 80-tih godina 20-tog vijeka, kroz rad Stephena Wolframa. On je otkrio mnoge fascinantne osobine ovih automata, što je kulminiralo u njegovoj knjizi "Nova vrsta nauke", u kojoj je Wolfram povezao čelijske automate sa praktično svim naučnim disciplinama (sociologija, biologija, fizika, matematika, ...), i pokazali su se vrlo povoljnim za modeliranje kompleksnih sistema, koristeći relativno jednostavna pravila.

Princip rada čelijskih automata je sljedeći: ćelije automata se nalaze u *univerzumu* automata, odnosno diskretnoj "rešetki" (engl. *lattice*) ćelija sa pravougaonom, šestougaonom, trougaonom ili drugom topologijom.



Slika 2: Različite vrste rešetki

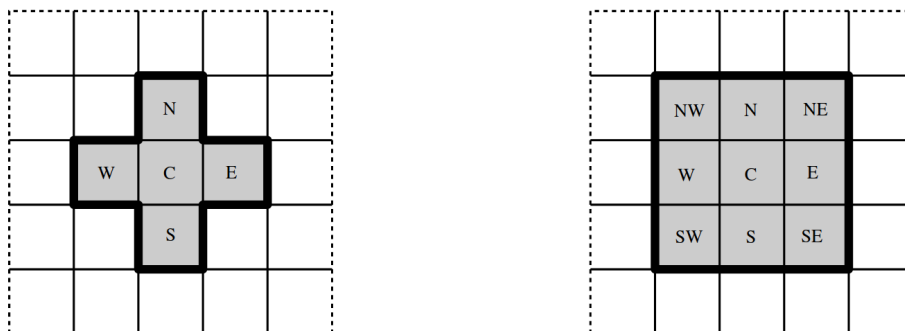
Rešetka može biti ograničena ili beskonačna, i može biti dimenzije 1 ili više. Tipično se koriste pravougaone rešetke, dimenzije jedan (linearne) ili dva (mreža). Za konačne mreže, nekada se uvode i granični uslovi za rub mreže. Čvrsta granica je kada ćelije na rubu imaju fiksno stanje, uglavnom 0. Još jedan popularan uslov jeste periodički, gdje se mreža "zavrće" nazad na početak. U 1D slučaju je to prsten, u 2D slučaju torus, mada su moguće i čudnije topologije kao što je Mobiusova traka ili Kleinova boca. Mi ćemo koristiti jedan poseban granični uslov za naš model, koji je hibrid čvrste granice i sistema redanja vozila.



Slika 3: Različite vrste granica

Boje odnosno stanja ćelija se tipično enkodiraju prirodnim brojevima, ali stanja nisu ograničena na diskretne skupove, mogu poprimiti i realne vrijednosti. Međutim, najčešće se koristi binarni skup stanja; da li je ćelija "živa" ili "mrtva".

Čelija unutar univerzuma čelijskog automata evoluirala na osnovu stanja ćelija u njenoj okolini. Okolina ćelije je iste veličine za svaku ćeliju. Za 1D slučaj, najjednostavnija okolina ćelije  $c$  je sama ćelija  $c$ , te prethodna i sljedeća ćelija na rešetki. U 2D slučaju imamo više izbora, von Neumannova, Mooreova, ili neka druga okolina.



Slika 4: Von Neumannova i Mooreova okolina 2D čelijskog automata

Primjetimo, da kako se dimenzionalnost rešetke povećava, to se broj članova okoline povećava eksponencijalno. Konačno, svaka ćelija nakon svakog diskretnog vremenskog skoka doživljava promjenu diktiranu lokalnim tranzicionim pravilom (funkcijom). Čelije uglavnom evoluiraju istovremeno, i deterministički. Ako tranzicija sadrži stohastične komponente (kao naš saobraćajni model), tada se čelijski automat naziva stohastičkim. Zadnja komponenta čelijskog automata jeste inicijalna konfiguracija, odnosno inicijalna stanja ćelija na mreži.

Vodeća komponenta čelijskih automata jeste funkcija tranzicije. Jasno je da dovoljno komplikovana funkcija tranzicije može prouzrokovati dovoljno kompleksno ponašanje sistema, no čelijski automati su poznati po svojoj jednostavnosti, te se pokazuje da čak i automati sa okolinom veličine 3 ispoljavaju nevjerovatne osobine.

Posmatrajmo Sliku 1 kao vizualizaciju jednog čelijskog automata. Prvi red (time=0) predstavlja inicijalnu konfiguraciju, one ćelije koje su crne su "žive". Ako odemo red niže (time=1), vidimo kako su sve ćelije evoluirale na osnovu prethodnog stanja, i tako rađa kompleksne uzorke. Kako idemo na niže, vidimo kako automat evoluirala kroz vrijeme, i možemo odabrati kada želimo da stane. Još jedan dobar način vizualizacije su i animacije. Čelijske automate možemo posmatrati i kao diskretne dinamičke sisteme, te neki

od njih dostižu stanja ekvilibrijuma i stabilnosti. Međutim, njihovo ispitivanje zahtjeva jednu čitavu novu teoriju te je van opusa ovog rada, i modela koji ćemo proučavati.

## 2.1 Formalna definicija

Definišimo ćelijske automate onako kako je to urađeno u teoriji automata.

Ćelijski automat je diskretni dinamički sistem kojeg možemo predstaviti uređenom četvorkom

$$CA = (\mathcal{L}, \Sigma, \mathcal{N}, \delta)$$

Gdje je  $\mathcal{L}$  diskretna rešetka (lattice) koja predstavlja fizičko okruženje,  $\Sigma$  skup svih mogućih stanja automata, pri čemu svaka  $i$ -ta komponenta (ćelija) rešetke  $\mathcal{L}$  u vremenskom trenutku  $t$  ima stanje  $\sigma_i(t) \in \Sigma$ . Dalje, okolina ćelije  $i$  je parcijalno uređen skup ćelija  $\mathcal{N}_i(t) \in \mathcal{N}$ . Konačno,  $\delta$  je tranziciona funkcija

$$\delta : \Sigma^{|\mathcal{N}|} \longrightarrow \Sigma, \quad \delta : \bigcup_{j \in \mathcal{N}_i(t)} \sigma_j(t) \longmapsto \sigma_i(t+1)$$

koja stanju ćelije  $i$  u vremenskom trenutku  $t+1$  dodjeljuje vrijednost na osnovu vrijednosti njenih susjeda u vremenskom trenutku  $t$ . Ovo preslikavanje se uglavnom zadaje kao tabela pravila, gdje se izlazno stanje zapiše za svaku moguću konfiguraciju okoline ćelije.

Također se može posmatrati i globalna konfiguracija ćelijskog automata

$$\mathcal{C}(t) = \bigcup_{j \in \mathcal{L}} \sigma_j(t)$$

Primjenom pravila  $\delta$  na svaku ćeliju rešetke dobijamo globalu funkciju tranzicije

$$G : \Sigma^{|\mathcal{L}|} \longrightarrow \Sigma^{|\mathcal{L}|}, \quad G : \mathcal{C}(t) \longmapsto \mathcal{C}(t+1)$$

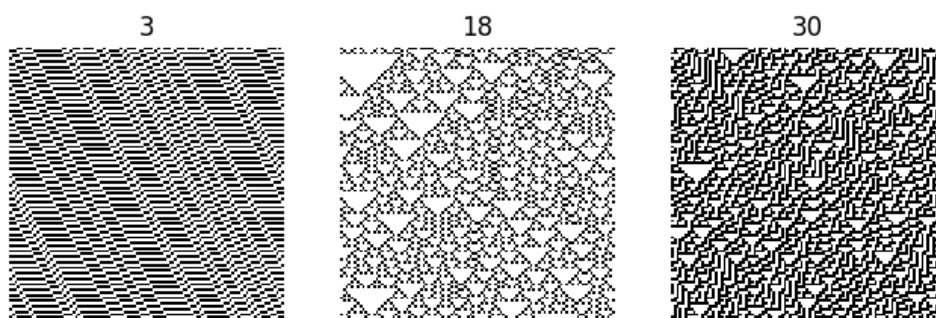
Za ćelijski automat kažemo da je invertibilan ukoliko je  $G$  invertibilna funkcija.

Globalna funkcija  $G$  se iterira od neke početne vrijednosti  $\mathcal{C}(0)$ , što vodi do niza konfiguracija  $\{G^n(\mathcal{C}(0))\}_{n=1}^{\infty}$ , kojeg nazivamo trajektorija ćelijskog automata.

## 2.2 Elementarni ćelijski automati

Elementarnim nazivamo one jednodimenzionalne ćelijske automate sa binarnim stanjima ( $|\Sigma| = 2$ ) i "najbliži susjed" okolinom - prethodna i sljedeća ćelija, zajedno sa ćelijom čiju okolinu posmatramo. Ime im je nadao S. Wolfram 1983. godine, i to su najjednostavniji ćelijski automati koji se mogu pojmiti. U ovakvim automatima, veličina okoline ćelije je  $|\mathcal{N}_i| = 3$ , te svaka ćelija može poprimiti ukupno dva moguća stanja iz skupa  $\Sigma = \{0, 1\}$ . Stoga je broj mogućih okolina ćelije  $2^3$ . Kako ćelijski automat mora specificirati tranziciono pravilo  $\delta : \{0, 1\}^3 \rightarrow \{0, 1\}$  za svaku moguću okolinu i dodijeliti jednu od dvije vrijednosti, zaključujemo da je mogućih elementarnih automata  $2^{2^3} = 256$ . Svaki od njih se indeksira binarnim brojem zvanim *Wolframov kod* koji ga jedinstveno identificira. Wolframov kod dobijamo tako što poredamo sve moguće konfiguracije okolina kao 3-bitne binarne brojeve, počevši od najvećeg (111, 110, ..., 000). Zatim, za svaku trojku  $b$  preračunamo novo stanje središnje ćelije  $\delta(b)$ , i na taj način dobijamo traženi 8-bitni binarni kod. Decimalna reprezentacija ovog koda se zove **pravilo**, i služi kao ime automata. Tako naprimjer automat "Pravilo 184" je najprimitivniji ćelijski model saobraćaja.

Iako ovih automata ima 256, pokazuje se da su mnogi od njih ekvivalentni do na jednostavnu geometrijsku transformaciju, kao što je refleksija oko ose ili komplementarnost (zamjena stanja 1 i 0). Ukupno jedinstvenih automata pod ovim transformacijama je 88.



Slika 5: Primjeri elementarnih ćelijskih automata

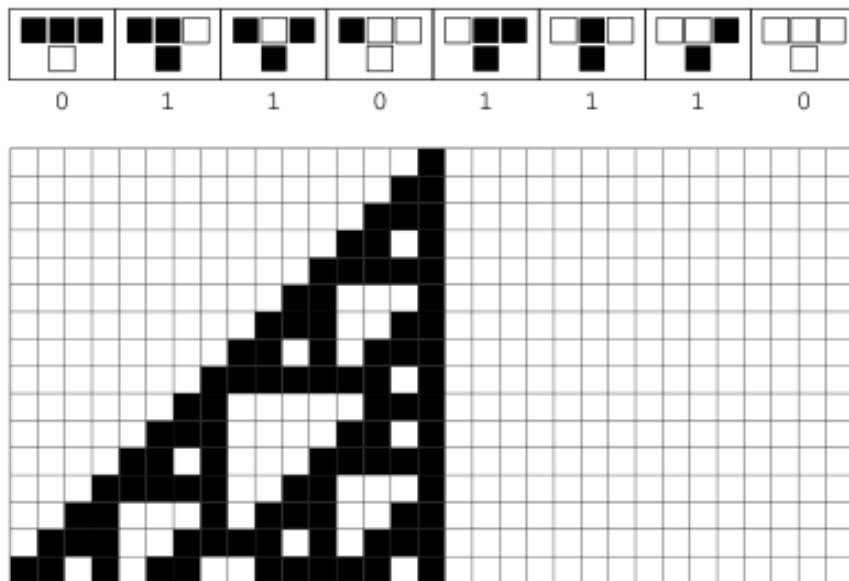
## 2.3 Ćelijski automati i izračunljivost

Kao što smo vidjeli u prethodnim primjerima, čak i naizgled jednostavni ćelijski automati mogu rezultovati veoma kompleksnim uzorcima, što ih čini vrlo povoljnim za



modeliranje. Postavlja se pitanje, koliko su ovi automati zapravo moćni, i koja su njihova ograničenja? Čelijski automati su originalno izmišljeni kao modeli kompjacije, koje proučava teorija automata, odnosno teorija kompleksnosti i izračunljivosti. Ova grana teorijske kompjuterske nauke se bavi pitanjima kompjacije i izračunljivosti, odnosno problemima koje je moguće riješiti nekim algoritmom. Ova teorija definiše *izračunljive* probleme (jezike) kao one koji se mogu odlučiti koristeći Turingovu mašinu. Turingova mašina jeste automat koji se sastoji od neograničene trake jedinica i nula, i glave koja može čitati i pisati jedinice i nule. Svaka Turingova mašina ima i funkciju tranzicije, gdje glava na osnovu pročitane vrijednosti i pozicije na traci, mijenja poziciju i opcionalno piše neku vrijednost. Ovaj model kompjacije se pokazao toliko univerzalnim, da su po njemu izračunljivi jezici i dobili ime. Jednostavnije rečeno, ako imamo neki izračunljiv problem (što je vrlo široka klasa problema), i dovoljno talentovanog programera koji programira Turingovu mašinu (kompjuter sa dovoljno memorije), taj problem se može riješiti unutar ograničenog vremenskog intervala. Ako zamislimo čelijske automate kao kompjacione modele, vrlo lako vidimo da su neki od njih ekvivalentni Turingovoj mašini. Uzmimo jednodimenzionalni binarni čelijski automat, sa neograničenom mrežom, i neograničenom okolinom svake ćelije koja obuhvata čitavu mrežu. Aktivne ćelije su one koje imaju 1 na Turingovoj mašini, a neaktivne koje imaju 0. Svaka promjena na traci (pisanje glave) odgovara promjeni ćelije. Odnosno, ovakav čelijski automat je Turingova mašina sa neograničeno glavom, gdje je svaka glava na svakom polju trake. Međutim, ovakav čelijski automat bi bio vrlo komplikovan, što anulira njihovu korisnost. Pitanje je da li jednostavni čelijski automati mogu biti Turing kompletni, odnosno implementirati Turingovu mašinu? Pokazuje se da zapravo postoje vrlo jednostavni čelijski automati sa ovom osobinom. Jedan od najpopularnijih čelijskih automata jeste John Conway-ova Igra života (engl. *Game of Life*), što je 2D čelijski automat koji proizvodi ponavljajuće strukture sa nevjerovatnim osobinama. Postoje takozvani "glideri", skupovi ćelija koji se pomiču po mreži, koje možemo shvatiti kao prenosnike signala. Također, određeni elementi mogu reagovati na 2 glidera, na isti način kao što logička kola reaguju na električne signale, što znači da je game of life sposoban implementirati kompjuter. Ova osobina je

čak dokazana i za jedan elementarni ćelijski automat, Rule 110.



Slika 6: Pravilo 110

Iako filozofske prirode, zaključak je da čak i jednostavni ćelijski automati mogu modelirati sve što je moguće modelirati. U ovome i leži pojašnjenje o tome kako je moguće simulirati tako kompleksan sistem kao što je saobraćaj, sa samo nekoliko pravila.

## 3 Modeliranje saobraćaja

Naš cilj je da na što bolji način (bolji u odnosu na tradicionalne metode) matematički modeliramo cestovni saobraćaj, i da budemo u stanju isti simulirati. Jedna od najvećih prepreka jeste stohastička priroda saobraćaja, te što vozilima upravljaju ljudi, čije je ponašanje često nepredvidivo, ili neprirodno. Dobar primjer ovoga jeste Braessov paradoks. Ovaj paradoks je primjećen kada se u gusto naseljenim područjima sa velikim protokom saobraćaja zatvore neke bitne saobraćajnice. Primjećeno je da se često u tim uslovima protok saobraćaja povećava, što prkosi zdravoj logici. Međutim, Braessova ideja jeste da vozači (ljudi) prave odluke optimalne za njih same, odnosno djeluju prema svom interesu. To znači da se prečice mogu birati previše često, od strane svakog vozača koji vidi mogućnost, što zna proizvesti zagušenje. Odnosno matematičkim rječnikom, Nashov ekvilibrij (pojam iz teorije igara, stanje u kojem svi "igrači" donose optimalne odluke za

sebe) ne odgovara optimalnom protoku saobraćaja.

Ovo je jedan ekstreman slučaj, te izvan granica ovog rada i kursa, međutim, ćelijski automati zapanjujuće dobro modeliraju "standardno" ponašanje vozača, i to koristeći vrlo jednostavna pravila.

Kao što smo već spomenuli, modeli saobraćaja se generalo dijele na **makroskopske** i **mikroskopske**, a postoje i **mezoskopski** koji su kombinacija prethodna dva.

Makroskopski modeli se ne fokusiraju na individualna vozila nego na globalne karakteristike saobraćaja kao što su protok, brzina, gustina i slično. Jedan pristup jeste da se saobraćaj posmatra kao dinamika fluida, te je pokazano da se gust saobraćaj može porediti sa kretanjem vode u rijeci za vrijeme poplave.

Mikroskopski modeli s druge strane posmatraju pojedinačna vozila, i njihovu dinamiku koju određuju pozicija i brzina auta. Godinama je smatrano da je mikroskopsko modeliranje saobraćaja kompleksan, dug, mukotrpan, i kompjutaciono skup proces. Međutim, 90-tih godina su se pojavili modeli saobraćaja koristeći ćelijske automate. Ovi modeli su bili lagani za implementirati, simulirati, i upravljati. Na prvi pogled može izgledati kao da ovi modeli nisu dovoljno dobri jer su diskretni a ne kontinualni, i kao mikroskopski nisu u stanju detaljno modelirati pojedinačna vozila. Međutim, to im nije ni cilj, nego ćelijski automati kroz mikroskopske interakcije očituju makroskopske saobraćajne fenomene iz stvarnog svijeta.

### 3.1 Kvantitativne karakteristike saobraćaja

Kako bismo mogli uporediti karakteristike našeg modela sa karakteristikama realnog saobraćaja, moramo uvesti neke veličine kojee koriste pri proučavanju saobraćaja.

Brzinu pojedinačnih vozila je teško izmjeriti, zato se inače koristi *srednja brzina u vremenu*, koja se mjeri tako što se postave detektori brzine na fiksnu tačku kolovoza, i mjere brzine auta koje prolaze u nekom vremenskom intervalu. Srednja brzina se zatim računa kao

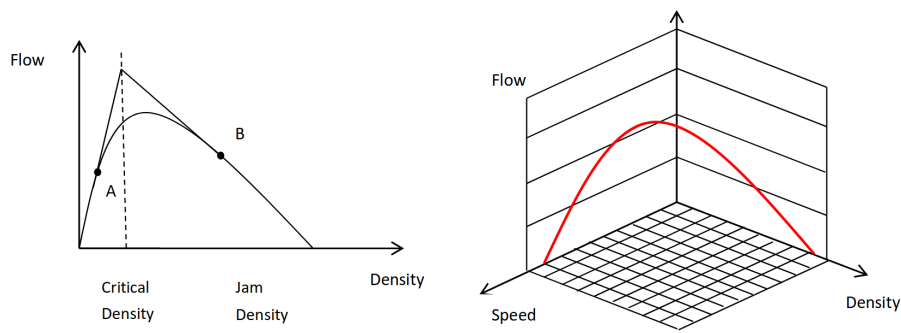
$$V = \frac{1}{N} \sum_{i=1}^N v_i$$

Gdje je  $N$  broj vozila koji je prošao kroz tačku, a  $v_i$  brzina svakog pojedinačnog vozila. Gustina saobraćaja  $\rho$  se mjeri kao broj vozila po jedinici kolovoza. Postoje dvije važne karakteristične vrijednosti gustoće saobraćaja, kritična gustoća  $\rho_c$  koja predstavlja maksimalnu gustoću saobraćaja za neometan tok, i gustoća u gužvi  $\rho_g$ .

Najbitnija karakteristika saobraćaja koju u praksi želimo da maksimizujemo jeste tok ili protok saobraćaja  $Q$ , koja predstavlja broj vozila koji prolaze kroz neku fiksnu tačku unutar nekog vremenskog intervala. Generalno, veza između ove tri veličine glasi:

$$Q = V \cdot \rho$$

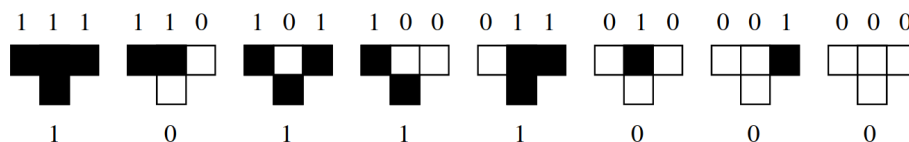
Što možemo vidjeti na Slici 7, zajedno sa kritičnom gustoćom  $\rho_c$



Slika 7: Odnosi gustoće, protoka i brzine saobraćaja

## 3.2 Pravilo 184

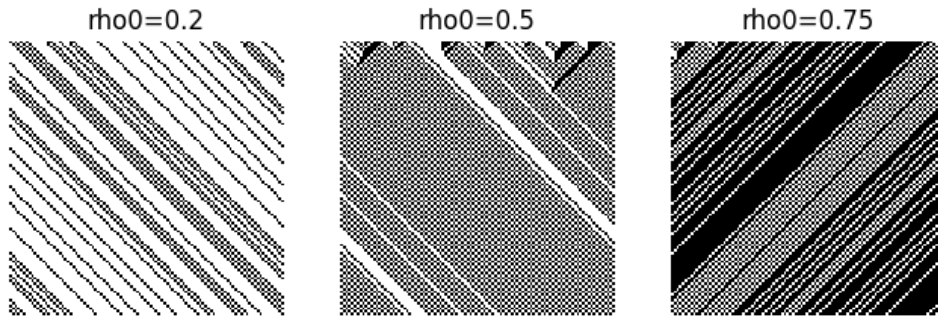
Prije našeg modela, osvrnimo se na najjednostavniji mogući model saobraćaja, koji se koristi elementarnim ćelijskim automatom Pravilo 184. Kako je  $(184)_{10} = (10111000)_2$ , tranziciono pravilo za ovaj automat vidimo na Slici 8 Neka je mreža automata dužine  $L$ .



Slika 8: Odnosi gustoće, protoka i brzine saobraćaja

Početnu konfiguraciju ćelija biramo na osnovu početne gustine automata  $\rho_0 = \frac{N_0}{L}$ . Za

ovaj primjer biramo periodičnu granicu. Na Slici 9 vidimo evoluciju automata za različite vrijednosti početnih gustoća.



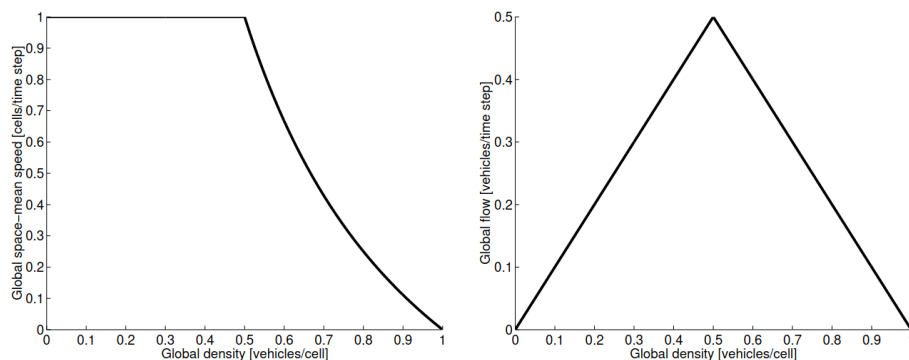
Slika 9: Odnosi gustoće, protoka i brzine saobraćaja

Primjetimo da za veće gustoće već vidimo realistično pojavljivanje stanja kongestije (neprekidan niz crnih ćelija na slici), i to koristeći jedan elementaran automat! Proučavajući tabelu pravila, vidimo da je  $\delta(100) = 1$ , što znači da će se vozilo koje nema vozila ispred sebe pomaknuti za jedno mjesto ispred (kao i za stanje 101!). S druge strane,  $\delta(110) = 0$ , što predstavlja vrijeme reakcije vozača nakon što vozač ispred njega krene, u ovom slučaju jedan diskretni vremenski korak. Označimo poziciju  $i$ -tog po redu vozila (upaljene ćelije koja se "miče") u vremenskom trenutku  $t$  sa  $x_i(t)$ , a brzinu sa  $v_i(t)$ . Iz ostatka tablice funkcije tranzicije vidimo da ponašanje ovog automata možemo opisati koristeći jednostavno pravilo:

$$x_i(t+1) = x_i(t) + \min\{1, x_{i+1}(t) - x_i(t) - 1\}$$

Ili jednostavnije, *"Ako postoji slobodno mjesto ispred, pomaknuti se, inače ostati na svom mjestu."*

Pored stanja kongestije, ovaj model očituje postojanje kritične gustoće  $\rho_c = 0.5$ , što vidimo iz Slike 10.



Slika 10: Odnosi gustoće, brzine i protoka za Pravilo 184

### 3.3 Nagel Schreckenberg (NaSch) model

1992. godine, Nagel i Schreckenberg su predložiti svoj stohastički model ćelijskih automata. Iako je ovo vrlo jednostavan model sa samo 4 pravila, on očitava osobinu spontane pojave zagušenja, i vrlo realistično ponašanje. Slično prethodnom primjeru, funkciju tranzicije nećemo zadati tabelarno niti za sve moguće konfiguracije mreže, nego kao pravila koja poštuju pojedinačna vozila. To nam omogućava lakše razumijevanje i implementaciju, uz iste rezultate. Specifično, ovaj projekat se bavi NaSch modelom sa otvorenom granicom i sistemom redanja (engl *queing*) vozila.

Kao u Pravilu 184, naš univerzum je jednodimenzionalna rešetka dužine  $L$ , ćelije mogu biti u dva stanja (vozilo jeste ili nije na lokaciji ćelije), i funkcija tranzicije ima stohastičku komponentu. Ovaj model generalizuje Pravilo 184 tako što uvodi parametar maksimalne brzine vozila  $v_{max}$ , gdje se pod brzinom misli na broj ćelija koje se pređu za jedan vremenski trenutak. Ako je trenutno u mogućnosti, svako auto ubrzava sa konstantnim ubrzanjem od jedne jedinice brzine. U prethodnom modelu je bilo  $v_{max} = 1$ , dok se za NaSch uglavnom bira  $v_{max} = 5$ . Ako auto nema dovoljno ćelija ispred sebe, usporit će na odgovarajuću brzinu kako ne bi izazvalo nesreću. Okolina ćelije je praktično određena maksimalnom brzinom, međutim, inače se formalno ne definiše, nego se tranzicija dešava na osnovu vozila koja dolaze jedan iza drugog i na osnovu zadanih pravila. Stohastička komponenta ovog modela jeste vjerovatnoća usporenja  $p$ , koja određuje da li će vozilo u trenutnom vremenskom koraku usporiti za jednu jedinicu. Ovaj parametar je ključan, i

odgovoran za realistično ponašanje sistema, i odgovara prekomjernoj reakciji vozača pri kočenju ili kočenju zbog drugih ljudskih faktora. Podaci iz stvarnog svijeta pokazuju da je ta vjerovatnoća na autocestama  $p = 0.3$ , a u gradovima  $p = 0.5$ . Također, moguće je da se u vremenskom trenutku  $t$  desi nesreća na ćeliji  $i$ , koja će trajati  $T$  vremenskih jedinica. Početna konfiguracija je diktirana početnom gustoćom  $\rho_0 = \frac{N_0}{L}$ , i bira se nasumično. Cesta je spojena na neki drugi put, i vjerovatnoća da će se novo auto uključiti na cestu jeste  $q$ . Rezimirajmo sve ove činjenice, i konstruišimo naš model.

Parametri modela su:

- $L$  - dužina ceste (rešetke)
- $v_{max}$  - maksimalna brzina svih vozila
- $p$  - vjerovatnoća usporavanja
- $q$  - vjerovatnoća uključenja auta na cestu
- $x_A$  - pozicija nesreće (opcionalno)
- $t_A$  - trenutak kada se nesreća desila (opcionalno)
- $T_A$  - trajanje nesreće (opcionalno)
- $\rho_0$  - početna gustoća saobraćaja

Početnu konfiguraciju rešetke biramo tako da ona sadrži  $N_0 = \lfloor \rho_0 \cdot L \rfloor$  živih ćelija, čije su pozicije izabrane nasumično. Svaka od živih ćelija predstavlja jedno vozilo, čiju poziciju označavamo sa  $x_i(t)$ , a brzinu sa  $v_i(t)$ . Na početku, svako auto ima nasumičnu cjelobrojnu brzinu između 0 i  $v_{max}$ .

U svakoj iteraciji, svako vozilo  $i$  poštuje sljedeća pravila:

**Ubrzanje** Ako je brzina vozila  $i$  manja od  $v_{max}$ , vozilo se ubrzava za jedan. Formalno,

$$v_i \longrightarrow \min\{v_{max}, v_i + 1\}$$

**Usporavanje** Vozilo će smanjiti svoju brzinu ako praznina ispred njega nije dovoljno velika.

$$v_i \longrightarrow \min\{v_i, x_{i+1} - x_i - 1\}$$

**Randomizacija** Bira se slučajan broj  $\xi_i(t) \in [0, 1]$ , uzorkovan iz uniformne raspodjele, i upoređuje

sa  $p$ . Ukoliko je  $\xi_i(t) > p$ , tada će vozilo smanjiti brzinu za 1 (ukoliko se miče).

$v_i \longrightarrow \max\{0, v_i - 1\}$ , sa vjerovatnoćom  $p$

**Kretanje** Vozilo se miče  $v_i$  ćelija unaprijed.

$$x_i \longrightarrow x_i + v_i$$

Pravilo **usporavanje** opciono može uključivati i čekanje saobraćajne nesreće. Granični uslovi koji biramo jesu otvorena granica zdesna, i redanje vozila slijeva. Odnosno, kada auto dođe do kraja ceste, ono je napusti, i više ne postoji u modelu. Svakog vremenskog trenutka, novo auto se priključuje na cestu sa vjerovatnoćom  $q$ . Ukoliko je početak ceste blokirano, formira se red auta koji žele da uđu na cestu. Sa svakim novim pojavljivanjem auta (s vjerovatnoćom  $q$ ), dodaje se novo auto na red čekanja.

Konačno, simulirajmo nekoliko instanci ovog modela koristeći popratnu Python biblioteku *traffic-ca*. Dio koda automata koji je zadužen za implementaciju navedenih pravila je:

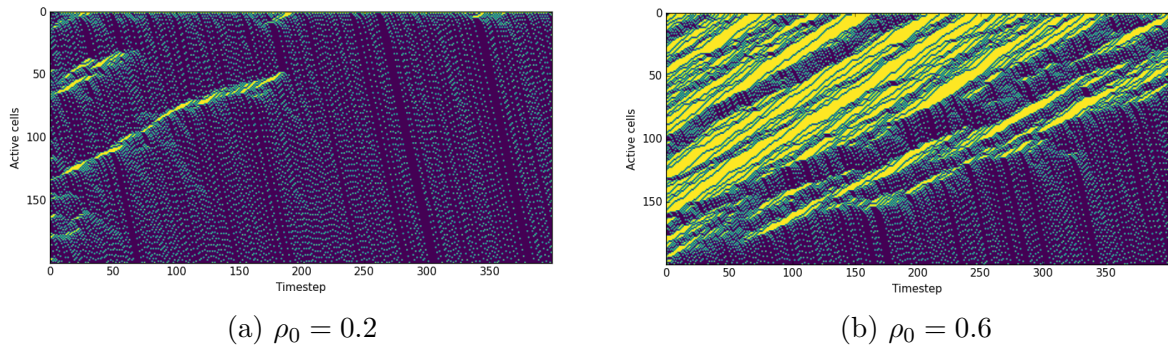
```
def __update_car(self, car: Car, next_car: Optional[Car]):  
    # Acceleration  
    car.accelerate()  
    # Deceleration  
    reaction_objects = [next_car]  
    if self.accident is not None and self.accident.occuring:  
        reaction_objects.append(self.accident)  
    for _object in reaction_objects:  
        car.react(_object)  
    # Randomization  
    if random_bool(self.slowdown_probability):  
        car.decelerate()  
    # Move  
    car.move()  
    if car.position >= self.highway_len:  
        self.__cars.remove(car)
```



Globalni parametri za sve modele, koje nećemo mijenjati su:

- $L = 200$
- $v_{max} = 5$
- $q = 0.75$

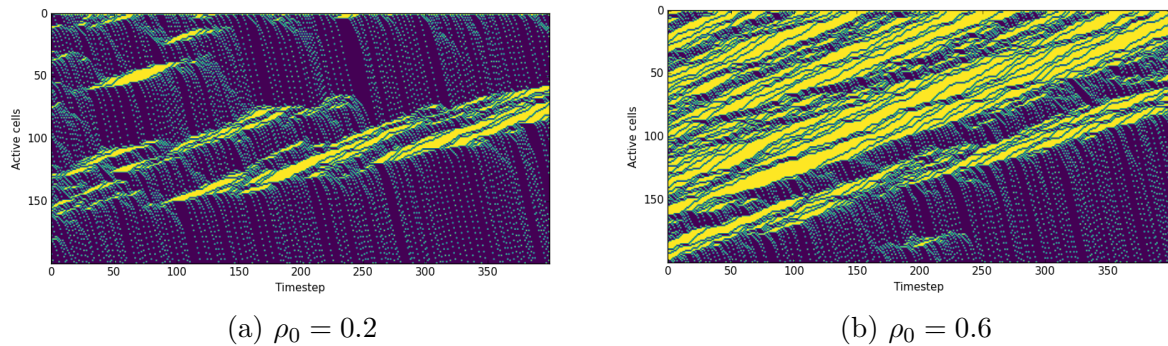
Za početak, posmatrajmo modele bez saobraćajnih nezgoda, sa varirajućim parametrima  $\rho_0$  i  $p$ , koji su intuitivno najodgovorniji za nepoželjno stanje kongestije. Najprije posmatrajmo prostorno-vremenske grafike modela autoputa na Slici 11.



Slika 11: Simulacija autoputa ( $p = 0.3$ )

Zagušenje saobraćaja na grafiku vidimo kao mase žutih ćelija koje se "kreću" prema gornjem desnom ćošku, što znači da se kongestija propagira unazad u prostoru. Na autoputima rijetko dolazi do spontane kongestije (koja nije uzrokovana početnom gustoćom), ali kada se desi, puno brže prolazi u uslovima sa manjom gustoćom. Ali ako je početna gustoća velika, gužve postaju katastrofalne i propagiraju sve do početka autoputa.

Posmatrajmo sada simulaciju gradskog saobraćaja na Slici 12.



Slika 12: Simulacija gradskog saobraćaja ( $p = 0.5$ )

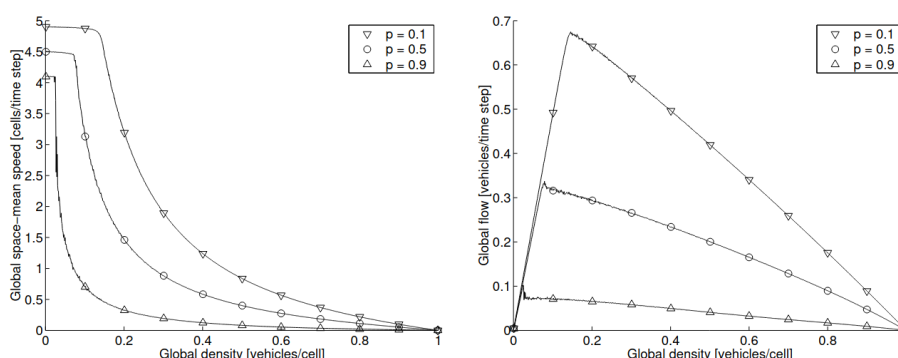
Čak i za male gustoće, spontane kongestije se često dešavaju, i znaju imati katastrofalne posljedice, bez očitog uzroka. Dok su za veću gustoću gužve ogromne, i miču se sporije nego kao na autoputu.

Za NaSch model sa periodičnom granicom, eksperimentalno se pokazuje, da kongestije koje se pojave u stanju gustoće manje od kritične traju konačno dugo, i neovisno o  $L$ . Ali ako gustoća pređe kritičnu, gužve traju beskonačno. Naš model je bolji iz razloga što ima realistične granice, te gužve ne traju beskonačno nego se razrješavaju na početku autoputa. Iako se može činiti da je ovo jednostavno rješenje za gužve, ovo uzrokuje velikim redom čekanja za ulazak na autoput.

Iz ovih modela, o parametrima  $p$  i  $\rho_0$  zaključujemo dvije stvari:

1. Često usporavanje vodi do spontanih kongestija, koje se mogu izbjeći
2. Kritična gustoća prouzrokuje kongestije koje je "nemoguće" popraviti.

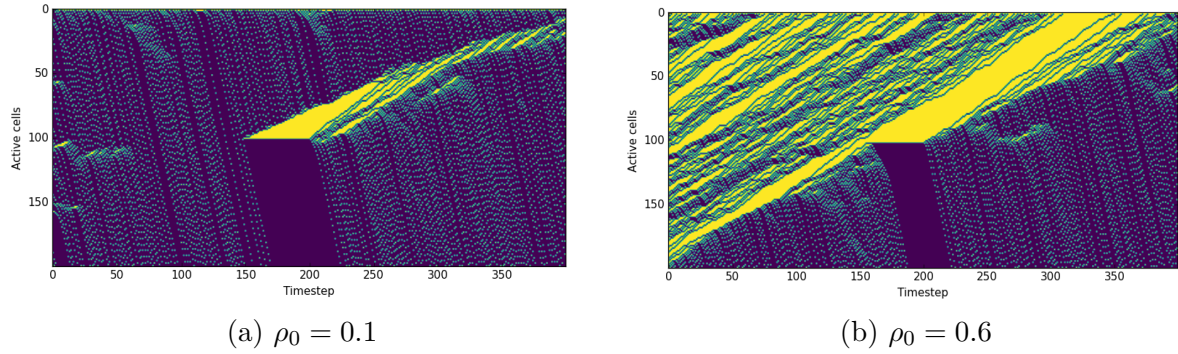
Odnose brzine, gustoće, i protoka za različite vrijednosti  $p$  vidimo na Slici 13.



Slika 13: Odnosi gustoće, brzine i protoka za NaSch model

I sa ovog grafika vidimo posljedice parametara  $p$  i  $\rho_0$ . Jedan od načina da interpretiramo parametar  $p$  jeste loše stanje puteva, što prouzrokuje da vozači često koče. Međutim, vidimo da čak i kada se spontane kongestije dese, u stanjima manje gustoće se brzo razriješe zbog slobodnog protoka vozila.

Posmatrajmo sada kako model reaguje na saobraćajnu nesreću, odnosno umjetni zastoj na nekoj koordinati, koji traje određen vremenski period. Izaberimo  $x_A = 300$ ,  $t_A = 100$ ,  $T_A = 150$ ,  $p = 0.3$ , te dvije različite vrijednosti  $\rho_0 = 0.1$  i  $\rho_0 = 0.6$ , na Slici 14. I ovdje

Slika 14: Simulacija autoputa sa nesrećom ( $p = 0.3$ )

vidimo da se velike kongestije ipak razrješavaju slobodnim protokom vozila. Primjetimo da je parametar  $q$  isti za oba modela, i ima relativno visoku vrijednost. Što znači da je broj vozila koja nastupaju na autoput isti u oba modela, jedina razlika je početna gustoća. Interpretacija ove činjenice jeste; ako moramo da blokiramo važnu saobraćajnicu, bolje je to učiniti u uslovima minimalne **trenutne** gustoće, bez obzira na to koliko će se auta uključiti na saobraćajnicu.

# 4 Zaključak

Cilj ovog projekta je bio dvostran:

1. Uvesti ćelijske automate kao novi način matematičkog modeliranja i simulacije, te pojasniti njihove prednosti i moć
2. Uspostaviti model saobraćaja koristeći ćelijske automate, koji je "bolji" od tradicionalnih modela sa diferencijalnim ili diferentnim jednačinama

Pokazali smo kako su ćelijski automati fascinantni jer prouzrokuju kompleksne uzorke iz jednostavnih pravila. Neki od njih, čak i elementarni, imaju sposobnost univerzalne kompjutacije. Relativno su novi u svijetu modeliranja, ali su našli primjenu u praktično svim granama nauke.

Konačno, koristeći stohastičke ćelijske automate smo uspostavili vrlo realističan model saobraćaja u jednoj traci. Taj model očituje realistična ponašanja kao što su spontane kongestije, gušći saobraćaj u gradu, te postojanje kritične gustoće. Upoređujući različite modele i simulirajući ih, mogli smo izvesti zaključke o pravom svijetu. Ali ipak, ovaj naš model ima pretpostavku o samo jednoj saobraćajnoj traci, što inače nije slučaj u pravom svijetu. Postavlja se pitanje, da li on zbog toga gubi na vrijednosti? Kao prvo napomenimo da se NaSch lagano generalizuje na model sa dvije trake, proširivajući dimenziju rešetke i dodavajući pravilo o preticanju. Također, mnoge Evropske države imaju zakon koji kaže da se samo desna saobraćajnica koristi za vožnju, a lijeva isključivo za preticanje. Eksperimenti pokazuju da često stanje desne trake odgovara stanju saobraćajnice sa samo jednom trakom. Isto tako, ako u desnoj traci postoje vozila koja se sporo kreću (kamione), tada lijeva traka pokazuje osobine saobraćajnice sa jednom trakom. Dakle, ovaj model jeste relevantan, i uspio je očitovati osobine koje tradicionalni modeli ne očituju.

---

## 5 Literatura

1. Ding Ding (2011). *Modeling and simulation of highway traffic using a cellular automaton approach* U.U.D.M. Project Report 2011:25, Department of Mathematics, Uppsala University.
2. Sven Maerivoet, Bart De Moor (2005). *Cellular Automata Models of Road Traffic*. arXiv:physics/0509082 [physics.soc-ph], Physics Reports, vol. 419, nr. 1, pages 1-64, November 2005
3. Rolf Pfeifer Rudolf M. Fuchslin (2013). *Cellular Automata - Dynamical Systems*. Zurich University of Applied Sciences, School of Engineering, ZAMP Center for Applied Mathematics and Physics [Internet] Dospuno na [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiipsHU\\_aH\\_AhUJ\\_bsIHcPUA4EQFnoECBQQAQ&url=https%3A%2F%2Fwww.ifi.uzh.ch%2Fdam%2Fjcr%3A000000000-2826-155d-ffff-ffffafe06077%2FLecture\\_7\\_CA\\_DynSys\\_2013.pdf&usg=A0vVaw2FTpjZYpECqcKA16yxIgEp](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwiipsHU_aH_AhUJ_bsIHcPUA4EQFnoECBQQAQ&url=https%3A%2F%2Fwww.ifi.uzh.ch%2Fdam%2Fjcr%3A000000000-2826-155d-ffff-ffffafe06077%2FLecture_7_CA_DynSys_2013.pdf&usg=A0vVaw2FTpjZYpECqcKA16yxIgEp) [pristupljeno 1.juni 2023]
4. Wolfram Mathworld, *Cellular Automation* [Internet] Dostupno na: <https://mathworld.wolfram.com/CellularAutomaton.html> [pristupljeno 1.juni 2023]
5. Wikipedia, *Elementary Cellular Automata*, [Internet] Dostupno na: [https://en.wikipedia.org/wiki/Elementary\\_cellular\\_automaton](https://en.wikipedia.org/wiki/Elementary_cellular_automaton) [pristupljeno 1.juni 2023]
6. Wikipedia, *Rule 184*, [Internet] Dostupno na: [https://en.wikipedia.org/wiki/Rule\\_184](https://en.wikipedia.org/wiki/Rule_184) [pristupljeno 1.juni 2023]
7. Wikipedia, *Traffic Model*, [Internet] Dostupno na: [https://en.wikipedia.org/wiki/Traffic\\_model](https://en.wikipedia.org/wiki/Traffic_model) [pristupljeno 1.juni 2023]
8. Wikipedia, *Braess's paradox*, [Internet] Dostupno na: [https://en.wikipedia.org/wiki/Braess%27s\\_paradox](https://en.wikipedia.org/wiki/Braess%27s_paradox) [pristupljeno 1.juni 2023]
9. Antunes, L. M. (2021). CellPyLib: A Python Library for working with Cellular Automata. Journal of Open Source Software, 6(67), 3608.