

Matematički fakultet  
Univerzitet u Beogradu

# Fazi kontroler za rad dinamičkog semafora

*Seminarski rad u okviru kursa  
Računarska Inteligencija*

Boris Karanović  
500/2017  
Dimitrije Sekulic  
500/2017

# Apstrakt

Sa povećanjem automobila u velikim gradovima, gužve u gradu su sve učestalije. To postaje veliki problem u svim gradovima širom sveta. Efektivno rešenje ovog problema je da se napravi sistem koji bi optimizovao kontrolu semafora nadgledajući saobraćaj u raskrsnicama.

Kod klasičnog semafora, svetla na semaforu se smenjuju u konstantnom vremenu kreirajući ciklus smenjivanja svetla na semaforima. Ovakvi sistemi ne pružaju optimalnu kontrolu nad raskrsnicom.

Rešenje zasnovano na fazi logici daje optimalnu kontrolu nad semaforima u raskrsnici, zavisno od trenutne gužve u raskrsnici. Cilj je da se poveća protok vozila i smanji čekanje u koloni. Zasnovano na Fazi Pravilima sistem odlučuje da li je potrebno da produži trajanje zelenog svetla.

Sistem je implementiran korišćenjem jezika C++ i alata Qt5 u kome je urađena cela fazi logika i napravljena je simulacija semafora na nekoliko tipova raskrsnica. U simulaciji se može videti i klasični statički tip semafora i videti jedan na jedan poređenje rada dva tipa semafora. Takođe se može videti neke statistike i metrike rada semafora.

## Uvod

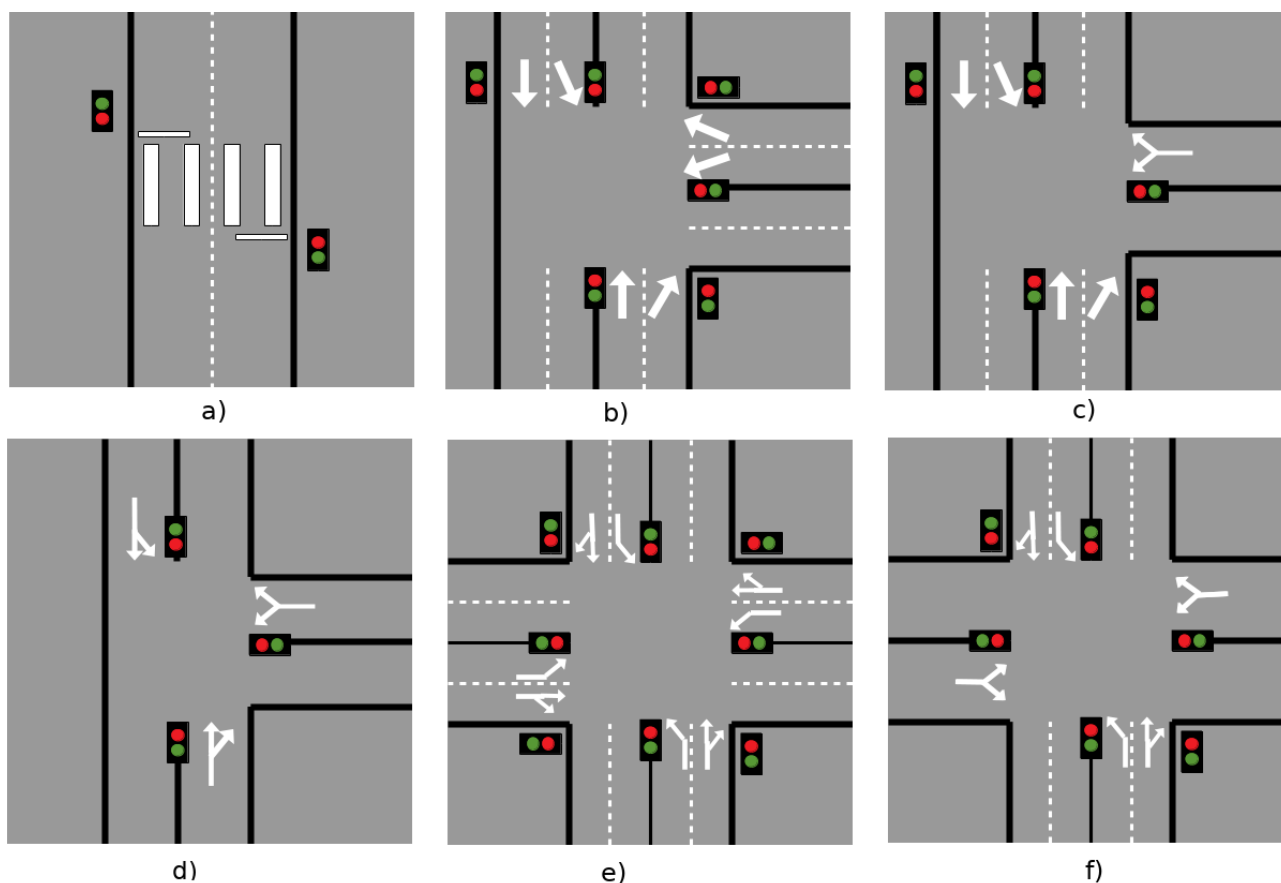
Kontrola saobraćaja je postao problem u mnogim zemljama. Sa konstantnim porastom broja vozila na ulicama moralo se osmisliti bolji način za rešavanje problema. Pored izgradnja novih puteva, proširivanja ulica, izgradnja mosteva, obilaznica kao i uvođenjem žutih traka, gužve u gradu kao da ne mogu da se rasterete. Gužve u saobraćaju se ne javljaju više samo u "špicu" i samo u centru grada već se rasprostiru tokom celog dana i na periferijama grada. U Beogradu se svake godine broj novo registrovanih vozila poveća za 10%, prema zavoda za statistiku Republike Srbije.

Kontrola saobraćaja nad opterećenim raskrsnicama je veliki problem. Glavni cilj je da se napravi sistem koji će da poveća broj vozila koji prođu kroz raskrsnicu i smanji čekanje na semaforima. Ovakvo poboljšanje bi imalo dosta uticaja na sve ljude koji žive u gradu i poboljšalo bi efikasnost saobraćaja. Standardni način rada semafora nije dobro rešenje, jer se gužve u gradovima javljaju. Promene u saobraćaju, kao što su zatvaranje određenih ulica, ne bi uticale na rad semafora. One su zanovalne samo na određenom ciklusu unapred zadatom fiksnih intervala bez ikakvog uticaja stvarne situacije saobraćaja. Ovakav sistem ne razmatra situaciju ako je jedna traka puna da li treba da produži zeleno svetlo njima.

U ovom radu rešavamo opisan problem implementacijom pametnog semafora korišćenjem fazi logike, tehnologija koja omogućava oponašanje kako čovek razmišlja. Na primer, čovek bi razmišljao na sledeći način: "Ako je saobraćaj gušći u nekoj traci u odnosu na ostale onde bi trebalo produžiti semafor u traci gde ima više vozila". Ovo je sličan način kao kada u raskrsnici stoji saobraćajni policajac koji diriguje koja traka je trenutno akrična, a koja stoji.

Fazi logikom možemo da zadajemo lingvističke promenljive pomoću kojih dalje zadajemo pravila. Skup ovih promenljivih i pravila čini mozak pametnog semafora, na osnovu njih možemo dobiti izlaz i odlučiti da li treba da zeleno svetlo traje duže.

U ovom radu razmatrano je šest tipova raskrsnica i poređeno je paralelno pametni semafor sa standardnim semaforom. Svih šest tipova možete videti na Slici 1. Sa ovim tipovima raskrsnica pokušali smo da ilustrujemo rad semafora na više slučajeva. U daljem tekstu opisaćemo ukratko fazi logiku, fazi promenljive koje su korišćene za raskrsnice, fazi pravila kao i opis simulacije gde će detaljnije biti reči o raskrsnicama.



Slika 1: Tipovi raskrsnica

## Fazi logika - ukratko

Klasična binaena logika je dobro definisana i korišćena. Mogli smo videti i upotrebu iskazne logike i logike prvog reda u Veštačkoj Inteligenciji gde su se mogu implemetirati razni rešavači i dokazivači. Međutim, ne mogu se svi problemi modelovati klasičnom logikom, kao neke neprecizne, nekompletne ili nesigurne informacije. Na primer, kako bi opisali klasičnom logikom "gužva na semaforu je velika". Ovakve stvari možemo modelovati fazi logikom. Sa fazi logikom nismo ograničeni nad binarnim vrednostima, već može uzimati vrednosti koja je između 0 i 1.

U fazi logici, domeni se opisuju lingvističkim promenljivama. Na primer, u rečenici "dužina reda na semaforu je mala", reč mala je lingvistička promenljiva. Ljudski mozak ima mogućnost da razume ovakve stvari i donositi zaključke.

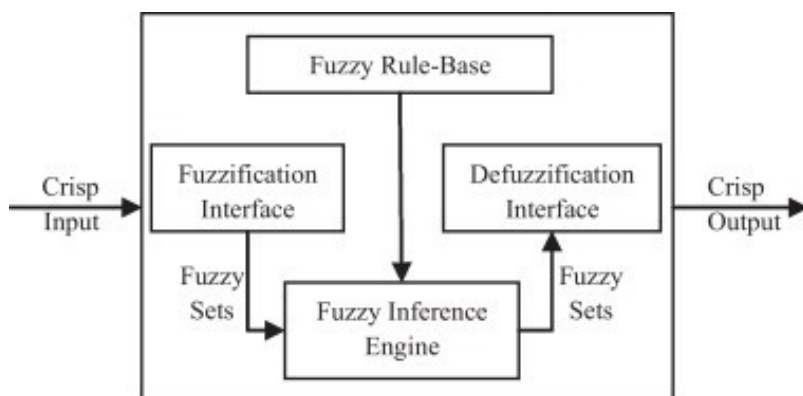
Kod klasične logike određivanje da li nešto pripada ili ne pripada skupu može da ima samo dva slučaja, ono ili pripada ili ne pripada. Razmotrimo primer visokih ljudi, koji je standardni primer u fazi logici, neka skupu niskih ljudi pripadaju svi koji su niži od 1.80m, a skupu visokih svi koji su viši. Tada bi čovek koji je visok 1.83m visok podjednako pripadao skupu visokih kao i košarkaš koji je visok 2.30m. Takođe, može se primetiti da skupu visokih pripada čovek visok 1.81m, a ne pripada čovek visine 1.79m iako je razlika samo 2cm. U fazi skupovima ovakvi problemi ne postoje. U ovom slučaju svi se nalaze u skupu visok i u skupu nizak samo sa različitim stepenom pripadnosti. Na primer, čovek visine 2.07m bi pripadao skupu visok sa stepenom 0.84, dok bi čovek visine 1.92 pripadao sa 0.67.

Uz pomoć fazi logike i fazi skupova može da se konstruiše softver koji može da oponaša kako čovek donosi zaključke.

Funkcija pripadnosti suština fazi skupova. Funkcija pripadnosti, još se naziva i karakterističnom funkcijom fazi skupa, definiše fazi skup. Koristi se za povezivanje stepena pripadnosti svakog od elemenata u domenu fazi skupa.

Operacije nad fazi skupovima:

- Ekvivalentnost: Dva skupa su ekvivalentna ako oba skupa imaju domen i ako za svaki element imaju istu funkciju pripadnosti.
- Podskupovi: Skup A je podskup skupa B ako važi  $\mu_A(x) \leq \mu_B(x)$  za svako  $x \in X$ .
- Komplement:  $A^c$  je komplement skupa A ako važi za svako  $x \in X$   $\mu_A(x) = 1 - \mu_{A^c}(x)$ .
- Presek: Dva standardna načina su:
  - Preko minimuma:  $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$ ,  $\forall x \in X$
  - Preko proizvoda:  $\mu_{A \cap B}(x) = \mu_A(x)\mu_B(x)$ ,  $\forall x \in X$
- Unija: Dva standardna načina su:
  - Preko maksimuma:  $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$ ,  $\forall x \in X$
  - Preko sume:  $\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x)$ ,  $\forall x \in X$



Slika 2: Ilustracija fazi zaključivanja

Fazifikacija je proces koji se brine o nalaženju fazi reprezentacije ne-fazi ulaza. Ovo se postiže primenjivanjem funkcije pripadnosti nad ulaznim podacima. Cilj je primeniti fazi pravila nad fazi ulaznim vrednostima. Defazifikacija je proces vraćanja fazi vrednosti u ne-fazi izlaz.

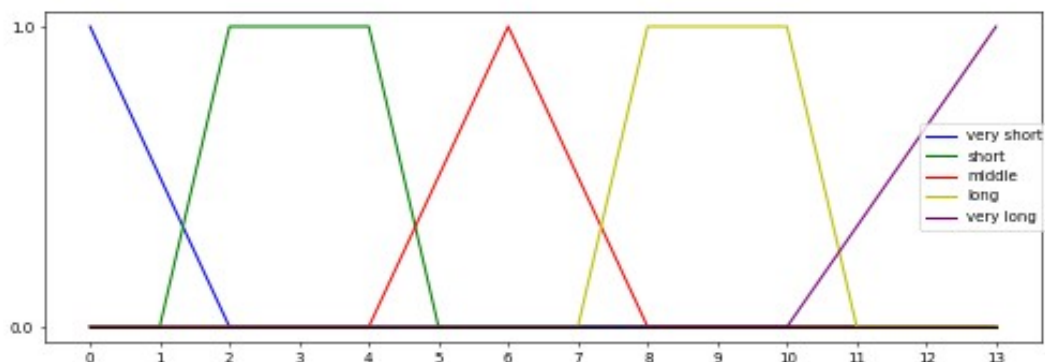
# Fazi promenljive

Za dinamički semafor koji smo implementirali koristili smo tri funkcije pripadnosti. To su Red (broj kola na semaforu), Pristizanje(predstavlja stepen kolko često dolaze nova vozila) i Produžetak(prestavlja za koliko je potrebno da se produži trajanje zelenog svetla). Tablela ispod prikazuje vrednosti koje uzimaju, kao i notacija za te promenljive korišćene u programu.

Red		Pristizanje		Produžetak	
veoma kratak	vshort	mali	small	veoma mali	vsmall
kratak	short	srednji	average	mali	small
srednji	middle	veliki	big	veliki	big
dugačak	long			veoma velik	vbig
veoma dugačak	vlong				

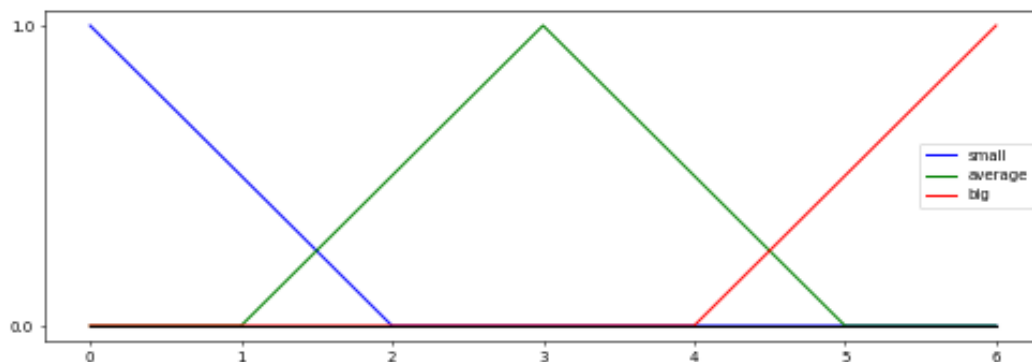
Grafičke reprezentacije i kratki opisi funkcija pripadnosti mogu da se vide ispod. Odatle mogu da se vide vrednosti funkcije pripadnosti za date lingvističke promenljive. Sve vrednosti koje smo stavili smo pokušali da zamislamo u realnoj situaciji šta bi mi smatrali da je na primer dugačak red na semaforu.

Na Slici 3 je predstavljena jedna od fazi ulaznih promenljivih koja predstavlja dužinu reda na semaforu. Dužina reda može biti veoma mala, mala, srednje, dugačka i veoma dugačka. Tako na primer za 3 vozila u koloni kažemo da je kolona kratka.



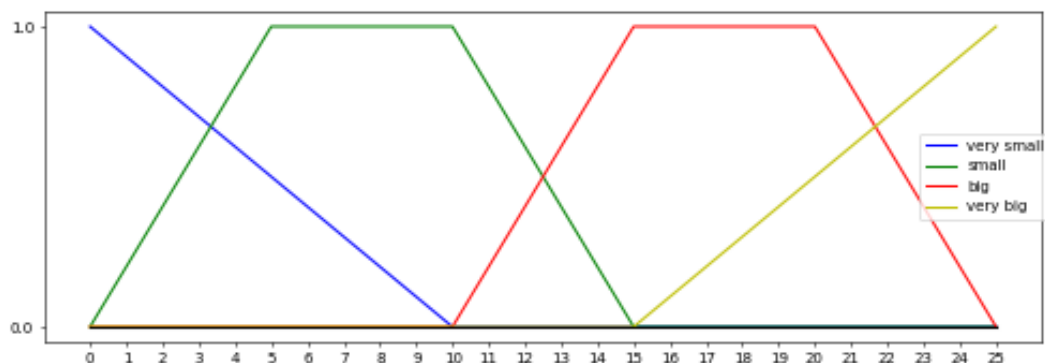
Slika 3: Fazi ulaz - dužina reda

Na Slici 4 smo predstavili drugu ulaznu fazi promenljivu na kojoj prikazujemo stepen pristizanja kola na semaforu. Vrednosti koje može imati su mali stepen pristizanja, srednji stepen pristizanja i veliki stepen pristizanja. Gde bi stepen pristizanja zapravo označavalo učestanost pristizanja kola u raskrsnicu. Ovde na primer



Slika 4: Fazi ulaz - stepen pristizanja

Na Slici 5 je predstavlja fazi izlaz, on prestavlja da li semafor treba da produži trajanje zelenog svetla. Uzima vrednosti veoma mali, mali, veliki i veoma veliki produžetak semafora. Ideja je da u zavisnosti od vrednosti fazi izlaza semafor može da odluči da li je potrebno da na tekućem zelenom svetlu produži trajanje zelenog svetla i na taj način propusti veći broj vozila.



Slika 5: Fazi izlaz - produžetak semafora

## Fazi Pravila

Fazi pravilima opisujemo način na koji pametni semafor treba da razmišlja. To je skup pravila koji treba da bude ispunjen. Ova pravila su nalik onome kako bi čovek ili najbolje opisano kako bi saobraćajni policajac donosio odluke iz koje ulice treba da se propuste kola. Na primer, saobraćajni policajac bi video da se stvara kolona vozila više na jednoj strani i onda bi propuštao više njih.

Prestavićemo skup pravila koji smo koristili u našoj implementaciji. Ovim pravilima smo pokušali da zamislimo kako je u nekoj realnoj situaciji. Oznake funkcija pripadnosti i vrednosti promenljivih su ista kao u prethodnom odeljku.

Skup pravila:

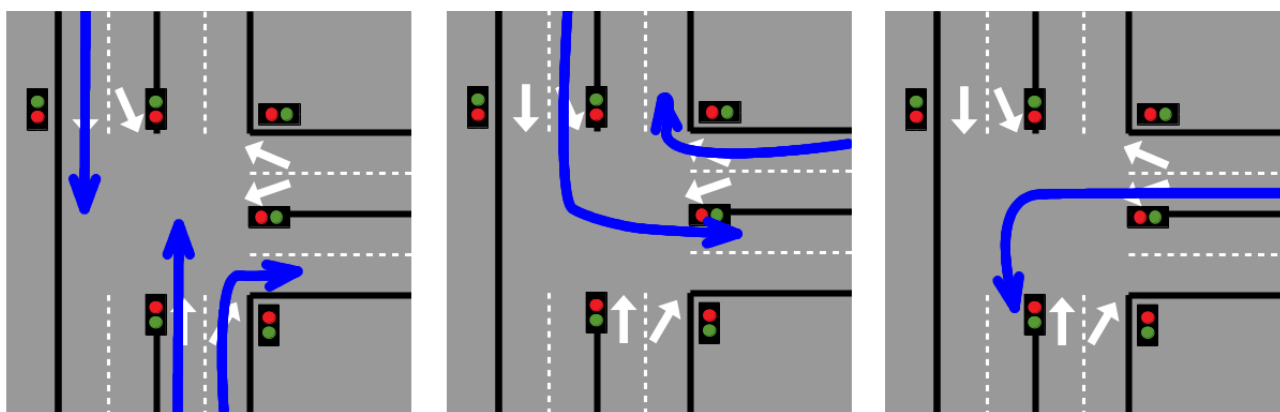
1. Ako je Red veoma dugačak onda je Produžetak veoma velik,
2. Ako je Red dugačak i Pristizanje je veliko onda je Produžetak veoma velik,
3. Ako je Red dugačak i Pristizanje je srednje onda je Produžetak velik,
4. Ako je Red dugačak i Pristizanje je malo onda je Produžetak mali,
5. Ako je Red srednji i Pristizanje je veliko onda je Produžetak velik,
6. Ako je Red srednji i Pristizanje je srednje onda je Produžetak mali,
7. Ako je Red srednji i Pristizanje je malo onda je Produžetak veoma mali,
8. Ako je Red mali i Pristizanje je veliko onda je Produžetak mali,
9. Ako je Red mali i Pristizanje je srednje onda je Produžetak veoma mali,
10. Ako je Red mali i Pristizanje je malo onda je Produžetak veoma mali,
11. Ako je Red veoma mali onda je Produžetak veoma mali.

## Opis Simulacije i Programa

Program koji smo implementirali sadrži sirovu implementaciju fazi kontrolera u opštem slučaju. Tako da iz simulacije možemo direktno da kreiramo promenljive i pravila po želji. Mi smo se odlučili da radimo sa istim skupom promenljivih i pravila na svim raskrsnicama. I da na taj način simuliramo rad istog fazi sistema na svim raskrsnicama. Implementacija je rađena u C++ i korišćena je Qt biblioteka. Pored same fazi logike, implementirali smo i simulator rada fazi kontrolera na raskrsnicama. Na Slici 1 još u uvodnom delu se može videti svi tipovi raskrsnica koje smo razmatrali. Kada se sam program pokrene otvara se prozor baš nalik Slici 1. Odatle je moguće odabrati željenu raskrsnicu na kojoj želimo da testiramo simulaciju. Kada smo selektovali željenu raskrsnicu otvara se novi prozor i u tom prozoru možemo da vidimo odabranu raskrsnicu nacrtanu dva puta, sa leve strane je raskrsnica sa pametnim fazi semaforom, a sa desne strane je sa standardnim ne-fazi semaforom. Simulacija se započinje pritiskom na start simulation.

Šta se desi kada se pokrene simulacija za odabran tip raskrsnice? Prvo što se desi je da se započne globalni tajmer, nakon toga je moguće da se zakazuju kola za svaku ulicu, kola se zakazuju ili pririskom dugmeta na ekranu ili preko određenog tastera sa tastature. Zakazana kola se pojave u željenoj ulici, i u levoj i u desnoj raskrsnici, kola se zatim kreću ka raskrsnici gde će na dalje da se zaustave i čekaju na semaforu ili samo prođu kroz raskrsnicu.

Svetla na semaforu se smenjuju u ciklusu. Objasnimo šta su ciklusi na Slici 6, prvo se uključuje zeleno svetlo na semaforima tako da je prolaznost kao na Slici 6 levo, posle isticanja vremena tog semafora on prelazi u stanje crvenog svetla, a u stanje zelenog svetla prelaze semafori tako da je prolaznost kao na Slici 6 sredina, nakon isticanja trajanja ovog semafora prelazi se na prolaznost kao na Slici 6 desno, nakon toga se ciklus vraća na početak.



Slika 6: Raskrsnica Ta

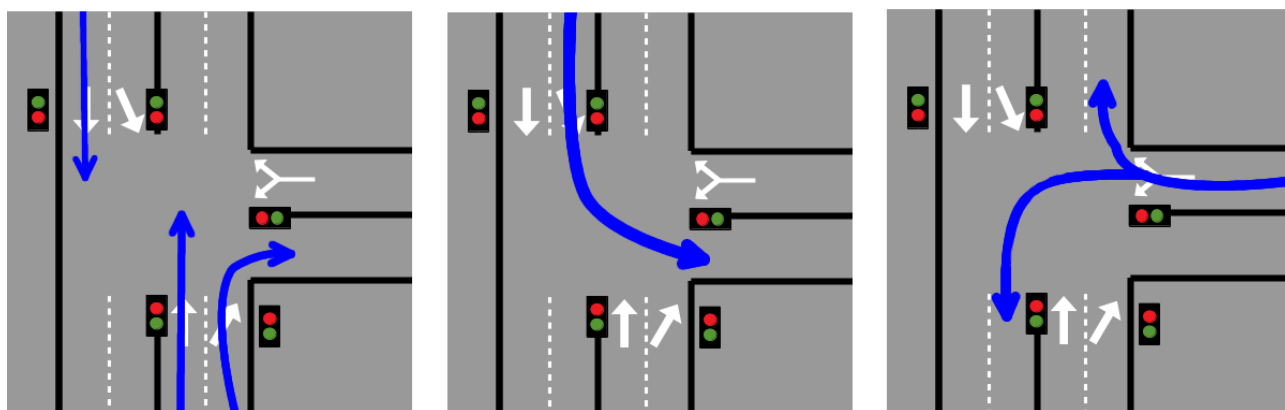
Kod standardnog semafora vreme trajanja zelenog na semaforima je fiksno, dok kod pametnog semafora trajanje zelenog svetla se menja dinamički u zavisnosti od dužina kolone i učestalosti pristizanja vozila.

Kolone su implementirane kao vektor koji sadrži binarne vrednosti koje označavaju da li ima kola u koloni ili nema. U svakom trenutku kada se doda automobil u kolonu on se pomera u koloni, odnosno vrednost iterira kroz vektor, dok ne dođe do kola koja već čekaju na semaforu ili dok ne dođe do semafora gde će ili čekati ili proći.

Dužina reda je senzor koji prebrojava koliko je kola u koloni, u programu je implementiran kao broj vrednosti 1 u koloni, gde 1 označava kola. Na pravoj ulici, ne u simulaciji, ovako nešto bi se moglo izvesti pomoću dva senzora, jedan koji bi brojao koliko kola je ušlo u kolonu i jedan koji bi brojao koliko je prošlo kroz raskrsnicu iz te kolone. Broj kola u raskrsnici bi označavao razliku dva brojača.

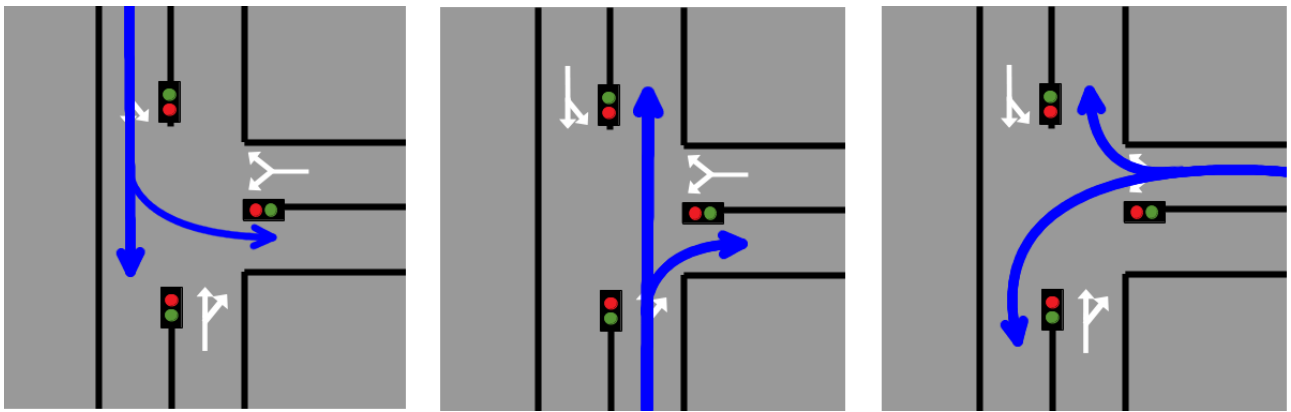
Stepen pristizanja je senzor koji se nalazi na neposrednoj udaljenosti od semafora i broji koliko je novih vozila prošlo kraj tog senzora kroz svaki ciklus tajmera koji pomera datu kolonu.

Na slikama 7 i 8 prikazane su nešto jednostavniji slučajevi raskrsnice sa slike 6.



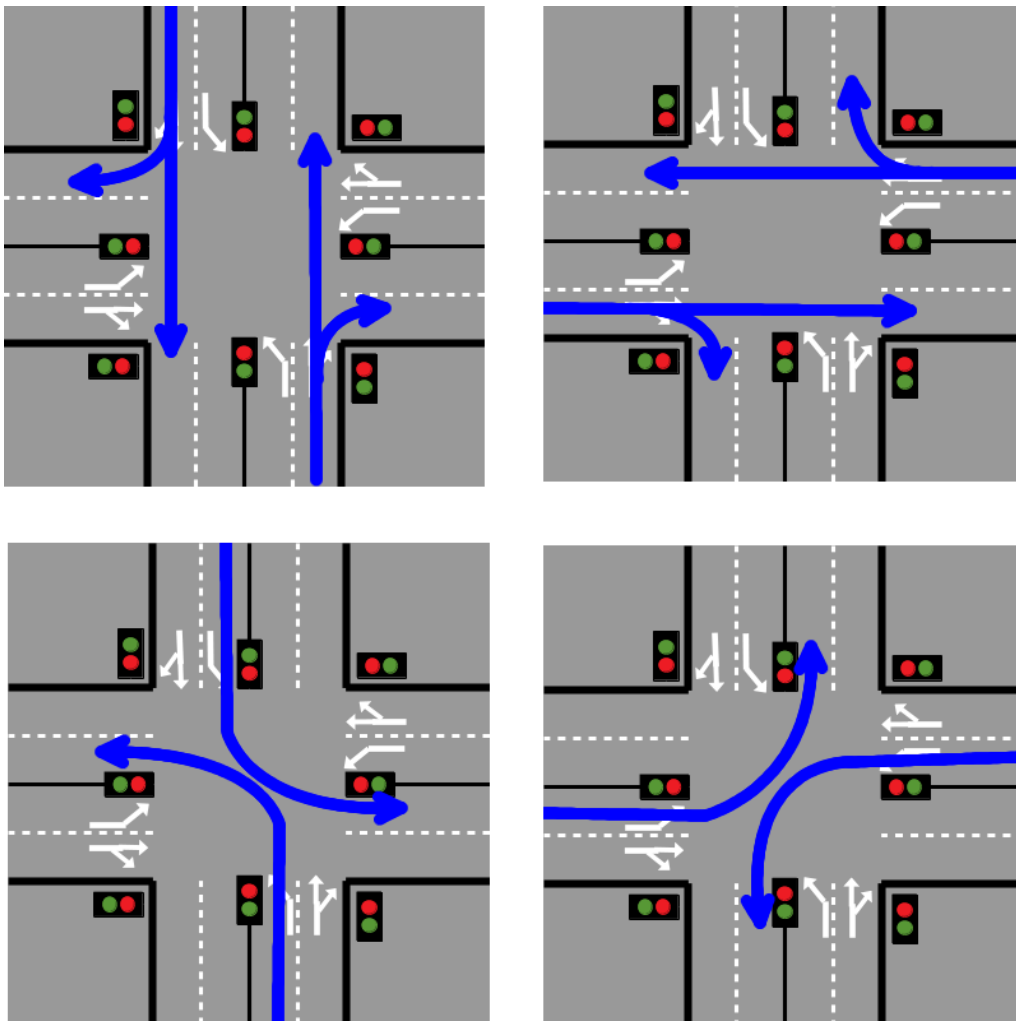
Slika 7: Raskrsnica Tb



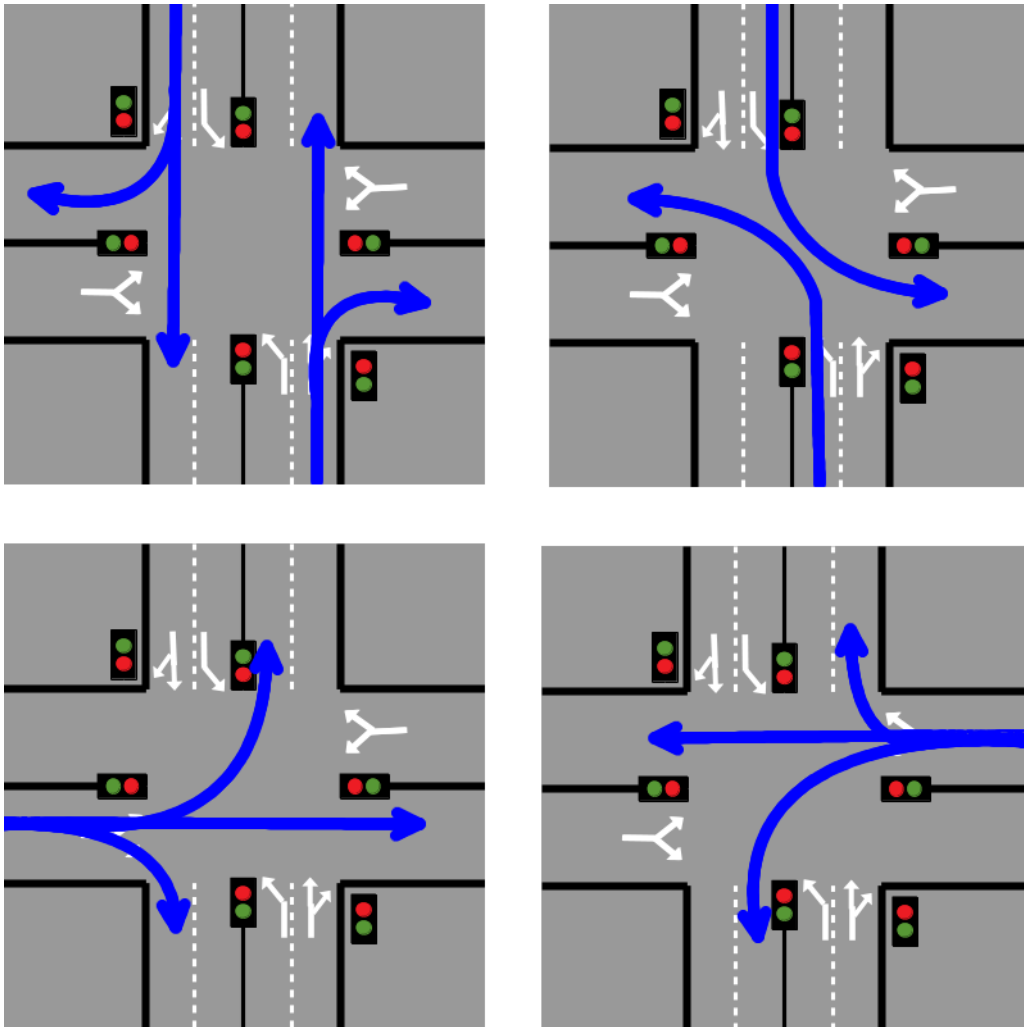


Slika 8: Raskrsnica Tc

Na slikama 9 i 10 prikazujemo nešto kompleksije raskrsnice, koje se odlikuju sa više ulica, semafora, a samim tim i broj kola koji može da stane u raskrsnicu. Na ovim raskrsnicama smo uspeli da dobijemo i najbolje rezultate o čemu će biti reči u narednom odeljku.



Slika 9: Raskrsnica Xa



Slika 10: Raskrsnica Xb

## Rezultati/Zaključak

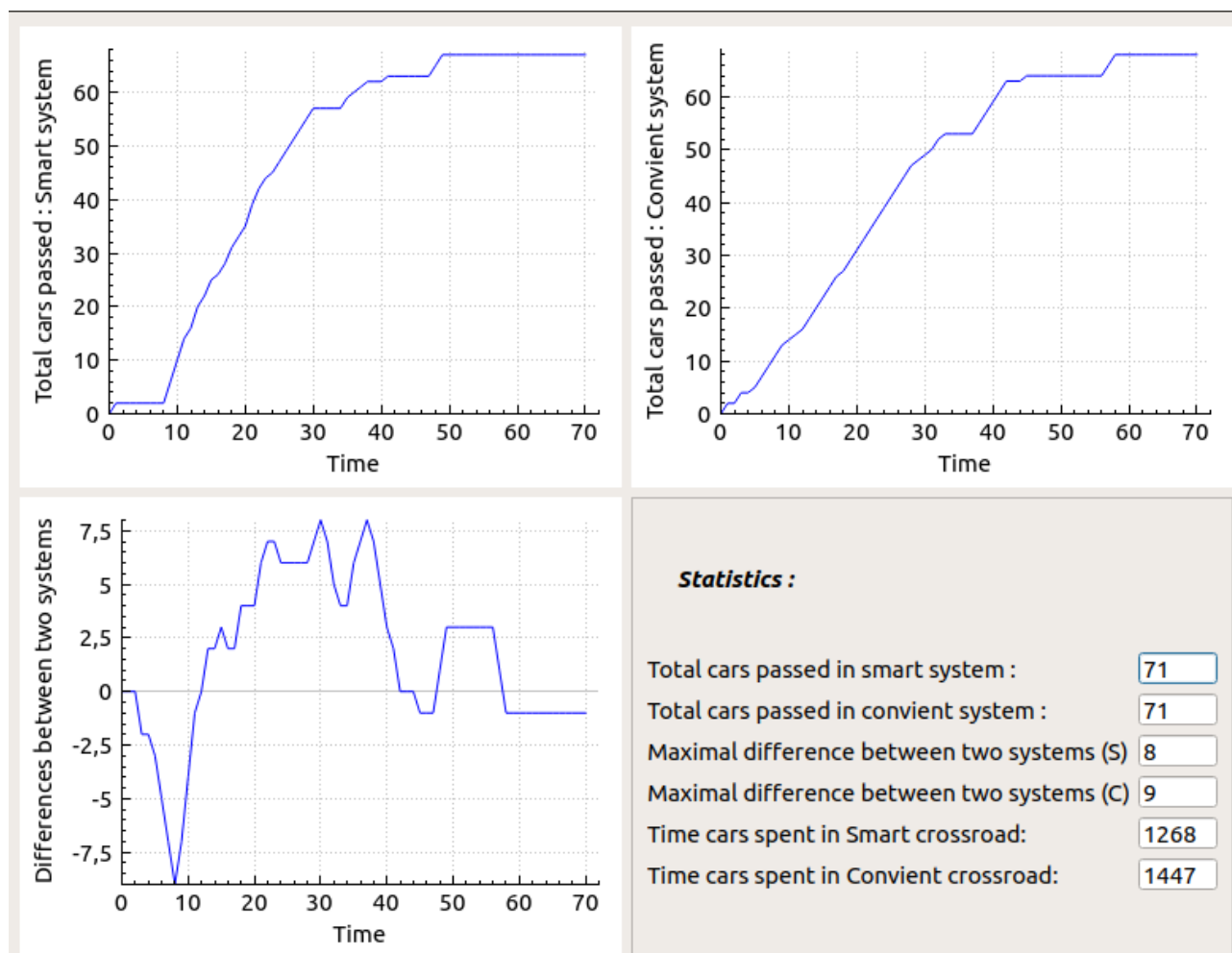
Može se primetiti da čak i bez direktnih statistika da zaista jeste veća prolaznost na pametnom semaforu i to samo gledajući paralelno dva tipa semafora na raskrsnici. Što je i bila čitava ideja da se stave oba tipa semafora jedan pored drugog i da se sve pokrene u identičnom okruženju, sa istim automobilima koji ulaze u raskrsnice u isto vreme.

Međutim, pored toga napravili smo da nakon završetka simulacije izađe statistika i metrike na kome mogu da se vide prednosti pametnog semafora. Prikazujemo koliko je kola prošlo kroz jedinicu vremena na pametnom i na standardnom semaforu. Kao i razlika u broju kola koje je prošlo kroz raskrsnicu. Kao i neke mere, koliko je kola ukupno prošlo na oba tipa semafora, kolika je najveća razlika bila i koliko su kola ukupno dugo čekala.

Kada smo upoređivali semafore, simulirali smo kola u kolonama na više načina, podjednako opterećenje na svim ulicama, opteretimo više jednu ulicu, opteretimo više ulicu za koju nismo smatrali da je prometnija (ovde standardni semafor baš kasni).

Na Slici 11 može da se vidi ilustracija ove statistike za raskrsnicu Xa, koja nam i daje najbolje rezultate, verovatno zato što može više kola da stane u raskrsnicu. Pustili smo

simulaciju za relativno mali broj kola i može se primetiti da su sva kola brže prošla kroz raskrnicu sa pametnim semaforom nego sa standardnim, to se može videti i poređenjem prva dva grafikona, na trećem grafikonu se može videti i da je razlika skoro sve vreme davala prednost pametnom semaforu. I iz metrika se može videti očigledno da su automobili manje ukupno čekali na pametnom semaforu.



Slika 11: Ilustracija statistike

Ono što smo još primetili da se nekad vidi da više kola prolazi kroz standardni semafor nego kroz pametni ali da u trenutku kada nova kola prestanu da pristižu pametni semafor brže raščisti gužvu.

Fazi dinamički semafor za kontrolu svetla se definitivno bolje pokazao od standardnog ne-fazi semafora. Takođe smo primetili koliko je semafor efikasniji ne samo za broj vozila koji prođe kroz semafor već i vreme koje su proveli u raskrnici.

# Literatura

- [1] International Journal of Advanced Research in Computer Science and Software Engineering, "Fuzzy Logic Controller Based Traffic Light System", Navneet Kaushal, Vithika Joshi
- [2] "Intelligent Traffic Lights Control By Fuzzy Logic", Article in Malaysian Journal of Computer Science, January 1997
- [3] "An Optimal Dynamic Control Method for an Isolated Intersection Using Fuzzy Systems", Iranian Journal of Optimization
- [4] "Smart Pedestrian Crossing Management at Traffic Light Junctions through a Fuzzy-Based Approach", G.Pau, T.Campisi, A.Canaele, A.Severino, M.Collotta, G.Tesoriere
- [5] "Computational Intelligence - An Introduction", A. Engelbrecht, J. Willey
- [6] Slajdovi sa predavanja
- [7] Materijali sa vežbi