

Simulacija in izboljšava prometnih tokov: primer na dveh izbranih slovenskih križiščih

Sandi Gec, Vlado Stankovski, Marko Bajec, Slavko Žitnik

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana

sandi.gec@fri.uni-lj.si, vlado.stankovski@fri.uni-lj.si, marko.bajec@fri.uni-lj.si, slavko.zitnik@fri.uni-lj.si

Izvleček

Prometna omrežja zaradi povezanosti trgov, potrošništva, potovanj, ... postajajo vse bolj zasičena. Izgradnja dodatnih povezav ni vedno možna, poleg tega pa je potrebno izvesti vse možne optimizacije za večjo pretočnost prometa. Eden od pristopov zahteva vpeljavo semaforizirane logike, ki skrbi za vzpostavitev prednostne logike na posameznih križiščih oz. vozliščih. Obstoječo deterministično logiko je potrebno posodobiti s prilagodljivimi sistemi. V delu se osredotočamo na študijo primerjave semaforiziranih logik in sicer fiksnega programa ter naprednejšega stohastičnega polno prometno odvisnega programa ter adaptivnega programa. Analiza uspešnosti algoritmov poteka na slovenskih križiščih Šoštanj predstavljen z enim semaforiziranim križiščem in Podutik Ljubljana, ki sestoji iz štirih zaporednih semaforiziranih križišč. Ugotovitve nakazujejo, da vpeljava adaptivne logike omogoča boljšo pretočnost prometa ter posledično zmanjšuje onesnaževanje.

Ključne besede: simulacija prometa, fiksni program, polno prometno odvisni program, adaptivni program, semaforizirano križišče, semaforne faze.

Simulation and improvement of traffic flows: A use case study on two selected Slovenian crossroads

Abstract

Transportation networks are becoming increasingly congested due to markets interconnection, consumers' needs, travelling etc. Implementation of additional links is not always possible, and therefore all possible optimisations need to be taken into account to enable more fluid traffic flows. One of the options is introduce traffic light logic that takes care of prioritization at individual intersections or nodes. It is necessary to ensure flowing traffic by upgrading the existing deterministic logic of fixed networks. The paper focuses on the analysis and the comparison of traffic lights logic programmes, specifically the fixed programme, the more advanced stochastic fully traffic-dependent programme and the adaptive programme. The analysis takes place at the Slovenian intersections Šoštanj, with one traffic lights intersection, and Podutik Ljubljana, which consists of four consecutive traffic lights intersections. The findings suggest that the introduction of adaptive logic enables improved traffic flow and consequently reduces pollution.

Keywords: Traffic simulation, fixed programme, full traffic dependent programme, adaptive programme, traffic light intersection, traffic light phases

1 UVOD

Slovenija je iz urbanističnega vidika ena redkih držav, kjer večji del prebivalstva živi v manjših mestih in vaseh, pri čemer imamo le peščico večjih mest. Pretežni del prebivalstva si za zagotovitev hitrega prehoda najpogosteje izbere lasten avtomobilski prevoz. Posledica takšnega trenda je število registriranih

vozil, kar se odraža v večji prometni obremenitvi ter onesnaževanju¹, zastojih, zmanjšani pretočnosti vozil, zamudah javnega cestnega prometa ter tovarnega prometa in druge cestne dogodke. V našem delu se osredotočamo na optimizacijo semaforne logike,

¹ <http://kazalci.arso.gov.si/sl/content/lastnistvo-osebni-avtomobilov-2>
(Dostopano dne: 18. julij 2022).

s katero želimo doseči večjo pretočnost prometa ter posledično nižjo onesnaževanje okolja. Za namene povečanja pretočnosti vozil je smotrna vpeljava rešitev pametnega prometa, predvsem na nivoju semaforne logike. Ker je neposredna vpeljava rešitev na infrastrukturi tvegana in cenovno draga, je smiselno proučiti rezultate posodobitve prometnih tokov semaforških križišč na podlagi realnih podatkov. V te namene je bila izvedena študija, ki se osredotoča na analitiko prometnih tokov na dveh različnih scenarijih, in sicer samostojno semaforizirano križišče ter nabor medsebojno odvisnih križišč. Analitika poleg numeričnih rezultatov omogoča tudi prilagajanje parametrov posameznega scenarija ter vizualizacijo prometnega poteka.

V tem prispevku obravnavamo prometna omrežja, ki so predstavljena s tremi osnovnimi elementi, in sicer: (a) cestišča, (b) semaforizirana križišča in (c) vozila. V prometna omrežja se dodajajo vozila, ki vozijo po vnaprej določeni poti na eno, dvo ali večpasovnih cestiščih, pri čemer je najbolj obremenjen del križišč semaforiziran. Cilj našega dela je definicija adaptivne logike semaforiziranih križišč z namenom izboljšanja pretočnosti v primerjavi s **fiksno semaforško logiko**.

Vodenje prometa v samostojnih izoliranih križiščih je potrebno analizirati ločeno od tistih za mrežno oz. arterijsko vodenje prometa, v conah in s sekvenco semaforiziranih križišč. Slednje se v osnovi nanaša na prometna omrežja z dvema ali več semaforiziranih križišč na isti cesti, kjer je potrebno upoštevati soodvisno semaforško logiko. Na izoliranih križiščih je bistveno več prostora za prilagajanje dolžine trajanj posameznih prikaznih luči (t.i. »splitov«), zamenjave vrstnega reda ter preskakovanja celotne sekvence faz, ker na teh križiščih ni potrebno upoštevati koordinacije, kot je to nujno v conah. V našem delu obravnavamo obe vrsti omrežij - samostojno izolirano omrežje v Šoštanju in več soodvisnih semaforiziranih križišč na prometnem omrežju Ljubljana-Podutik.

Delo se osredotoča na optimizacijo prometnih tokov z vidika definicije logike oziroma algoritmov semaforiziranih križišč. Vmesni algoritmi med fiksnimi in adaptivnimi so polno prometno odvisni. Ti so nadgradnje fiksnih algoritmov z vnaprej določenimi dolžinami posameznih faz, splitov in ciklov, kjer ima vsak split minimalno in maksimalno dolžino trajanja, ta pa je odvisen od količine prometa beležene-

ga na podlagi induktivnih zank ali detektorjev, postavljenih na vseh vhodnih poteh pred semaforjem. Najbolj napredni **adaptivni algoritmi** za kontrolo semaforjev ne potrebujejo vnaprejšnje definicije fiksnih prometnih programov, saj se dolžina cikla, »spliti« in zaporedje faz dinamično izračuna, in sicer z namenom, da se v križiščih poveča pretočnost prometa. Podatke za izračune podobno kot pri polno prometno odvisnih algoritmi pridobivamo iz detektorjev. V nadaljevanju so predstavljeni algoritmi in študije, ki nakazujejo, da se adaptivna kontrola obnese boljše od polno prometno odvisne.

V nadaljevanju najprej analiziramo sorodna dela. V poglavju kasneje opredelimo algoritme, ki se v našem delu uporabljajo ter orišemo našo rešitev, vključno z arhitekturo. Evalvacija ter rezultati simulacije z diskusijo so predstavljeni v četrtem poglavju. V zaključnem poglavju podamo sklepne ugotovitve in nadaljne možnosti za razvoj.

2 SORODNO DELO

Težnja k optimizaciji prometnih tokov sega v čase druge polovice prejšnjega stoletja, ko so se začela množično graditi prometna omrežja in povečano proizvodnjo vozil. V nadaljevanju se osredotočamo na sorodna dela in sicer:

- Pregled trendov raziskav mehke logike prometnih tokov skozi čas, ki razvoj pogojuje tudi z razvojem senzorske in druge tehnološke opreme,
- razvoja algoritmov adaptivne logike ter način delovanja in
- analizo računalniških orodij za simulacijo prometnih tokov vključno z izdelavo prometnega toka, definicije semaforne logike in simulacijskih entitet (npr. vozila, pešci itd.).

2.1 Pregled raziskav mehke logike prometnih tokov

Prvi znani poskus uporabe mehke logike pri upravljanju prometne signalizacije sta leta 1977 izvedla Pappis in Madani. Predlagala sta teoretično simulacijsko študijo mehkega upravljalnika v izoliranem semaforiziranem križišču dveh enosmernih cest z dvema voziščema in enakim prometnim pretokom (Pappis in Madani, 1997). Njuno delo se uvršča v po-voje aplikativne predstavitve mehke logike. Avtorja sta primerjala svojo mehko metodo s prilagodljivim sistemom upravljanja signalizacije, ki minimizira zamude z optimalno dolžino cikla. Mehka logika je bila

vsaj tako dobra kot primerjalni prilagodljiv sistem.

V devetdesetih letih prejšnjega stoletja so se začele vrstiti raziskave in študije uporabe mehke logike pri upravljanju prometne signalizacije. Sodobne mehke sisteme v prometnem inženirstvu je predstavil Teodorovic, ki se med drugim osredotoča tudi na prilagodljivo upravljanje prometne signalizacije (Teodorovic, 1999). Preučevanje mehke logike so predstavili avtorji, ki so preučevali mehke algoritme v izoliranih križiščih (Kim, 1997). Avtorji so dolžino zelenega signala prilagodili razmeram prometa ob koncu vsake faze. Favilla in sod. so leta 1993 predstavili dve različni metodi ostrenja in kriterije pri sprejemanju odločitev (Favilla in sod., 1993). Trabia in sod. so leta 1999 predstavili mehki sistem upravljanja prometne signalizacije za izolirano križišče, ki je temeljil na dvonivojski proceduri, ki je odločala o podaljšanju ali prekinitvi faze (Trabia in sod., 1999). V prvi fazi je sistem ocenil intenzivnost prometa, nato pa rezultat upošteval v drugi fazi, kjer se odloča o prekinitvi ali podaljškju. Sistem je vračal boljše rezultate, predvsem za manjše zakasnitve vozil in enak delež ustavljanja kot v primeru metode na zaznavo vozil. Sayers in sod. so se ukvarjali z mehkim upravljanjem prometne signalizacije z vidika več ciljev in tudi z ozirom na to, kje naj bi se upravljalni sistem uporabljal (Sayers in sod., 1998). Uporabili so genetski algoritem (MOGA) kot optimizacijsko tehnologijo. Wei in sod. so predstavili pristop za zmanjševanje zamud in kontrolo faz - koncept »glavna nujna faza« in »manjša nujna faza« skupaj z vsemi pravili mehke logike (Wei in sod., 2001). Pristop algoritma mehke logike z agenti, ki imajo sposobnost odločanja, je leta 2003 predstavil Kosonen (Kosonen, 2003). Avtorja Akiyama in Okushima sta leta 2006 predlagala pristop kako v algoritmu spreminjata tok prihodov vozil s spremenljivkami za optimizacijo lingvističnih spremenljivk v definiciji modela mehke logike (Akiyama in Okushima, 2006). Avtorja sta pripravila napredni adaptivni algoritem, ki je dodatna funkcionalnost prometnemu informacijskemu sistemu za prometne tokove na hitrih vpadnicah na Japonskem. Drugačen pristop, in sicer z uporabo Mamdani metode za optimizacijo in skrajševanjem povprečnih časov ter zmanjševanje dolžine kolon so pripravili Hu in sod., kjer so leta 2007 definirali algoritem z mehko logiko za določanje dolžine zelenega časa. Evalvacijo so izvedli za 5-krako križišče s 14 pasovi, ki vključujejo 6 zavijalnih pasov in dva prehoda za pešce (Hu

in sod., 2007). Algoritem uporablja prometne podatke iz detektorjev v križišču. Avtorja Zhang in Ye sta predstavita metodologijo za napoved prometnih tokov v križišču z uporabo dvojnega detektorja (Zhang in Ye, 2008). Metoda se je izkazala za bolj natančno in robustno, saj so bile napovedi natančne za različne prometne tokove.

2.2 Algoritmi semaforke logike prometnih tokov

Adaptivna kontrola temelji na analizi napovedi prihajajočega prometnega toka, za kar obstajata dve vrsti napovedi. Prva napoved meri podatke v realnem času na detektorjih, druga pa iz historičnih podatkov napove gibanje prometnega toka. Z analizo podatkov v realnem času lahko nastavljamo cikel med odvijanjem in se sproti odločamo o podaljševanju ali zaključevanju posamezne faze. Z optimizacijo na nivoju cikla med samim izvajanjem se zelo dobro prilagajamo hipnim nihanjem v prometu. Za optimizacije, ki potekajo vsakih 30 do 60 sekund, se lahko uporablja podatke v realnem času.

COP algoritem (Sen in Head, 1997) iz sistema RHODES (Mirchandani in Fei-Yue Wang, 2005) optimizira vrstni red in trajanje faz vsakih 30 do 40 sekund, in sicer odvisno od lokacije detektorjev. RHODES razdeli potovanje od oddaljenega detektorja do črte stop na dva dela, in sicer od oddaljenega detektorja do bližnjega ter od bližnjega detektorja do stop črte. Bližnji detektorji so definirani precej daleč, in sicer okvirno 100 m od stop črte, da ima krmilnik dovolj časa da se odzove na podatke iz detektorjev. Ker je detektor precej oddaljen, se ocenjuje trajanje poti in hitrosti od tega detektorja do črte stop, saj pri gostem prometnem toku prihaja do nabiranja kolon.

OPAC strategija (Gartner, 1990) temelji na obdelavi podatkov zadnjih 50 do 100 sekund. Prihodi vozil se merijo na oddaljenih detektorjih. Algoritem mora izračunati predviden čas potovanja od oddaljenega detektorja do stop črte. Na podoben način tudi OPAC definira potovalne čase potovanj od bolj k manj oddaljenim detektorjem, tako da tudi pri tej metodologij ostajajo problemi z nastajanjem kolon in oceno njihove dolžine.

PODE (Passive Opposition Differential Evolution) (Cheng, 2017) je strategija s hibridno funkcijo, t.j. funkcija definirana z večimi podfunkcijami, ki optimira dolžine signalov v kratkih časovnih intervalih. Kot ostali algoritmi, tudi PODE za vhodne podatke jemlje število prihodov vozil in kolone na črti stop. Poseb-

nost sistema je, da ima funkcionalnost spremenljivega intervala optimizacije in samoprilagodljivega mehanizma. Optimizacija lahko poteka le na nekaj sekund lahko pa poteka le vsake pol minute. Če je potrebno, se na začetku vsakega intervala popravi napaka ocene trajanja faz v trenutnem ciklu. Vse mogoče kombinacije dolžine in vrstnega reda faz se analizirajo za vsak interval, t.j. če imamo 8 faz in dolžino intervala 15 sekund, pomeni to 120 različnih kombinacij za analizo. Po primerjavi vseh 120 izračunov indeksov za vse kombinacije se izbere najboljša. Cilj algoritma je minimizacija zamud vozil za celotno križišče.

ACS-LITE (Zhanbo in sod. 2018) je reaktivni adaptivni sistem (sistem na osnovi povratne informacije zanke pri procesiranju podatkov) za nadzorovanje prometa. Za spremembo »splitov« uporablja podatek o zasedenosti detektorjev na stop črti. Zasedenost detektorjev se primerja z dolžino pripadajoče zelene faze, da se analizira koliko zelene faze je bilo porabljeno, oziroma za koliko časa bi se lahko skrajšala. Analiza računa povprečje v 3 do 5 ciklih. Po dobljenem rezultatu se »spliti« spremenijo. Cilj algoritma je doseči enakomerno stopnjo nasičenosti priključkov v križišču. Spreminjanje »splitov« se v ACS-LITE izvaja lokalno na vsakem krmilniku posebej in neodvisno. Vsaka optimizacija »splita« se najprej vrši po 3 končanih ciklih in 5 minutah. »Split« in zamik se spreminja v majhnih korakih, t.j. v časovnih intervalih 2 do 5 sekund. Zaključevanje faz pred maksimalno zeleno in preskakovanje faz deluje normalno v opisanem algoritmu.

Če povzamemo, potrebujemo za dobro adaptivno kontrolo na izoliranih križiščih naslednje:

- Podatke za optimizacijo, ki se pridobivajo v realnem času na podlagi podatkov s postavljenih detektorjev. Uporaba historičnih podatkov ni zaželena.
- Model napovedi prihoda na črto stop, ki mora biti zanesljiv za redek, gost in nasičen prometni tok ter mora pokrivati tudi pojavljanje zastojev.
- Adaptivni sistem, ki mora vsebovati samo prilagodljivi mehanizem za spremljanje podatkov in prilagajanje dejanskemu stanju.
- V našem delu primerjamo tri krovne pristope algoritmov in njihovo empirično delovanje na dveh različnih prometnih omrežjih z različnimi metrikami.

2.3 Orodja za simulacijo

Pri pripravi simulacije prometnih tokov smo se osredotočili na pregled odprtokodnih rešitev, ki omogočajo učinkovito simulacijo. Kot najbolj ustrezna kandidata smo identificirali orodji SUMO in Aimsun, ki sta dve izmed programskih orodij za izdelavo simulacij prometa. V Tabeli 1 prikazujemo njuno primerjavo.

3 SISTEM SIMULACIJE PROMETNIH TOKOV

V tem poglavju povzamemo algoritme aplicirane na izbranih prometnih omrežjih. Čeprav se širom sveta še vedno uporabljajo standardni fiksni programi, se zaradi preobremenjenosti omrežij pojavljajo potrebe po novih adaptivnih pristopih, ki merijo vozila neposredno pred semaforiziranimi križišči ali globalen pristop meritve na večjem območju prometnega omrežja in primerno ukrepanje. V našem primeru se osredotočamo na lokalni pristop, saj so omrežja dovolj zgoščena ter se njihova komunikacija podatkov razprostira do največ štirih semaforiziranih križišč.

V nadaljevanju so povzeti trije algoritmi za katere imamo razpoložljive vhodne podatke semaforske logike ter na voljo podrobne opise prihodov vozil glede na čas v omrežje.

3.1 Algoritmi

V tem razdelku so navedene le ključne lastnosti treh glavnih algoritmov. Podrobnosti algoritmov so poslovna skrivnost industrijskega partnerja projekta.

3.1.1 Fiksni program

Fiksni program lahko opredelimo kot časovno togo krmiljenje semaforjev. »Spliti« oz. podaljševanja zelenih faz so v vseh ciklih enaki, ne glede na nihanja v prometu. Cikel, zamik in split so definirani na podlagi historičnih podatkov in praviloma opredeljujejo obdobja kot so jutranji in popoldanski časi prometnih konic, nočni, vikend in ostale režimi. Z urnikom je definirana izbira prometnega programa glede na čas v dnevu oz. tednu. Takšno krmiljenje je primerno za cone, kjer so prometne obremenitve zelo predvidljive in so odstopanja majhna. Detekcija pri takšnem vodenju ni potrebna, zato se podatki na indukcijских zankah ne uporabljajo.

Tabela 1: Analiza funkcionalnosti med orodji SUMO in Aimsun

Glede na preprostost uvoza omrežij in naborom funkcionalnosti, širokim naborom metrik in možnostjo brezplačne odprtokodne uporabe, smo se odločili za orodje SUMO.

Funkcionalnost	Orodja	
	SUMO ¹	Aimsun ²
Uvoz omrežij	NETCOVERT (skripta za uvoz in avtomatsko prilagoditev cestnih omrežij v okolje SUMO) in druga neuradna orodja.	Neposredno z aplikacijo, omogoča naprednejše možnosti in podpira integracijo z OpenStreetMap ⁴ .
Modeliranje povpraševanja	Matrično, povpraševanje glede na aktivnosti (angl., Activity based demand) kot so dnevne poti na delo, v mesto, šole ter dogodki, naključne poti, računanje poti na podlagi uteži v omrežju. Različne funkcije za računanje uteži. Podpira razrede in tipe vozil.	Podobno kot SUMO, z nekaterimi dodatnimi funkcijami za računanje uteži povezav. Lažje generiranje prometa z isto aplikacijo.
Nadzor semaforške logike	Podpira običajno nastavljanje semaforjev. Vpliv detektorjev na semaforje in tudi programske vodene semaforje (angl., actuated traffic lights).	Enako kot SUMO. Sistem se upravlja preko grafičnega uporabniškega vmesnika. Podpira tudi grupiranje vozlišč s semaforji, torej nastavljanje pravil za več vozlišč naenkrat. Podpira kontrolerje, ki nadzorujejo dinamiko v enem križišču (detektorji na cestah, pešci, programi za nadzor semaforjev)
Javni prevoz	Javni promet so vozila s primernim razredom. Za ustvarjanje linij je potrebno specificirati poti. Ni urnikov, oz. jih je potrebno drugače implementirati.	Zelo natančno ustvarjanje in urejanje javnega prevoza, z linijami in urniki.
Izjemni dogodki	SUMO nima posebne sekcije, ki bi se ukvarjala s dogodki. Vse kar ima Aimsun, bi se dalo sicer implementirati.	Ima možnost nastavljanja množic pravil, ki določajo način kako naj se spremeni dinamika v omrežju (podprte so že nesreče, zaprtje pasov, sprememba namembnosti pasov, prisilno obračanje).
Upravljanje modelov omrežja	SUMO nima posebnih orodij za to.	Aimsun ima različne urejevalnike omrežja, prometa, javnega prometa. Pravzaprav gre za nastavljanje parametrov omrežja in prometa, da kasneje lepše delujejo kot celota. Omogoča izvajanje poskusov in ustvarjanje scenarijev.
Izbira poti	SUMO ima kar nekaj algoritmov za izbiro optimalne poti na podlagi uteži v omrežju. To je izvedeno z dodatnimi orodjem DUAROUTER (skripta za nastavljanje zahtev in omejitev pri usmerjanju prometa).	Aimsun ima vsaj na prvi pogled izredno obširno podprto usmerjanje. Vsebuje ogromno funkcij. V dokumentaciji je to zbrano pod t.i. razdelkom Dynamic Traffic Assignment.
Obnašanje voznikov	/	Ponuja precej modelov za simulacijo obnašanja voznikov na cesti, kot je npr. varnostna razdalja, spreminjanje voznih pasov, prilagajanje hitrosti glede na ostale voznike in druge.
Analiza podatkov in vizualizacija	Vsebuje orodja za vizualizacijo podatkov, ki so razširitve Python knjižnice Matplotlib. Množica podatkov, ki jih lahko pridobimo iz simulacije in statistika, je kar obsežna.	Zelo močno podprta vizualizacija in pridobitev in analiza podatkov. Sicer je precej podobno kot pri orodju SUMO.
Storitve	Del funkcionalnosti dostopen preko knjižnice Traffic Control Interface (TRACI), ki omogoča neposredno programsko upravljanje s simulacijo.	Vse funkcionalnosti dostopne preko storitve.

3.1.2 Polno prometno odvisni program

Značilnost tega prometnega programa je ta, da se krmiljenje izvaja na podlagi dodatne senzorike, ki je integrirano v prometno omrežje. Detektorji oz. induktivne zanke so nameščeni na vseh smereh za

vse faze. Prav tako je obvezna najava pešcev v primerih, ko s tem podatkom razpolagamo. »Spliti« na nekoordiniranih fazah se izvedejo tako, da se preveri pretočnost križišča v zadnjih treh fazah ter izračuna dve možnosti:

- podaljševanje zelene faze ali
- skrajševanje zelene faze.

² <https://www.eclipse.org/sumo/> (Dostopano dne: 18. julij 2022).

³ <https://www.aimsun.com> (Dostopano dne: 18. julij 2022).

⁴ <https://www.openstreetmap.org> (Dostopano dne: 18. julij 2022).

Z detekcijo na koordinirani fazi lahko pri manjših obremenitvah definiramo čas od dejanskega konca zelene faze do predefiniranega maksimalnega splita drugim fazam. Prometne programe umestimo v čas izvajanja z urnikom, podobno kot v sorodnih pristopih (Gartner, 1990).

Pri polno prometno odvisnih programih kjer imamo na voljo več semaforiziranih križišč, lahko opredelimo tudi njihove prioritete. Na primeru prometnega omrežja Podutik imamo na voljo 4 križišča, pri katerih je najbolj obremenjeno križišče Š-32 (prikazano na Sliki 8), saj se čez to križišče zapelje največ vozil in je posledičnost obremenjenost najvišja. Na podlagi primarnega križišča, ki ga tudi imenujemo dominantno, se »spliti« zelenih faz ostalih sekundarnih križišč lahko računajo na dva načina: (a) glede na dominantno križišče ali (b) neodvisno od dominantnega križišča. V našem primeru računamo »splite« vseh križišč neodvisno od dominantnega križišča.

3.1.3 Adaptivni program

Glavna naloga adaptivnih sistemov in algoritmov za krmiljenje prometa je z dobro analizo in interpretacijo vhodnih - detektorskih podatkov ponuditi primerno izbiro, prilagoditev obstoječega, oz. definicijo novega prometnega programa za zagotavljanje čim manjših zamud in čimvečje pretočnosti v križiščih. Za to je nujno potrebno vzpostaviti kompleksen sistem detektorjev, zmogljiv krmilnik za obdelavo in arhiviranje realnočasovnih podatkov podobno kot pri algoritmičnih adaptivne logike sorodnega dela ACS-LITE (Zhanbo in sod. 2018). Med križišči v koordinaciji mora obstajati zanesljiva dovolj zmogljiva povezava za prenos podatkov in sinhronizacijo med križišči. Večina adaptivnih sistemov uporablja modele za generiranje prometa v simulatorju, za distribucijo prihodov vozil, prihodov vozil v kolonah, oceno zamud in stoječih kolon. Spreminjanje cikla, splita in zamika je rezultat analiz, ki lahko služijo pri minimiziranju zamud, stopnje ustavitve vozil ter maksimizaciji zelenega vala pri križiščih v koordinaciji.

Efektivnost adaptivne strategije je odvisna tudi od parametrov, ki določajo dolžino kolone, zamude pri speljevanju in praznjenje vozil iz križišča. Zadnja parametra sta lahko odvisna od časa v dnevu, geometrije križišča, vzdolžnih naklonov, vremena in drugih dejavnikov. Večina obstoječih adaptivnih sistemov ne vsebuje samo prilagodljivih mehanizmov, ker so zgornji parametri modelirani kot statične spremenljivke.

Adaptivno krmiljenje prometa, ko na potek signalnega programa vplivajo vsi v križišče usmerjeni tokovi s sosednjih križišč, krmilimo preko nadzornega sistema, na način, da se poleg prometa na zeleni smeri hkratio analizira tudi promet na rdeči smeri. Tako se lahko »split« optimizira in prilagaja na osnovi podatkov v realnem času. Primer definicije adaptivne semaforke logike prometnega omrežja Podutik je prikazan na Sliki 1, kjer signalne glave opredeljujejo vse razpoložljive signalizacije udeležencev v prometu (vozila, pešci, kolesarji, semaforji za zavijanja in drugo). Za vsako signalno glavo je opredeljen fiksni ali prilagodljiv čas v sekundah posameznih faz semaforja, ki so navedeni v legendi na dnu slike.

Glavne funkcionalnosti na nivoju mikrokontrol pri fazni organizaciji so:

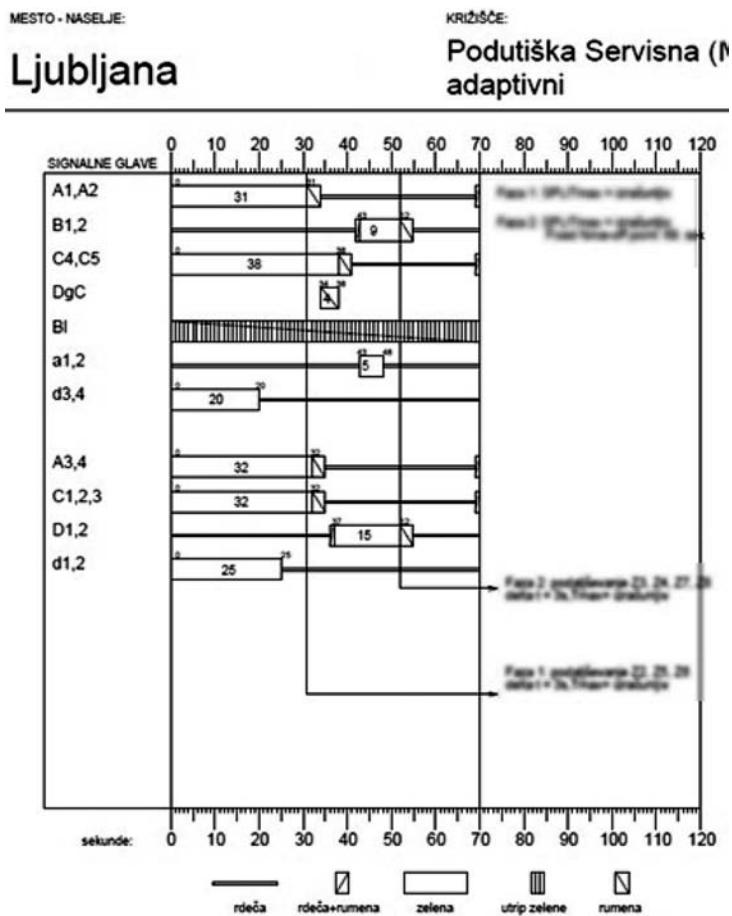
- minimalna zelena (angl., phase minimum green),
- zadrževanje faze (angl., phase rest), ki ga potrebujemo za mirovna stanja,
- podaljševanje faze (angl., phase extension), potrebujemo za podaljševanje minimalne začetne zelene,
- skrajševanje faze (angl., phase (early) cut-off), potrebujemo pri signalnih grupah z daljšimi vmesnimi časi (npr., pešci), vpeljavi prioritetenih voženj, spremembi prometnega programa križiščem v coni in prilagajanjem zamika,
- preskok faze (angl., skip-phase), uporabimo, če za preskočeno fazo ni najave na detektorjih in
- fiksiran najkasnejši možen zaključek faze (angl., phase fixed force-off), potrebujemo, da določimo kateri fazi se prerazporedi neporabljeni zeleni čas predhodne faze.

3.2 Opis rešitve

3.2.1 Funkcionalne zahteve

Simulacijsko okolje temelji na odprtokodnem orodju SUMO. V fazi zasnove sistema je bilo potrebno preučiti funkcionalne in nefunkcionalne zahteve, ki so opredeljene kot:

1. Visoka stopnja skalabilnosti sistema, za kar je potrebno ustrezno programsko ogrodje (angl., framework), ki omogoča podporo in možnost dodajanja različnih programskih paketov ter funkcionalnosti (npr. protokoli, podpora branju Microsoft Excel dokumentov).
2. Osnovna komunikacija naj bo kar se da standardna, zato temelji na standardnem protokolu REST (angl., Representational State Transfer).



Slika 1: Primer adaptivne semaforke logike prometnega omrežja Podutik. Deli, ki predstavljajo poslovno skrivnost so zamegljeni.

- Osnovna interakcija sistema za (končnega) uporabnika naj bo omogočena preko spletnega (angl., Web) grafičnega uporabniškega vmesnika, ki omogoča interakcijo z osnovnimi funkcionalnostmi sistema preko protokola REST.
- Sistem naj bo interoperabilen in prenosljiv, saj so pričakovane povezljivosti preko zunanjih sistemov.
- Posamezne komponente naj bodo neodvisno razvite ter naj jih bo mogoče tudi neodvisno poganjati na različnih infrastrukturah.
- Odzivnost simulacije naj omogoča vsaj 8-kratno hitrost simulacije glede na dejansko trajanje. Če celovit scenarij traja 24ur, naj bo torej maksimalno trajanje simulacije 3 ure.

3.2.2 Arhitektura rešitve

Celoten sistem je zasnovan modularno na podlagi osnovnih funkcionalnih zahtev projekta, ki so zajete v komponento za simulacijsko okolje ter izbrano

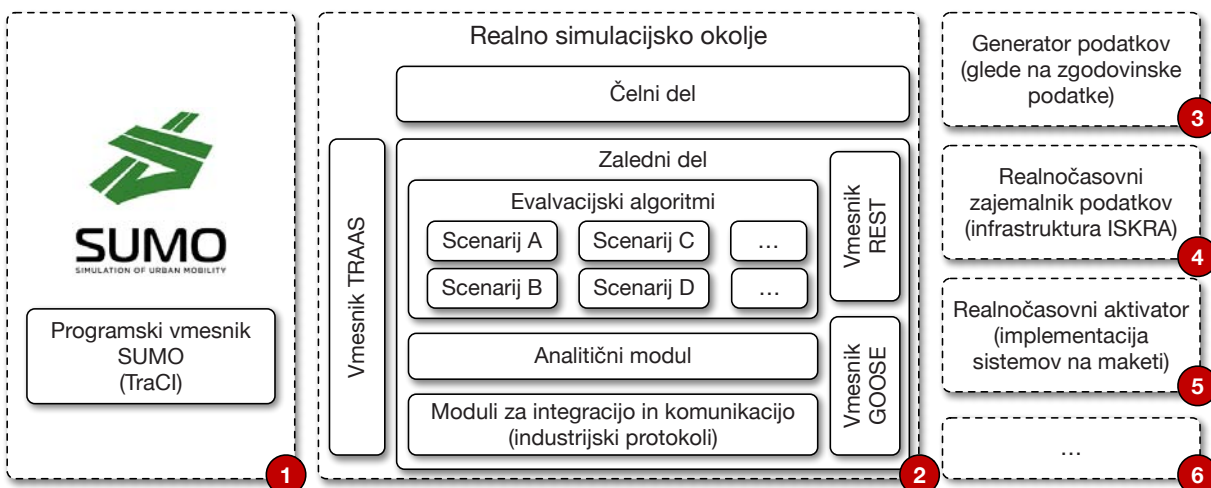
okolje simulatorja SUMO. V teku projekta so se pojavile nove želje po integraciji, zato smo omogočili dodatne protokole (npr. OpenIEC61850 Goose in MQTT) ter preko vmesnikov omogočili povezljivost s komponentami za generiranje podatkov. Komponente za realnočasovno komunikacijo z našim sistemom so bile tudi neposredno integrirane s testno infrastrukturo podjetja ISKRA. Zasnova arhitekture je predstavljena na Sliki 2.

Glavna komponenta realnega simulacijskega okolja je sestavljena iz zalednega sistema, ki temelji na programskem jeziku Java in vsebuje modul z vsemi tremi simulacijskimi algoritmi, opisanimi v prejšnjem poglavju. Algoritmi komunicirajo s SUMO orodjem preko protokola SOAP (angl. Simple Object Access Protocol) z uporabo Java knjižnice TraaS, ki neposredno komunicira z vtičnikom SUMO. Takšen način komunikacije omogoča interakcijo v obe smeri in sicer:

- podatki s simulacije SUMO se lahko pošiljajo v naš zaledni sistem (npr. informacije o stanju indukcijskih zankah) ter
- zaledni sistem lahko spreminja lastnosti med izvajanjem simulacije (npr. logika semaforških križišč).

Zaledni sistem omogoča HTTP REST protokol komunikacije, ki čelnem delu oz. spletnemu grafičnemu uporabniškemu vmesniku omogoča pripravo parametrov ter pregledovanje metrik simulacije že med samim izvajanjem. Poleg tega na željo konzorcijskega partnerja ISKRA, zaledni sistem podpira tudi protokol OpenIEC61850 Goose, ki omogoča izjemno hitro komunikacijo med strojno opremo na realnemu prometnemu omrežju (v primeru, da se komponento za simulacijo vozil priklopi na realno omrežje).

Grafični uporabniški vmesnik je bil razvit z namenom upravljanja funkcionalnosti, ki niso podprte v orodju SUMO ter predstavitvi trenda simulacije v realnem času z izrisom rezultatov metrik v grafični obliki, ter prikazu ostalih pomembnih podatkov (npr. stanja induktivnih zank). Za namene predstavitve rezultatov je pripravljena tudi stran, ki grafično ponazorijo vse razpoložljive metrike zaključenih simulacij.



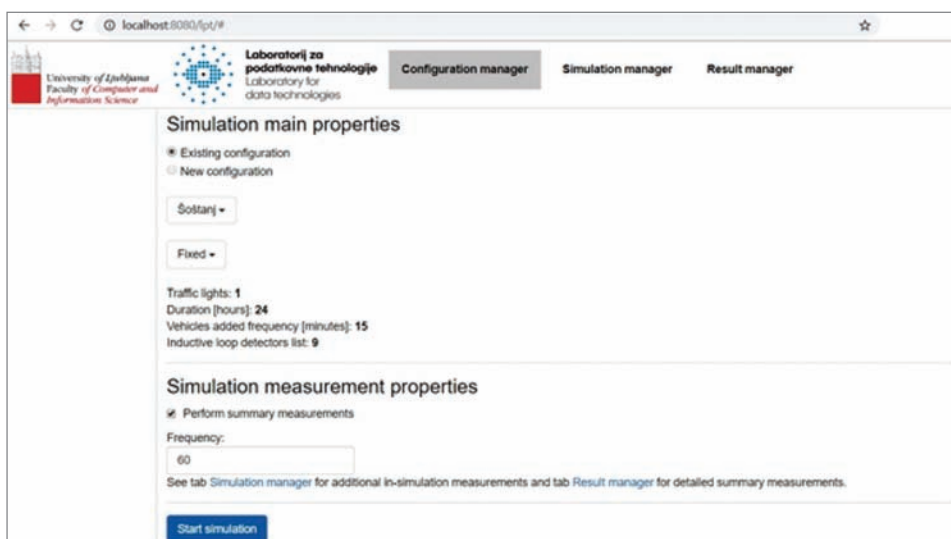
Slika 2: Prikaz razvitega sistema za evalvacijo. Sistem sestoji iz (1) simulatorja prometa SUMO, (2) ogrodja za izvedbo simulacij, komunikacijo in analizo, (3) generatorja podatkov, (4) zajemalnika iz realnih sistemov in (5) sistema za proženje akcij. Možna je vključitev poljubnega dodatnega sistema s komunikacijo preko protokolov REST/GOOSE (6).

3.2.3 Spletni uporabniški grafični vmesnik

Spletni grafični uporabniški vmesnik je bil razvit z namenom poenostavljenega upravljanja simulacije, njeno interakcijo v realnem času in intuitiven grafični ter numerični pregled rezultatov simulacije. Vmesnik je osnovan na podlagi treh zaslonskih form, ki so dostopne preko zavihkov, in sicer:

- konfiguracijski vmesnik (angl., configuration manager),
- simulacijski vmesnik (angl., simulation manager) in
- vmesnik rezultatov (angl., result manager).

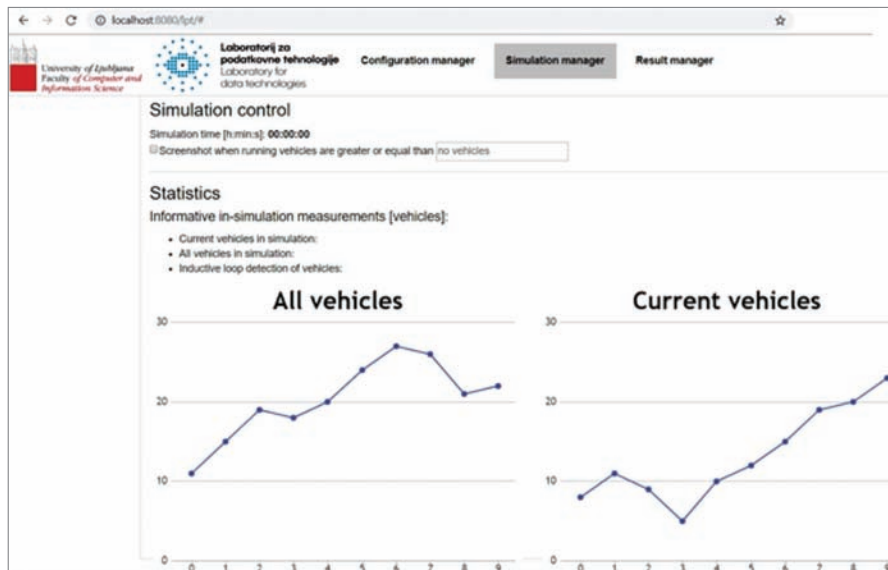
Konfiguracijski vmesnik je namenjen pripravi parametrov želene konfiguracije in njenemu zagon. V osnovi vmesnik omogoča izbiro obstoječe konfiguracije, kjer se ob izbiri prikažejo osnovne lastnosti konfiguracije na podlagi XML metapodatkov konfiguracije ter vhodnih podatkov o vozilih. Te osnovne lastnosti opišejo število semaforiziranih križišč v simulaciji, trajanje simulacije, periodo dodajanja vozil (npr. na vsakih 15 minut) in število vseh induktivnih zank v konfiguraciji. Pred zagonom lahko opredelimo tudi dodatne parametre, kot je npr. način shranjevanja končnih rezultatov simulacije po zaključku ali definiramo periodo beleženja podatkov. Primer konfiguracijskega vmesnika prikazuje Slika 3.



Slika 3: Primer konfiguracijskega vmesnika pri izbiri simulacije prometnega omrežja Šoštanj.

Simulacijski vmesnik služi uporabniku kot orodje za grafični pregled metrik tekom izvajanja simulacije. Poudarek je pri metrikah, ki se nanašajo na vozila, kot so npr. globalen prikaz števila vozil v simulaciji ali lokalni prikaz za krajše časovno obdobje (npr. v zadnjih 15 minutah). Za lažjo analizo kritičnih obremenitev nam vmesnik omogoča tudi dolo-

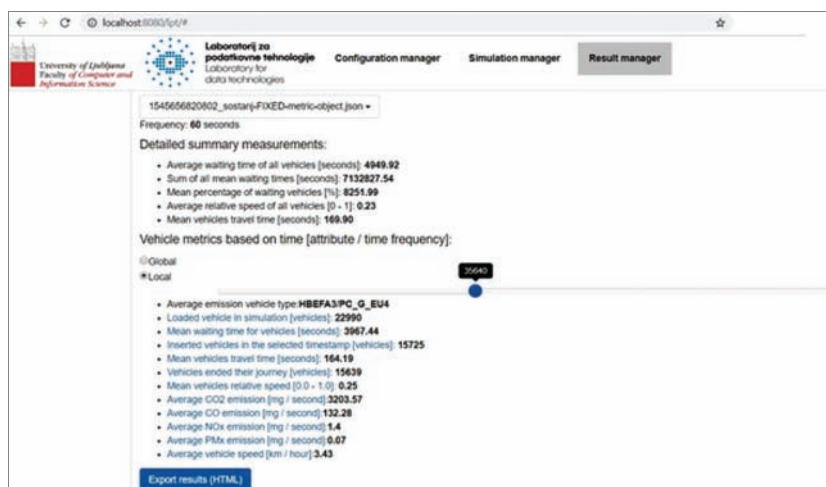
čitev praga nad katerim se shrani zaslonska maska prometnega omrežja, ko je ta prag presežen. Na tak način nam simulator omogoča prikaz stanja ob visokih obremenitvah prometa med izvajanjem in tudi globalnih maksimumov števila vozil v prometnem omrežju. Primer simulacijskega vmesnika prikazuje Slika 4.



Slika 4: Primer simulacijskega vmesnika, ki omogoča prikaz osnovnih podatkov o vozilih med izvajanjem simulacije ter možnost beleženja zaslonskih mask ob preseženem pragu – št. vozil v prometnem omrežju.

Vmesnik rezultatov služi za pregled metrik iz zaključenih simulacij. Uporabnik izbere željeno konfiguracijo ter dobi na voljo osnovne povzetke simulacije (npr. kumulativno povprečje mirovanja vozil v simulaciji) ter možnost pregleda globalnih ter lokalnih metrik. Lokalne metrike so prikazane za celotno obdobje

trajanja simulacije, v našem primeru 24ur (86.400 sekund). Lokalne metrike nam omogočajo še podrobnejši pregled metrik v izbranem trenutku glede na frekvenco vzorčenja. Primer vmesnika rezultatov fiksne scenarija Šoštanj prikazuje Slika 5. Podrobnejši opis metrik pa je podan v naslednjem poglavju.



Slika 5: Primer vmesnika rezultatov pri pregledu lokalnih metrik ob 9uri 54min (sekunda 35.640).

4 EVALVACIJA IN DISKUSIJA

V tem poglavju je opisan postopek evalvacije s predstavitvijo metrik, metodologijo ter opisom testnih križišč, ki so osnova za pripravo rezultatov simulacije testnih križišč s primerjavo algoritmov. Rezultati zajemajo predvsem kvantitativne podatke razpoložljivih metrik, kjer so razvidne razlike delovanja posameznih algoritmov.

4.1 Predstavitev metrik

Simulator prometa SUMO ponuja nabor metrik, ki jih glede na vrsto vzorčenja delimo na dvoje - globalne ter podrobne. Globalne metrike nam nudijo informacije v realnem času tekom izvajanja simulacije in v povzetku simulacije. Podrobnih metrik ni mogoče pridobivati v orodju SUMO v realnem času, saj se metrike zapišejo v obliki dnevniških zapisov (angl., log), pri čemer so izračunane ob zaključku simulacije. Njihov izračun je definiran glede na obdobje oz. periodo beleženja metrik (npr. vsakih 60 sekund).

Razpoložljive globalne metrike so:

- Trenutno število vozil v simulaciji [vozila],
- trenutno število vseh vozil umeščenih v simulacijo [vozila] in
- število zabeleženih vozil na (induktivnih) zankah v posameznem semaforškem ciklu [vozila].

Razpoložljive podrobne metrike so:

- Povprečen čas vseh vozil v mirovanju (čakanje) [sekunde],
- kumulativen seštevek časa čakanja vseh vozi [sekunde],
- razmerje vozil indikatorskih vrednosti ali je vozilo čakalo ali ne [število],
- povprečna hitrost vseh vozil čez simulacijo [km / uro],
- atributi vozil v posameznem trenutku [atribut / časovna frekvenca] kot npr.:
 - število naloženih vozil v simulaciji [vozila],
 - trenutno število vozil v simulaciji [vozila],
 - povprečen čas čakanja vozila [sekunde],
 - dodana vozila glede na predhodno časovno frekvenco [vozila],

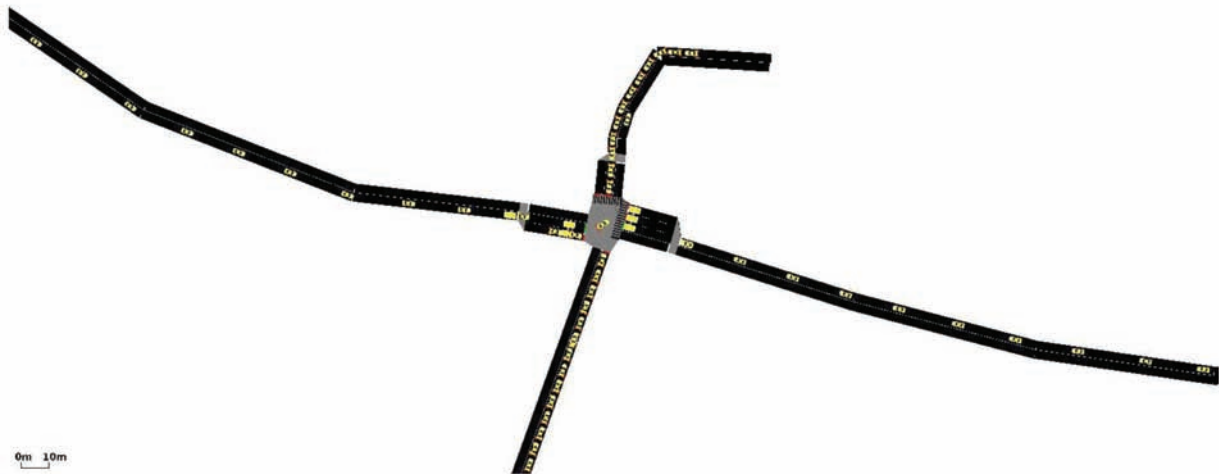
- povprečen čas potovanja vozila (od začetka do cilja) [sekunde],
- število vozil z zaključenim potovanjem glede na predhodno časovno frekvenco [vozila],
- povprečna relativna kumulativna hitrost vozil glede na omejitve [0.0 - 1.0],
- povprečen čas potovanja na posameznih cestnih odsekih [sekunde] in
- podrobne informacije individualnega vozila z možnostjo povprečiti na posamezno časovno frekvenco ali globalno:
 - CO₂ izpusti [mg / sekundo],
 - CO izpusti [mg / sekundo],
 - NO_x izpusti [mg / sekundo],
 - PM_x oz. izpusti trdih delcev [mg / sekundo],
 - Emisijski tip vozila (npr. EURO5, EURO6 itd.),
 - Hitrost vozila [km / uro] in
 - kategorija vozila (npr. osebni avtomobil, tovornjak, kolesar, pešec itd.).

4.2 Opis testnih križišč

Simulacija optimizacijskih algoritmov poteka na dveh različnih scenarijih oz. prometnih omrežjih. V prvem primeru gre za samostojno križišče v Šoštanju, v drugem primeru pa za prometno omrežje v Podutiku, grajeno s štirimi križišči. V nadaljevanju so povzete lastnosti obeh prometnih omrežij ter njihovo preslikavo v orodju SUMO.

4.2.1 Šoštanj

Prvi scenarij je sestavljen le iz enega semaforiziranega križišča. Značilnost križišča je povišana pretočnost križišča na podlagi cepitve prometnih pasov iz enega na dva z zahoda in cepitve iz enega prometnega pasu na tri iz vzhoda ter severa. Križišče za osnovno delovanje zadostuje pogojem opisano v vhodnih podatkih, za adaptivna algoritma pa so dodane še indukcijske zanke na vzhodu, zahodu ter jugu, skupno 8 zank. Pri gradnji križišča je bilo potrebno upoštevati ustrezne dolžine cepitev prometnih pasov in pripraviti možnosti prehoda vozil s prepovedjo polkrožnega obračanja. Prikaz scenarija v orodju SUMO je prikazan na Sliki 6.

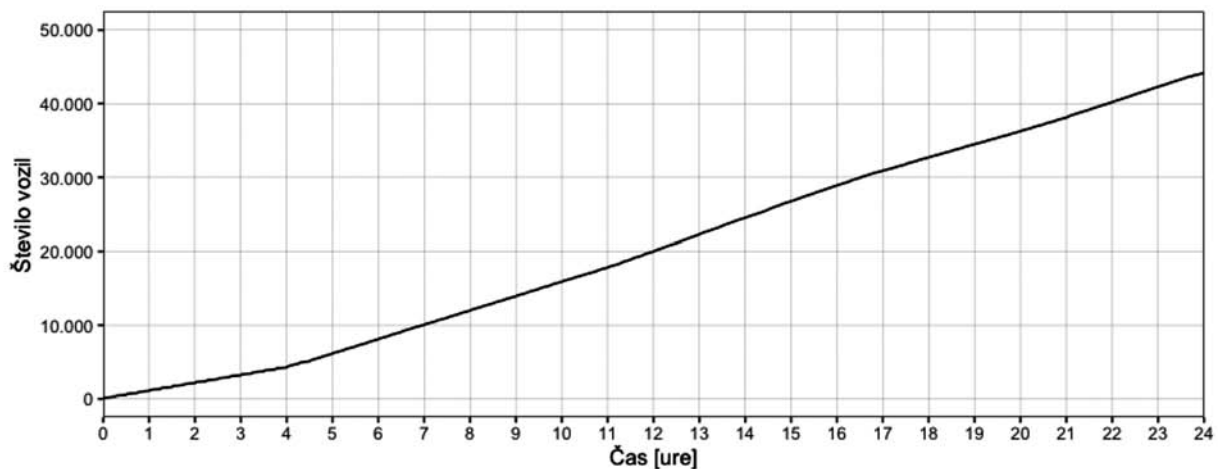


Slika 6: Prikaz scenarija križišča v Šoštanju, ki je vozlišče na lokalni cesti ter povezuje večje kraje kot Lokovica, Šoštanj in Velenje ter premogovnik.

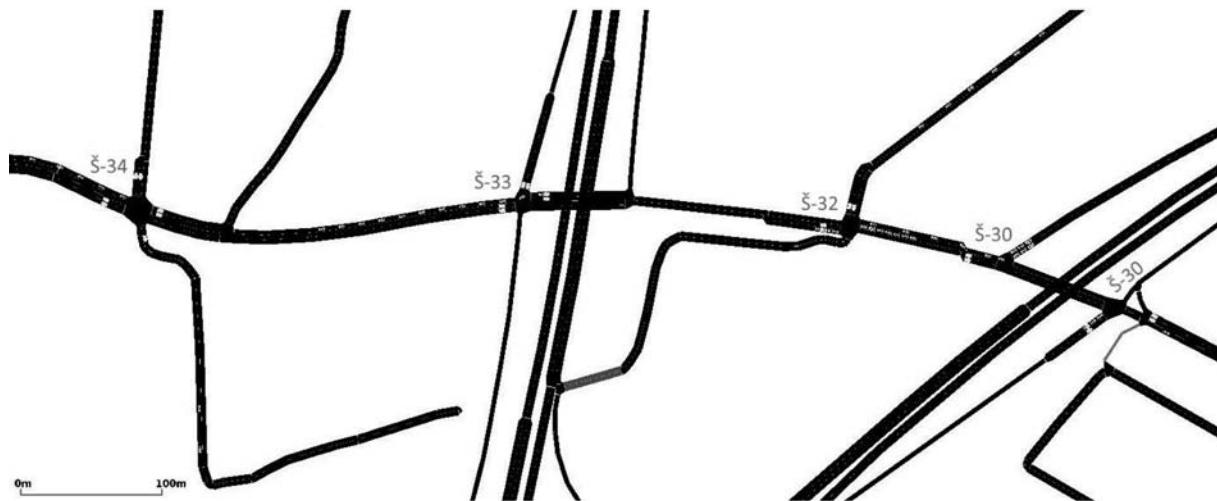
Ne glede na izbrano vrsto algoritma se vozila v simulaciji dodajajo na podlagi predhodnih meritev pridobljenimi v realnem okolju. Na podlagi meritev se določijo vozila, ki vstopajo v simulacijo s smerjo prihoda ter številom vozil, ki v simulacijo vstopajo enakomerno vsako sekundo iz posamezne smeri kot prikazuje Slika 7. Dodajanje vozil je deterministično določeno s strani podjetja Iskra d.d. na podlagi predhodnih študij obremenjenosti dotičnih omrežij in tako predstavljajo dovolj dober približek običajni prometni obremenitvi v obdobju enega dne.

4.2.2 Podutik

Drugi scenarij predstavlja prometno omrežje sestavljeno iz 4 križišč pri čemer je eno križišče deljeno na dve (Š-30). Glede na količino prevoženih vozil v križiščih je najbolj obremenjeno križišče Š-32. Takšna križišča se v notaciji algoritmov imenujejo dominantna križišča, pri čemer so vsa ostala križišča v prometnem omrežju odvisna in se temu primerno obravnavajo pri izračunih v adaptivnih algoritmih. Vsako križišče ima induktivne zanke na vseh vhodnih prometnih pasovih z izjemo križišča Š-30, ki se obravnava kot eno križišče z enim skupnim programom krmiljenja semaforjev.



Slika 7: Število dodanih vozil v simulacijo Šoštan na X osij za časovno obdobje 24ur, kjer Y os ponazarja število vseh vozil, ki so v prometnem omrežju nastopali do določenega časa v dnevu.



Slika 8: Prikaz scenarija križišč v Podutiku, ki povezuje Podutik, Kamno Gorico ter ljubljansko obvoznico s Kranjem.

Značilnost tega prometnega omrežja je prehod iz večjih hitrosti izvozov z avtoceste ljubljanske obvoznice na lokalni medkrajevni promet s temu primernimi omejitvami hitrosti. Grafična predstavitev scenarija v orodju SUMO je prikazana na Sliki 8.

Ne glede na izbrano vrsto algoritma se vozila v simulaciji dodajajo v vseh primerih enako, kar prikazuje Slika 9.

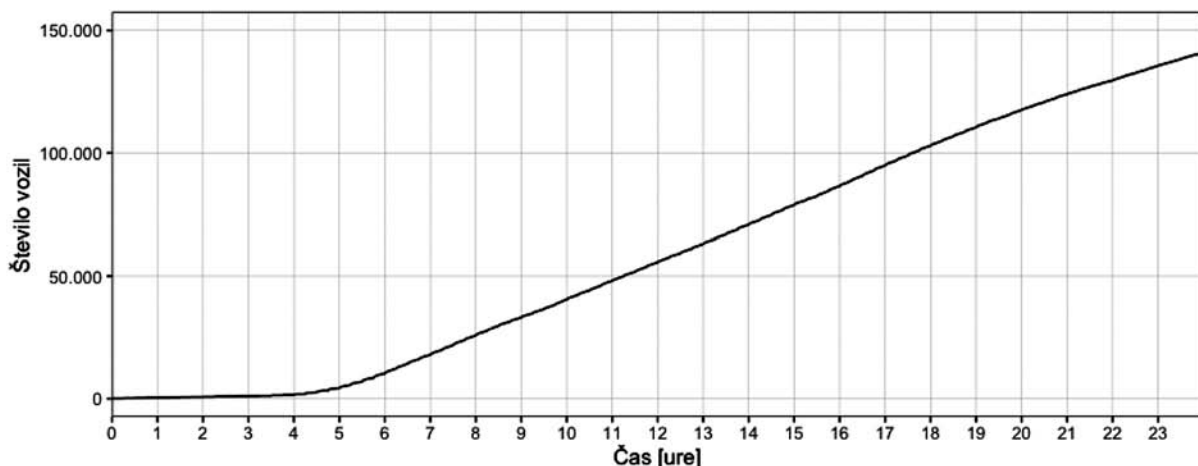
4.3 Metodologija

Metodologija se osredotoča na empirično analizo treh razpoložljivih algoritmov na predstavljenih prometnih omrežjih, pri čemer je število prihodov vozil v prometno omrežje vhodni podatek poleg topologije posameznih semaforiziranih križišč.

Pri testiranju algoritmov smo rezultate pridobili s pomočjo orodja SUMO, kjer smo beležili vse razpoložljive metrike v celotnem času simulacije celega dne. Količina razpoložljivih (surovih) rezultatov metrik vseh XML strukturiranih podatkov prometnega omrežja Šoštanj je velikostnega reda približno 10 GB. V primeru prometnega omrežja Podutik, ki je po zasnovi kompleksnejše, je količina surovih podatkov velikostnega reda 30 GB za celotno simulacijo.

4.4 Rezultati in diskusija

Pri zagonu simulacij smo z orodjem SUMO pripravili surove zabeležbe simulacij v XML obliki. Iz teh smo izločili osnovne metrike simulacij v JSON obliki za izbrano periodo vzorčenja 60 sekund, ki so pred-



Slika 9: Število dodanih vozil v simulacijo Podutik na X osij za časovno obdobje 24ur, kjer Y os je ponazarja število vseh vozil, ki so v prometnem omrežju nastopali do določenega časa v dnevu.

stavljene v nadaljevanju. Vizualizacija rezultatov je v celoti pripravljena s spletnim grafičnim vmesnikom. Glede na to, da so simulacijski algoritmi deterministični, so bile vse simulacije zagnane le po enkrat.

4.4.1 Šoštanj

Prometno omrežje Šoštanj je sestavljeno iz enega semaforiziranega križišča, ki povezuje največja okoliška mesta in industrijo. Simulacije so bile izvedene z enakimi vhodnimi podatki dodajanja vozil - čez celotno simulacijo kumulativno približno 42.000 vozil. Osnovne časovne metrike so prikazane v Tabeli 2. Čas polno prometno odvisnega programa je najdaljši, saj očitno prihaja do zasičenja vozil, pri ostalih dveh programih pa se razlikuje glede na izbrano metriko.

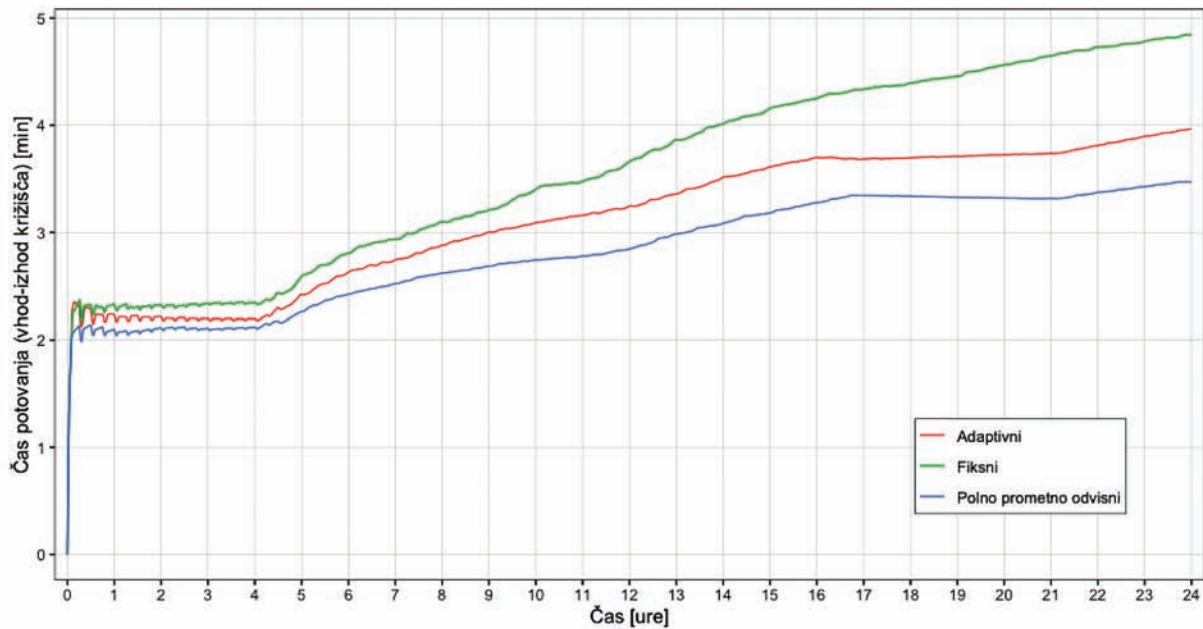
Iz rezultatov simulacij prometnega omrežja Šoštanj je s Slike 10 razvidno, da fiksni program z najbolj togo logiko deluje najslabše oz. ima povprečno vozilo najdaljši potovalni čas pri večjih obremenitvah omrežja od jutranje konice naprej. Polno prometno odvisni program v tem primeru dosega najboljše rezultate, pri čemer je značilnost algoritma, da omogoča podaljšanja zelenih ciklov glede na vnaprej določen maksimum. Adaptivni program v tem primeru

podaljšanja manj ugodno obravnava, saj s prevelikim podaljševanjem vozil iz vzhoda proti zahodu in obratno, obenem podaljšuje čas potovanja vozil s severa ali južne poti. Vse rezultate bi bilo mogoče dodatno izboljšati pri podaljševanju poti prometnega omrežja, saj sta severna in južna stran bistveno krajši, kar povzroča manj enakomerno hitrost vozil zaradi nenehnih pospeševanj ter zaviranj vozil. Dodatna možnost za izboljšavo bi zagotovo bila vpeljava dodatne induktivne zanke za vozila, ki vstopajo s severne strani.

Relativna hitrost se glede na čas simulacije razlikuje na posameznem programu v normirani obliki, ki je grafično povzet na Sliki 11. Fiksni program se pričakovano pretežno izkazuje kot najslabši, saj dosega najvišje vrednosti in posledično najnižjo povprečno hitrost vozil. Ostala programa se pa glede na stopnjo zasičenosti dosegata primerljive rezultate. V primerih, ko pride do izrazite spremembe v relativni hitrosti v kratkem časovnem obdobju pomeni, da prihaja do nenadnih zastojev. Ob višanju relativne hitrosti zastoji prehajajo v tekoč promet. To je najbolj izrazito v obdobju po popoldanski prometni konici po 17:00 uri, ko promet upada.

Tabela 2: Rezultati časov trajanja vozil glede na različne prometne algoritme na prometnem omrežju Šoštanj.

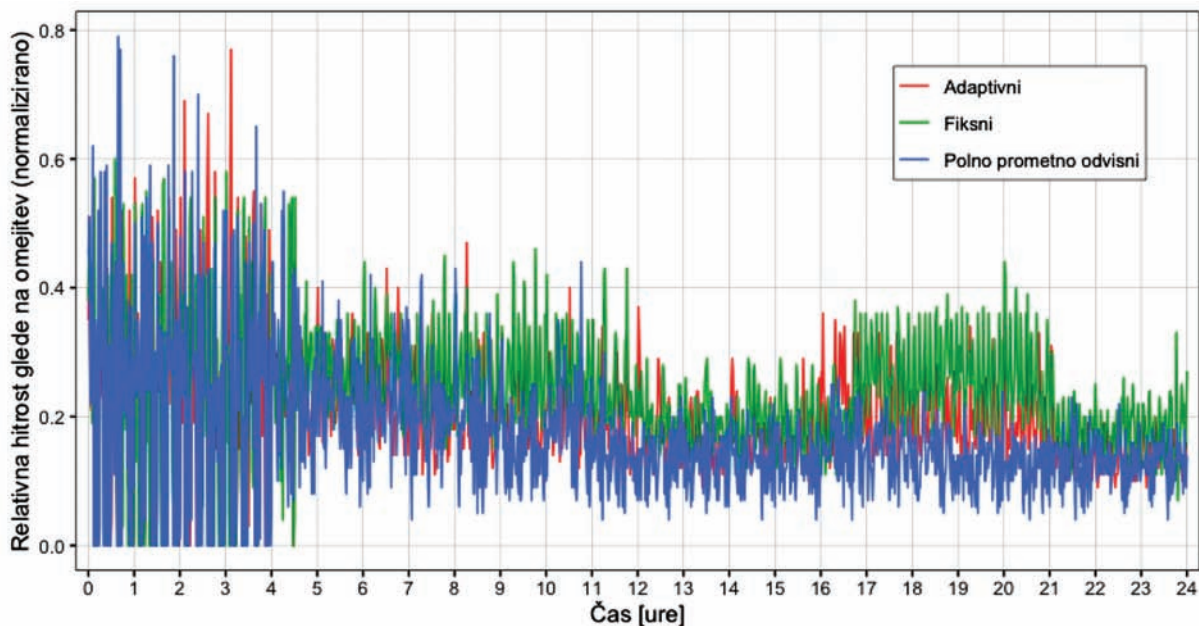
Metrike	Fiksni program	Polno prometno odvisni program	Adaptivni program
Povprečni čas čakanja vseh vozil [sekund]	6.734,82	4.949,92	5.976,91
Vsota vseh srednjih časov (mediana) v celotni simulaciji [sekund]	9.704.875,62	7.132.827,54	8.612.728,82
Srednji časi čakanja vozil [sekund]	8.190,52	8.251,99	9.447,20
Povprečna relativna hitrost vseh vozil v razponu [0-1]	0,17	0,23	0,20
Mediana potovanja vozila [sekund]	215,2	196,90	188,43



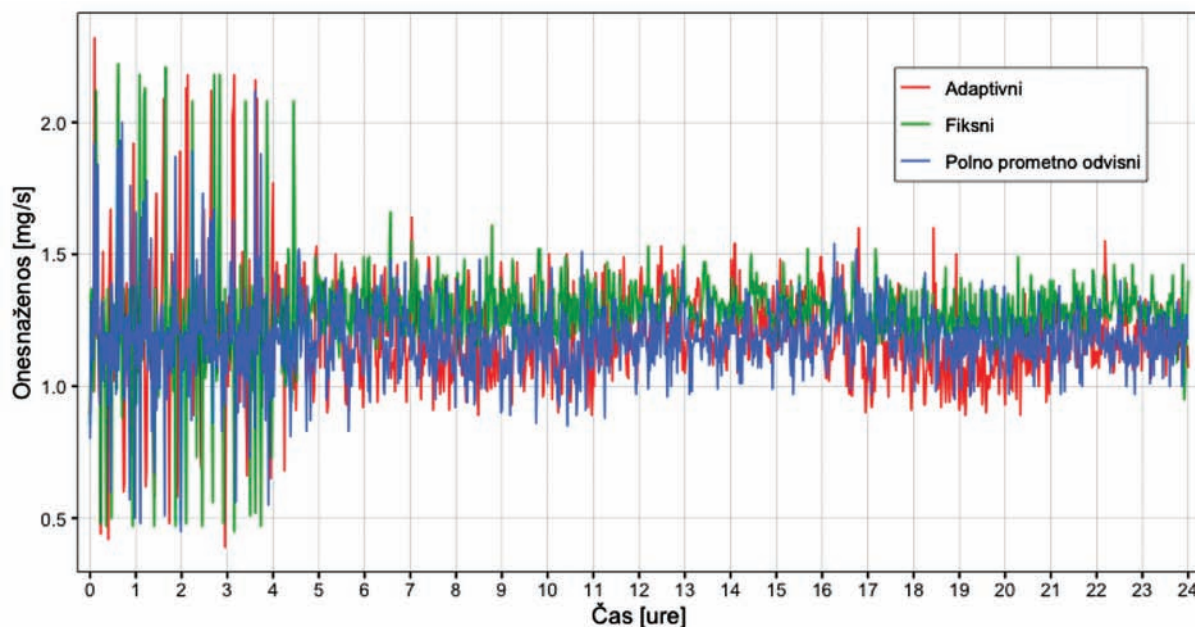
Slika 10: Srednja vrednost časa potovanja vozil za simulacijo Šoštanj. Višja kot je vrednost na Y osi nižje so hitrosti in posledično višja so zasičenja v omrežju.

Na Sliki 12 je predstavljena emisijska onesnaženost z delci NO_x pri vseh prometnih programih, pri predpostavki, da vozila vozijo z motorji standarda Euro5, ki je najpogosteje zastopan na slovenskih cestah. Stopnja onesnaženosti je pogojena s pospeše-

vanjem in zaviranjem vozil, kjer spremembe hitrosti terjajo višjo porabo goriva in posledično večjo onesnaženost. Rezultati nakazujejo, da fiksni program povzroča največjo onesnaženost v primerjavi z ostalima dvema programoma.



Slika 11: Srednja vrednost relativne hitrosti vozil za simulacijo Šoštanj. Višja kot je vrednost na Y osi višje so hitrosti in posledično manjša so zasičenja v omrežju.



Slika 12: Povprečna onesnaženost vozil v prometnem omrežju Šoštanj za čas trajanja simulacije.

4.4.2 Podutik

Prometno omrežje Podutik je sestavljeno iz štirih semaforiziranih križišč. Simulacije so bile izvedene z enakimi vhodnimi podatki dodajanja vozil, čez celotno simulacijo približno 140.000 vozil kumulativno. Osnovne metrike osredotočene na čas, so prikazane v Tabeli 2. Časi programov se glede na metriko razlikujejo, pri čemer se togi fiksni program ne izkaže najslabše v vseh primerih.

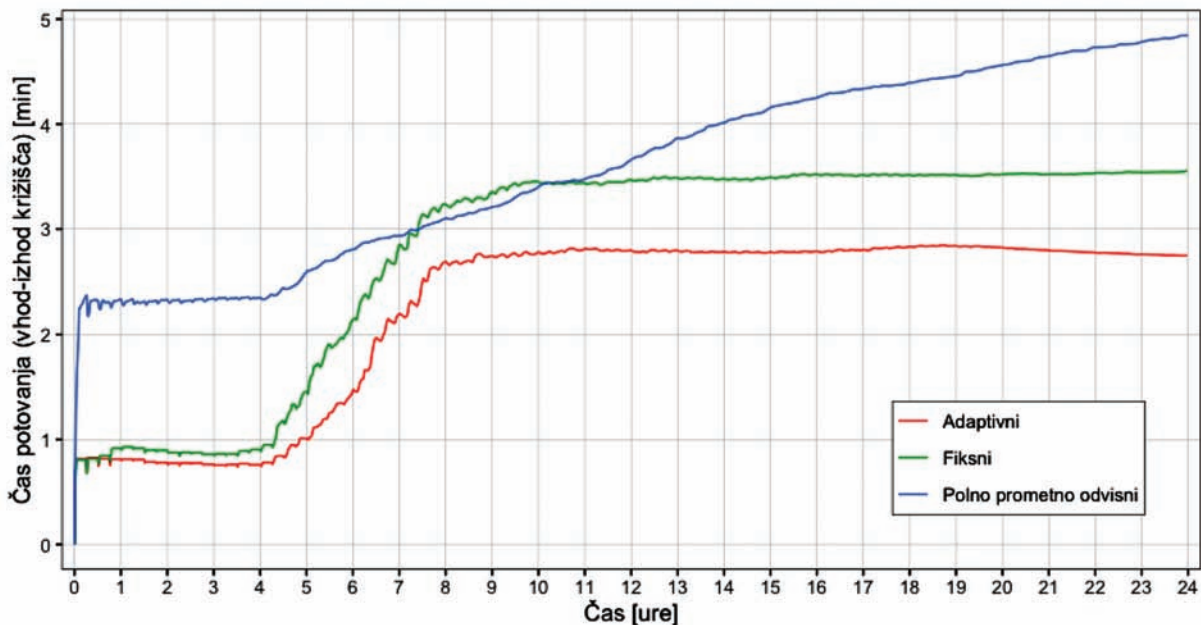
4.4.3 Primerjava rezultatov algoritmov

Iz rezultatov simulacij prometnega omrežja Podutik je s Slike 15 razvidno, da fiksni program s togim delovanjem deluje primerljivo s polno prometno odvisnim programom, kjer ima polno prometno odvisni

program hitrejša časa potovanja pri manjših obremenitvah prometnega omrežja. Polno prometno odvisni program pri večjih prometnih obremenitvah s podaljševanjem zelenih faz zapira hiter prehod križišč iz drugih strani. Kot najbolj obetaven algoritem se je v tem primeru izkazal adaptivni program, ki v povprečju bolj optimalno upravlja cikle in vidno skrajša potovalne čase vozil. Smiselno je omeniti, da tudi v tem prometnem omrežju ni natančno definirano, kje natančno vozila vstopajo v simulacijo, zato bi bilo bolj objektivno simulacijo izvesti na prometnem omrežju z daljšimi cestnimi odseki. V vseh treh primerih je na podlagi podatkov orodja SUMO srednja vrednost relativne hitrosti prikazala anomalije v simulatorju pri uporabi različice 0.31.

Tabela 3: Rezultati časov trajanja vozil glede na različne prometne algoritme na prometnem omrežju Podutik.

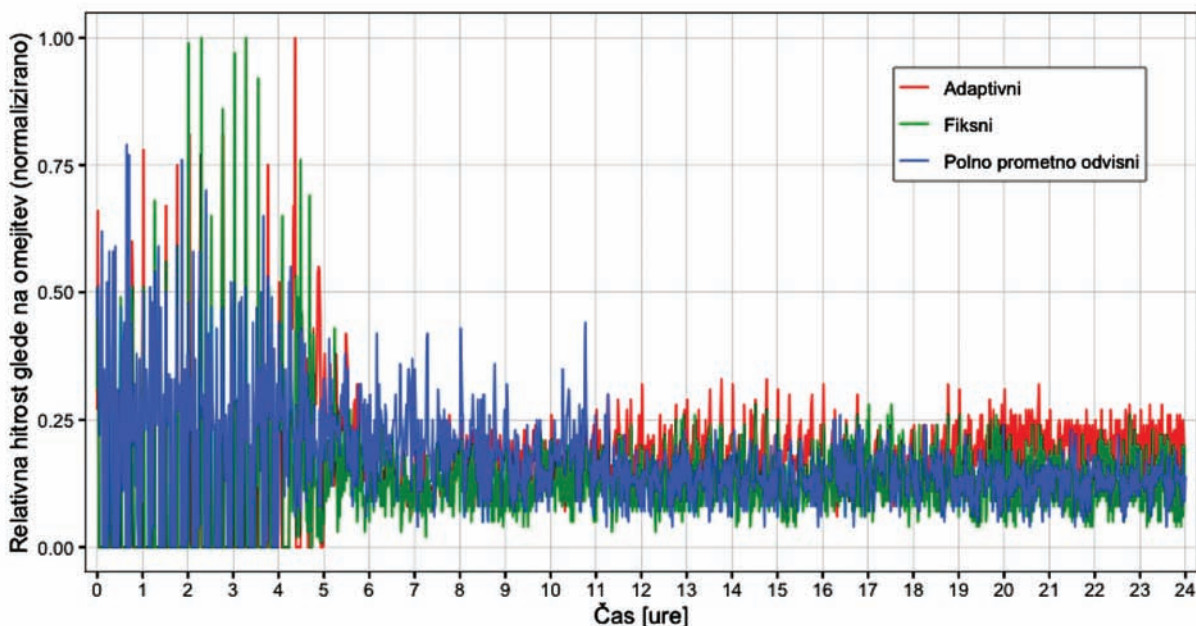
Metrike	Fiksni program	Polno prometno odvisni program	Adaptivni program
Povprečni čas čakanja vseh vozil [sekund]	3.511,92	1.513,51	3.477,96
Vsota vseh srednjih časov (mediana) v celotni simulaciji [sekund]	5.057.168,26	5.751.135,37	5.008.257,33
Povprečna relativna hitrost vseh vozil v razponu [0-1]	0,14	0,15	0,18
Mediana potovanja vozila [sekund]	169,20	159,52	135,19



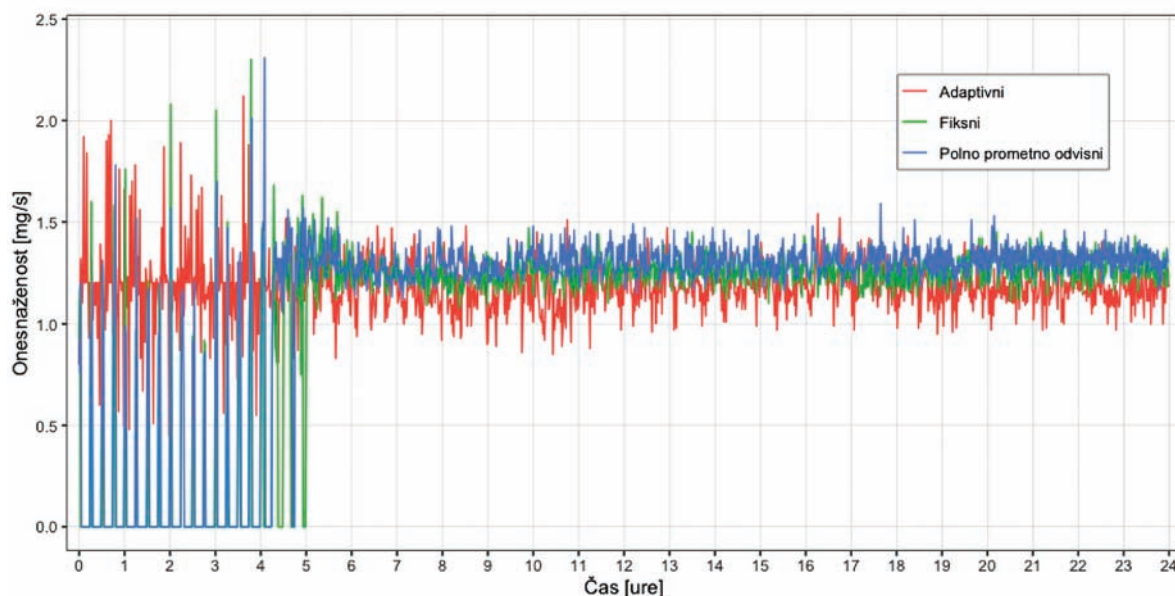
Slika 13: Srednja vrednost časa potovanja vozil za simulacijo Podutik. Višja kot je vrednost na Y osi nižje so hitrosti in posledično višja so zasičenja v omrežju.

Relativna hitrost se glede na čas simulacije razlikuje na posameznem programu v normirani obliki, ki je grafično povzet na Sliki 13. Polno prometno odvisni program se že na začetku zasiči in vrača najslabše rezultate. Adaptivni program ima podobna zasičenja kot fiksni program, pri čemer se pri adaptivnem čas potovanja v povprečju skrajša za 30 sekund.

Relativna hitrost se glede na čas simulacije razlikuje na posameznem programu v normirani obliki, ki je grafično povzet na Sliki 14. Kot najslabši se izkaže polno prometno odvisni program, saj privzeti pragovi niso primerni za konkretno simulacijo na podlagi vozil, ki se konstantno linearno kopičijo v simulaciji. Fiksni program dosega relativno dobre



Slika 14: Srednja vrednost relativne hitrosti vozil za simulacijo Podutik.



Slika 15: Povprečna onesnaženost vozil v prometnem omrežju Podutik za čas trajanja simulacije.

rezultate kjer pride pri adaptivnem programu do dodatnega izboljšanja za povprečno 30 sekund pri potovanju vsakega vozila.

Na Sliki 15 je predstavljena emisijska onesnaženost z delci NO_x pri vseh prometnih programih, pri predpostavki, da vozila vozijo z motorji standarda Euro5, ki je najpogosteje zastopan na slovenskih cestah. Stopnja onesnaženosti je pogojena s pospeševanjem in zaviranjem vozil, ki se v prometnem omrežju odražajo z višjo onesnaženostjo z začetkom jutranje prometne konice po 5:00 uri. Rezultati nakazujejo, da fiksni program povzroča največjo onesnaženost v primerjavi z ostalimi programi.

5 SKLEP

Na Svetu predstavlja preučevanje optimizacije prometnih omrežij še vedno velik izziv zaradi velike količine dejavnikov, ki na lahko simulacijo vplivajo v realnem okolju. Pri vnaprej znanih vhodnih podatkih lahko z orodjem SUMO preučujemo odzive različnih algoritmov ali pa druge dejavnike, kot so maksimalne prometne obremenitve posameznih cestnih odsekov ali semaforiziranih križišč, in s tem zmanjšamo čas potovanja vozil ter obenem znižamo onesnaženost okolja.

V delu je predstavljen celovit sistem za simulacijo optimizacijskih algoritmov na dveh različnih prometnih tokovih. V delu smo se najprej posvetili definiciji podatkov ter predstavitvi posebnosti prometnih omrežij, na katerih smo testirali optimizacijske algo-

ritme. Na podlagi podatkov, razpoložljivih tehnologij in potencialnih možnosti za integracijo, smo predstavili arhitekturo rešitve in podporna orodja kot spletni grafični vmesnik za namene evalvacije. Implementirana rešitev podpira tri optimizacijske algoritme ter dve testni prometni omrežji z možnostjo nadaljnjih nadgradenj. Na podlagi razvitega spletnega orodja za analitiko smo povzeli osnovne značilnosti algoritmov ter njihove globalne lastnosti obnašanja algoritmov v obdobju trajanja posamezne simulacije 24-urnega cikla.

Pri pripravi sistema za simulacijo prometnih tokov smo naleteli na številne izzive s strani komunikacijskih protokolov med posameznimi komponentami sistema. Implementirali smo tudi globalno analitiko rezultatov za celotno obdobje trajanja simulacije, češar orodje SUMO v osnovi ne podpira. Pri integraciji algoritmov je bila priprava vhodnih podatkov semaforске logike otežena zaradi nestandardiziranega formata opisa semaforских logik, ki je pripravljen na rastrski sliki v formatu PDF. Iz simuliranih prometnih omrežij smo ugotovili, da adaptiven algoritem v simulaciji Šoštanj ne deluje bolje od polno prometno odvisnega programa. Pri prometnem omrežju Podutik so rezultati pričakovani. V osnovi ugotovitve kažejo, da adaptivni algoritmi vračajo boljše rezultate od togega fiksnega determinističnega algoritma. V prihodnjem delu se bomo osredotočili na izboljšavo obstoječih algoritmov ter razširili testne podatke z novimi prometnimi omrežji.

ZAHVALA

Raziskava je bila finančno podprta s sredstvi programa EkoSmart s strani Republike Slovenije in Evropske unije iz Evropskega sklada za regionalni razvoj. Raziskava je bila mogoča s prenosom znanja in domenskih izkušenj podjetja ISKRA d. o. o., Stegne 21, 1000 Ljubljana.

LITERATURA

- [1] Akiyama, Takamasa & Okushima, Masashi. (2006). Implementation of cordon pricing on urban network with practical approach. *Journal of Advanced Transportation*. 40. 221 - 248. 10.1002/atr.5670400208.
- [2] Cheng, Y. (2017). Increasing Robustness of Differential Evolution by Passive Opposition, volume 454, pages 85–94.
- [3] Gartner, N. (1990). Opac: Strategy for demand-responsive decentralized traffic signal control. *IFAC Proceedings Volumes*, 23(2):241 – 244. *IFAC/IFIP/IFORS Symposium on Control, Computers, Communications in Transportation*, Paris, France, 19-21 September.
- [4] Hong Wei, Wang Yong, Mu Xuanqin and Wu Yan, »A cooperative fuzzy control method for traffic lights,« *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, 2001, pp. 185-188.
- [5] Hu, Y., Thomas, P., & Stonier, R. (2007). Fuzzy control of traffic signals accompanying pedestrian crossings. In: *Proceedings of the 2007 WSEAS international conference on computer engineering and applications*, Gold Coast, Australia, January 17–19 (pp. 288–292).
- [6] J. Favilla, A. Machion and F. Gomide, »Fuzzy traffic control: adaptive strategies,« *Second IEEE International Conference on Fuzzy Systems*, 1993, pp. 506-511 vol.1, doi: 10.1109/FUZZY.1993.327519.
- [7] Kim, J., A fuzzy logic control simulator for adaptive traffic management, *IEEE International Conference on Fuzzy Systems*, vol. 3, 1997, pp. 1519- 1524.
- [8] Kosonen, I. (2003). Multi-Agent Fuzzy Signal Control Based on Real-Time Simulation. *Transportation Research Part C: Emerging Technologies*, 11(5), 389-403.
- [9] Mirchandani, P. and Fei-Yue Wang (2005). Rhodes to intelligent transportation systems. *IEEE Intelligent Systems*, 20(1):10–15.
- [10] Pappis C. and Mamdani E. (1977). A Fuzzy Logic Controller for a Traffic Junction. *IEEE Transactions on Systems, Man and Cybernetics*. Vol. SMC-7, No. 10, pp. 707–717.
- [11] Sayers, T., Anderson, J. and Bell, M. (1998), »Traffic Control System Optimisation: A Multiobjective Approach«, Griffiths, J.D. (Ed.) *Mathematics in Transport Planning and Control*, Emerald Group Publishing Limited, Bingley, pp. 37-46.
- [12] Sen, S. and Head, K. (1997). Controlled optimization of phases at an intersection. *Transportation Science*, 31(1):5–17.
- [13] Teodorovic, D., 1999. »Fuzzy logic systems for transportation engineering: the state of the art,« *Transportation Research Part A: Policy and Practice*, Elsevier, vol. 33(5), pages 337–364, June.
- [14] Trabia, M., Kaseko, M. S., Ande, M. (1999). A Two-Stage Fuzzy Logic Controller for Traffic Signals. *Transportation Research Part C: Emerging Technologies*, 7(6), 353-367.
- [15] Zhanbo Sun; Wan Li; Xuegang (Jeff) Ban; and Tianyu Huang: *An Adaptive Traffic Signal Control System (ACS-Lite) in Heavily Congested Arterial Traffic: Experiences and Lessons Learned; CICTP 2018 : Intelligence, Connectivity, and Mobility*. 2018.
- [16] Zhang, Y. and Ye, Z. (2008). Short-term traffic flow forecasting using fuzzy logic system methods. *Journal of Intelligent Transportation System* 12 102–112.

Sandi Gec je zaposlen kot asistent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. V raziskovalno-razvojnih projektih se je ukvarjal z uporabo semantičnih tehnologij pri razvoju oblčnih sistemov, bazami znanja ter integracijo podpornih rešitev v oblaku. Svoje znanje je apliciral na Horizon 2020 projektih SWITCH, ENTICE in DECENTER. Trenutno se v okviru Horizon 2020 projekta ONTOCHAIN ukvarja z novimi pristopi tehnologije veriženja blokov, predvsem s pametnimi pogodbami ter komunikacijo med verigami in zunaj verige.

■

Vlado Stankovski je redni profesor računalništva in informatike. Ima bogate izkušnje na področju programskega inženirstva, računalništva v oblaku, na robu in v megli, porazdeljenih sistemov, semantike ter tehnologij umetne inteligence (strojno, globoko učenje). Sodeloval je pri načrtovanju, razvoju in integraciji tehnologij vmesne programske opreme. Sodeloval je pri več nacionalnih in mednarodnih projektih, v konzorciju Superračunalniški center Slovenije, na projektu pametne specializacije IQ DOM ter v gruči za programsko inženirstvo projektov Obzorje 2020 kot predstavnik projektov ENTICE, SWITCH in DECENTER. Vlado Stankovski je znanstveno-tehnični koordinator projekta Naslednje generacije interneta ONTOCHAIN.

■

Marko Bajec je redni profesor na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Je vodja Laboratorija za podatkovne tehnologije in lot Demo Centra. Raziskovalno in aplikativno se ukvarja s podatkovno intenzivnimi sistemi, njihovim razvojem in obvladovanjem. Sodeloval je pri razvoju različnih podatkovnih platform, vključno s platformo za mobilno in elektronsko zdravstvo, platformo za analizo medijev, platformo za simulacijo in upravljanje prometa itd. V zadnjem času se poglobljeno ukvarja z govornimi tehnologijami. V svoji karieri je vodil ali koordiniral več kot 30 raziskovalnih in aplikativnih projektov ter prejel več priznanj in nagrad za prenos znanja v prakso.

■

Slavko Žitnik je docent na Fakulteti za računalništvo in informatiko Univerze v Ljubljani, kjer poučuje predmete s področja podatkovnih baz in obdelave podatkov. Raziskovalno se ukvarja z obdelavo naravnega jezika, predvsem na semantični ravni. Je član več strokovnih združenj, sodeluje pri organizaciji konferenc s področja informatike in pri projektih, povezanih z obdelavo podatkov na področju interneta stvari.