

Optimiziranje prometnega toka semaforiziranega križišča

Šemso Hrnjičič

Univerza v Mariboru, Fakulteta za organizacijske vede

E-mail: semso.hrnjicic@gmail.com

Povzetek

Razvoj simulacijskega okolja pripomore k razumevanju določene problematike, ponuja hitro in cenovno ugodno eksperimentiranje ter ponuja izdelavo podatkov katere lahko uporabimo pri analiziranju in izboljšavi. Cilj projekta je razviti simulacijsko okolje, znotraj tega izboljšati semaforizirano križišče in analizirati različna stanja križišča. Na pretok semaforiziranih križišč vpliva precej veliko faktorjev. Čas, dela na cesti, delovanje drugih semaforiziranih križišč, nesreče, navigacije itd., to vse ima vpliv na križišča v okolici. Zato je tudi toliko pomembno, da ustvarimo nek dinamični sistem, kateri se bo odzival na okolje, učil iz opazovanj ter se skozi čas izboljševal. Opisan sistem smo razvili in ga implementirali v naši simulaciji.

Ključne besede: vizualizacija, razvoj, simulacija, optimizacija, strojno učenje

1 Uvod

Za izdelavo simulacijskega modela smo uporabili orodje Unity. Izbrali smo si določeno semaforizirano križišče ter izmerili pretočnost ob določenem času. Podatke smo uporabili v simulaciji ter pridobili rezultate in jih analizirali. Simulacijski model smo v nadaljevanju optimizirali na tri različne načine.

V primeru nepričakovanih zastojev (na primer zaradi nesreč) obveščeni potniki skrajšajo svoj potovalni čas z zamenjavo poti, vendar zaradi tega alternativne poti postanejo bolj polne, zaradi česar se lahko podaljšujejo potovalni časi za neobveščene voznike. (Wiering, van Veenen, Vreeken, & Koopman, 2004)

V nadaljevanju bomo predstavili vizualizacijo semaforiziranega križišča, delovanje senzorjev, pridobljene podatke, spodbujevalno učenje ter primerjavo rezultatov.

2 Vizualizacija

Vizualizacija je precej pomemben del, saj nam pomaga pri razumevanju problematike in lažjemu predstavljanju različnih sprememb. Z uporabo programske rešitve Blender

smo izdelali 3D model našega izbranega križišča in ga uvezli v Unity. Poleg križišča smo za simulacijo uporabili satelitsko sliko, različne modele avtomobilov in semaforje.

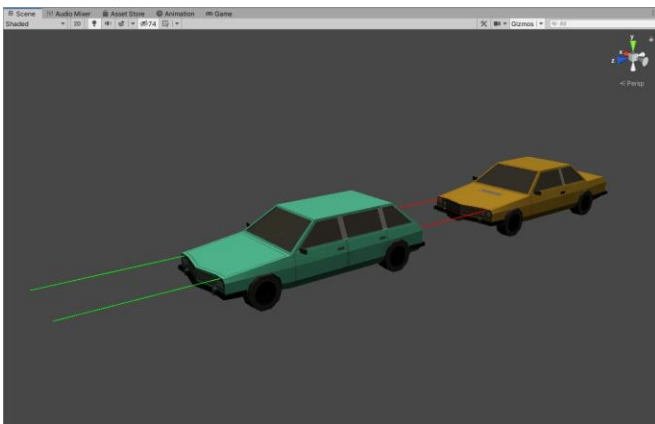
Vse omenjene elemente vidimo na sliki 2.1. Za piko na i smo semaforjem dodali še delujoče zelene, rumene in rdeče luči.



Slika 2.1: Končna vizualizacija križišča.

Da bi avtomobili upoštevali cestno prometne predpise smo jim dodali senzorje s katerimi bodo prepoznali druge avtomobile, kontrolne točke in prednostne točke na cestah.

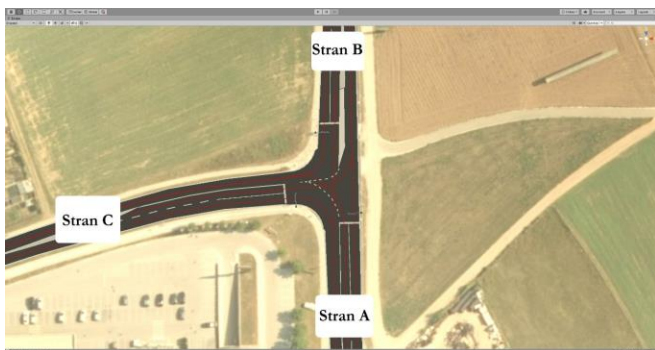
Na sliki 2.2 imamo prikazana dva avtomobila z izrisanimi senzorji. Avtomobil na levi strani ne zazna pred seboj nobene prepreke, zato ima žarka senzorja obarvana v zeleno barvo. Senzor avtomobila na desni strani zazna avtomobil pred seboj, zato se njegova žarka obarvata v rdečo barvo, ter informacijo posreduje avtomobilu. Avtomobil se potem pravilno odzove na posredovane informacije.



Slika 2.2: Vizualizacija senzorjev

3 Simulacija

Na sliki 3.1 vidimo naše izbrano križišče od zgoraj. Na sliki imamo označene tri strani križišča. Stran A, B in C. Strani A in B sta na prednostni cesti, stran C pa je na stranski.



Slika 3.1: Poimenovane strani križišča.

V simulaciji smo uporabili podatke o pretoku, katere smo izmerili na obravnavanem križišču. Podatke smo merili eno uro. S strani A prihaja največ avtomobilov, kar 690 na uro. S strani B prihaja 646 avtomobilov in s strani C jih je samo 120.

V tabeli 3.1 vidimo izmerjeno število avtomobilov ene ure z vseh strani, ter v odstotkih prikazane odločitve voznikov z vsake strani.

Tabela 3.1: Podatki o pretoku križišča.

	Število avtomobilov/h	Levo/h	Naprej/h	Desno/h	Levo v %	Naprej v %	Desno v %
A	690	108	582	0	16	84	0
B	646	0	564	82	0	87	13
C	120	84	0	36	70	0	30

Ko smo končali z razvojem simulacijskega modela smo uporabili izmerjen podatke ter simulirali pretok semaforja za čas ene ure. Vsako stanje semaforiziranega križišča smo simulirali desetkrat in povprečne rezultate analizirali. V tabeli 3.2 imamo prikazane povprečne rezultate simulacij v formatu »hh:mm:ss«. Skupne čakalne čase avtomobilov smo zabeležili za vse smeri posebej. Avtomobili s smeri A so skupaj v povprečju največ časa čakali na semaforju, kar 1 uro in 37 sekund.

Tabela 3.2: Povprečni čakalni čas.

hh:mm:ss	Čakalni čas A	Čakalni čas B	Čakalni čas C	Skupni čakalni čas
	01:00:37	00:57:04	00:51:36	02:49:18

4 Optimizacija

Prvi način je sprememba časov semaforiziranega križišča. To smo naredili tako, da smo vse možne kombinacije prednostne in stranske ceste testirali. Tukaj nam simulacijski model omogoča hitro in cenovno ugodno eksperimentiranje.

Druga optimizacija vključuje implementacijo senzorjev, s katerimi v realnem času spremljamo stanje na cesti ter omogočamo semaforiziranemu križišču, da se odziva na promet bolj dinamično.

Tretja optimizacija je s pomočjo strojnega učenja. S pomočjo odprtokodnega projekta ML-Agents ter odprtokodne knjižnice TensorFlow smo izdelali model nevronske mreže, kateri se odziva na promet.

Vse optimizacije smo testirali in rezultate zapisovali. Povprečne simulacij vseh različnih stanj semaforiziranega križišča smo uporabili pri analiziranju ter primerjavi.

5 Strojno učenje

Z uporabo odprtokodnega paketa ML-Agents bomo trenirali našega agenta (v tem primeru semaforizirano križišče), da se samo odloča in usmerja promet. Spodbujevalno učenje temelji na povratnih informacijah. Za nas je pomembno, da zmanjšamo čakalni čas avtomobilov. To smo tudi uporabili kot oceno uspešnosti agenta.

Spodbujevalno učenje je eno najbolj vznemirljivih in tudi eno izmed najstarejših področij strojnega učenja danes. Obstaja že od petdesetih let dvajsetega stoletja in je čez leta proizvedlo veliko zanimivih aplikacij, zlasti na področju iger (npr. TD-Gammon, Backgammon-playing program) in na področju nadzora stroja, vendar so bile aplikacije redko uvrščene med naslovne novice. Leta 2013 pa se je zgodila revolucija, ko so raziskovalci britanskega podjetja DeepMind pokazali sistem, ki bi se lahko naučil igrati skoraj katero koli igro Atari in je sčasoma postal boljši od ljudi. Pri delovanju uporablja samo neobdelane slikovne pike kot vhod brez predhodnega poznavanja pravil igre. (Géron, 2019)

Na sliki 5.1 imamo prikazane elemente spodbujevalnega učenja. Naš semafor preko senzorjev opazuje križišče, in se odloča kateri strani bo dodelil zeleno luč. Od simulacijskega modela dobiva povratne informacije v obliki nagrade. Nagrado agentu dodelimo v pozitivni ali negativni obliki, glede na skupni čakalni čas avtomobilov simulacije.

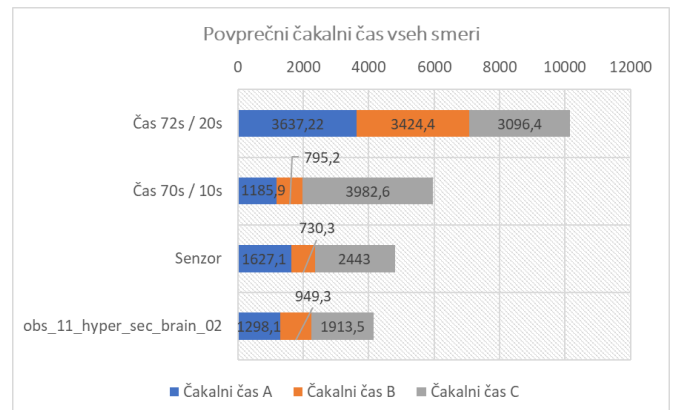


Slika 5.1: Elementi spodbujevalnega učenja.

6 Rezultati

Na sliki 6.1 imamo prikazane vse rezultate zlatih stanj semaforiziranega križišča. Prva vrstica, poimenovana »Čas 72s / 20s« prikazuje nespremenjeno stanje križišča. Druga vrstica »Čas 70s / 10s« prikazuje prvo optimizacijo s

spremenjenim časom, tretja vrstica »Senzor« prikazuje rezultate križišča z implementiranim senzorjem. V zadnji vrstici imamo prikazano izboljšavo semaforiziranega križišča z izdelanim modelom nevronske mreže.



Slika 6.1: Primerjava povprečnih čakalnih časov.

S spremembo časa semaforiziranega križišča smo skupni čakalni čas zmanjšali za približno 41 %. Potem ko smo semaforiziranemu križišču implementirali senzorje, smo zmanjšali skupni čakalni čas za približno 53 %. Na koncu, ko smo uspešno izdelali nevronske mreže ter jo uporabili, smo skupni čakalni čas zmanjšali za približno 59 %.

7 Zaključek

Pri izdelavi simulacijskega modela smo se precej omejili. Osredotočili smo se na promet, izbrali smo si eno križišče in simulirali promet za določen dan ob določeni uri. Resnični svet je precej bolj kompleksen in tudi veliko več faktorjev upliva na promet.

Zato je tudi toliko pomembno, da razvijamo bolj dinamične sisteme, kateri se lahko odzovejo in prilagajajo na različna stanja na cestah. Naslednji korak je vključiti sosednja križišča, saj vsako križišče s svojim delovanjem vpliva na sosednja križišča. To pomeni, da so povezana med seboj in vplivajo na pretočnost drug drugega.

Bibliography

- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. Newton: O'Reilly Media, Inc.
- Wiering, M., van Veenen, J., Vreeken, J., & Koopman, A. (2004). *Intelligent Traffic Light Control*. Utrecht: institute of information and computing sciences, utrecht university.