

Názov: Simulácia adaptívneho riadenia križovatky
Autory: Michal Gabonay (xgabon00), Martin Kocour (xkocou08)
Date: 19. 11. 2018
Fakulta informačných technológií, VUT Brno

1 Úvod

V dnešnej dobe čelia veľké moderné mestá niekoľkým dopravným problémom. Jedným z takých problémov sú jednoznačne dlhé kolóny áut, čakajúce na križovatkách počas dopravnej špičky. Táto práca pojednáva o riešení tohoto problému pomocou adaptívneho riadenia križovatky. Toto riadenie je modelované ako umelý agent, ktorý neustále monitoruje situáciu na križovatkách, a ktorého cieľom je zabezpečiť plynulú premávku.

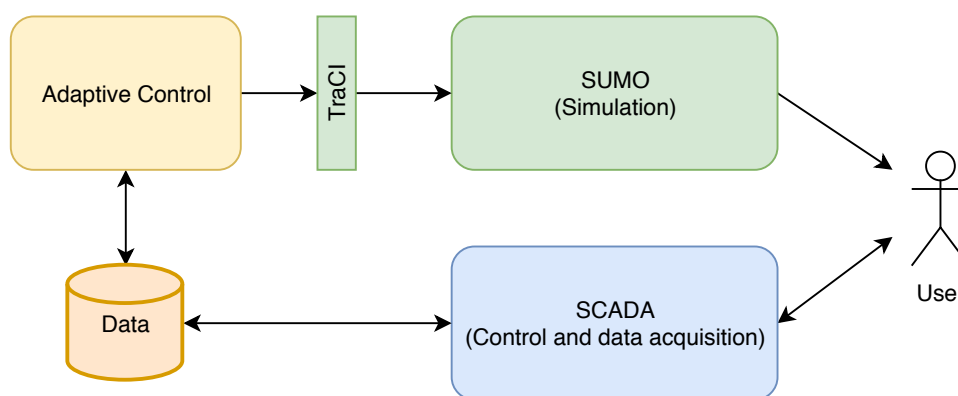
V tomto projekte modelujeme križovátku s niekoľkými senzormi umiestnenými na vozovkách, ktoré umožňujú križovátke zbierať informácie o premávke. Adaptívne riadenie je ovplyvnené najmä počtom áut v jednotlivých pruhoch a umiestnením prioritizovaných dopravných prostriedkov (autobusy).

Navrhovaný prístup kombinuje viaceré nástroje. Umelý agent je implementovaný v jazyku Python. Pre účely simulácie premávky bol použitý nástroj SUMO (Simulation of Urban MObility). Agent a simulácia komunikujú medzi sebou pomocou rozhrania TraCI (Traffic Control Interface), ktoré je súčasťou balíka SUMO. Nazhromaždené dáta sa ukladajú do Redis databázy. Celý systém je možné monitorovať a riadiť pomocou SCADA rozhrania, ktoré komunikuje s touto databázou.

2 Implementácia

Architektúra simulácie adaptívneho riadenia križovatky je znázornená na obrázku 1. Beh križovatky a premávka sú simulované pomocou nástroja SUMO¹. Adaptívne riadenie komunikuje so simuláciou pomocou rozhrania TraCI². Cez toto rozhranie získava informácie zo senzorov o aktuálnej premávke na križovatkách. Nasledovne na základe aktuálneho stavu premávky vyberá najvhodnejší program, viď podkapitola 2.1 a 2.2.

Adaptívne riadenie ukladá do databázy stavy semaforov (červená, oranžová, zelená) a zvolený program. V rámci projektu bola zvolená jednoduchá Redis³ databáza. Táto databáza ukladá jednotlivé informácie do pamäte vo forme dvojíc (kľúč a hodnota). S týmito informáciami ďalej pracuje SCADA⁴ rozhranie, ktoré okrem vizualizácie dát umožňuje aj manuálne riadiť celý systém, viď podkapitola 2.3.



Obr. 1: Architektúra aplikácie.

¹<http://sumo.dlr.de>

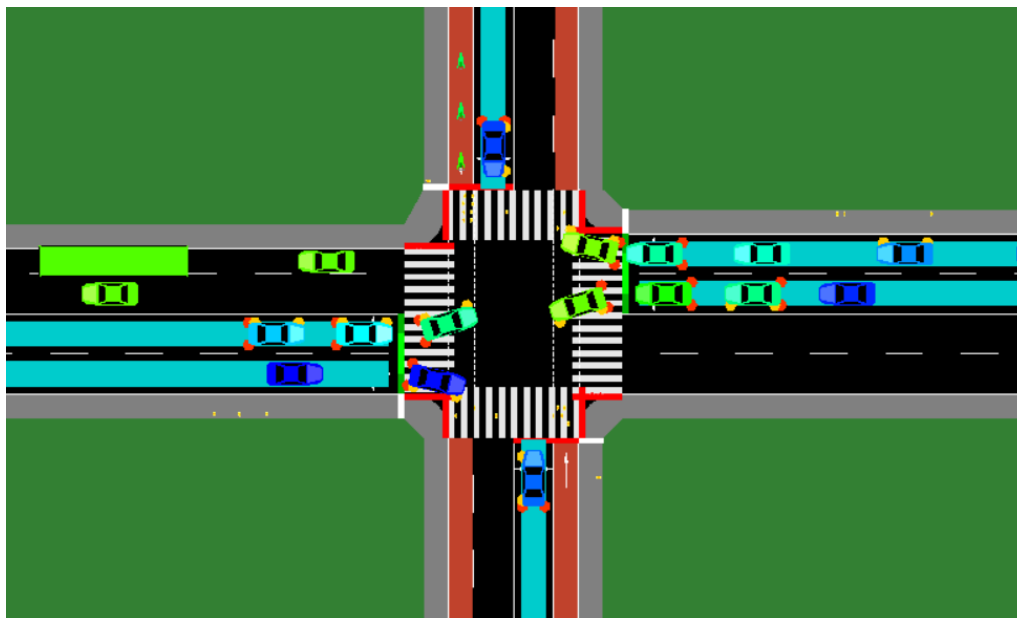
²<http://sumo.dlr.de/wiki/TraCI>

³<https://redis.io>

⁴<http://www.scadasystems.net>

2.1 Križovatka v prostredí SUMO

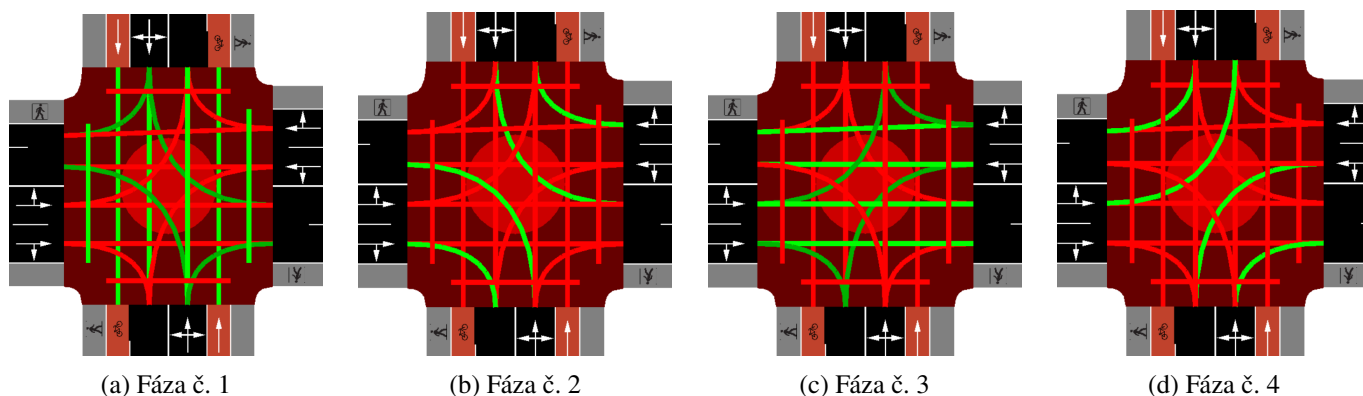
V rámci projektu sme sa rozhodli simulovať klasickú priesečníkovú križovatku riadenú semaformi. Okrem hlavnej a vedľajšej cesty križovatka obsahuje aj cyklistickú cestu a príslušné chodníky pre chodcov. V križovatke sú modelované 4 typy účastníkov premávky, osobné autá, autobusy, cyklisti a chodci. Model križovatky je znázornený na obrázku 2.



Obr. 2: Modelovaná križovatka s hlavnou a vedľajšou cestou.

Systém svetelnej signalizácie sa cyklicky prepína v niekoľkých riadiacich fázach, kde v každej fáze sa púšťa iný smer vozidiel. Štyri hlavné fázy je možno vidieť na obrázku 3. Pred a za každou hlavnou fázou sa spúšťa pomocná fáza, ktorej úlohou je zaistiť plynulosť prechodu medzi hlavnými fázami. Tieto hlavné fázy spolu s pomocnými fázami tvoria jeden cyklus programu križovatky.

Modelovaná svetelná križovatka obsahuje 4 programy. Program č. 0 je predvolený program s rovnomernými a spravodlivými časmi fáz pre každý smer jazdy. Program č. 1 je nastavený tak, aby zvýšil priepustnosť vedľajších ciest a cyklistických ciest vo smeroch sever-juh a obrátene. Program č. 2 podporuje priepustnosť hlavnej cesty vo smeroch západ-východ a obrátene. Posledný program č. 3 zvyšuje priepustnosť áut ktoré chcú odbočovať z hlavnej na vedľajšiu cestu a naopak. Bližšie informácie o jednotlivých programoch a ich fázach sa čitateľ dozvie v tabuľke 1.



Obr. 3: Hlavné riadiace fázy semaforov.

	Trvanie [krokov]			
	Program č. 0	Program č. 1	Program č. 2	Program č. 3
Fáza č. 1	47	58	36	41
Fáza č. 2	11	10	9	10
Fáza č. 3	41	34	54	34
Fáza č. 4	16	13	16	30
Pomocná	15	15	15	15
Spolu	130	130	130	130

Tabuľka 1: Trvanie jednotlivých fáz programov.

2.2 Adaptívne riadenie

Cieľom adaptívneho riadenia križovatky je vybrať taký program s fázami semaforov, ktorý pre danú situáciu umožní plynulejší prechod áut križovatkou. Výber programu prebieha automaticky na základe informácií o počte áut a autobusov v jednotlivých jazdných pruhoch. Tieto informácie sa zbierajú pomocou senzorov, ktoré sú umiestnené v každom jazdnom pruhu. Výber a zmena programu nastáva vždy až po uplynutí všetkých fáz aktuálneho programu. Adaptívne riadenie križovatky zvyhodňuje jazdné pruhy s autobusmi.

Nasledujúce rovnice matematicky popisujú chovanie adaptívneho riadenia križovatky. Rovnica 1 znázorňuje výpočet množstva dopravných prostriedkov na križovatke pre jednotlivé jazdné pruhy l . Pruhy s autobusmi sú zvyhodnené s váhou 3.

$$veh(l) = cars(l) + 3 * buses(l) \quad (1)$$

Skóre programu p sa spočíta podľa rovnice 2,

$$score(p) = \sum_{l \in L_p} \frac{w_a * all(l)}{A} + \frac{w_v * veh(l)}{V} \quad (2)$$

$$all(l) = \sum_{r \in R} veh(l) \quad (3)$$

kde

- L_p značí množinu jazdných pruhov, ktoré zvyhodňuje program p ,
- $all(l)$ značí počet všetkých vozidiel, ktoré prešli jazdným pruhom l v priebehu 1 cyklu aktuálneho programu križovatky,
- $veh(l)$ značí aktuálny počet vozidiel v jazdnom pruhu l
- A a V značia celkový počet vozidiel v priebehu 1 cyklu aktuálneho programu križovatky resp. aktuálny počet všetkých vozidiel na križovatke,
- w_a a w_v sú váhy, ktoré je možné meniť,
- R je množina fáz aktuálneho programu.

Na záver sa vyberie program s najvyšším skóre, tak ako to popisuje rovnica 4,

$$program = \begin{cases} \arg \max_{p \in P} score(p) & \text{if } \Delta_{score} \geq MIN_{\Delta} \\ p_d \in P \end{cases} \quad (4)$$

$$\Delta_{score} = \max_{p \in P} score(p) - \min_{p \in P} score(p) \quad (5)$$

kde P je množina všetkých programov križovatky. V prípade, že je rozdiel Δ_{score} v najmenšom a najväčšom skóre zanedbateľný t.j. jazdné pruhy sú rovnako vyťažené, tak sa vyberie predvolený program p_d (program č. 0).

2.3 SCADA rozhranie

Súčasťou projektu je aj jednoduchý SCADA systém pre monitorovanie a riadenie križovatky. Realizovaný SCADA systém zobrazuje aktuálny stav každého semaforu (červená, oranžová, zelená) a aktuálny semaforový program v ktorom je aktuálne križovatka riadená. Tento program je možné manuálne zmeniť. Grafické používateľské rozhranie je znázornené na obrázku 4.



Obr. 4: Grafické používateľské rozhranie pre monitorovanie a riadenie križovatky.

3 Experimenty

V rámci projektu bolo vykonaných niekoľko experimentov. Počas experimentovania boli použité tie isté podmienky tak, aby sa ten istý experiment dal kedykoľvek zopakovať (generovanie premávky). V tabuľke 2 sú zaznamenané jednotlivé trvania s manuálnym riadením križovatky. Trvaním sa myslí doba behu systému od začiatku simulácii až do konca simulácie, kedy križovatkou prejde posledné auto. V tomto kontexte je treba podotknúť, že autá sa generovali až do 3600 krokov simulácie.

Ako z tabuľky 2 vyplýva. Dĺžka trvania simulácie sa výrazne líši pre jednotlivé programy. Ako sa dalo predpokladať, program č. 0 je spravodlivo nastavený. Preto je jeho dĺžka trvania najkratšia. Naopak program č. 2 je prudko nevyvážený. Problémom sú najmä cyklisti. Tento program im prideluje veľmi málo času so zelenou.

Program	0	1	2	3
Trvanie [krokov]	3951	4410	4985	4014

Tabuľka 2: Experiment s manuálnym riadením

Tabuľka 3 znázorňuje trvania s adaptívnym riadením križovatky. Z tabuľky vyplýva, že až tak veľmi nezáleží na nastavení váh w_a a w_v . Viac záleží na nastavení prahu MIN_{Δ} . Z tabuľky tiež vyplýva, že takmer vo všetkých prípadoch adaptívne riadenie pomáha pri zvýšení plynulosti premávky.

Experiment	1	2	3	4	5	6	7	8	9	10	11	12	13
w_a	1.0	1.0	1.0	0.9	0.9	0.9	0.5	0.5	0.5	0.2	0.2	0.2	0.3
w_v	0.0	0.0	0.0	0.1	0.1	0.1	0.5	0.5	0.5	0.8	0.8	0.8	0.7
MIN_{Δ}	0.0	0.05	0.1	0.0	0.5	0.1	0	0.025	0.1	0	0.025	0.1	0.175
Trvanie [krokov]	3939	3951	3936	3939	3938	3944	3962	3937	3944	3939	3936	3953	3978

Tabuľka 3: Experiment s adaptívnym riadením

4 Záver

V pôvodnom návrhu sme adaptívne riadenie navrhli ako umelého agenta v prostredí JADE. No počas implementácie sme narazili na problém s kompatibilitou JAVA knižnice TraCI4J⁵, ktorej úlohou bolo zaistiť komunikáciu so SUMO simuláciou. Problém spočíval v zastaranom formáte správ protokolu, ktorý táto knižnica používala na komunikáciu so SUMO rozhraním. Z toho dôvodu sme nepoužili JADA technológiu. Aktuálna implementácia umelého agenta je síce postačujúca pre účely tohoto projektu. Avšak pre potreby simulovania súčinnosti viacerých križovatiek je nutné adaptívne riadenie implementovať ako umelého agenta v multiagentnom prostredí. Napriek tomu sme sa oboznámili so základnou problematikou adaptívneho riadenia cestnej premávky a rozšírili si obzor o nové technológie zaoberajúce sa danou témou.

V projekte sa nám podarilo navrhnúť možný spôsob riešenia problému s dopravnými zápchami na križovatkách. Navrhli sme architektúru, ktorá okrem adaptívneho riadenia križovatky je schopná jednoducho zbierať a zobrazovať dáta ďalším procesom. Práca je stavaná tak, aby bola ľahko rozširovateľná a s potenciálom pre ďalšie optimalizácie adaptívneho riadenia alebo zasadenie do zložitejších prípadov. V rámci tejto práce sa nám podarilo vytvoriť funkčnú simuláciu križovatky s adaptívnym riadením. Z experimentov vyplýva, že automatické adaptívne riadenie je účinnejšie než manuálne riadenie križovatky.

⁵<https://github.com/egueli/TraCI4J>