



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar

Intelligens gépkocsik városi forgalomban

Kommunikáción alapuló kooperáció vizsgálata

Tudományos Diákköri Konferencia – Dolgozat

Készítette: Alekszejenkó Levente (UTEJVZ)

**Konzulens: Dobrowiecki Tadeusz (Méréstechnika és Információs Rendszerek
Tanszék)**

2018.

Absztrakt

Ma a városi közlekedés legnagyobb problémáját a forgalmi torlódások, dugók jelentik. A torlódások kialakulásának egyik fontos oka, hogy a közlekedési lámpák programozása nem az aktuális forgalmi igényeknek megfelelő, illetve, hogy sok a felesleges sávváltás.

Az előbbire megoldás lehet, ha valamilyen formában az aktuális igényeknek megfelelő, adaptív logikával látjuk el a csomópontok forgalomirányítóját.

Míg utóbbira jó megoldás lehet az, hogy a járműveket csoportokba szervezzük, a csoportokat pedig egy egységként kezeljük.

A dolgozatban¹ e két megközelítés együttes alkalmazását vizsgálom, budapesti forgalmi helyzeteket modellező szimulációkkal. A járműveket és a csomóponti forgalomirányítókat kommunikáló kooperatív intelligens ágensekként fogom fel, és dolgozom ki a kooperáció protokollját.

A szimulációt a nyílt forráskódú SUMO szimulációs platformra alapozva, saját fejlesztésű szimulátorral végzem.

¹Ez a dolgozat EFOP-3.6.2-16-2017-00013 és FIKP-MI-SC projektek támogatásával készült.

Tartalomjegyzék

| | |
|---|-----------|
| Ábrák jegyzéke | iv |
| Táblázatok jegyzéke | vi |
| 1. Bevezetés | 1 |
| 1.1. Motiváció | 1 |
| 1.2. Feladatértelmezés | 1 |
| 1.2.1. Elvégzendő kísérletek, elérendő célok | 2 |
| 1.3. Kapcsolódó irodalom | 3 |
| 2. Algoritmusok és kommunikációs protokolljavaslatok | 5 |
| 2.1. Elvi áttekintés | 5 |
| 2.2. Definíciók | 6 |
| 2.2.1. Statikus infrastrukturális elemek | 6 |
| 2.2.2. Okosautók és képességeik | 7 |
| 2.2.3. Csoportosítások definíciói | 8 |
| 2.2.4. Bírók | 9 |
| 2.3. Csoportosítás | 9 |
| 2.3.1. Csoportformálás | 9 |
| 2.3.2. Csoport képességei és előnyei | 9 |
| 2.3.3. Csoportok felbontása | 10 |
| 2.4. Sávváltások | 10 |
| 2.4.1. A SUMO sávváltásának ismertetése | 10 |
| 2.4.2. A módosítások ismertetése | 11 |
| 2.5. Bírók | 11 |
| 2.5.1. Csomóponti áthaladás protokollja | 12 |
| 2.5.2. Forgalmirányítási algoritmus | 12 |
| 2.5.3. Ütemezési algoritmusok | 13 |
| 3. Szimulációs platform | 18 |
| 3.1. A SUMO rövid bemutatása | 18 |
| 3.1.1. A forgalomszimuláció elméletéről tömören | 18 |
| 3.1.2. A SUMO felépítése | 18 |

| | | |
|-----------|---|-----------|
| 3.2. | A SUMO-n alapuló keretrendszer | 19 |
| 3.2.1. | Rétegmodell | 19 |
| 3.2.2. | Üzenetküldő rendszer | 20 |
| 3.2.3. | Szimulációabsztrakciós eszköz, sávváltási menedzser | 20 |
| 3.2.4. | Csoportok és a markerrendszer | 20 |
| 3.2.5. | A konfliktusosztályok | 21 |
| 3.2.6. | A bírók | 22 |
| 3.2.7. | Ismert problémák, furcsaságok és limitációk | 22 |
| 4. | Mérések | 24 |
| 4.1. | Mérési elrendezések | 24 |
| 4.1.1. | Mérések pályái | 24 |
| 4.1.2. | Lámpaprogramok | 25 |
| 4.1.3. | Normális és rendhagyó forgalom | 27 |
| 4.1.4. | (Intelligens) lámpavezérlés | 27 |
| 4.2. | Mérendő mennyiségek | 28 |
| 4.2.1. | Áthaladó autók mennyisége | 28 |
| 4.2.2. | Időeredmények | 28 |
| 4.2.3. | Egyéb eredmények | 28 |
| 4.3. | Mérési módszerek | 28 |
| 4.3.1. | Tesztpályán végzett mérések | 28 |
| 4.3.2. | BAH-csomópontban végzett mérések | 29 |
| 4.4. | Mérési eredmények | 30 |
| 4.4.1. | Tesztpályán végzett mérések | 30 |
| 4.4.2. | BAH-csomóponton végzett mérések | 30 |
| 4.5. | Mérési eredmények értékelése | 34 |
| 4.5.1. | Átlagsebességek összehasonlítása | 34 |
| 4.5.2. | Áthaladási idők összehasonlítása | 35 |
| 4.5.3. | Átlagos várakozási idők összehasonlítása | 36 |
| 4.5.4. | Ütemezések időviszonyai | 37 |
| 4.5.5. | Az ütemezési módszerek érzékenysége | 38 |
| 5. | Összefoglalás, konklúzió | 40 |
| 5.1. | Az eredmények kritikus értékelése | 40 |
| 5.2. | Továbbfejlesztési lehetőségek, újabb kísérletek | 41 |
| 6. | Irodalomjegyzék | 42 |
| A. | Pontos protokolleírások | 44 |
| A.1. | Csoportosítás | 44 |
| A.1.1. | Csoportba belépés | 44 |
| A.1.2. | Csoportból kilépés | 45 |
| A.2. | Bíróval kapcsolatos üzenetváltás | 46 |

| | |
|---|-----------|
| A.2.1. Konfliktusosztályba belépés | 46 |
| A.2.2. A csomóponti áthaladás protokollja | 46 |
| B. Algoritmusleírások | 48 |
| B.1. P-szabályozók | 48 |
| B.1.1. Adaptív tempomat csoporttagok között | 48 |
| B.1.2. Megállás bírónál | 49 |
| B.2. Preemptív RR ütemező | 50 |
| B.3. MDDF ütemező | 51 |

Ábrák jegyzéke

| | |
|--|----|
| 2.1. Elvi áttekintés. Az egyforma színnel jelölt járművek egy csoportot alkotnak. A kereszteződés forgalmát egy elvont entitás, a bíró szabályozza, amely üzenetek formájában közli, hogy egy autó(csoport) számára éppen szabad-e az áthaladás. | 16 |
| 2.2. A statikus elemek szemléltetése. A kritikus pont, megállási távolság és bejelentési távolság gyakorlatilag egy-egy kör mentén helyezkednek el. Ennek okáért lásd a 2.2.1. fejezetet! | 17 |
| 2.3. Példa két konfliktusosztályra | 17 |
| 3.1. A rétegmodell szemléltetése | 20 |
| 3.2. A SUMO bővítésének áttekintése. A korábban említett szimulációs réteghez a vörössel, a szimulációabsztrakciós réteghez a narancssárgával, míg az intelligens logikai réteghez a zölddel jelölt modulok tartoznak. A SmartSL2015 sávváltási modell egymagában alkotja a sávváltási réteget. | 21 |
| 4.1. A tesztpálya szemléltetése. A kék nyilakkal jelzett útszakaszok egyirányúak. A pálya be- és kilépési pontjai kelet-nyugat irányú utak végpontjai, illetve az észak-dél irányú utak közül a középső két végpontja. Az útszakaszok vastagsága egyenesen arányos az útszakaszon lévő sávok számával. | 25 |
| 4.2. A BAH-csomópont és környékének térképe | 26 |
| 4.3. A tesztpályán végzett mérések szemléltetése kategóriánként | 31 |
| 4.4. A BAH-csomópontbeli átlagsebességek alakulása a forgalom jellege és az ütemező típusa szerint | 34 |
| 4.5. A BAH-csomópontbeli átlagos áthaladási idők alakulása a forgalom jellege és az ütemező típusa szerint | 35 |
| 4.6. A BAH-csomópontbeli átlagos várakozási idők alakulása a forgalom jellege és az ütemező típusa szerint | 36 |
| 4.7. A hagyományos, statikus ütemezés időviszonyai | 37 |
| 4.8. Az RR ütemezés időviszonyai | 37 |
| 4.9. Az MDDF ütemezés időviszonyai | 38 |
| 4.10. Az ütemezők érzékenysége a terhelés jellegére | 38 |

| | |
|---|----|
| A.1. A csoporthoz történő csatlakozás szekvenciadiagramja | 44 |
| A.2. A csomóponti áthaladás szekvenciadiagramja | 46 |
| B.1. Az MDDF ütemező felépítése | 51 |

Táblázatok jegyzéke

| | |
|---|----|
| 4.1. A BAH-csomópont és a környékének topologikus beállítása | 27 |
| 4.2. Tesztpályán végzett mérési eredmények I. | 30 |
| 4.3. Tesztpályán végzett mérési eredmények II. | 30 |
| 4.4. BAH-csomóponton, kis terheléssel végzett mérési eredmények I. | 31 |
| 4.5. BAH-csomóponton, kis terheléssel végzett mérési eredmények II. | 32 |
| 4.6. BAH-csomóponton, nagy terheléssel végzett mérési eredmények I. | 32 |
| 4.7. BAH-csomóponton, nagy terheléssel végzett mérési eredmények II. | 32 |
| 4.8. BAH-csomóponton, <i>abnorm1</i> esettel végzett mérési eredmények I. | 32 |
| 4.9. BAH-csomóponton, <i>abnorm1</i> esettel végzett mérési eredmények II. | 32 |
| 4.10. BAH-csomóponton, <i>abnorm2</i> esettel végzett mérési eredmények I. | 32 |
| 4.11. BAH-csomóponton, <i>abnorm2</i> esettel végzett mérési eredmények II. | 33 |

1. Bevezetés

1.1. Motiváció

Napjainkra az emberek széles körének vált elérhetővé a gépkocsival való közlekedés. Sajnos amikor úthálózatunkat a múlt század közepén megtervezték, még álmodni sem mertek a mérnökök a mai forgalom méretéről. A forgalom növekedést az elmúlt nagyjából két évtizedben jól mutatja [16] és [10] összehasonlítása. Míg 1995-ben nagyjából 15000 járművet számláltak naponta az M1-es autópálya egyik szakaszán, 2017-ben majdnem ugyanazon a szakszon már közel 60000 jármű haladt át naponta. Amennyiben hihetünk [15] előrejelzésében, ez a szám a jövőben még nőni is fog, ami sajnos meg is látszik, az útjaink állandóan telítettek, mindennapossá váltak a kilométeres forgalmi torlódások.

Mivel a torlódásokat, dugókat sokszor az emberi figyelmetlenség – vagy egyszerűen csak a lassabb reakcióidő – okozza, a járműgyártók igyekeznek minél több és minél kifinomultabb vezetéstámogató rendszert szerelni autóinkba. Ez a technológia kezd elég fejletté válni ahhoz, hogy a gépjárművek akár saját magukat is vezessék, akár emberi beavatkozás nélkül is. Másfél-két évtizeden belül valós elképzelésnek tűnik, hogy a mindennapi forgalom egy részét már ilyen, önvezető járművek adják.

Ha ezen autókat felruházzuk megfelelő kommunikációs képességekkel, akkor minden lehetőség adott egy kooperatívan működő ágensrendszer kialakításához is. Lehetővé válhat általa az útfelület jobb kihasználása kisebb követési távolságokkal. Megnövelhető lenne a forgalmi csomópontok áteresztő képessége új, kommunikáción alapuló forgalomirányító mechanizmusokkal. A sávváltások forgalomcsökkentő hatása is csökkenthető lenne, ha a járművek egy csoportja váltana egyszerre sávot és nem egyesével kellene sávot váltania az egyes autóknak. Emellett persze nem mehetünk el olyan előnyök mellett sem, mint a forgalom biztonságának növekedése, vagy az üzemanyagfogyasztás – ezzel együtt a károsanyag-kibocsátás – esetleges csökkenése.

1.2. Feladatértelmezés

A forgalom folyamatosságának javítása érdekében szeretnék kidolgozni egy kommunikációs protokollt járművek–járművek, illetve járművek–infrastrukturális elemek között.

Ez a protokoll tartalmazni fog olyan üzeneteket, amelyek a kommunikációban résztvevő járművek csoportba szervezését segíti. Továbbá olyan üzeneteket is, amelyek a

csoportok sávváltását segítik elő. Végül pedig olyan üzeneteket, melyek egy „okos csomóponton” való átjutását segíti a csoportoknak.

Ha már létezik egy csomóponti áthaladást segítő kommunikációs protokoll, érdemes megvizsgálni azt is, hogy a közlekedési csomópont, mint egy ritka erőforrás, és az okosautók, ágensek, mint erőforrásért folyamodó processzek analagóióából kiindulva lehetséges-e valamilyen (operációs rendszerekben alkalmazott) ütemező algoritmus módosított használata forgalomirányítási célra.

Két ilyen ütemezési algoritmust, a körforgó (*Round Robin, RR*) és az MDFF (*Minimal Destination Distance First*) [1] algoritmust szeretném kipróbálni. Mindkét módszer igen elterjedt az operációs rendszerek világában. Az operációs rendszerek körforgó ütemezése szinte módosítások nélkül adaptálható egy forgalomszabályozó algoritmusnak is. Az operációs rendszerek világában a *legrövidebb taszkot előre* (Shortest Job First) stratégiáról bizonyított az, hogy optimális megoldást ad átlagos válaszüdőre. Ennek a közutakra alkalmazható verziója lehet például az MDFF ütemező.²

A megalkotott protokollt a Simulation of Urban MObility (SUMO) nevű forgalom-szimulátorra [9] alapozva kísérleti jelleggel implementálni fogom, és segítségével szimulációkat fogok futtatni.

1.2.1. Elvégzendő kísérletek, elérendő célok

Mivel az európai jellegű városi úthálózat alapvetően eltér az amerikai típusútól, ezért két teszt pályán fogom kipróbálni a rendszert. Az első teszt pályára egy fiktív sakktábla-szerű elrendezés lesz (néhány utat egyirányúnak választva), ez hivatott az amerikai típusú úthálózatot modellezni. A másik pályára a budapesti BAH-csomópontot³ és környékét fogja szimulálni.

Az egyik legfontosabb mérőszám ezekben a szimulációkban a szimuláció időtartama alatt *áthaladt járművek száma*. Ezen kívül mérni fogom az áthaladás időviszonyait, így például:

- *Az átlagos áthaladási időt.*
- *Az átlagos várakozási időt.* Ez azt mutatja meg, hogy egy-egy autó mennyi időt töltött „piros lámpánál” történő várakozással.
- *Az elvesztegetett időt.* Azaz azt az időt, amikor egy autó lassabban haladt, mint ahogyan haladhatott volna akkor, ha rajta kívül mások nem lettek volna az úton, és minden „közlekedési lámpán” szabad jelzést kapott volna.

Ezen kívül megvizsgálom, hogy ezek a módszerek mennyire érzékenyek a forgalom nagyságára és irányára.

²Ha teljesen precíz akarnunk lenni, akkor ez a Shortest Job First preemptív verziójának, a Shortest Remaining Time First algoritmusnak feltethető meg.

³Budaörsi – Alkotás – Hegyalja utak kereszteződése, kiegészülve a Jagelló és Villányi utak kereszteződéseivel.

Feltehetőleg mindkét (RR és MDDF) algoritmus minden paraméterében legalább olyan eredményeket fog produkálni, mint amit a hagyományos forgalomirányítás elér. Elméletileg az MDDF algoritmus optimális kimenetet fog adni átlagos várakozási idők tekintetében. Az intelligens algoritmusok (RR és MDDF) elvileg kevésbé lesznek érzékenyek a forgalom irányára, mint a hagyományos algoritmus.

1.3. Kapcsolódó irodalom

A járművek csoportosításának (*platooning, konvoj*) igen sok előnyét vizsgálták már. Bár mi városi forgalomban és személyautókkal végzünk vizsgálatot, és a fogyasztási adatok mérése jelenleg nem célunk, megjegyzendő, hogy [2] megmutatta azt, hogy kb. 4-7% üzemanyagmegtakarítást is hozhat a haszongépjárművek konvojban haladása autópályán.

Emellett a csoportosítás megnöveli a járműfolyamok nagyságát egy-egy útszakaszon, ahogy arra [6] rámutat. Érdekes kérdés viszont, hogy mi történik, ha [6] ellentétben nem egy főpálya mellett lévő mellékvágányon lévő állomás metaforája mentén, hanem egy valós környezetben (pl. a budapesti BAH-csomópontban) vizsgáljuk a csoportosítás előnyeit.

Mivel [8] eredménye, hogy egyetlen sávváltás is 8-18%-os kapacitáscsökkenést tud okozni egy útszakaszon, ezért érdemesnek tűnik vizsgálni annak a lehetőségét, hogy egy-egy járműcsoport együttes sávváltása vajon hatékonyabb közlekedést eredményez-e, mintha az egyes járművek külön-külön próbálnának sávot váltani. Feltehetőleg igen, ugyanis egy járműcsoportot elképzelhetünk úgy, mint egy sávot váltó teherautót, nyergesvontatót, ezekről pedig a [14] megmutatta, hogy még egy duplapótkocsis teherautó is legfeljebb 3 személyautóval egyenértékű, holott a helyére több, mint három személyautó elfér egy csoportban. Így tehát egy csoport sávváltása kisebb hatással lehet a forgalomra, mint az egyes autók sávváltásai külön-külön.

Egyetlen járműcsoport biztonságos sávváltását vizsgálta [7], illetve [13] leír egy hatékony módszert két csoport egymásba olvadására úgy, hogy egyik csoportot sem kell felosztani. Viszont a mindennapi forgalomban inkább az a szituáció jellemző, hogy egy csoportnak két másik csoportok közé kell beférnie, anélkül, hogy összeolvadnának, így az én szimulációim ezt manővert fogják modellezni.⁴

[3] igen érdekes matematikai bizonyításokat ad járműcsoportok ütemezéséről előre ismert csoportok és útvonalak esetén, ám nem vizsgálják a futási időben jelentkező csoportok igényeinek kiszolgálását, így egy valós forgalmi helyzetben vagy annak szimulációjában ezek az elvek nem igazán alkalmazhatóak. [1] előáll egy online időben is alkalmazható megoldással (MDDF ötlete), ám mérési eredményei egy tesztpálya al-

⁴Például egy forgalmi sávból balra és egyenesen lehet továbbhaladni. A mi csoportunk a jobbra forduló sávban halad, de egyenesen szeretne továbbmenni. Előttünk már van egy csoport, de messze, velük nem előnyös összeolvadni. Mellettünk pedig halad egy csoport, akik balra akarnak fordulni, velük pedig az eltérő cél miatt nem szeretnénk összeolvadni. A feladatunk az lenne, hogy sávot váltjunk, és mindenki a lehető legzavartalanabban haladhasson tovább.

kalmazásával kapta, kérdéses, hogy valós úttopológiával megismételhetőek-e az elért eredményeik. További probléma ezzel az megoldási javaslattal, hogy nem fair, azaz előfordulhat benne kiéheztetés, az, hogy egy jármű nagyon sokáig (akár örökre is) várakozni kényszerül. Az alapötletet felhasználva készíték egy fair MDDF-megvalósítást, amit szimulációval ki is fogok próbálni.

Az ismertetésre kerülő algoritmusoknál, protokolloknál az üzenetek továbbítását kiszolgáló infrastruktúrát ugyan absztraktnak tekintem, de erre gondolhatunk úgy is, mint a VANET-ek (*Vehicle Ad-Hoc Network*, [4]) legfelső, alkalmazási rétegében futó programokra. Ezek a hálózatok univerzálisan felhasználhatóak, így alkalmazási rétegükben futhatnak útdíjfizető programok, flottamenedzsment, közölhetnek közlekedési információkat, vagy éppen kooperációt elősegítő üzeneteket is válthatnak egymással a járművek.

2. Algoritmusok és kommunikációs protokolljavaslatok

2.1. Elvi áttekintés

Az általam javasolt megoldások törekszenek használni a már meglévő közúti infrastruktúrát (pl. sávjelzések), ám szükséges lehet néhány új jelzés bevezetése. Ilyen jelzés lehet egy forgalmi csomópont megközelítését vagy éppen elhagyására figyelmeztető marker elhelyezése (ahogyan az pl. a 2.1. ábrán látható).

Járműveink, hívjuk őket *okosautóknak*, ezen jelzések között *csoportokba* tömörülnek. Azok a járművek alkothatnak egy csoportot, akik a csomóponton azonos mozgási pályán kívánnak átmenni. Egy-egy csoportot a csoport élén haladó *csoportvezető* vezeti, a többiek, a *csoporttagok* pedig a csoportvezetőt követve próbálnak átkelni a kereszteződésen.

A csoportvezetők döntései nyomán történnek a sávváltások is. Amennyiben egy csoportvezető úgy határoz, hogy a csoportja sávot fog váltani, a manőver megkezdése előtt közli ezt a szándékot a célsávban lévő csoport vezetőjével. A két csoportvezető ezután megegyezik abban, hogy melyikük csoportja lesz a manőver végén előrébb. Ha szükséges, a célsávban lévő csoport beengedi maga elé a másik csoportot. A sávváltást a csoporttagok egymás után, láncszerűen mozogva kivitelezik, így ahol a csoportvezető sávot váltott, ott az egész csoportja is sávot tud váltani.

A csomópontot megközelítve pedig a csoportvezető jelenti be a csomópontban az áthaladási jogot osztó *bíró* számára csoportja érkezését. A bíróját logikája szerint az egyes csoportokat olyan „nagyobb csoportokba” sorolja, amely nagyobb csoportok egyszerre áthaladhatnak az adott csomóponton. Hívjuk ezeket a „nagyobb csoportokat” *konfliktusosztályoknak!* A bíróját a konfliktusosztályok közül választja ki azt, amely éppen áthaladhat a csomóponton.

Vegyük észre, hogy a bíróját hasonló döntést hoznak, mint az operációs rendszerek ütemezői: egy szűkös erőforrás elosztásán dolgoznak egymással „versengő” kliensek számára. Innen jött az ötlet, hogy a már jól ismert algoritmusokat (némi módosítással) a bíróját konfliktusosztály-váltó logikájaként fel lehetne használni.

2.2. Definíciók

2.2.1. Statikus infrastrukturális elemek

Közlekedési csomópont

Közlekedési csomópont, *csomópont*, *kereszteződés* alatt értem azokat a területeit az úthálózatnak, ahol több út találkozik, még hozzá úgy, hogy ott az elsőbbségi viszonyokat tisztázni kell.

Így például *közlekedési csomópont* egy „T” vagy „+” alakú kereszteződés, illetve ezeknél összetettebb formák. Nem számítom *közlekedési csomópontnak* viszont az úthálózat azon területeit, ahol megnyílik vagy elfogy egy-egy közlekedési sáv, ugyanis ott nem több út találkozik. Mivel pl. az autópálya-lehajtók is gyakorlatilag sáv bővülésnek tekinthetők, ezek sem *forgalmi csomópontok* a főpálya szempontjából.

Belépési marker

A *belépési marker* egy néhányszor tíz méter hosszú szakaszt⁵ jelöl a csomópontba befelé vezető oldalon, de a csomóponttól viszonylag messzebb kezdve (lásd a 2.2. ábrán).

A belépési marker jelzi az okosautók számára, hogy össze kell állni csoportokba.

Bejelentési távolság

A bejelentési távolság a csoportvezető számára hordoz információt. Ekkora távolságból (közel 100 m) kell ugyanis bejelentkeznie a bírónál. A bíró ekkor határozza meg, hogy egy csoport melyik konfliktusosztályba tartozik.

Megállási távolság

Ez gyakorlatilag a Kreszből ismert STOP-vonal megjelenítése. Technikai okokból a csomóponttól előre definiált távolságra van, a csomópont topológiájától függően.⁶

Kritikus pont

A megállási távolságtól egy-két méter távolságra (a csomóponthoz közelebb) elhelyezett pont. Amelyik autó ezen a ponton áthalad, annak a csomóponton is minél előbb át kell haladnia. Az okosautók mindig bejelentik, ha áthaladnak ezen a ponton, így tudja a bíró, hogy éppen van-e autó a csomópontban.

⁵A pontos hosszt nem érdemes definiálni. Azt az aktuális topológia befolyásolja ugyanis.

⁶A preempció miatt előfordulhat olyan eset, hogy egy járműnek relatíve intenzíven kell fékeznie ahhoz, hogy ne fusson bele a csomópontba...

Kilépési marker

Amint egy autó áthaladt a csomóponton, a kilépési markerhez ér. Itt egyfelől jelzi a bíró számára, hogy áthaladt a csomóponton. Másrészről a jármű kilép a csoportjából és konfliktusosztályából is.

Mivel az csoport élén halad a csoportvezető, ezért természetesen ő lép ki először a csoportból. Ilyenkor gondoskodunk az új csoportvezető kiválasztásáról is.

Magyarázat a megállási távolság és kritikus pont léte

A mindennapi közlekedésünk során is elő-előfordul az a jelenség, hogy hiába észleljük időben a sárga lámpát, már nem tudunk biztonsággal megállni a lámpa előtt, ezért továbbhaladunk. Eközben elképzeltető, hogy a lámpa már pirosra is váltott.

Ilyen esemény természetesen egy intelligens közlekedési rendszerben is bekövetkezhet. Csakhogy itt az algoritmusok érzékenyebbek arra, hogy éppen van-e valaki a kereszteződésben, akinek már elvileg nem lenne szabad ott lennie. Valószínűleg bonyolult implementációval ez a helyzet is számítható lenne, de ennél egyszerűbb megoldás, ha a következők szerint járunk el:

1. Csak bizonyos idő (pl. 3 s) letelte után indítjuk el a következő konfliktusosztályt. Ez nagyjából a sárga lámpa esete.
2. Csak akkor indítjuk el az új konfliktusosztályt, ha már letelt az előző pontban említett idő, és a kereszteződés is üres. Ez pedig a pirosba belecsúszó autó esete.

A megállási távolság és a kritikus pont egymáshoz való helyzetének megértéséhez szükséges a szimulációs platform, a SUMO (bővebben lásd a 3. fejezetet) felépítésének ismerete. Tömören fogalmazva itt az autók egy gráf élein (ezek nagyjából a sávok) mozognak. A gráf pontjai pedig a sávok kapcsolódási pontjai, viszont a SUMO-ban ezek fizikai kiterjedéssel nem rendelkeznek. Tehát egy kereszteződés képe a SUMO-ban egy grafikai trükk, a csomópont pontos határainak ismerete így nehézkes volna. Ezért nem esik egybe a megállási távolság és a kritikus pont, ugyanis nem volna egyszerű rájuk egy konkrét értéket meghatározni. Ez a mi szempontunkból jelenleg nem is érdekes kérdés.⁷

2.2.2. Okosautók és képességeik

Feltételezzük, hogy az *okosautók* olyan gépjárművek, amik alapvetően rendelkeznek mindazon vezetési képességekkel, amit ma egy humán vezetőől elvárunk: Tőle elvárható módon betartja a közlekedési szabályokat. Felismeri a táblákat, a forgalmi sávok határain belül halad, megfelelő követési távolságot tart. Tudja, hogy merre kell mennie, melyik sávba kell beállnia egy kereszteződésben. Ha van rá (legális) lehetőség, akkor

⁷A kritikus pont, a megállási és bejelentési távolságok egy-egy koncentrikus kör mentén helyezkednek el, melyek középpontja a SUMO csomópontjának pozíciója.

megelőzi az őt feltartó járművet. Ezt a feltételezést egy-két évtizeden belül jó eséllyel teljesíteni tudják majd az új autók.

Ezen kívül feltételezzük azt is, hogy a VANET-eknek hála létezik egy ad-hoc kommunikációs hálózat is a járművek között. Ez a kommunikációs csatorna elég kapacitással rendelkezik ahhoz, hogy az általunk javasolt üzeneteket átvigye gyakorlatilag késleltetésmentes módon.

A járművek fel vannak szerelve távolságtartó tempomattal, és rendelkezünk egy interfésszel, amin keresztül olyan manővereket kérhetünk, mint:

- haladási sebesség beállítása,
- megállás,
- sávváltás jobbra vagy balra

A járművek továbbá képesek detektálni a 2.2.1. fejezetben ismertetett infrastrukturális jelöléseket. És képes végrehajtani ezen jelölések által kiváltott protokollokat, illetve a kommunikáció útján kapott információkra, utasításokra reagálni.

2.2.3. Csoportosítások definíciói

Csoportok

Amikor az *okosautók* egy csomópont belépési markerére érnek, csoportformálásba kezdenek. Minden okosautó pontosan egy csoportnak lesz vagy tagja vagy vezetője. A csoportok élén a *csoportvezető* halad, őt követik egymás mögött a *csoporttagok*.

Egy csoportba olyan okosautók tartozhatnak, akik azonos pályán haladnak végig egy csomóponton. Azaz ugyanabban a sávban érkeznek a belépési markerre, együtt haladnak a csomópontig, majd a ugyanazon a kilépési markeren, ugyanabban a sávban hagyják el a csomópontot.

A csoportvezető gondoskodik a csoportja szükséges sávváltásairól: A többi csoport vezetőjével ő egyezteteti a sávváltásokat kommunikáció révén. Ha sávot akar váltani, akkor megkérhet egy másik csoportot, hogy engedjék be az ő csoportját. Avagy, ha őt kéri meg arra, hogy engedjen be egy másik csoportot, akkor ezt lassítással, esetleg megállással lehetővé teszi.

A csoporttagok a csoportvezetőt és egymást követik, nagyjából 10 méteres⁸ távolságban. A távolság alapú követés hatékonyabban támogatja az együtt mozgás lehetőségét is, szemben az időalapú követéssel. Időalapú követésnél (pl. 2 másodpercnyi távolságból követik egymást az autók) ez azért előnyösebb, mert a járművek fix távolságra követik egymást. Így induláskor késleltetésmentesen indulhatnak egymáshoz képest az autók, nem kell megvárni, hogy a pl. 2 másodperces követési idő kialakuljon.

A távolság tartásáról egy P-szabályozón alapuló algoritmus gondoskodik. Részletekért lásd a B.1. fejezetet!

⁸A 10 m távolság a szimulációs platform sajátossága, ugyanis a járművekhez egy 6,65 méteres szakasz tartozik benne, amin belül néha ütközést érzékel a szimulátor.

Konfliktusosztályok

A *bírók* számítási igényük alacsonyán tartása céljából nem az egyes okosautó-csoportokat, hanem a csoportok egy halmazának áthaladását ütemezi. Ezek a csoportthalmazok a *konfliktusosztályok*.

Egy konfliktusosztály olyan csoportok halmaza, amelynek tagjai egyszerre haladhatnak át a kereszteződésen, azaz pályájuk nem metszi egymást. Konfliktusosztályt egyetlen csoport egyetlen okosautója is képezhet, sőt konfliktusosztály lehet az üreshalmaz is, ha éppen a konfliktusosztálynak egyetlen csoport sem tagja.

A 2.3. ábrán látható például két konfliktusosztály: a piros és a zöld. A két konfliktusosztály láthatóan nem haladhat át egyszerre a kereszteződésen, mert a pályáiknak több közös pontja is van. Viszont látható az is, hogy az egyszínű pályák nem metszik vagy érintik egymást.

2.2.4. Bírók

A *bírók* gyakorlatilag a forgalomirányító lámpáknak feleltethetőek meg. Három feladatuk van:

1. Meghatározzák, hogy egy *csoport* melyik *konfliktusosztályba* tartozzon. Ez a *konfliktusosztály-választó logika*.
2. Kijelölik, hogy pillanatnyilag melyik *konfliktusosztály* haladhat át a kereszteződésen. Ezt *ütemező logikának* nevezem.
3. A korábban említett két logika számításai alapján pedig eldönti, hogy egy adott okosautó éppen áthaladhat-e a csomóponton avagy várakoznia kell.

2.3. Csoportosítás

2.3.1. Csoportformálás

Ha egy okosautó egy csomópont belépési markeréhez ér, kötelező belépnie egy csoportba. Ez a csoport olyan, hogy abba az irányba szeretne a csomópont elhagyása után továbbhaladni, mint az újonnan érkező jármű.

Ha a jármű egy megfelelő csoport, mint menetoszlop végéhez tud csatlakozni, akkor az A.1.1. fejezetben ismertetett protokollt követve a csoport tagjává válik.

Ha viszont nincs megfelelő csoport, akkor a jármű létrehoz egy új csoportot, annak csoportvezetővé válik, és lehetővé teszi, hogy újabb járművek csatlakozzanak hozzá.

2.3.2. Csoport képességei és előnyei

A csoportosítás több célt is szolgál. Egyfelől ez egy számítási és kommunikációs spórolás: a csoportvezető egész csoportja nevében jelzi pl. a bíró számára az áthaladási igényt vagy a konfliktusosztályhoz való csatlakozását.

A csoportosítás eredményeképpen a sávváltási események is megkönnyebbíthetők. Egy csoport sávváltásának együttes hatása a forgalomra kisebb lehet, mint amilyen az egyes autók sávváltásának külön-külön lenne.

Nem is beszélve arról, hogy a csoportosítás, idegen szóval *platooning* milyen további lehetőségeket is nyújt. Így például az együtthaladást, a minimális késleltetésű elindulást, nagytempójú haladásnál pedig az üzemanyagfogyasztás csökkenését.

2.3.3. Csoportok felbontása

Amikor a csoportvezető kilépési markerre ér, kilép a csoportjából, és a csoport vezetését a mögötte haladó okosautó kapja meg.

A csoport végét az az esemény jelenti, ha az utolsó jármű is kilépett a csoportból.

Ez az egyszerű felbontás a bírók ütemezőjének preemptivitását szolgálja. Így a csoportban az első okosautó, aki már nem kap áthaladási engedélyt, az kis idő elteltével garantáltan csoportvezető lesz, ugyanis az előtte lévők még áthaladnak a csomóponton, majd elhagyják a csomópontot.

A pontos protokoll leírásáért lásd még az A.1.2. fejezetet.

2.4. Sávváltások

2.4.1. A SUMO sávváltásának ismertetése

A *Simulation of Urban MObility* (SUMO, [9]) egy nyíltforráskódú forgalomszimuláló program, melyet méréseimhez használok. A 3. fejezetben részletesen is ismertetem ezt a szoftverrendszert, de a sávváltások megértéséhez idekíváncozik néhány gondolat a SUMO által alkalmazott sávváltási modellről.

A vizsgálatunk fókuszában lévő sávváltási modell a SUMO nyelvezetében SL2015 névre hallgat. Ennek a modellnek az alapját az [5] adja. Ez a sávváltási modell a sávokon kívül ún. alsávokat is kezel (azaz a sávon belüli pozíciót is számítja).

A modell érzékeli a jármű környezetében lévő közlekedési helyzetet, ismeri, hogy a jármű merre akar haladni valamint ismeri a jármű *hosszát* is. Ezek alapján kalkulálja, hogy melyik sávot érdemes éppen választania az autónak. Ez a modell a prioritási sorrendben a következő esetekben dönt sávváltás mellett:

1. *Sürgős stratégiai váltás kell* annak érdekében, hogy a jármű kijelölt útvonalán haladhasson. Ilyen pl. egy megszűnő sáv esetén a sávváltás vagy egy kanyarodó sávba való besorolás.
2. *Egy jármű egy sürgős stratégiai sávváltást igénylő járművet blokkol.*
3. *Előzés:* ha egy lassabban haladó jármű(sor) mögötti haladásnál van jobb alternatíva, akkor váltunk.
4. *Jobbratartási kötelezettség teljesítése.*

5. Nem sürgős stratégiai váltás kell.

Amennyiben a sávváltás mellett szavazunk, a sávváltási modell a környező autók sebességét megpróbálja összehangolni, hogy ne következzen be baleset és megtörténhessen a sávváltás.

2.4.2. A módosítások ismertetése

Az SL2015 modell alapján létrehoztam a SmartSL2015 névre keresztelt modellt.

Ez az új modell tisztában van a környezetén kívül azzal is, hogy az okosautó, amelyet vezérel (hívjuk most egónak) tagja-e valamilyen csoportnak és vezetői vagy tagsági minőségben teszi-e ezt. Ez a körülmény ugyanis befolyásolja, hogyan kell értelmeznie a modellnek a környezetét, és hogyan kell viselkednie.

A modell új viselkedése a csoporttagság minőségétől függően a következő:

- Ha egy okosautó nem tartozik egyetlen csoportba sem éppen, akkor az eredeti SL2015-ös modell működésének megfelelően viselkedik.
- Ha egy okosautó egy csoport csoportvezetője, akkor döntéseit az SL2015 döntései alapján hozza. Viszont a környezetet más módon érzékeli: nem autók hosszával, hanem csoportok hosszával kalkulál. Amennyiben sávváltásra van szükség, egó megkéri az esetleg akadályozó csoport vezetőjét (altert), hogy engedje be egó csoportját maga elé. Alter ezt lehetővé teszi. Ezután egó sávot vált, majd szól az őt követő csoporttagnak, hogy váltson sávot.
- Ha egy okosautó éppen egy csoport csoporttagja, és az előtte lévő csoporttagtól vagy csoportvezetőtől értesítést kapott, hogy sávot kell váltania, akkor ő is sávot vált. Majd a sávváltási felszólítást továbbküldi. Ha egó egy csoport leghátsó tagja, akkor ő szól alternek, hogy egó csoportja sikeresen sávot váltott, alter csoportjával most már zavartalanul folytathatja útját.

A fentebb vázolt algoritmus eredményeként a csoportok tagjai láncszem-szerűen váltanak sávot egymás után. Így biztosított a sávváltás helyigénye is, ugyanis ahol a csoportvezető átfért, ott a csoport többi tagja is el fog majd férni, ha felételezzük, hogy egy csoport homogén járművekből áll.

2.5. Bírók

A *bírók* az okos rendszer forgalomirányításáért felelnek. Hasonlóan az utcákon jelenlévő forgalomirányító lámpákhoz, ezek is azt az információt szolgáltatják, hogy jelenleg megengedett-e egy járműnek a csomóponton való átkelés.

2.5.1. Csomóponti áthaladás protokollja

Csatlakozás konfliktusosztályhoz

Amikor az *okosautók* a *bejelentési távolsághoz* érnek, csatlakozniuk kell egy konfliktusosztályhoz. Amennyiben a csoportvezető ér ide, az egész csoportját egyszerre csatlakoztatja a konfliktusosztályhoz. Persze elképzelhető olyan eset is, hogy egy későbbi *csoporttag* még nem tagja ekkor a csoportnak. Ilyenkor ennek a járműnek kell majd csatlakoztatnia saját magát a konfliktusosztályhoz.

A csatlakoztatáshoz az okosautó vagy csoportvezető a belépési marker segítségével kideríti a közlekedési csomópont címét. Ezután bejelenti saját maga vagy a csoportja érkezését a bíró felé. A bíró saját *konfliktusosztály-választó logikai feltétele* segítségével eldönti, hogy a jármű(csoport) melyik konfliktusosztályhoz csatlakozzon. Majd elvégzi a bíró a csatlakoztatást.

Áthaladás a csomóponton

A *bejelentési távolság* és a *megállási távolság* között az okosjárművek (a preempciót és a tudásbázis konzisztenciáját biztosítandó) periodikusan lekérdezik a bírótól, hogy áthaladhatnak-e a csomóponton.

Erre a kérdésre a bíró egy igen/nem jellegű választ ad, annak megfelelően, hogy a kérdező autó konfliktusosztálya éppen áthaladásra van-e jelölve.

Amennyiben az okosjármű megerősítő választ kap, a megengedett és lehető legjobb sebességgel közelíti meg a közlekedési csomópontot, és ha nem jön közbe semmi, át is haladhat a csomóponton.

Amennyiben nemleges válasz érkezik, akkor a *megállási távolságnál* megpróbálnak megállni. Néha persze ez nem lehetséges... Ilyenkor természetesen továbbhaladhat, behajthat a kereszteződésbe, de minél gyorsabban el kell hagynia azt.

A kritikus ponton az okosautók bejelentik a bírónak, hogy elérték ezt a pontot, azaz most már a közlekedési csomóponton belül vannak. Amikor az *okosautók* elérik a *kilépési markert*, akkor erről is tájékoztatják a bírót.

2.5.2. Forgalmirányítási algoritmus

Döntés áthaladásról

Az áthaladási döntés alapvetően egy nagyon egyszerű logikai feltételvizsgálat: a kérdező autó konfliktusosztálya megegyezik-e az áthaladásra ütemezett konfliktusosztállyal. (Klasszikus hasonlittal élve: „zöld-e számukra a lámpa?”)

Vannak azonban olyan helyzetek, amikor a bíró döntése ellenére behajtanak járművek a kereszteződésbe. Ennek az az oka, hogy fizikailag nincs elég idő már értelmes fékezéssel⁹ megállni. (Talán kreszórán hallottuk, hogy „*ha közel sárga, gáz, ha távol sárga, akkor fék.*”)

⁹azaz nem vészfékezve

Ha a konfliktusosztályok indítása késleltetés nélkül történne, akkor a „sárgán átcsúszó” járművek a frissen indított osztály járműveivel balesetveszélyt okoznának. Ezért bevezettem egy *sárgafázist*. Ez gyakorlatilag egy 3 s-nyi késleltetést jelent aközött, hogy a korábbi konfliktusosztály autói már nemleges jelzést kapnak a bírótól, és aközött, hogy az új konfliktusosztály járművei megkapnák az áthaladási engedélyt.

Konfliktusosztály váltása

Az áthaladásra jelölt konfliktusosztályok (nevezzük az aktuálisan áthaladásra jelölt konfliktusosztályt a könnyebbség kedvéért *aktív osztálynak*) váltásánál két dolgon érdemes elgondolkodni:

1. Mikor és milyen feltételek fennállása esetén kell váltani aktív osztályt?
2. És melyik legyen a konfliktusosztályok közül a következő aktív?

Utóbbi kérdés megoldásához az operációs rendszerek ütemezőit fogjuk segítségül hívni a 2.5.3. fejezetben.

Az első kérdés megválaszolásához pedig gondoljuk végig, hogy milyen feltételeket kell kielégítenünk!

Egyfelől ne pazaroljunk időt olyankor, amikor nem tudunk újabb autót átengedni már az aktuális konfliktusosztályból. Mondjuk tehát azt, hogy ha egy meghatározott időn belül, pl. 3 s-on belül nem hajt be új okosautó a kereszteződésünkbe, akkor új konfliktusosztályt fogunk aktívnak jelölni.

Triviálisan válthatunk aktív osztályt olyankor is, amikor az aktuális aktív osztályból az utolsó jármű is áthaladt a csomóponton.

Technikailag két lépésben váltunk aktív osztályt. Első lépésben észrevesszük, hogy lehetőségünk van a váltásra, és megjegyezzük, hogy kit fogunk új aktív osztállyá választani.

Majd a *sárga fázis* következik. Ilyenkor senki sem kap engedélyt áthaladásra. Amint a sárga fázis letelt, befejezzük az aktív osztályok váltását. A korábban megjegyzett konfliktusosztályt aktívvá tesszük, és engedélyezzük járművei számára az áthaladást.

2.5.3. Ütemezési algoritmusok

Rokonság operációs rendszerek ütemezőjével

Ha átgondoljuk, hogy milyen hasonlóságok vannak egy kereszteződés egy operációs rendszer ütemezője között, akkor könnyen figyelhetünk arra, hogy a két terület igen hasonló. Mindkét esetben egy szűkös erőforrás (kereszteződés vagy processzor) kiosztását kell megoldani egymással versenyző entitások (konfliktusosztályok vagy taszkok) között.

Persze van eltérés is. Míg független taszkok közül szabadon választhatunk, előrébb rendezhetünk később érkezett taszkokat is, ha éppen erre lenne szükség. Ezzel szemben

egy kereszteződésben viszont ezt a műveletet, az átrendezést nem lehet kivitelezni. Ezt szem előtt kell tartanunk az algoritmusok átvételekor illetve implementálásakor is.

Hagyományos forgalomirányítás, mint statikus ütemezés

A hagyományos forgalomirányítást is egyfajta ütemező algoritmusnak tekinthetjük. A beágyazott rendszerek világában ismert statikus ütemezés ehhez igen hasonlóan működik. Egy előre ismert taszkkészletből előre ismert időzítéssel jelöl ki folyamatokat futásra.

Preemptív Round Robin ütemező

A preemptív Round Robin ütemező¹⁰ egy körforgó ütemező. A következő konfliktusosztály mindig a soron következő osztály lesz, amelyben vannak autók és az autókat nem akadályozzák más konfliktusosztály járművei abban, hogy behajthassanak a kereszteződésbe. Aktív osztály váltásánál a jelenlegi aktív osztály mindig a sor végére kerül.¹¹

Ez egy preemptív ütemező, ugyanis az áthaladás jogát megvonhatja, ha

1. Lejárt a konfliktusosztály maximális áthaladási ideje egy ciklusra.¹²
2. Bekövetkezik bármelyik feltétel a 2.5.2. fejezetben felsoroltak közül.

Az algoritmus pontos leírása megtalálható a B.2. fejezetben.

MDDF ütemező

Bizonyított tény, hogy operációs rendszerek esetén a *legrövidebb löketidejét előre* Shortest Job First, SJF) elnevezésű ütemezés optimális válaszidőre nézve. Ennek megöröülve szeretnénk egy olyan forgalmi *ütemezőt* alkotni, amely ennek az algoritmusnak az ötletét követi.

Mint ahogyan ismeretes, a probléma a löketidő becslésével van. Ez egy forgalom-szimulációban sincsen másképp. Több módon is becsülhetnénk a löketidőt (pl. konfliktusosztályban lévő járművek száma és azok fizikai paraméterei alapján, sebességbecslésekkel stb.). [1] javaslata a löketidőbecslésre az, hogy használjuk a járművek célig hátralevő euklidészi távolságát, mint löketidőbecslést. Ez az egész hálózatra vetítve (és amerikai típusú, sakktábla elrendezésű utcaszerkezeteket véve) egy jó löketidőbecslésnek tekinthető.

Azonban ennél pontosabb képhet kaphatunk egy európai (szabálytalan) utcaszerkezetre, ha a pontos hátralevő távolságot vesszük a löketidőbecslés alapjául.¹³

¹⁰Dolgozatomban gyakran RR ütemezőként hivatkozok rá.

¹¹Természetesen itt logikai értelemben kerülnek a sor végére, az útpályán elfoglalt pozíciójukat megtartják.

¹²Ez gyakorlatilag 100%-os kihasználtságnál megegyezik egy rendőrlámpa vezérlésével.

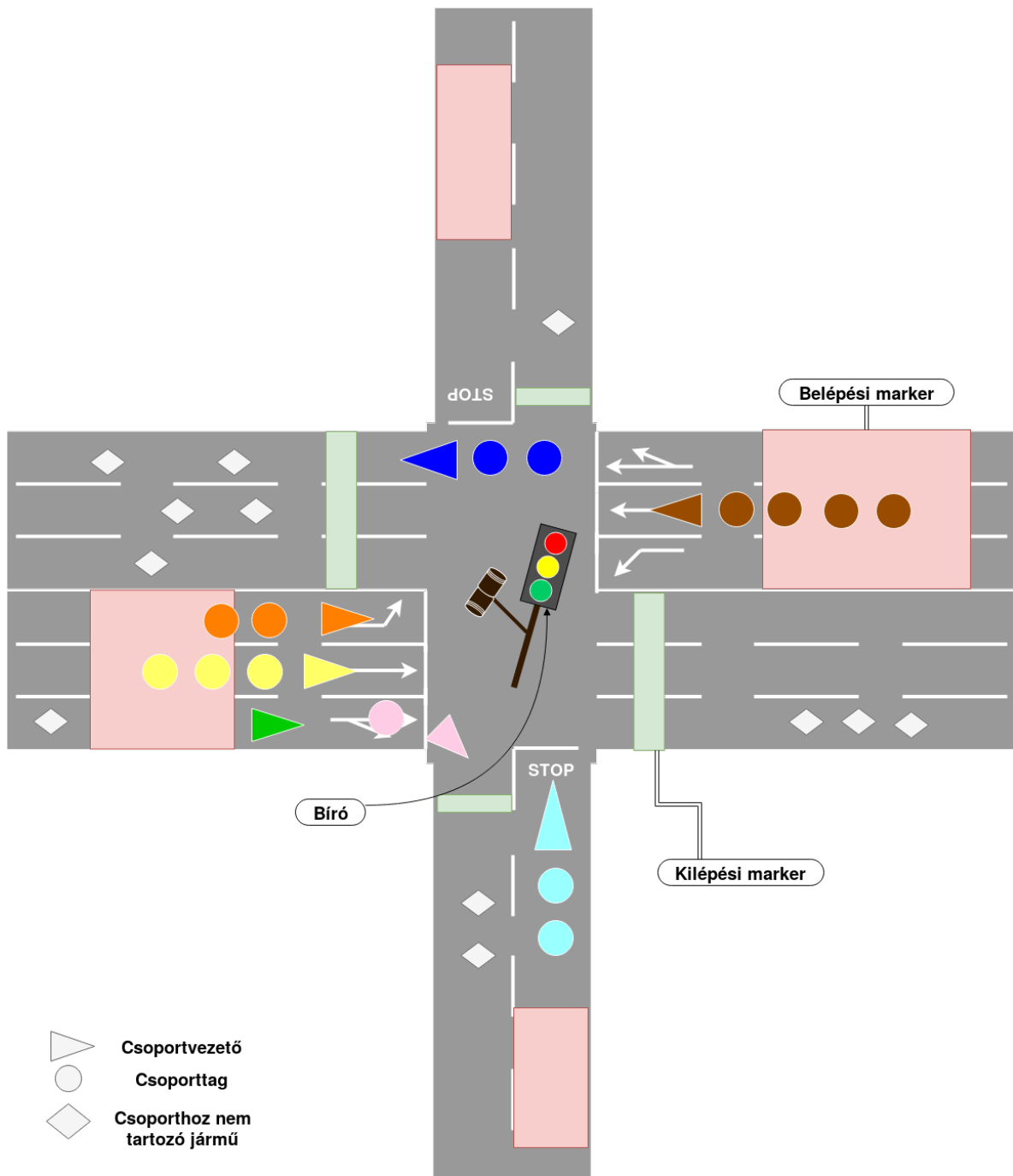
¹³A 3. fejezetben ismertetett platformban ez a paraméter rendelkezésre áll.

Másik probléma lehet az [1] által vázolt algoritmussal, hogy nem fair. Tegyük fel a következőt! Éjszaka a BAH-csomópontba a budai hegyek felől, a Hegyalja útról egyedül érkezik egy jármű, *ego*, ami Portugáliába, Európa legnyugatibb pontjához szeretne eljutni, azaz a hátralévő távolsága nagyjából 3000 km. Az Alkotás és Budaörsi utakról folyamatosan érkezik egy-egy autó, akik csak Budapesten belülrre igyekeznek. Ez az egy-két autó érkezése azonban elegendő, hogy *ego* sose tudjon behajtani a csomópontba, ugyanis egyedül alkotna egy konfliktusosztályt, amiben a minimális távolság 3000 km.

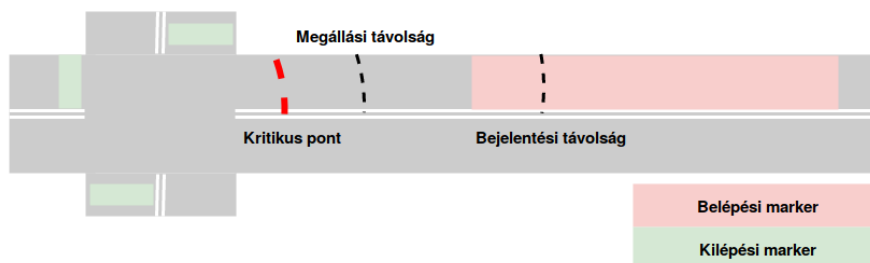
Ezen az igazságtalanságon enyhíteni kell. Ezért az én megvalósításomban az MDDF-ütemező egy kétszintű ütemező. A magasabb prioritási szinten egy körforgó ütemező található, amibe a régen (90 s-on kívül) ütemezésre került konfliktusosztályok találhatók. Az alacsonyabb szinten pedig a már említett [1]-ből átvett MDDF algoritmusötlet alapján ütemezünk.

A prioritásos ütemező és a round robin szint biztosítja, hogy az algoritmus fair lesz, és előbb utóbb a Portugáliába induló autónk is átjuthat a BAH-csomóponton.

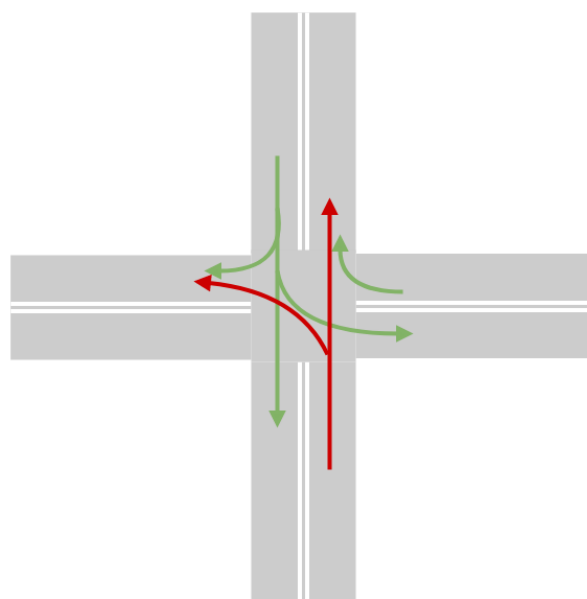
Az algoritmus pontos leírása megtalálható a B.3. fejezetben.



2.1. ábra. Elvi áttekintés. Az egyforma színnel jelölt járművek egy csoportot alkotnak. A kereszteződés forgalmát egy elvont entitás, a bíró szabályozza, amely üzenetek formájában közli, hogy egy autó(csoport) számára éppen szabad-e az áthaladás.



2.2. ábra. A statikus elemek szemléltetése. A kritikus pont, megállási távolság és bejelentési távolság gyakorlatilag egy-egy kör mentén helyezkednek el. Ennek okáért lásd a 2.2.1. fejezetet!



2.3. ábra. Példa két konfliktusosztályra

3. Szimulációs platform

3.1. A SUMO rövid bemutatása

3.1.1. A forgalomszimuláció elméletéről tömören

A forgalommodellezésnek két alapvetően különböző módszere van [11], a mikro- és a makroszkopikus szimuláció.

A makroszkopikus szimuláció lényege, hogy a forgalmat, mint áramló közeget, folyamatot definiáljuk. Itt a fókuszban a forgalomsűrűség és a forgalom nagysága van az egyes útszakaszokon.

Másik megközelítés a mikroszkopikus forgalomszimuláció. Itt a járművek viselkedését külön-külön vizsgáljuk, és olyan modelleket alkotunk, amelyben megjelennek olyan elemek, mint a járműkövetés vagy a sávváltás modellje.

A Simulation of Urban MObility [9] (röviden SUMO) egy nyílt forráskódú mikroszkopikus forgalomszimulátor eszköz. Nyílt forráskódjának köszönhetően elterjedten alkalmazzák különféle kutatási projektekből, illetve ipari (közútkezelői) környezetben is. Széleskörű bővíthetősége jó alapot biztosít az általam javasolt módszerek kipróbálásához is.

3.1.2. A SUMO felépítése

A SUMO az úthálózatot egy gráfként modellezi. A gráf élei a forgalmi sávok, míg a gráf pontjai reprezentálják a közlekedési csomópontokat, a kereszteződéseket. Az élekhez a szimulátor rendel kiterjedést, viszont a gráfpontokhoz nem. Ez egyébként tud érdekességeket okozni, részletekért lásd a 2.2.1. fejezetet még.

Az így felépülő hálózaton mozognak a járművek. A járművek mozgását két (függetlenül kiválasztható) logika befolyásolja. Az egyik a *járműkövetési modell*, amely nevéhez híven a járművek egymás közötti távolságtartási, gyorsítási, lassítási műveleteiért felelős. A másik mozgást befolyásoló modell a *sávváltási modell*. Ez azon kívül, hogy a sávváltásokról dönt, képes valamilyen szintű kooperációt is kialakítani annak érdekében, hogy egy-egy sávváltást kivitelezni tudjon.

A járművekhez kapcsolódhatnak ún. *eszközök (device)* is, ezek tetszőleges funkciókat betölthetnek. Ilyen funkció lehet például egy hálózati kapcsolatot megvalósító modul,

egy elektromos autó esetén az akkumulátor. Ezeket a felhasználó által létrehozott eszközökön át lehet a legegyszerűbben bővíteni a SUMO tudását.

3.2. A SUMO-n alapuló keretrendszer

Ahhoz, hogy érdekes kérdésekkel is foglalkozhassunk az intelligens gépkocsi-forgalom és intelligens forgalomirányítás témakörében, szükségünk van egy biztos alapra, egyfajta keretrendszerre. Egy ilyen keretrendszer az alapként használt forgalomszimulátor mély rétegeit elfedi, így könnyebbé teszi az implementációs folyamatokat. A mérések elvégzéséhez lehet majd felparaméterezni, konkrét algoritmusok implementálásával konkretizálni a működést.

Ezzel a keretrendszerrel szemben támasztunk néhány alapvető követelményt:

- A keretrendszer biztosítson alapvető üzenetküldő és fogadó mechanizmusokat gépjármű–gépjármű, illetve gépjármű–közlekedési csomópont között.
- A keretrendszer adjon interfészt intelligens csomópontok implementációjához.
- A keretrendszer a lehető legnagyobb mértékben fedje el a szimulátor mélyebb rétegeit.
- A keretrendszer oldja meg az intelligens gépjárművek csoportformálását és sáv-váltásait. Tegye ezt olyan módon, melyet később esetleg módosíthatunk.

A keretrendszert első körben absztrakt szinten (pl. konkrét ütemező algoritmusok nélkül) kódoltam le, ezzel lehetőséget teremtve későbbi vizsgálatokhoz elvégzéséhez is.

3.2.1. Rétegmodell

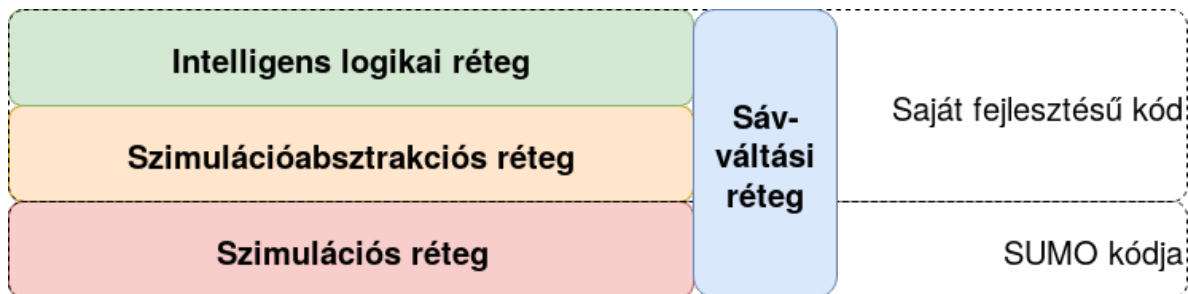
Azért, hogy jól elkülöníthessük a saját logikánkat a szimulátorétól, bevezettem három szoftverarchitektúra-réteget (lásd a 3.1. ábrát).

A legalsó réteg a *szimulátor réteg*. Ez az a szint, ahol csak olyan kódok szerepelnek, melyek a SUMO forráskódbázisából származnak.

Erre az alsó rétegre épül a *szimulátorabsztrakciós réteg*. Ez egyfajta kapocs a felsőbb rétegek és a szimulációs réteg között. A rétegben lévő kódok ismerik a SUMO kódbázisából származó osztályokat, metódusokat. Viszont ezeket az osztályokat elfedik a keretrendszer felsőbb szintjei elől.

A keretrendszer legfelsőbb rétege két részből áll. Egyik részét a keretrendszert használó újabb logikai működést megvalósító modulok fogják adni. A másik részét pedig az egyelőre nem módosításra szánt kódok (absztrakt őosztályok, interfészek) adják. Ezt a szintet összefoglaló néven *intelligens logikai szintnek* nevezem.

Még egy érdekesség, hogy a sáv-váltásokért felelős szoftverrész egy saját réteget kapott *sáv-váltási réteg* néven. Erre azért van szükség, mert ez a kódrész átível a másik három rétegen is, a szimulátor mélyén is dolgozik, hogy megvalósíthasson olyan funkciókat, amelyek egyértelműen az intelligens logika szintjére tartoznának.



3.1. ábra. A rétegmodell szemléltetése

3.2.2. Üzenetküldő rendszer

Üzenetküldő rendszerrel mérlegelnünk kell, hogy érdemes-e expliciten lekódolni egy üzenetküldési mechanizmust, avagy egyszerű függvényhívások szintjén maradjunk. Első esetben a valóságot konkrétabban ábrázoló modellt kapunk, de sajnos ennek a számításigénye sem elhanyagolható. Ezért úgy döntöttem, hogy az üzenetküldést függvényhívásokon keresztül oldom meg.

Univerzalitást a közös, absztrakt alaposztály interfésze ad a rendszernek (pl. közlekedési csomópontok ösosztálya, járművek ösosztálya stb.).

Ugyanakkor elméleti szinten továbbra is beszélek üzenetekről, ugyanis ez filozófiájában, leírásában igen szemléletes, csupán forráskód szintjén csak implicit módon jelenik meg.

3.2.3. Szimulációabsztrakciós eszköz, sáv váltási menedzser

Az intelligens keretrendszer legfőbb kapcsolódási pontja az alap szimulátorhoz a *szimulációabsztrakciós eszköz* (lásd a 3.2. ábrán). Ez az eszköz minden okosautóra „fel van szerelve”.

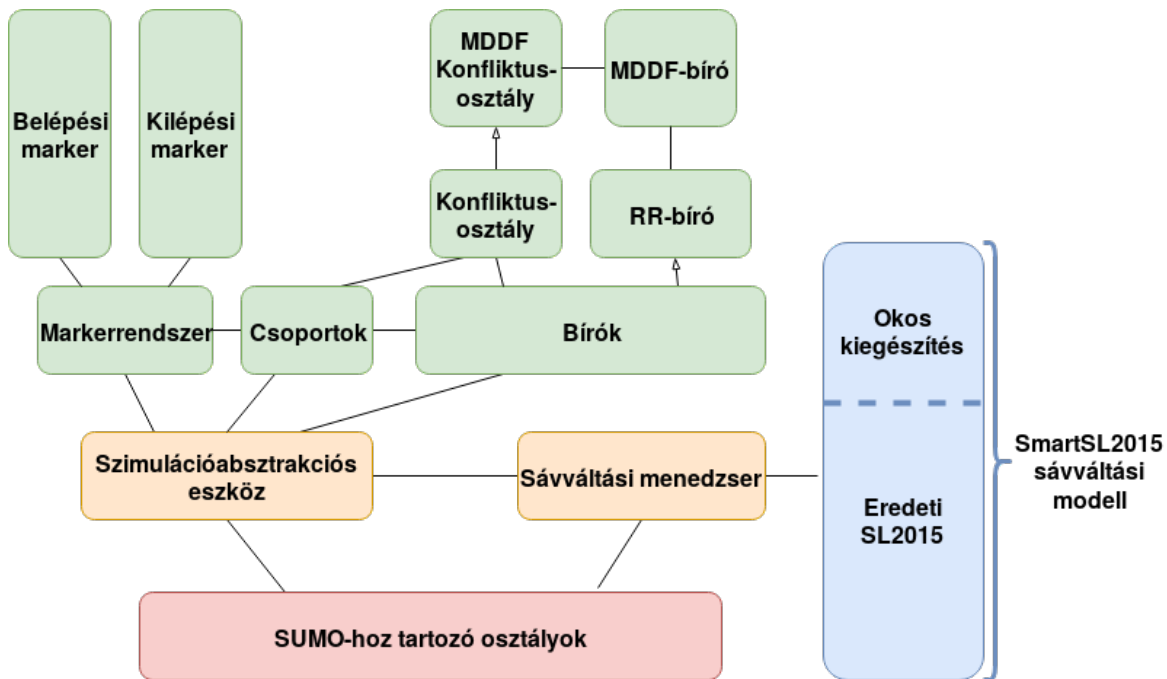
Lényege, hogy az ő megfelelő metódusait a szimulátor minden egyes lépésében meghívja. Ezek az eljárások ismerik a szimulátort, felismerik az okosautók helyzetét, és ezek alapján futtatják le a szükséges algoritmusokat, kezdeményezik az egyes protokollokat.

A Simulation of Urban MObility sáv váltási modellje¹⁴ igen bonyolult. Éppen ezért szükség van valamilyen köztes interfészre az intelligens szimulátorplatform és sáv váltási modell között. Ez az interfész a *sáv váltási menedzser* modul.

3.2.4. Csoportok és a markerrendszer

A *csoportok* és a *markerrendszer* a markerekkel együtt természetesen megjelennek a szimulációs platformban is. Arról, hogy egy jármű egy markerre ér, a szimulációabsztrakciós eszköz értesül, mert a SUMO alapvetően frissíti az eszközök állapotát.

¹⁴Részletekért lásd a 2.4. fejezetet!



3.2. ábra. A SUMO bővítésének áttekintése. A korábban említett szimulációs réteghez a vörössel, a szimulációabsztrakciós réteghez a narancssárgával, míg az intelligens logikai réteghez a zölddel jelölt modulok tartoznak. A SmartSL2015 sávváltási modell egymagában alkotja a sávváltási réteget.

A markerrendszer és a markerek gyakorlatilag információlekérdezésre szolgálnak. Ilyen információ a nevük, a csomópont bírójának címe például.

Ha egy okosautó *belépési markerre* lép, megpróbál belépni egy csoportba, ahogyan az a 2.3.1. fejezetben olvasható.

A szimulátorplatformban megjelenő *csoportok* egy listát tárolnak a csoportjukba tartozó okosautókról. Ez a lista egy FIFO, aki legelőször lépett be a csoportba, az van a legelején (praktikusan a csoportvezető).

3.2.5. A konfliktusosztályok

A *konfliktusosztály*¹⁵ alapvetően egy osztályként is példányosítható interfészt, őosztályt nyújt további konfliktusosztályok leszármaztatásához.

Azon túl, hogy a konfliktusosztály egy listát vezet a beletartozó járművekről, van arra is egy függvénye, hogy csatlakozhat-e hozzá egy-egy újabb csoport vagy autó avagy sem. Alapesetben az őosztály mindenkinek megengedi a csatlakozást.

A konfliktusosztályok rendelkezhetnek például költség számító függvényekkel, melynek konkrét logikája konfliktusosztálytól függ. Így például az *MDDF-bíróhoz* szükséges

¹⁵Definíciót lásd a 2.2.3. fejezetben!

MDDF-konfliktusosztály a csoportjában a minimális céltávolságú okosautó céltávolságával tér vissza.

3.2.6. A bírók

A *bírók* a *konfliktusosztályhoz* hasonlóan egy absztrakt őosztályként léteznek, azzal a különbséggel, hogy ezen őosztályt nem tudjuk példányosítani. Minden intelligens kereszteződéshez¹⁶ tartozik egy bíró.

A bírók a 2.5. fejezetben olvasható módon kiszámítják, hogy egy autó(csoport) melyik konfliktusosztályhoz csatlakozhat, hogy mikor kell új konfliktusosztályt kijelölni áthaladásra, illetve, hogy egy jármű áthaladhat-e éppen a kereszteződésen.

Alapvetően mind a *konfliktusosztály-választó*, mind az *ütemező* logika a konkrét leszármaztatás implementációjától függ. A bírók az okosautók áthaladásról érdeklődő lekérdezéseire ugyanaz az egyszerű logika ad választ: a kérdező autó éppen az áthaladásra kijelölt konfliktusosztályba tartozik-e.

Két leszármazottimplementációt készítettem, az egyik a körforgó, round-robin algoritmuson alapuló *RR-bíró*. A másik a fair MDDF-algoritmuson ([1] módosított formája, részletekért lásd a B.3. fejezetet) alapuló MDDF-bíró. Míg első esetben egyszerű körforgó választást valósít meg a *ütemező* logika, addig az MDDF esetén egy többszintű ütemező eredménye adja a megvalósítás alapját. Viszont mindkét esetben azonos a *konfliktusosztály-választó* logika: topológiai alapon választja ki azt a konfliktusosztályt, amelybe a járművünknek csatlakoznia kell.

Az MDDF-bíróhoz már nem elegendő az alapvető konfliktusosztály-interfész, ezért készítettem neki egy új MDDF-konfliktusosztály objektumot is.

3.2.7. Ismert problémák, furcsaságok és limitációk

Sajnos minden szoftverrendszernek megvannak a saját problémái. A SUMO-val kapcsolatban sincs ez másképp. Az alapul használt verzióban¹⁷ több problémával is szembesültem, röviden ismertetném ezeket.

Mivel a SUMO nem tud hátramenetet szimulálni, ezért elő tud fordulni az, hogy sávváltáskor egy-egy jármű holtpontra kerül. Ezalatt azt értem, hogy olyan sokáig nem sikerül sávot váltania, hogy elfogy a sáv, azaz már nem áll rendelkezésre elegendő hely arra, hogy a sávváltást kivitelezze. Ilyen probléma néha kezdeti szimulációimban is előfordult. Noha én is és a szimulátor alapkódja is igyekszik elkerülni ezeket a helyzeteket, tökéletesen sajnós nem minden esetben lehet ezt megakadályozni.

Másik érdekes probléma az eredeti SL2015-ös modell sajátossága: a járművek képesek „oszcillálni” két sáv között. Az történik ugyanis, hogy a két sáv közötti döntésben egyik pillanatban az egyik forgalmi sáv jobb, a következő pillanatban pedig a másik. Mivel a kedvezőbb sávban akar haladni a járművünk, hol az egyik, hol a másik sáv felé indul el.

¹⁶Ezek definícióját a szimulátorplatform bementként várja.

¹⁷Pontos verziószám: *v0_32_0+2041-7b1f5dd99e*.

Harmadik furcsaság, hogy a járművek nem feltétlenül a hosszabb, de gyorsabb utat választják, hanem inkább a rövidebbet. Például a BAH-csomópontban a Hegyalja út Erzsébet híd felőli oldala és a Budaörsi út között van egy felüljáró. Józan ésszel megmagyarázhatatlan, hogy ha egy jármű ismeri a környéket, akkor miért nem a felüljárót preferálja az utcaszinten lévő nagy-nagy kereszteződéssel szemben.

Programozástechnikai szempontból szomorú tény még, hogy a SUMO grafikus környezete és a szimulátor belső logikája két külön szálon fut. Ezek a szálak használnak (írnak) közös erőforrást, de nem védik azokat megfelelően kritikus szakaszokkal. Ez persze könnyedén tud nem igazán determinisztikus hibákat, és többnyire programleállást is okozni. Szerencsére a SUMO-nak létezik grafikus felület nélküli verziója is, ami szálkezeléstől is mentes.

4. Mérések

4.1. Mérési elrendezések

Az európai városokra a történelem során kanyargósan kialakult úthálózat jellemző, melyek a terület beépítettségétől függően hol szélesebbek, hol szűkebbek. Sok európai nagyvárosban ez egy körutas-sugárutas városszerkezetben nyilvánul meg, így például Budapesten is. Ebben az utcaszerkezetben (a város két széle között) létezik egy legjobb, legrövidebb útvonal, ez az útvonal például két sugárúton és esetleg egy körúton át vezet, melyek a lehető legközelebb vannak a két pontot összekötő húrhoz.

Ezzel ellentétben Amerikában (vagy Ausztráliában) a sakktábla jellegű utcaszerkezet a gyakori, ahol derékszögű kereszteződésekben találkoznak az utak, egymástól nagyjából egyforma távolságra. Itt nem feltétlenül létezik egy legrövidebb út a város két pontja között. Egészen pontosan több azonos hosszú út is létezhet közöttük.

A két eset közötti alapvető különbség miatt két pályán tesztelem a korábban ismertett algoritmusokat, protokollokat. Mivel jelenleg nem egy teljes hálózat optimalizálásán dolgozunk, ezeknek csak egy-egy kisebb szeletét használom a szimulációkhoz.

4.1.1. Mérések pályái

Tesztpálya

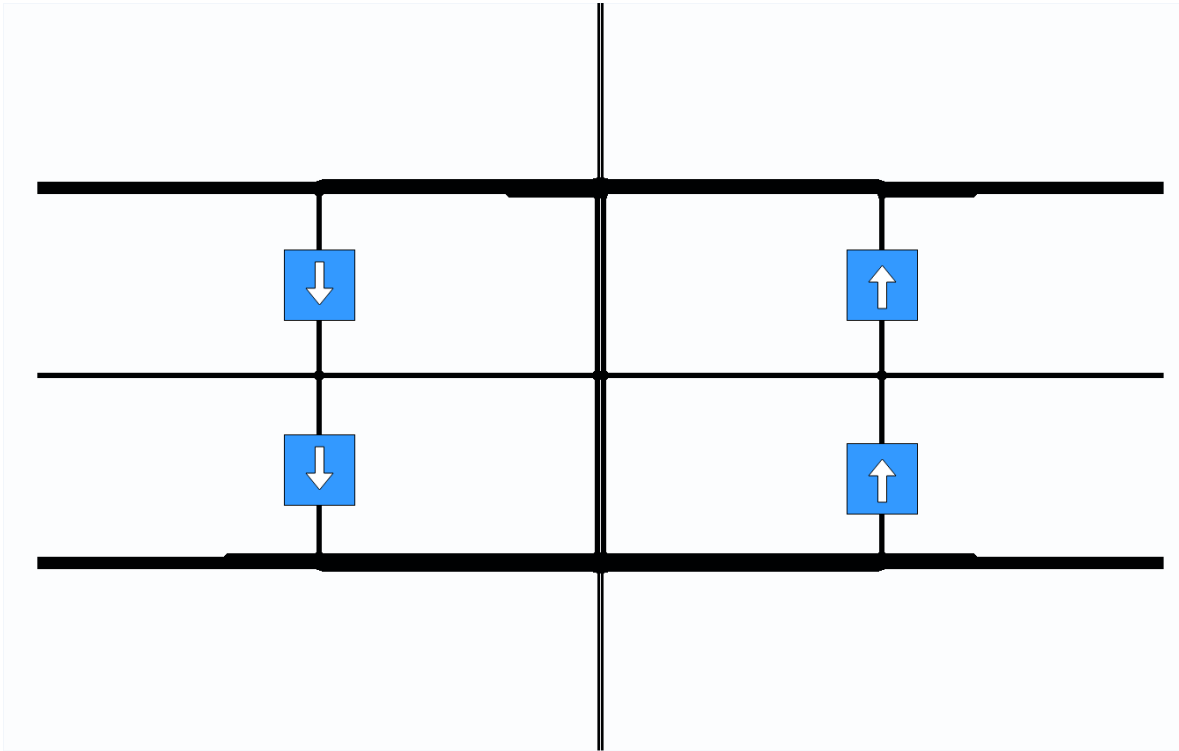
Az amerikai stílusú pálya szemléletésére és az algoritmusok kipróbálására készítettem el a *Tesztpályát*, lásd a 4.1. ábrán.

A Tesztpályát 3-3 kelet-nyugat, illetve észak-dél irányú út alkotja, tehát 9 kereszteződés található a pályán. Négy útszakasz egyirányúsított, hogy változatosabb közlekedési helyzeteket lehessen rajta szimulálni.

A középső észak-dél irányú út, valamint a kelet-nyugat irányú utak közül a két szélső nagyobb forgalmat képes elvinni, ezek párhuzamos közlekedésre alkalmas szakaszok.

BAH-csomópont és környéke

Az európai úthálózatot a budapesti BAH-csomópontot, és annak szűk környékét szimulálom. BAH-csomópontnak a Budaörsi – Alkotás – Hegyalja utak kereszteződését nevezem, beleértve a Villányi és Jagelló utak kereszteződését is.



4.1. ábra. A tesztpálya szemléltetése. A kék nyilakkal jelzett útszakaszok egyirányúak. A pálya be- és kilépési pontjai kelet-nyugat irányú utak végpontjai, illetve az észak-dél irányú utak közül a középső két végpontja. Az útszakaszok vastagsága egyenesen arányos az útszakaszon lévő sávok számával.

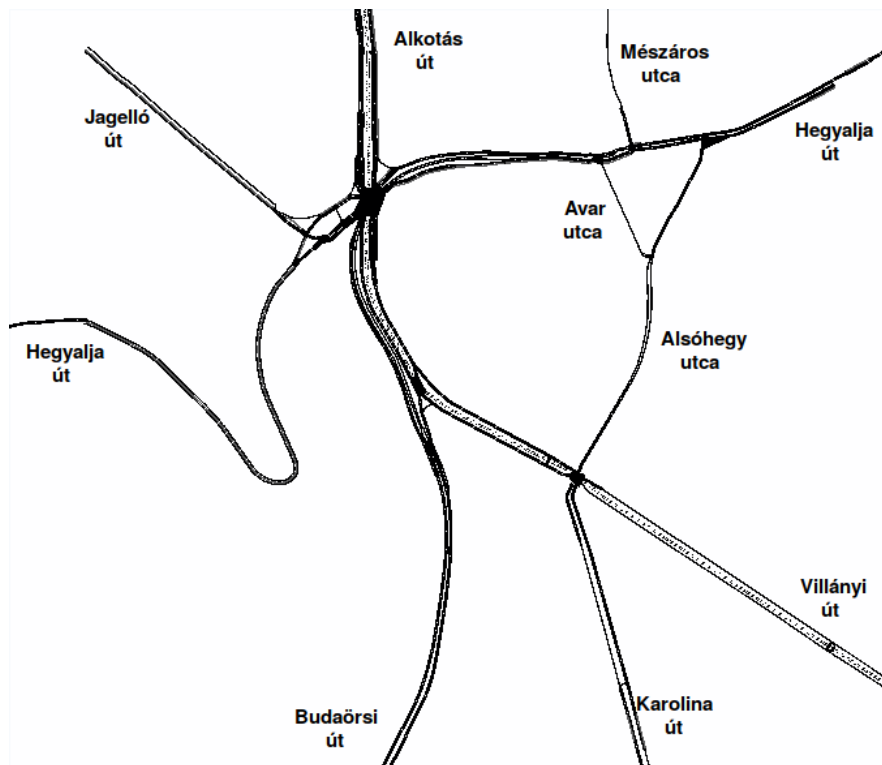
A fent említett utak szimulációjára természetesen szükség van. Ezen kívül a Karolina út és Mészáros utca kereszteződése is fontos a szimuláció szempontjából, ugyanis ezek befolyással vannak a BAH-csomópontba adott idő alatt beáramló járművek szabályozásában. Az Alsóhegy utca azért került bele a szimulált útszakaszok körébe, mert a BAH-csomópont megkerülhető rajta keresztül. Hogy ez a kerülőút a Hegyalja út felől is járható legyen, az Avar utca is bekerült a szimulációba, lásd a 4.2. ábrán.

Az eredeti térkép az OpenStreetMap-ről származik, melyet a SUMO által nyújtott konvertáló eszköz segítségével töltöttem le, majd a fel nem sorolt utcákat eltávolítottam a térképről.

4.1.2. Lámpaprogramok

Ahhoz, hogy az intelligens rendszer által szolgáltatott mérési eredmények összehasonlíthatóak legyenek a jelenlegi forgalmi helyzettel, ezért elvégzek olyan méréseket is, ami a hagyományos közlekedésen alapszik.

Ezekhez a mérésekhez szükség van arra, hogy a pályákon lévő forgalomirányító lám-



4.2. ábra. A BAH-csomópont és környékének térképe

pákra valamilyen lámpaprogramot konfiguráljunk. Ezt a következőképpen tettem meg:

Tesztpálya

A Tesztpályán minden csomópontban a középső kelet-nyugat irányú út és a két egyirányú út kereszteződésén kívül minden más csomópontjában van egy forgalomirányító lámparendszer.

Ezek programozása a következő logikát követi: A két szélső kelet-nyugat irányú úton kelet-nyugat irányban zöldhullámot hozok létre ($50 \frac{km}{h}$ -s sebességet feltételezve). Ez meghatározza a középső észak-dél irányú út két lámpájának váltási fázisait is. Ehhez pedig beállítottam a középső utak kereszteződésében lévő lámpát is úgy, hogy ez is zöldhullámot hozzon létre.

BAH-csomópont

A BAH-csomópont kereszteződéseinek lámpaprogramjai [17] közül az éjszakai, reggeli és napközbeni lámpaprogramokat használok fel.

Ezeket a kereszteződéseken kívül a Villányi – Karolina út valamint a Hegyalja – Mészáros, Hegyalja – Avar, Hegyalja – Alsóhegy utcák kereszteződésében vannak forgalomirányító lámpák. Ezeket a lámpákat úgy konfiguráltam fel, hogy a BAH-csomópontba befelé zöldhullámot hozzanak létre (most is $50 \frac{km}{h}$ -s sebességet feltételezve).

4.1.3. Normális és rendhagyó forgalom

A BAH-csomópont modelljében szeretnék szimulálni egy érdekes dolgot. Tétélezzük fel, hogy a Budaörsi úton történik valami rendhagyó esemény (pl. baleset), ami megakadályozza, hogy a forgalom például a Budaörsi útról a felüljáróra vagy a BAH-csomópontba érkezzen. Ilyenkor a járművek persze kerülőutat keresnek az úticéljuk felé, és ezért szokatlan irányokból megnő a forgalom. Az ilyen és ehhez hasonló eseteket hívom *rendhagyó forgalomnak*.

Ezzel szemben *normális forgalom* van jelen, ha nincs semmilyen rendkívüli esemény. Persze gondoljuk végig, hogy ez reggel és este egy aszimmetrikus terhelés. Reggel a BAH-csomópontban az agglomeráció valamint a XI. és XII. kerület lakói igyekeznek a város központja felé (pl. Budaörsi út – Hegyalja út keleti vége vagy az Alkotás út irányában), míg este ők pont hazafelé, kifelé tartanak a városból. Nappal és éjszaka a forgalom ennél természetesen szimmetrikusabb.

4.1.4. (Intelligens) lámpavezérlés

A mérések fő célja természetesen az, hogy az RR és MDDF ütemezők valamint a teljes intelligens rendszer kipróbálásra kerüljön. Ehhez definiálnunk kell a topológiát a bírók számára. Ilyen topologikus adat a csomópont középpontjának koordinátái, a megállási távolság, valamint a kritikus pont távolsága.

Ezek a Tesztpályán jól definiálható távolságok, minden kereszteződésben megegyeznek, a megállási távolság 20 m-nél, a kritikus pont 15 m-nél található. A csomópontok közepei között pedig kelet-nyugat irányban 300 m-nyi, észak-dél irányban 200 m-nyi távolság van. Az észak-nyugati kereszteződés kezdőpozíciója: (−200; 300).

A BAH-csomópont koordinátáinak leírása ennél bonyolultabb, azt a 4.1. táblázat szemlélteti.

| Csomópont | Középpont koordinátái | Megállási távolság (m) | Kritikus pont távolsága (m) |
|-------------------------------|-----------------------|------------------------|-----------------------------|
| Alsóhegy utca – Hegyalja út | (1719; 1579) | 26 | 22 |
| Mészáros utca – Hegyalja út | (1589; 1571) | 12 | 7 |
| Avar utca – Hegyalja út | (1518; 1552) | 25 | 23 |
| Budaörsi – Alkotás – Hegyalja | (1101; 1474) | 59 | 57 |
| Jagelló út – Hegyalja út | (981; 1410) | 33 | 31 |
| Villányi út – Budaörsi út | (1189; 1124) | 57 | 55 |
| Villányi út – Karolina út | (1485; 965) | 25 | 21 |

4.1. táblázat. A BAH-csomópont és a környékének topologikus beállítása

4.2. Mérendő mennyiségek

4.2.1. Áthaladó autók mennyisége

A szimulátor eredményként megadja, hogy hány járművet indított el (*elindult*), illetve, hogy jelenleg hány jármű tartózkodik a pályán (*fut*). Ebből a két mennyiségből származtatom a *megérkezett* járművek számát:

$$\text{megérkezett} = \text{elindult} - \text{fut}$$

4.2.2. Időeredmények

A szimulátor kimeneteként a következő időeredményeket szolgáltatja:

- *Átlagos idő (s)*: egy járműnek átlagosan ennyi idejébe telt áthaladni a hálózaton.
- *Átlagos várakozás (s)*: egy jármű átlagosan ennyi időt töltött el forgalmi lámpánál (vagy bírónál) történő állással.
- *(Átlagos) elvesztegetett idő (s)*: Az az idő, amelyet a járművek átlagosan elvesztettek azzal, hogy nem egyedül voltak a hálózatban, és nem kaptak mindig 'szabad' jelzést a forgalomirányítástól. Ez az adat magában foglalja az *átlagos várakozást* is.

4.2.3. Egyéb eredmények

A szimuláció eredménye tartalmazza vagy abból származtathatók a következők:

- *Átlagos úthossz (km)*: Az ütemezésnek közvetett hatása van az úthosszra. Ha egy csomópontban torlódás alakul ki, akkor a járművek kerülőúton haladnak.
- *Átlagsebesség (km/h)*: Származtatott mennyiség a következők szerint:
$$\text{átlagsebesség} \left[\frac{\text{km}}{\text{h}} \right] = \frac{\text{átlagos úthossz}}{\text{átlagos idő}} \cdot 3,6$$
- *Érzékenység*: Egy ütemezési stratégia érzékenységén azt értem, hogy a módszer által átengedett autók arányát ($\frac{\text{megérkezett}}{\text{elindult}}$) mennyire befolyásolja a forgalom jellege (iránya, nagysága).

4.3. Mérési módszerek

4.3.1. Tesztpályán végzett mérések

A Tesztpályán egy előre meghatározott, fiktív forgalmat indítok el. A szimulált idő 500 s.

Három mérést végzek el:

1. Hagyományos forgalom mérése.
2. Intelligens rendszer mérése, minden bíró *RR* típusú.
3. Intelligens rendszer mérése, minden bíró *MDDF* típusú.

4.3.2. BAH-csomópontban végzett mérések

A BAH-csomópontbeli mérések szimulált idejét 300 s-ban határoztam meg, ugyanis a módszerek számítási igénye nem kevés, így egy-egy szimuláció lefuttatása is jelentős időt vesz igénybe. Továbbá 300 s alatt úgy tapasztaltam, hogy az alapvető karakterisztikák már megjelennek a mérési eredményekben.

Forgalombecslés

Sajnos a SUMO kétféle forgalomleírást fogad el. Az egyik módszer, hogy járműnek minden egyes csomópontot előírunk, amit érintenie kell. A másik, hogy megadjuk a szimulált útvonal be- és kilépési pontját.

Az első módszer ugyan a szimuláció szempontjából kívánatosabb, ugyanakkor a gyakorlati helyzetet nem jól modellezi (elvégett sávváltások, torlódások elkerülése stb.). A második módszerhez viszont nem léteznek valós bemeneti adatok, viszont ez szimuláció tekintetében pontosabban modellezi a valóságot, ezért ezt a módszert választottam.

A bemenő forgalmi adatokat tehát így csak becsülni tudom. Elkészítettem tehát egy éjszakai (kis forgalmú), reggeli (nagy forgalmú, aszimmetrikus) és egy napközbeni (közepes, szimmetrikus) forgalombecslést.

A generált forgalmat kizárólag egyforma, átlagos menettulajdonságokkal bíró személyautók alkotják.

Normális forgalom melletti mérések

A módszerek teszteléséhez egy kis forgalmú időszak (éjszaka) forgalmi becslései és egy nagy forgalmú időszak (reggel) forgalmi becslései alapján elvégzek 3-3 mérést. Ezek a következők:

1. Hagyományos forgalom mérése.
2. Intelligens rendszer mérése, minden bíró *RR* típusú.
3. Intelligens rendszer mérése, minden bíró *MDDF* típusú.

Rendhagyó forgalom melletti mérések

A statikus, hagyományos lámpaprogramok feltehetőleg nincsenek felkészülve arra, hogy rendkívüli forgalmi igények lépjenek fel. Így kíváncsi vagyok arra, hogyan teljesítenek egymáshoz képest a hagyományos és az intelligens módszerek.

Ezért készítettem két rendhagyó scenáriót:

- Abnormális forgalmat szimulálok oly módon, hogy a BAH-csomópont és a felüljáró felé a Budaörsi útról nem érkezik jármű. Helyette ezek a járművek a Villányi út felől jelennek meg. Ezt *abnorm1* mérésorozatnak nevezem.
- Abnormális forgalmat szimulálok oly módon, hogy a BAH-csomópontból kifelé, a Budaörsi útra nem haladnak járművek (a felüljárón sem). Helyette a járművek a Hegyalja út nyugati vége felé kívánnak eljutni. Ezt *abnorm2* mérésorozatnak nevezem.

Ezzel a két esettel is elvégzek 3-3 mérést, ezek a már megszokottak:

1. Hagyományos forgalom mérése.
2. Intelligens rendszer mérése, minden bíró *RR* típusú.
3. Intelligens rendszer mérése, minden bíró *MDDF* típusú.

4.4. Mérési eredmények

4.4.1. Tesztpályán végzett mérések

A tesztpályán végzett mérések eredményeit a 4.2. táblázat és a 4.3. táblázat tartalmazza, míg a mérési eredmények grafikus ábrázolását lásd a 4.3. ábrán.

| Mérési eset | Elindult | Fut | Megérkezett | Átlaghossz (m) | Átlagsebesség ($\frac{km}{h}$) |
|-------------|----------|-----|-------------|----------------|----------------------------------|
| Hagyományos | 398 | 197 | 201 | 1063,04 | 24,49 |
| RR | 398 | 104 | 294 | 1063,51 | 32,41 |
| MDDF | 398 | 114 | 284 | 1070,02 | 29,20 |

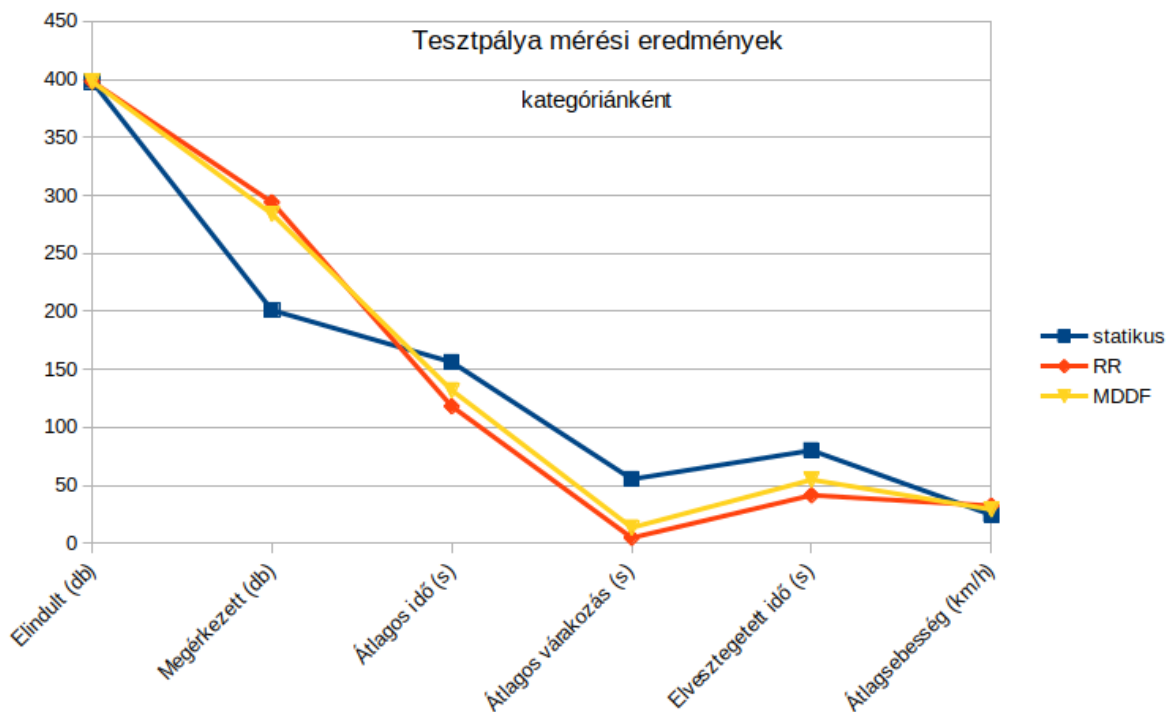
4.2. táblázat. Tesztpályán végzett mérési eredmények I.

| Mérési eset | Átlagos idő (s) | Átlagos várakozás (s) | Elvesztegetett idő (s) |
|-------------|-----------------|-----------------------|------------------------|
| Hagyományos | 156,25 | 55,41 | 80,16 |
| RR | 118,12 | 4,81 | 41,48 |
| MDDF | 131,94 | 13,6 | 54,84 |

4.3. táblázat. Tesztpályán végzett mérési eredmények II.

4.4.2. BAH-csomóponton végzett mérések

A BAH-csomópontban két mérésorozatot futtatam le. Az első sorozat normális forgalom mellett, különböző forgalomnagyságokkal zajlott. A második sorozat pedig a rendhagyó forgalmi helyzeteket szimulálja.



4.3. ábra. A tesztpályán végzett mérések szemléltetése kategóriánként

Kis és nagy terhelés összehasonlítása

A kis terhelés mellett, a BAH-csomópontban végzett mérések eredményeit a 4.4. táblázat és a 4.5. táblázat tartalmazza.

| Mérési eset | Elindult | Fut | Megérkezett | Átlaghossz (m) | Átlagsebesség ($\frac{km}{h}$) |
|-------------|----------|-----|-------------|----------------|----------------------------------|
| Hagyományos | 96 | 40 | 56 | 1799,96 | 38,24 |
| RR | 95 | 47 | 48 | 1815,6 | 39,66 |
| MDDF | 95 | 47 | 48 | 1886,82 | 37,89 |

4.4. táblázat. BAH-csomóponton, kis terheléssel végzett mérési eredmények I.

A nagy terhelés mellett, a BAH-csomópontban végzett mérések eredményeit a 4.6. táblázat és a 4.7. táblázat tartalmazza.

Rendhagyó forgalommal történő mérések

A 4.3.2. fejezetben már ismertetett két rendhagyó forgalmi esettel (*abnorm1* és *abnorm2* mérések) is lefutattam a szimulációt. Az *abnorm1* eredményét a 4.8. táblázat és a 4.9. táblázat, míg az *abnorm2* eredményét a 4.10. táblázat és a 4.11. táblázat tartalmazza.

| Mérési eset | Átlagos idő (s) | Átlagos várakozás (s) | Elvesztegetett idő (s) |
|-------------|-----------------|-----------------------|------------------------|
| Hagyományos | 169,45 | 22,6 | 39,85 |
| RR | 164,81 | 5,95 | 33,36 |
| MDDF | 179,28 | 12,77 | 42,67 |

4.5. táblázat. BAH-csomóponton, kis terheléssel végzett mérési eredmények II.

| Mérési eset | Elindult | Fut | Megérkezett | Átlaghossz (m) | Átlagsebesség ($\frac{km}{h}$) |
|-------------|----------|-----|-------------|----------------|----------------------------------|
| Hagyományos | 377 | 241 | 136 | 1760,93 | 37,40 |
| RR | 376 | 253 | 123 | 1673,96 | 35,43 |
| MDDF | 376 | 252 | 124 | 1679,37 | 35,15 |

4.6. táblázat. BAH-csomóponton, nagy terheléssel végzett mérési eredmények I.

| Mérési eset | Átlagos idő (s) | Átlagos várakozás (s) | Elvesztegetett idő (s) |
|-------------|-----------------|-----------------------|------------------------|
| Hagyományos | 169,48 | 22,37 | 42,71 |
| RR | 170,07 | 11,43 | 38,59 |
| MDDF | 172,01 | 14,43 | 49,21 |

4.7. táblázat. BAH-csomóponton, nagy terheléssel végzett mérési eredmények II.

| Mérési eset | Elindult | Fut | Megérkezett | Átlaghossz (m) | Átlagsebesség ($\frac{km}{h}$) |
|-------------|----------|-----|-------------|----------------|----------------------------------|
| Hagyományos | 346 | 229 | 117 | 1646,36 | 34,75 |
| RR | 346 | 245 | 101 | 1559,77 | 32,11 |
| MDDF | 346 | 243 | 103 | 1413,85 | 33,05 |

4.8. táblázat. BAH-csomóponton, *abnorm1* esettel végzett mérési eredmények I.

| Mérési eset | Átlagos idő (s) | Átlagos várakozás (s) | Elvesztegetett idő (s) |
|-------------|-----------------|-----------------------|------------------------|
| Hagyományos | 170,55 | 29,68 | 51,98 |
| RR | 174,87 | 12,11 | 61,97 |
| MDDF | 154,02 | 12,41 | 51,45 |

4.9. táblázat. BAH-csomóponton, *abnorm1* esettel végzett mérési eredmények II.

| Mérési eset | Elindult | Fut | Megérkezett | Átlaghossz (m) | Átlagsebesség ($\frac{km}{h}$) |
|-------------|----------|-----|-------------|----------------|----------------------------------|
| Hagyományos | 525 | 323 | 202 | 1824,95 | 32,95 |
| RR | 376 | 253 | 123 | 1673,96 | 35,43 |
| MDDF | 314 | 206 | 108 | 1727,22 | 35,19 |

4.10. táblázat. BAH-csomóponton, *abnorm2* esettel végzett mérési eredmények I.

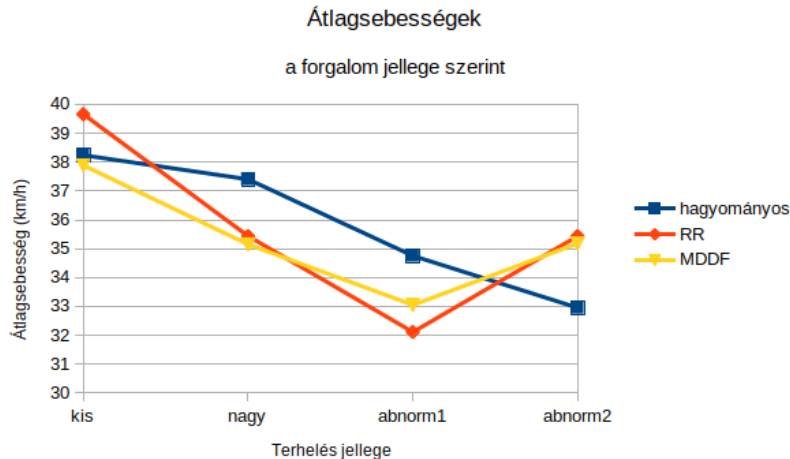
| Mérési eset | Átlagos idő (s) | Átlagos várakozás (s) | Elvesztegetett idő (s) |
|-------------|-----------------|-----------------------|------------------------|
| Hagyományos | 199,38 | 36,44 | 67,96 |
| RR | 170,07 | 11,43 | 48,59 |
| MDDF | 176,72 | 10,74 | 51,23 |

4.11. táblázat. BAH-csomóponton, *abnorm2* esettel végzett mérési eredmények II.

4.5. Mérési eredmények értékelése

4.5.1. Átlagsebességek összehasonlítása

A tesztpályán, a várt módon a hagyományos eset bizonyult a leglassabbnak ($24,49 \frac{km}{h}$ sebességgel), ezt követte az MDDF ($29,19 \frac{km}{h}$) és a legnagyobb átlagsebességet a RR ütemező produkálta ($32,41 \frac{km}{h}$).



4.4. ábra. A BAH-csomópontbeli átlagsebességek alakulása a forgalom jellege és az ütemező típusa szerint

A BAH-csomópontbeli méréseket megfigyelve (lásd a 4.4. ábrán) is az adódik, hogy az RR ütemező az esetek többségében nagyobb átlagsebességet biztosít, mint az MDDF. Ennek az lehet az oka, hogy az RR ütemező minden ütemezéskor olyan konfliktusosztályt választ áthaladásra, aki nincs blokkolva, míg az MDDF esetében előfordulhat az, hogy a legrövidebb hátralévő távolsággal rendelkező konfliktusosztály nem tud megmozdulni, mert állnak előtte. Ilyenkor kb. 3 s-nak el kell telnie ahhoz, hogy ez a bírónál újraütemezést okozzon.

Érdekesebb azonban, hogy a hagyományos lámpaprogram adott esetben jobb eredményeket ér el, mint az intelligens megoldások. Ennek alapvetően három oka lehet.

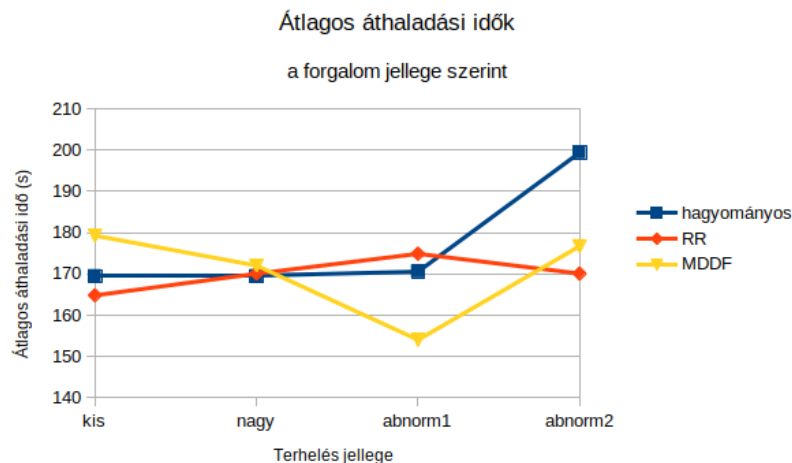
1. Az implementáció tökéletlensége okozhat némi sebességsökkenést.
2. Az átlagos terhelésre a valós, statikus ütemezés a végletekig optimalizálva van. Erről tanúskodik, hogy a „kis” esetben az RR, az „abnorm2” esetben az RR és az MDDF is nagyobb átlagsebességet biztosít, hiszen ilyenkor vagy nincs kihasználva a teljes kapacitás („kis”) vagy a kapacitást túlzott mértékben is használni szeretnék („abnorm2”). Az „abnorm1” esetben a járművek a Villányi út felől be tudnak hajtani a BAH-csomópontba, így pedig ki tudják használni az optimalizált lámpaprogramot.

- Végül pedig az is okozhat átlagsebességcsökkenést, hogy a tesztkonfiguráció nem tökéletes. Bizonyára létezik optimális megállási és kritikuspont-távolság is, úgy, mint optimális kilépési markerpozíció is. Ezeket jelenleg nem volt céлом megtalálni.

Még egy érdekes tény látszik a 4.4. ábrán. Méghozzá az „abnorm1” esetben az MDDF (a szokásostól eltérő módon) nagyobb átlagsebességet biztosít, mint az RR. Ez bizonyára annak köszönhető, hogy itt (főleg a Villányi út – Alsóhegy utcák kereszteződésénél) igen nagy szerepe van a hátralevő távolságnak, amit az MDDF számításához fel is használ, és itt előnyt is jelent ez neki.

4.5.2. Áthaladási idők összehasonlítása

A tesztpályán alternatív útvonalat nem érdemes választani, ezt mutatják a nagyjából megegyező távolságadatok is (két 1063 m-es és egy 1070 m-es átlagos eredmény). Így az átlagsebességeknek megfelelően a leglassabb a hagyományos módszer, melynél rendre gyorsabb az MDDF és az RR.



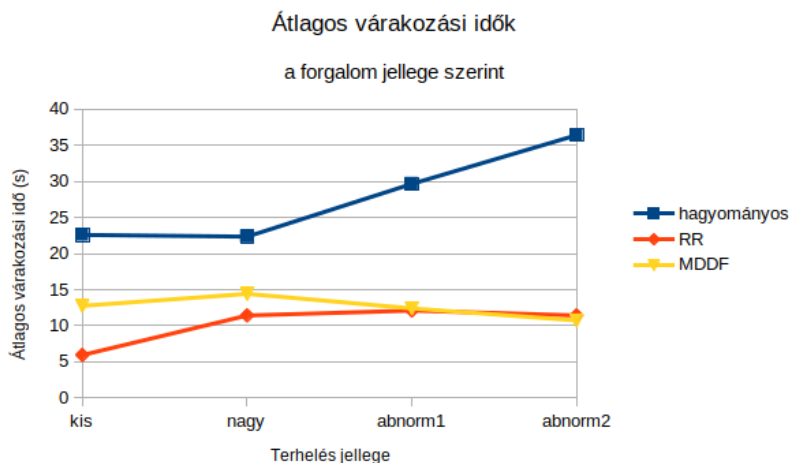
4.5. ábra. A BAH-csomópontbeli átlagos áthaladási idők alakulása a forgalom jellege és az ütemező típusa szerint

A 4.5. ábráról leolvasható, hogy olyan esetekben, amikor a csomópontok elvileg képesek megbirkózni az érkező forgalommal („kis”, „nagy” és „abnorm1” esetek), olyankor az RR és a hagyományos ütemező hasonlóan teljesít. Az MDDF főleg akkor teljesít jól, ha az ütemezési döntés számára fontos a hátralevő távolság („abnorm1”). Egyébként pedig nagyjából 10%-os határon belül ugyanúgy teljesít, mint az RR ütemező.

4.5.3. Átlagos várakozási idők összehasonlítása

A tesztpályán az MDDF és az RR is gyakorlatilag 0-ra képes csökkenteni az átlagos várakozási időt. Ezt valószínűleg a topológia biztosítja, ugyanis csak a középső észak-dél irányú úton fordulhat az elő, hogy autócsoportok pontosan egyszerre érkeznek egy kereszteződésbe. Ezt a hatást pedig a találkozó két konfliktusosztály az első kereszteződésben legyőzi, innentől kezdve az ütemezők őket sorba rendezik úgy, hogy a következő kereszteződésbe már megint egymás után érkeznek.

Ez egy zöldhullámnál nem feltétlenül van így, ugyanis ha három zöldhullám van (északi és déli végen egy-egy, és a középső észak-déli úton egy) a rendszerben, akkor két hullámhatáron át kell menni, és az első hullámhoz csatlakozni, ami legrosszabb esetben akár három periódusnyi várakozást is jelenthet.



4.6. ábra. A BAH-csomópontbeli átlagos várakozási idők alakulása a forgalom jellege és az ütemező típusa szerint

A 4.6. ábrán a BAH-csomópontbeli átlagos várakozási idők alakulását láthatjuk. RR ütemezés esetén a várakozási idők konvergensenek tűnnek a forgalom nagyságától függően, de annak irányától függetlenül.

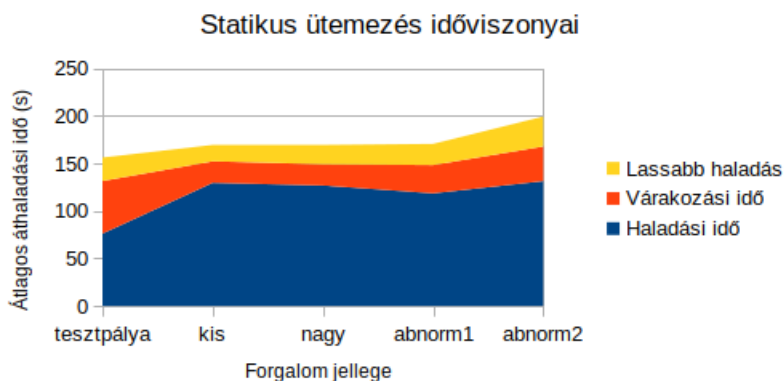
Az MDDF hasonlóan az operációs rendszerek legrövidebb taszkot előre (Shortest-Job-First) várakozási időre nézve optimális megoldás¹⁸, ezt a mérési adatok is alátámasztják. Persze azzal számolnunk kell, hogy itt nem vehetünk előre tetszőleges autócsoportot, mint ahogyan azt a „kis” adatsornál látható kiugró érték is mutatja.

Hagyományos esetben (nem meglepő módon) folyamatosan romlik az átlagos várakozási idő, ahogy egyre nagyobb vagy egyre szokatlanabb forgalmi igények merülnek fel.

¹⁸Egy közlekedési csomópontbeli áthaladási időt (ami az operációs rendszerek esetében a végrehajtási idő) konstansként értelmezve.

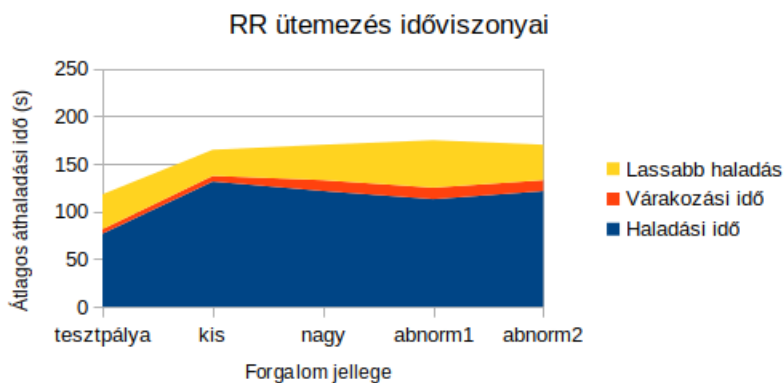
4.5.4. Ütemezések időviszonyai

A mérések elvégzésével a következőket tudjuk kimutatni az egyes ütemezések időviszonyairól.



4.7. ábra. A hagyományos, statikus ütemezés időviszonyai

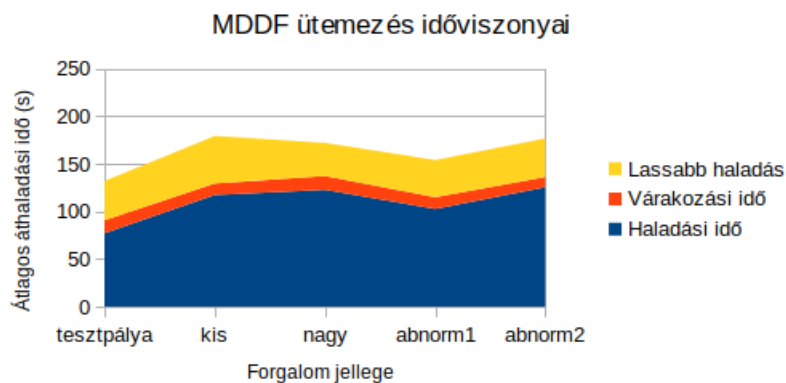
A hagyományos rendszerben (statikus ütemezés) megszokott módon egyre kevésbé haladunk, és egyre többet állunk piros lámpánál, ahogyan nő a forgalom és minél rendhagyóbb irányból érkezik. Ezt mutatja a 4.7. ábra is.



4.8. ábra. Az RR ütemezés időviszonyai

Ahogyan azt már korábban elemeztük, az RR ütemezés várakozási ideje egy adott értékhez konvergál. Viszont a forgalmat ettől függetlenül lassítja („abnorm1” esetén nagyobb a „lassabb haladással” töltött időtartam), ahogyan azt a 4.8. ábráról leolvashatjuk.

Ahogyan azt az ütemezők elméletéből ismerjük, az MDDF optimális ütemezést biztosít. Ezt mutatja a 4.9. ábra is. A várakozási és a lassabb haladási idő szinte konstans-

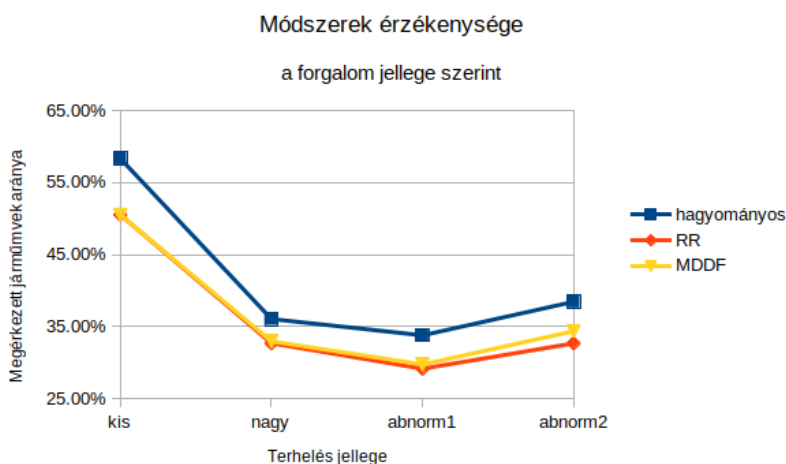


4.9. ábra. Az MDDF ütemezés időviszonyai

nak tekinthető a különböző esetekben, a teljes áthaladási időt jóformán csak a tényleges haladási idő befolyásolja.

4.5.5. Az ütemezési módszerek érzékenysége

Utolsó kérdésként megvizsgáljuk, hogy az egyes módszerek mennyire érzékenyek a terhelés jellegére vonatkozóan. Ehhez a BAH-csomópontbeli mérési eredményeket veszem alapul, ugyanis az itteni mérésekből több adatsor is rendelkezésre áll.



4.10. ábra. Az ütemezők érzékenysége a terhelés jellegére

Az érzékenységet a 4.10. ábrán ábrázoltam. Az ábrán jól látható, hogy a hagyományos ütemezés érzékenyebb a forgalomra, mint az intelligens megoldások. Leolvasható az is, hogy az RR és az MDDF között nincsen számottevő különbség érzékenységben.

A görbék alakja igen hasonló, ennek valószínűleg az az oka, hogy a járműveknek a kereszteződésen kívül is van helyigénye, azaz tetszőlegesen sok jármű nem fér el véges sok felületen.

Mivel a függvények alakja igen hasonló, ráadásul a két különböző intelligens megvalósítás gyakorlatilag egybeesik, ezért feltehetően valamilyen konfigurációs tökéletlenség okozza azt, hogy nagyjából 10%-kal kevesebb járművet engedünk át a hagyományos megoldásnál.

5. Összefoglalás, konklúzió

5.1. Az eredmények kritikus értékelése

Csoportosítás-megvalósításom a [12] által említett PATH projekt platoondefinícióhoz hasonlít leginkább. Itt is csak kétféle szereppel (csoportvezető és csoporttag) találkozhatunk. A csoportok képességeit is a sávváltás, a sávban történő haladás, a csatlakozás és a csoport felbontása képezi.

Láthatjuk, hogy az intelligens rendszer működését alapvetően befolyásolja a pálya topológiája, a rendszer konfigurációja. Megfelelő környezetben és jó konfigurációval mind az RR ütemezőre, mind az MDDF ütemezőre épülő forgalmi hálózat eredményei minden mért paraméter szempontjából messze meghaladják a hagyományos módszer eredményeit.

Ez főleg a kommunikáción alapuló forgalomirányításnak köszönhető, aminek a munkáját a csoportosítás segíti: Az összeállt autócsoportok ugyanis csökkentik a „kontextusváltások”, azaz a konfliktusosztályok közötti váltások számát. Ezek a csoportok kialakulásuk után együtt maradnak, és az okos sávváltásnak köszönhetően maguk közé nem engednek be újabb járművet. Így könnyen beláthatjuk, hogy legfejlebb annyi konfliktusosztály-váltásra van szükség, mint hagyományos esetben. Ha ugyanis legalább két okosautóból álló csoportok érkeznek, a potenciális konfliktusosztály-váltások száma felére csökken. Akár olyan is előfordulhat, hogy csak egy óriási csoport érkezik be egy közlekedési csomópontba. Ilyenkor nincs is szükség konfliktusosztály-váltásra. A másik véglet, amikor egy autóból álló csoportok érkeznek. Nos, ez a hagyományos helyzet, tehát a csoportosítással valóban csökkentjük a konfliktusosztály-váltások számát.

Sajnos a BAH-csomópontban végzett mérések esetében az intelligens módszerek többnyire nem nyújtották a várt eredményeket. Ez feltehetően a nem optimális konfigurációnak köszönhető. Viszont az eredmények így is azt mutatják, hogy az intelligens módszerek átlagos áthaladási idejét, illetve várakozási idejét kevésbé befolyásolja a megjelenő forgalom nagysága és iránya, mint a hagyományos rendszerét.

Megfigyelhettük azt is, hogy az MDDF algoritmus valóban optimális eredményt ad az átlagos várakozási időkre.

A módszerek érzékenységét vizsgálva kitűnik, hogy számottevő különbséget nem tapasztalunk az RR és az MDDF módszerek között. A BAH-csomópontban mindkét intelligens rendszer alulmúlja érzékenység szempontjából a hagyományos rendszert. Ezt megint csak a konfiguráció optimalizálatlanságára vezethető vissza.

Bár léteznek olyan forgalomirányító berendezések, amelyek figyelembe veszik az aktuális forgalmat, ezek többnyire drága, bonyolult és nem túl precíz eszközökkel (pl. hurrokdetektorok, infraszenzorok, súlymérők stb.) mérik a forgalmat. A tapasztalat azt mutatja, hogy ezek az eszközök általában nem veszik észre, ha egy kerékpáros vagy akár egy motorokerékpáros közeledik a kereszteződéshez. Ráadásul az útburkolatba épített érzékelők javítása sem egyszerű feladat.

A kommunikáción alapuló intelligens ütemezés ezzel szemben mindössze egy rádiómodul meglétét feltételezi. Ez megbízhatóbb, olcsóbb, egyszerűbben javítható komponens, mint a ma használt megoldások. Akár egy kerékpáros is magával tudja vinni, így garantált, hogy előbb-utóbb ő is ütemezésre kerül. Tehát megvalósítás szempontjából is előnyös tud lenni a kommunikáción alapuló forgalomirányítás.

5.2. Továbbfejlesztési lehetőségek, újabb kísérletek

Már többször szóba került a konfiguráció optimalizálatlansága. Valószínűleg optimális beállítások mellett el lehetne érni a BAH-csomópontban is, hogy az intelligens megoldások legalább annyira hatékonyan működjenek áteresztés szempontjából, mint a hagyományos megoldás.

Az ütemezési algoritmusokat módosítani lehetne úgy, hogy előnyben részesítsék a megkülönböztető jelzést használó (pl. mentő, tűzoltó, rendőr) járműveket az átlagos járművekkel szemben.

Jelenleg a BAH-csomópontot három külön csomópontként modellezem (a Villányi – Budaörsi utak, a Budaörsi – Alkotás – Hegyalja utak és a Hegylja – Jagelló utak kereszteződéseiként). Persze logikailag az egész egy nagy csomópont, így érdekes, hogy van-e értelme a kialakult csoportokat minden kereszteződés után felbontani. Elképzelhető lenne egy hierarchikus felépítés, ahol magas szinten lenne egy BAH-csomópont, és ennek alcsomópontjai lennének a fent említett keresztezések. A csoportok pedig a magas szintű BAH-csomóponton belül nem bomlanának fel.

Akár hierarchikus felépítést tekintve, akár független csomópontonként modellezve érdekes kérdés lehet, hogy mennyiben javítaná a forgalom paramétereit az, ha a bírók megosztanák valamilyen módon egymással tudásukat.

A vázolt algoritmusok feltételezik az okosautók teljes kooperativitását. Érdeemes volna megvizsgálni azt is, hogy mennyire befolyásolja helyzetet az, ha a teljes kooperativitás nem áll fenn. Akár úgy, hogy valaki nem vesz részt a kooperációban (pl. egy emberi vezető), akár úgy is, hogy ártó szándékkal fellép a kooperáló ágensekkel szemben.

6. Irodalomjegyzék

- [1] F. Ahmad, S. A. Mahmud, G. M. Khan, és F. Z. Yousaf, „Shortest remaining processing time based schedulers for reduction of traffic congestion”, in 2013 International Conference on Connected Vehicles and Expo (ICCVE), Las Vegas, NV, USA, 2013, o. 271–276.
- [2] A. A. Alam, A. Gattami, és K. H. Johansson, „An experimental study on the fuel reduction potential of heavy duty vehicle platooning”, in 13th International IEEE Conference on Intelligent Transportation Systems, 2010, o. 306–311.
- [3] J. J. Besa Vial, W. E. Devanny, D. Eppstein, és M. T. Goodrich, „Scheduling Autonomous Vehicle Platoons Through an Unregulated Intersection”, 2016.
- [4] C. Campolo, Vehicular ad hoc networks: standards, solutions, and research. New York, NY: Springer Science+Business Media, 2015.
- [5] J. Erdmann, „SUMO’s Lane-Changing Model”, in Modeling Mobility with Open Data, M. Behrisch és M. Weber, Szerk. Cham: Springer International Publishing, 2015, o. 105–123.
- [6] P. Fernandes, „Platooning of IVC-enabled Autonomous Vehicles Information and Positioning Management Algorithms for High Traffic Capacity and Urban Mobility Improvement”, 2012.
- [7] H. C. Hsu és A. Liu, „Kinematic Design for Platoon-Lane-Change Maneuvers”, IEEE Transactions on Intelligent Transportation Systems, köt. 9, sz. 1, o. 185–190, jan. 2008.
- [8] W.-L. Jin, „A kinematic wave theory of lane-changing traffic flow”, arXiv:math/0503036, márc. 2005.
- [9] D. Krajzewicz, J. Erdmann, M. Behrisch, és L. Bieker, „Recent Development and Applications of SUMO – Simulation of Urban MObility”, International Journal On Advances in Systems and Measurements, 5(3&4): 128-138. o, 2012.
- [10] Lajos J. és Gergely N., „Az országos közutak 2017. évre vonatkozó keresztmetszeti forgalma. Az országos közúthálózat napi forgalma. Összesítő táblázatok (országos

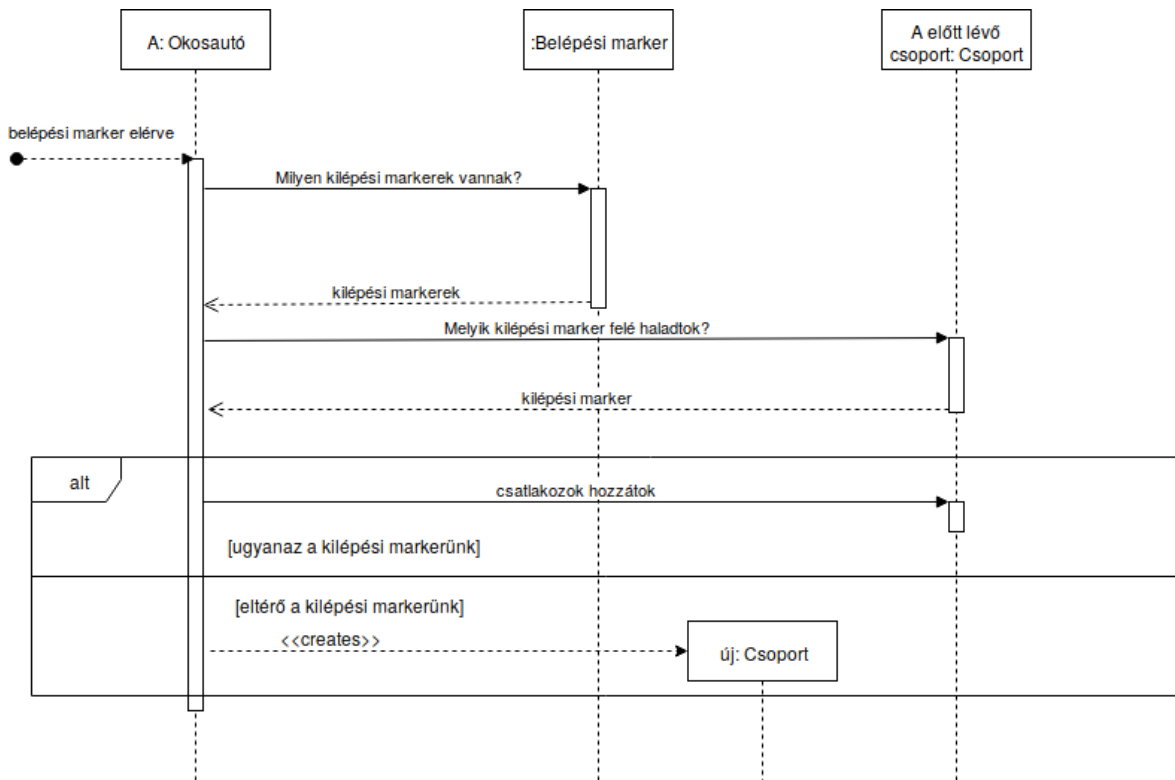
- és kezelőnkénti bontás)”, Magyar Közút Nonprofit Zártkörűen Működő Részvénytársaság, Budapest, 2018. július
- [11] Luspay Tamás, Tettamanti Tamás és Varga István, Forgalomirányítás – Közúti járműforgalom modellezése és irányítása. Budapest: Typotex Elektronikus Kiadó Kft., 2011.
- [12] S. Maiti, S. Winter, és L. Kulik, „A conceptualization of vehicle platoons and platoon operations”, Transportation Research Part C: Emerging Technologies, köt. 80, o. 1–19, júl. 2017.
- [13] X. Sun, R. Horowitz, és C.-W. Tan, „An Efficient Lane Change Maneuver for Platoons of Vehicles in an Automated Highway System”, in Dynamic Systems and Control, Volumes 1 and 2, Washington, DC, USA, 2003, köt. 2003, o. 355–362.
- [14] N. Webster és L. Elefteriadou, „A simulation study of truck passenger car equivalents (PCE) on basic freeway sections”, Transportation Research Part B: Methodological, köt. 33, sz. 5, o. 323–336, 1999.
- [15] Szenáriók D.-K., „1/3 kötet: A magyar autópálya korridorok forgalmi előrebecslései és díjbevételi prognózisai 2008.-2020.”, o. 170., BAUCONSULT Mérnökiroda Kft. Győr, 2006. május 31.
- [16] „AZ ORSZÁGOS KÖZUTAK 1995. ÉVRE VONATKOZÓ KERESZTMETSZETI FORGALMA”, Közlekedéstudományi Intézet Rt és Állami Közúti Műszaki és Információs KHT, Budapest, 1996
- [17] A Budapest Közút Zrt. által rendelkezésünkre bocsátott anyagok, 2018 tavasza

A. Pontos protokolleírások

A.1. Csoportosítás

A.1.1. Csoportba belépés

A csoportba történő belépés protokollját az A.1. szekvenciadiagram mutatja.



A.1. ábra. A csoporthoz történő csatlakozás szekvenciadiagramja

A protokollhoz köthető üzenetek a következők:

Milyen kilépési markereid vannak?

Cél: Információkérés a belépési markerből elérhető kilépési markerekről.

Feladó: Okosautó, aki belépési markerre ért.

Címzett: A belépési marker.

Válasz: Egy lista a kilépési markerekről.

Melyik kilépési marker felé haladtok?

Cél: Információkérés az okosautónk előtt, vele egy sávban haladó csoport haladási irányáról.

Feladó: Okosautó, aki belépési markerre ért.

Címzett: Az okosautó előtt haladó csoport (utolsó tagja).

Válasz: Egy konkrét kilépési marker.

Hatás: Az okosautónk el tudja dönteni, hogy számára megfelel-e a csoport.¹⁹

Csatlakozók hozzátok

Cél: Csoporthoz csatlakozás bejelentése

Feladó: Okosautó, aki belépési markerre ért.

Címzett: Az okosautó előtt haladó csoport (utolsó tagja).

Tartalom: Ki csatlakozik?

Hatás: A csoport regisztrálja, hogy új jármű érkezett bele. Ennek megfelelően módosítja a saját állapotát.

A.1.2. Csoportból kilépés

A csoportból való kilépés mindössze egyetlen üzenetváltásból áll, melyet a kilépési markerre lépés indít el. Ennél a szekvenciadiagramon történő ábrázolástól eltekintek.

Kilépek

Cél: Csoportból való kilépés bejelentése

Feladó: A csoportvezető

Címzett: A csoport (minden tagja, egyesével)

Hatás: A csoport regisztrálja, hogy elhagyta csoportvezetője, és hogy az új csoportvezetője az lesz, aki immár a csoport élén halad. Ezt az okosautót frissíti is, hogy tudjon csoportvezetői funkciójáról.

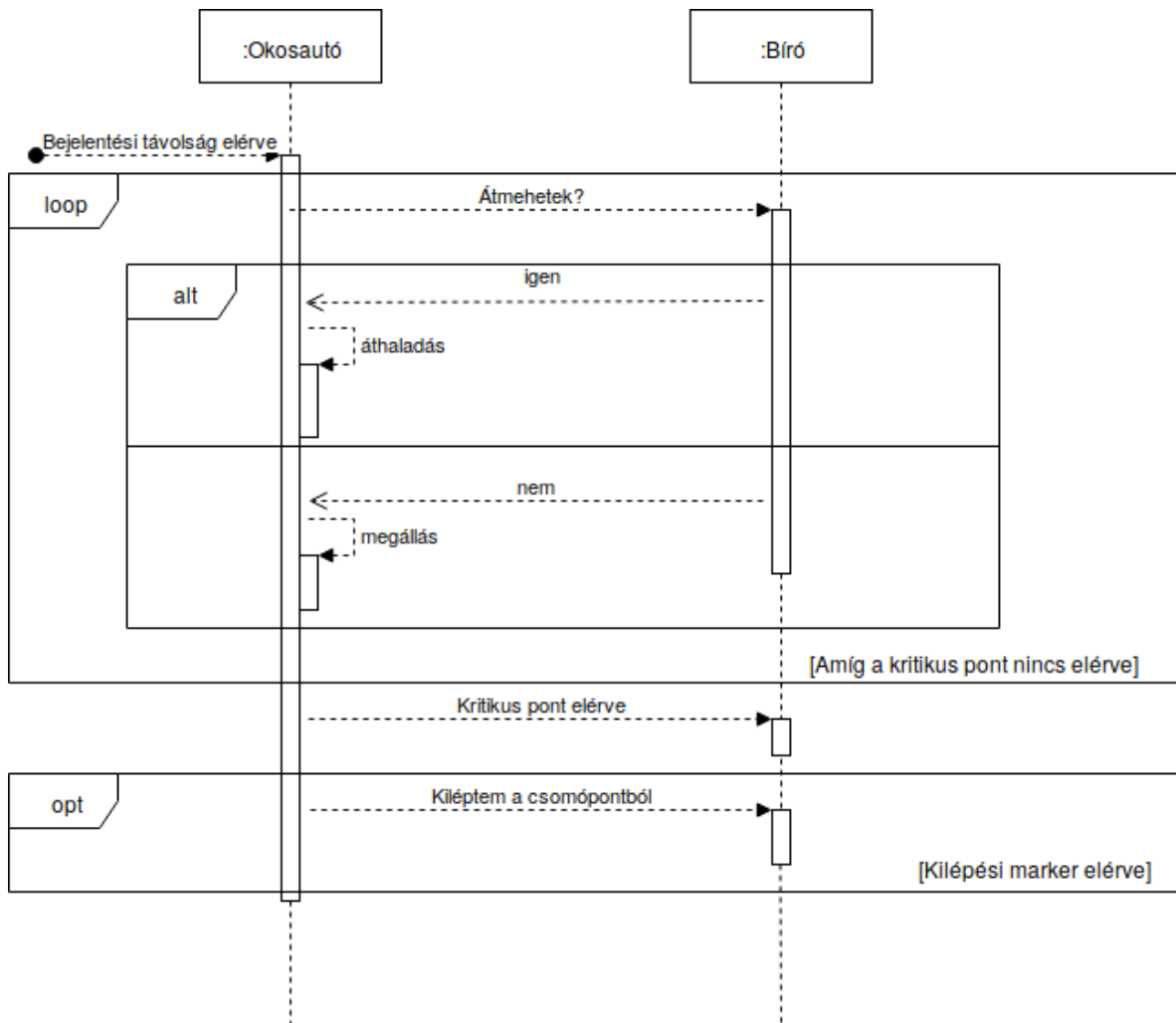
¹⁹Elvi lehetőség van egy csatlakozási szándék elutasítására is. Ilyenkor például olyan kilépési markert kell válaszként adni, ami egészen biztosan nem egy létező marker.

A.2. Bíróval kapcsolatos üzenetváltás

A.2.1. Konfliktusosztályba belépés

A.2.2. A csomóponti áthaladás protokollja

A csomóponti áthaladás protokollja az A.2. szekvenciadiagramon olvasható.



A.2. ábra. A csomóponti áthaladás szekvenciadiagramja

A protokollhoz kapcsolódó üzenetek pedig a következők:

Átmehetek

Cél: Információkérés arról, hogy egy okosautó éppen átmehet-e a csomóponton.

Feladó: Okosautó, aki a bejelentési távolságon belül van.

Címzett: A csomópont bírója.

Válasz: A bíró döntése arról, hogy az adott jármű áthaladhat-e.

Kritikus pont elérve

Cél: Értesítés arról, hogy egy okosautó elhagyta a kritikus pontot, azaz belépett a csomópontba.

Feladó: Okosautó, aki átment a kritikus ponton

Címzett: A csomópont bírója.

Hatás: A bíró jegyzi, hogy egy újabb autó beléérkezett.

Kiléptem a csomópontból

Cél: Értesítés arról, hogy egy okosautó elhagyta a csomópontot.

Feladó: Okosautó, aki elérte a kilépési markert.

Címzett: A csomópont bírója.

Hatás: A bíró jegyzi, hogy egy újabb autó elhagyta a csomópontját.

B. Algoritmusleírások

B.1. P-szabályozók

A korábban vázolt módszerek két ponton igényelnek valamilyen szabályozó algoritmust. Az egyik ilyen szabályozás a csoport tagjai közötti távolságtartásért felel. A másik pedig azt a célt szolgálja, hogy a járművek a megállási távolságnál valóban megálljanak.

Szabályozó algoritmusként egy P-szabályozót valósítottam meg. A P-szabályozó alatt a jól ismert PID-szabályozó integráló résztől és differenciáló tagtól mentes verzióját értem. Azért esett választásom erre, mert ez a legkevésbé számításigényes. A szimuláció mintavételezése 0,1 s-os, így a rendszer tehát viszonylag érzéketlen a munkapont körüli oszcillációra.

B.1.1. Adaptív tempomat csoporttagok között

Amennyiben egy *okosautó csoporttagi* minőségben tagja egy csoportnak, akkor megpróbálja a sebességét (legális és biztonságos keretek között) úgy megválasztani, hogy az autónk az ő előtte haladó csoporttaghoz vagy a csoportvezetőhöz képest állandó távolságot (pl. 12 m-t) tartson, akár csak egy adaptív (távolságtartó) tempomat.

Ezt a működést próbálja biztosítani a B.1 formula²⁰ alapján működő P-szabályozó. Ez a szabályozó egy lineáris függvény mentén állítja a sebességet az előttünk haladó csoporttag vagy csoportvezető v_l sebességének függvényében, ha a csoport tagjainak Δs távolsága a Δs_d kívánt távolság Δs_l sugarú környezetébe esik, azaz a $[\Delta s_d - \Delta s_l; \Delta s_d + \Delta s_l]$ tartományon belül van. A szabályozó V_{Pout} kimenete tehát lineáris arányossági tényező és a kívánt távolságtól való eltérés $(\Delta s - \Delta s_l)$ szorzataként áll elő.

$$v_{Pout} = \frac{v_l}{\Delta s_d - \Delta s_l} (\Delta s - \Delta s_l) \quad (\text{B.1})$$

Ez még az adaptív tempomatnak megfelelő működést nem tudja produkálni, ugyanis ettől elvárjuk azt, hogy egy beállított (jelen esetben $25 \frac{m}{s}$ -os) sebességet tartson, ha nincs előttünk bizonyos távolságban (15 m) másik tagja a csoportnak. Ha viszont a távolságtartási zónán (a tapasztalat azt mutatja, hogy ezt 9 m-re érdemes választani) belülre esnénk, akkor pedig vészfékezéssel üzemre kell váltanunk. Így a teljes

²⁰A képletekben szereplő fizikai mennyiségek SI-mértékegységek szerint értendők.

adaptív tempomat szabályozása a v_d kívánt sebességre a B.2 képlet szerint alakul.

$$v_d = \begin{cases} 25, & \text{ha } \Delta s \geq 15 \\ v_{Pout}, & \text{ha } \Delta s \in [9; 15[\\ 0, & \text{ha } \Delta s < 9 \end{cases} \quad (\text{B.2})$$

B.1.2. Megállás bírónál

A csomópontokat megközelítve az okosautóknak a megállási távolság által kijelölt vonalnál (körívnél) meg kell állniuk, ha a bírótól nem kaptak szabadjelzést a csomóponton történő áthaladásra. Ez a megállás is szabályozottan történik a B.3 képlet szerint.

$$v_d = \frac{v_l}{r_R}(x - r_S) \quad (\text{B.3})$$

Ez a szabályozó a kívánt v_d sebességet úgy állítja be, hogy az egyenletesen csökkenjen az r_R bejelentési távolság és az r_S megállási távolság között, az annak függvényében, hogy az okosautó és a csomópont középpontjának x távolsága mennyivel nagyobb még, mint az r_S megállási távolság (ez az eltérés). Az így kapott értéket az adott útszakaszra megengedett legnagyobb v_l sebességgel skálázzuk.

B.2. Preemptív RR ütemező

A preemptív RR ütemező egy körforgó ütemező (*Round Robin*), amely gyakorlatilag egy olyan rendőrlámpának felel meg, aki az aktuális forgalom alapján működik: ha valamerről nem érkezik autó, akkor annak az iránynak nem ad – feleslegesen – zöld jelzést.

A konfliktusosztályoknak van egy előre definiált áthaladási sorrendje, és van hozzájuk rendelve egy maximális időintervallum, amelynél több ideig egy körben nem kapnak szabad jelzést.

Ezen kívül a preemptív RR ütemező rendelkezik még azzal a képességgel, hogy egy újabb konfliktusosztályt csak akkor tesz aktívvá, és ad neki szabad jelzést, ha már a korábbi konfliktusosztály áthaladt a kereszteződésen.

Az preemptív RR ütemező pszeudó kóddal tehát az 1. algoritmus szerint írható le.

Algorithm 1 Preemptive RR scheduler

```
1: procedure UPDATE_ACTIVE_CONFLICTCLASS
2:   if (activeConflictClass.duration > maxDuration or
        timeOf(lastCar.arrived) > 3 s) or
        activeConflictClass =  $\emptyset$  then
3:     Select_new_RR()
4: procedure SELECT_NEW_RR
5:   if yellowPhase and timeOf(candidateConflictClass.since) > 3 then
6:     activeConflictClass  $\leftarrow$  candidateConflictClass.
7:     yellowPhase  $\leftarrow$  false.
8:     return ;
9:
10:  candidateConflictClass  $\leftarrow$  activeConflictClass.
11: loop:
12:  candidateConflictClass  $\leftarrow$  next from ConflictClasses.
13:  if candidateConflictClass =  $\emptyset$  and
        candidateConflictClass.isNotBlockedByOthers and
        candidateConflictClass  $\neq$  activeConflictClass then
14:    goto loop.
15:  if not junction.isEmpty then
16:    yellowPhase  $\leftarrow$  true.
17:    candidateConflictClass.since  $\leftarrow$  now.
18:  else
19:    activeConflictClass  $\leftarrow$  candidateConflictClass.
20:    yellowPhase  $\leftarrow$  false.
```

B.3. MDDF ütemező

Az MDDF ütemezőmegvalósításom egy kétprioritásos ütemező. A magasabb prioritási szinten gyakorlatilag egy körforgó ütemező (preemptív RR), az alacsonyabbikon pedig az [1] javasolt MDDF ütemező fut.

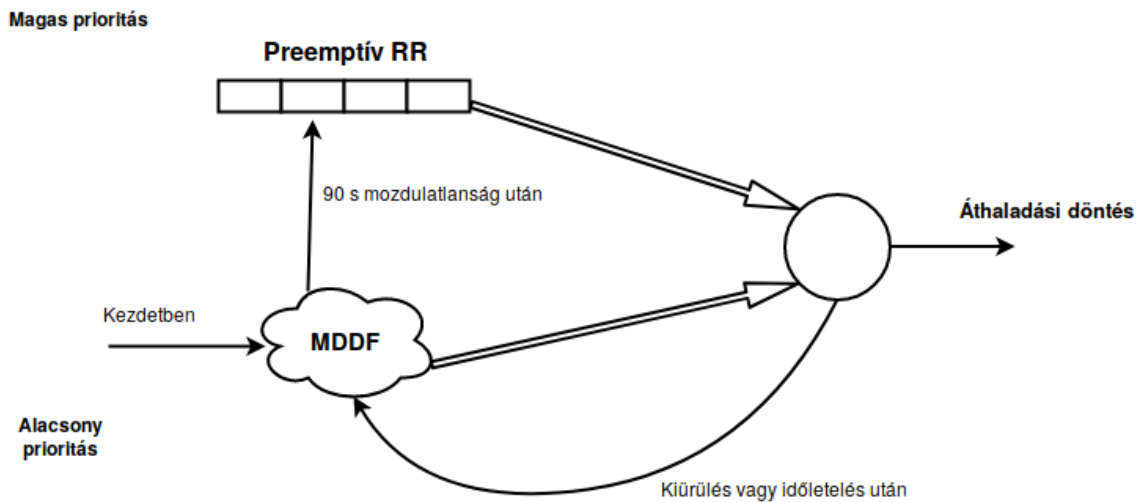
A konfliktusosztályok alapvetően az alsó prioritási szinten futnak. Amennyiben egy konfliktusosztályban van várakozó jármű, és az adott konfliktusosztály már több, mint 90 s óta nem került ütemezésre, átkerül a magasabb prioritási szintre. Ha valaki futásra kerül a magasabb prioritási szintről, az visszakerül az alacsonyabb, MDDF szintre (lásd a B.1. ábrán).

Ez az algoritmus fair, ugyanis garantált, hogy egy jármű másodpercben mért t_{wmax} várakozási ideje legfeljebb:

$$t_{wmax} = 90 + \sum_{i \in CC} d_{maxi} + t_Y$$

A képletben CC jelentése a konfliktusosztályok halmaza, d_{maxi} az i . konfliktusosztály számára fenntartott maximális áthaladási idő egy ciklusra. A t_Y pedig a sárgafázis hosszát jelöli. Ez a fázishossz egy 3 másodperces késleltetésnek és az előző konfliktusosztályt alkotó járművek csomópontelhagyási idejének összegeként adódik.

Az algoritmus pszeudokódját a 2. algoritmus mutatja, melynek 15. sorában szereplő „ScheduleByRR” függvénye a már leírt preemptív RR ütemező által adott ütemezési döntést szolgáltatja a paramétereként kapott konfliktusosztályhalmazon.



B.1. ábra. Az MDDF ütemező felépítése

Algorithm 2 MDDF scheduler

```
1: procedure UPDATE_ACTIVE_CONFLICTCLASS
2:   if (activeConflictClass.duration > maxDuration or
        timeOf(lastCar.arrived) > 3 s) or
        activeConflictClass =  $\emptyset$  then
3:     Select_new_RR()
4:
5: function SELECT_MINIMAL_CC
6:   minCC  $\leftarrow$  ConflictClasses[0].
7:   minPrice  $\leftarrow$  0.
8:   for each X  $\in$  ConflictClasses do
9:     minCar  $\leftarrow$  X.cars[0].
10:    for each C  $\in$  X.cars do
11:      if C.distanceToDestination < minCar.distanceToDestination then
12:        minCar  $\leftarrow$  C.
13:      if minCar.distanceToDestination < minPrice then
14:        minCC  $\leftarrow$  X.
15:        minPrice  $\leftarrow$  minCar.distanceToDestination.
16:    return minCC
17:
18: procedure SELECT_NEW_RR
19:   if yellowPhase and timeOf(candidateConflictClass.since) > 3 then
20:     activeConflictClass  $\leftarrow$  candidateConflictClass.
21:     yellowPhase  $\leftarrow$  false.
22:     return
23:
24:   for each X  $\in$  ConflictClasses do
25:     if X  $\neq$   $\emptyset$  and X.lastTime > 90 s then
26:       add X to RR_Candidates.
27:
28:   if RR_Candidates  $\neq$   $\emptyset$  then
29:     candidateConflictClass  $\leftarrow$  ScheduleByRR(RR_Candidates).
30:   else
31:     candidateConflictClass  $\leftarrow$  Select_Minimal_CC(ConflictClasses)
32:   if not junction.isEmpty then
33:     yellowPhase  $\leftarrow$  true.
34:     candidateConflictClass.since  $\leftarrow$  now.
35:   else
36:     activeConflictClass  $\leftarrow$  candidateConflictClass.
37:     yellowPhase  $\leftarrow$  false.
```
