



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és Információs Rendszerek Tanszék

Intelligens városi közlekedési rendszerek

Explicit torlódásértesítésen alapuló
jelzéskoordináció

TUDOMÁNYOS DIÁKKÖRI KONFERENCIA 2019. –
DOLGOZAT

Készítette
Alekszejenkó Levente

Konzulens
Dr. Dobrowiecki Tadeusz Pawel
(Méréstechnika és Információs Rendszerek Tanszék)

2019. október 28.

A dolgozatban ismertetett eredmények a Budapesti Műszaki és Gazdaságtudományi Egyetem Villamosmérnöki és Informatikai Kar Balatonfüredi Hallgatói Kutatócsoport szakmai közössége keretében jöttek létre a régió gazdasági fejlődésének elősegítése érdekében. Az eredmények létrehozása során figyelembe vettük a balatonfüredi központú Rendszertudományi Innovációs Klaszter által megfogalmazott célkitűzéseket, valamint a párhuzamosan megvalósuló EFOP 4.2.1-16-2017-00021 pályázat támogatásával elnyert „BME Balatonfüredi Tudáscentrum” térségfejlesztési terveit.

A kutatás az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg (EFOP-3.6.2-16-2017-00013, Innovatív Informatikai és Infokommunikációs Megoldásokat Megalapozó Tematikus Kutatási Együtműködések).

Tartalomjegyzék

1. Bevezetés	1
1.1. Motiváció	2
1.1.1. Előzmények	2
1.1.2. További kutatások motivációja	2
1.2. A dolgozat részletes célja és felépítése	3
1.3. Irodalmi áttekintés	4
2. Feltételezések	7
3. Megvalósítás	8
3.1. Áttekintés	8
3.2. Együtt vezérelhető kereszteződések	8
3.2.1. A csomópont-konglomerátumok	9
3.2.2. Algoritmus csomópont-konglomerátum keresésére	11
3.2.3. Példa az algoritmus futtatására	12
3.2.4. Konfliktusforrások csomópont-konglomerátumokban	13
3.3. Jelzésterv generálása	13
3.3.1. A pillanatnyi jelzések előállítása	13
3.3.2. Az időzítés	15
3.4. Az ECN-bíró	15
3.4.1. Az explicit torlódásértesítés	16
3.4.2. A torlódás valószínűsége	17
3.4.3. Bírók közti üzenetváltás	18
3.4.4. Torlódásjelzés figyelembe vétele jelzésterv-generálásnál	18
3.5. Kommunikációs protokollok	18
3.5.1. Kommunikáció a bírók között	18
3.5.2. Kommunikáció az okosautók és a bírók között	19
4. Mérési eredmények és összehasonlításuk	21
4.1. Mérési környezet	21
4.2. Mérési eredmények	21
4.3. A kapott eredmények értékelése	23

5. Összefoglalás	26
5.1. Irodalomjegyzék	27
A. A szimulációs keretrendszer	30
B. A BAH-csomópont konglomerátumának meghatározása	32
C. A jelzésterv-generáláshoz használt egészértékű programozási feladat	35

1. fejezet

Bevezetés

Az elmúlt évtizedekben tapasztalható közúti városi forgalomnövekedés új kihívások elé állította a közlekedésmérnöki szakmát. Mivel a közutak szélesítése igen költséges és semmiképpen sem tűnik egy fenntartható fejlesztési módszernek, ezért máshogy szükséges növelnünk a közútjaink kapacitását.

Ilyen kapacitásnövelő módszer az egyes jelzőlámpával ellátott kereszteződések közötti jelzéskoordináció, azaz a forgalomirányító berendezések jelzéseinek valamilyen módon történő összehangolása annak érdekében, hogy a járműcsoportok minél rövidebb idő alatt elhagyhassák a koordinációban részt vevő csomópontokat. Ennek évszázados módja [15] a *zöldhullámok* kialakítása. A zöldhullámokat általában egy kitüntetett, *arteriális irány* mentén lévő jelzőberendezések időzítésének összehangolásával érik el. Azonban, ha a főirány áteresztőképessége valamilyen okból (útfelújítás, baleset stb.) lecsökken, a mellékirányban lévő kerülőutak megközelítése is nehézkessé válik, ugyanis ezen rendszerek többnyire nem képesek önállóan arra, hogy egy ideiglenes főirányt alakítsanak ki.

Napjainkban ugyancsak elterjedt megoldás az *aktuált jelzőlámpák* telepítése is [10]. Ezek detektorok segítségével érzékelik azt, hogy éppen milyen forgalmi igények lépnek fel az úthálózaton, és ennek megfelelően változtatják az aktuális jelzéseiket. Ilyen csomópontok hálózatában úgy tűnik, hogy nincsen mód jelzéskoordináció kialakítására, azonban ne felejtjük el, hogy itt a járművek implicite képesek kialakítani „zöldhullámokat” csupán azáltal, hogy egymás után megjelennek a kereszteződések detektorainál.

Azonban egy aktuált kereszteződés bizonyos helyzetekben meglehetősen unfair tud lenni: Előfordulhat olyan helyzet, hogy egy mellékirányból érkező járműnek nagyon sokáig várakoznia kell ahhoz egy kereszteződésben, hogy szabadjelzést kapjon. Sőt, ennek az ellentétes helyzete is előfordulhat, méghozzá az, hogy egy mellékirányból érkező nagyobb járműcsoport tartja fel a főirányból érkezőket.

Az önvezető járművek és az IoT-eszközök terjedésével lehetőségünk nyílik új megoldások létrehozására is. Dolgozatom célja egy jármű–infrastruktúra (V2I) és az infrastruktúra elemei közötti kommunikáción alapuló új jelzéskoordinációs módszer megvalósítása, illetve szimulációval történő ellenőrzése. A megvalósítandó jelzésko-

ordinációs módszertől a következő tulajdonságokat várjuk el:

- nem igényli artériák definícióját,
- valósídejű, a pillanatnyi forgalomnak megfelelően viselkedik,
- várakozási időt tekintve fair,
- jól skálázódik a kereszteződések számával.

1.1. Motiváció

1.1.1. Előzmények

Bár úgy tűnik, hogy az aktuált (vagy az ezekkel gyakorlatilag megegyező, intelligens) forgalomirányító berendezések eleve képesek megoldani ezt a problémát [25], a szimulációs eredményeim nem feltétlenül ezt mutatják.

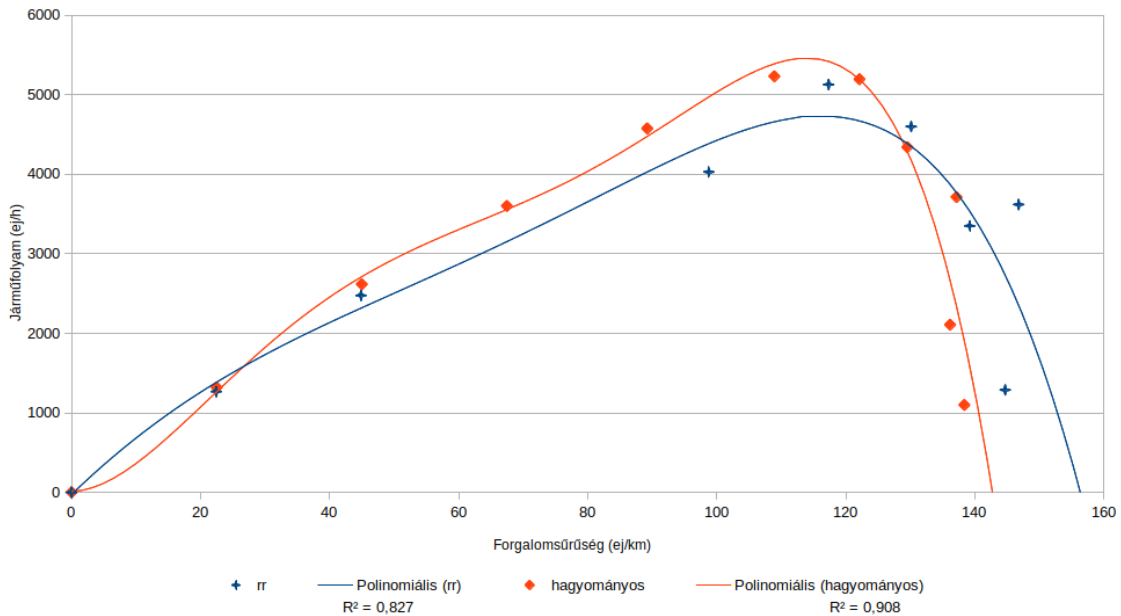
A SUMO mikroszkopikus forgalomszimulátoron alapuló intelligens keretrendszer segítségével (lásd az A. függelék) egy hétköznapi déli forgalomigény többszöröseire emelt nagyságával meghatároztam a forgalmat jellemző, ún. *Macroscopic Fundamental Diagram*-ot, MFD-t [26], [23], amely a forgalomsűrűség függvényében mutatja a forgalom nagyságát. Összehasonlítottam a hagyományos, fix időzítésen alapuló lámpaprogramot egy körforgó, round-robin alapú¹ intelligens megoldással [2], [3], lásd az 1.1. ábrán. Az eredményen látszik, hogy a zöldhullámot kialakító hagyományos megoldás még nagy járműsűrűség mellett is képes felülmúlni az intelligens, ámde nem koordinált megoldást. Ugyanakkor az intelligens rendszer valamennyivel hatékonyabban tud működni a diagramon a csökkenő, azaz a rendszer vergődési fázisában [22], vélhetően azért, mert itt a hagyományos rendszerbeli főirány torlódása miatt a mellékirányok még kicsivel jobban járhatóak maradnak, hiszen itt – mellékirányok lévén – eleve nem volt erősen optimalizálva a rendszer.

1.1.2. További kutatások motivációja

Nagyobb forgalomsűrűség esetén feltehetőleg azért jut előnybe a hagyományos rendszer, mert az időzítései, a zöldhullámok segítségével közelítik a globálisan is optimális megoldást, míg az intelligens rendszer bírói csak lokális optimalizációra képesek. A lokálisan optimális megoldások összessége pedig ritkán egyezik meg a globálisan optimális megoldással.

Ahhoz tehát, hogy egy intelligens forgalomirányító rendszer által nyújtott teljesítmény ne maradjon el nagy forgalomsűrűség esetén sem (de még nem a vergődés

¹Az [1], [2], [3] MDDF-alapú megoldása már a korai időszakban átváltott volna RR működésre, így az MDDF által nyújtott eredmények gyakorlatilag megegyeznének a nála jóval egyszerűbb RR-alapú rendszer eredményeivel.



1.1. ábra. Intelligens és hagyományos rendszer összehasonlítása jelzéskoordináció nélkül. A kapott konkrét mérési adatpontok alapján egy jól illeszkedő negyedfokú polinomot, mint trendvonalat vettem fel mindkét adatsorhoz. A regressziós függvény súlyozása egyenletes, ugyanis a hibára vonatkozó a priori ismeretünk nicsen.

állapotában) a hagyományos rendszer teljesítményétől, elengedhetetlennek tűnik az intelligens jelzőrendszerek, *bírók* közötti jelzéskoordináció. Ez a koordináció elméletileg azt a célt szolgálná, hogy a megoldásunk minél jobban közelítse a globálisan optimális megoldást, azaz a közlekedési hálózat áteresztőképességét maximalizálná.

Ilyen rendszert már többen is megalkottak (lásd az 1.3. fejezetet), viszont ezek a rendszerek erősen támaszkodnak az előre definiált főirányokra, artériákra. Véleményem szerint egy intelligens megoldás fő előnye ugyanakkor az, hogy képes dinamikusan reagálni a forgalmi helyzet megváltozására, így például arra is, ha valami miatt egy artéria átjárhatatlanná válik. Éppen ezért célként tűzöm ki azt is, hogy a dolgozatban javasolt rendszer ne igényelje artériális irányok előzetese definícióját.

1.2. A dolgozat részletes célja és felépítése

Jelen dolgozat fő célja így egy artériafogalomtól mentes jelzéskoordinációs módszer megalkotása, amely segíthet növelni a hálózat áteresztőképességét nagy forgalom esetén is [6]. Ez egyben azt is jelenti, hogy nem csupán lokális optimalizációs problémákat kell megoldanunk, de törekednünk kell arra is, hogy a probléma megoldása minél jobban közelítse a globális probléma megoldását is.

Ahhoz, hogy egyáltalán megállapítsuk, hogy mit értünk globálisan a forgalom-

áteresztési képességen (azaz mely kereszteződéseket kell együtt optimalizálnunk), szükségünk van egy módszerre, amely el tudja dönteni, hogy mely csomópontok alkotnak valamiféle szorosan kapcsolt csoportot, *csomópont-konglomerátumot* egymással (azaz, mely csomópontok irányítása van közvetlen kihatással más csomópontokra is). Fontos megjegyeznünk, hogy egy ilyen csoportosulást nem a földrajzi elhelyezkedés határoz meg, hanem a csomóponthálózat topológiája, így ezeket a kereszteződéseket megtalálni nem triviális feladat. Erre egy algoritmikus megoldást fogunk a 3.2. fejezetben bemutatni.

Ezután persze el kell érünk azt is, hogy a csomópontkonglomerátum tagjai, a *szorosan kapcsolt kereszteződések* együtt tudjanak dolgozni. Ehhez három dolgot is meg kell terveznünk, amiből a legalapvetőbb igény az, hogy ezek a csomópontok képesek legyenek valós időben jelzéstervet generálni, részletekért lásd a 3.3. fejezetet.

Ez a jelzésterv-generáló algoritmus képes lesz figyelembe venni egyfelől a lokális forgalmi helyzetet, másfelől a csomópont-konglomerátumra jellemző globális helyzetet is. Viszont ahhoz, hogy ezt a helyzetet jellemezni tudjuk, illetve reagálni tudjunk rá, szükségünk van egy újabb algoritmusra. Egy ilyen algoritmus például a számítógépes hálózatoknál jól ismert explicit torlódásértesítés (*Explicit Congestion Notification, ECN*) eljárás [9]. Az ECN eljárást az internetes világban többek között azért vezették be [14], hogy a torlódás kialakulása előtt még egy stabil állapotban tarthassák a hálózatot, ráadásul úgy, hogy ezt a rendszer csomópontjai biztosítják. Előnye, hogy alkalmazásával egy alacsonyabb késleltetésű hálózatot is kialakít ez a módszer, és így gyakorlatilag egybevág² a mi célunkkal is. Az ECN-algoritmus forgalomirányításra használható megoldását mutatja a 3.4. fejezet.

1.3. Irodalmi áttekintés

Jelzéskoordináció ötlete nem új gondolat, számtalan eljárás ismert, amely képes valamilyen szempontból optimalizálni egy közlekedési hálózat jelzőberendezéseinek időzítését. A TRANSYT és a SCOOT néven ismert eljárások [20] is ilyenek, ezek optimalizációs célja, hogy a csomópontok előtt kialakuló sorok minél rövidebbek legyenek. A TRANSYT egy jelzéskoordinációs tervet készít (tervezési időben) egy (akár több évvel korábban felvett) adatsor felhasználásával. A SCOOT az egyik első intelligensnek tekinthető forgalomirányító algoritmus. Az utolsó néhány perc adataival számol és ezeket hasonlítja a pillanatnyi helyzethez. Az összehasonlítás eredményeképpen kiszámolja, hogy érdemes-e pár másodperccel módosítani a jelenlegi időzítéseket valamelyik irányba avagy érdemesebb-e megtartani a jelenlegi időzítést. Azonban a SCOOT csak a lokális megoldások változtatásával halad a globális optimalizálás felé, a döntések itt lokálisak. A jelzések koordinálását maga a járműfolyam végzi. A SCOOT nagy hátránya, hogy költséges, ugyanis sok-sok

²Az alacsonyabb késleltetés rövidebb átfutási időt eredményez, tehát magasabb átlagsebességet is. Ha viszont magasabb az átlagsebesség, akkor az a forgalmi folyamat is növeli, ami megegyezik azzal a célkitűzéssel, hogy növeljük meg a hálózat áteresztő képességét.

szenzor telepítését igényli, ráadásul a szenzorok 15%-ának kiesésével már nem okoz érzékelhető javulást.

A SCOOT megjelenése óta a kutatók rengeteg ötlettel álltak elő intelligens forgalomirányításra. Ezen eljárások alapulhatnak ütemezésméleten [2], vagy a SCOOT-hoz hasonlóan a forgalmi helyzet vizsgálatán is [4]. Érdekes, hogy az ERIS [12] módszer pedig képes figyelembe venni a szomszédos csomópontok kibocsátási értékeit is, ám ezt csak a beérkező járművek csoportosításánál használja, magához az áthaladási döntéshez már nem.

A [8] által felvázolt módszer pedig gyakorlatilag szakít a hagyományos közlekedéstervezési módszerekkel. A javasolt algoritmusuk lényege, hogy minden jármű próbáljon meg magának lefoglalni egy időpontot az áthaladásra, éppen úgy, mint ahogyan szállodai szobát foglalunk magunknak. Ezt a foglalási igényt a csomóponti vezérlő elfogadhatja vagy visszautasíthatja (ilyenkor egy másik javaslatot küldhet a kezdeményező járműnek). Elméleti szempontból ez a módszer akár alkalmazható lenne a dolgozatom céljaként kitűzött feladat megoldására is: egy csomóponti vezérlő megkérdezheti a szomszédaitól, hogy ők hogy tudják majd kiszolgálni az ebből a csomópontból kifutó járműveket. Ha valamiért egy ilyen kiszolgálás akadályozva lenne, akkor ez a csomópont is visszautasíthatja az eredeti foglalási igényt. Egy ilyen rendszer legfőbb hátránya a hatalmas állapottere: egy csomópont-rendszerben akár néhány ezer autó is jelen lehet, ezek összes foglalását és a foglalások függőségeit nyomonkövetni túlságosan számítás-, illetve tárigényes feladat lenne.

Kevésbé erőforrásigényes, intelligens jelzskoordinációt megvalósító eljárásokból is igen sokat találhatunk. A [19] által leírt módszer végső soron azon alapszik, hogy dinamikusan módosítja a kereszteződések jelzésidőzítését (hasonlóan a SCOOT-hoz, csak itt több csomóponté egyszerre). Azonban figyeljük meg, hogy ez a módszer impliciten artériákra optimalizál, ugyanis a kiindulási időzítés nagy valószínűséggel egy artériákra optimalizált időzítés volt. Továbbá ez a módszer nem vagy csak nagyon lassan képes arra, hogy egy tetszőleges újabb artériát alakítson ki szükség esetén.

A dolgozatomban vázolt problémához nagyon hasonló feladatot old meg az [5]. A fő különbség az, hogy ez a munka, szemben a dolgozattal, az artériák menti optimalizálást tűzte ki célul, ezért a csatolt kereszteződéseket is másként definiálja, mint ahogyan az a dolgozatomban a 3.2. fejezetében olvasható: azaz itt a csatolt kereszteződéseknek az egymás melletti kereszteződéseket tekintik.

A [11] egy olyan megoldást mutat be, amely képes arra, hogy nem csak időben, hanem térben is elossa a járműveket egy már túltelített forgalmi helyzetben. Ehhez persze kihasználja azt, hogy egyszerre több lehetséges útvonal is létezik két pont között. Ez a feltételezés praktikusán igaz egy amerikai stílusú, négyzetácsos úthálózatra, azonban egy szabálytalan európaira nem feltétlenül (részletekért lásd még a 3.2. fejezetet).

Egy ágens alapú megközelítéssel találkozhatunk a [27] munkában. Zöldidőt számító módszerükben figyelembe veszik az összes szomszédos csomópont állapotát is.

Ez a megoldás magában hordozza az instabilitás és a pillangó-effektusként ismert jelenség veszélyét is: könnyen előfordulhat egy olyan állapot, hogy valamely csomópont időzítésének kicsiny megváltoztatása azt okozza, hogy a forgalmi helyzet közelíteni kezd a torlódási állapothoz. Erre a változásra reagálnak a kiindulási csomópont szomszédjai is, és itt is megindul a torlódási folyamat, amire pedig ezen csomópontok fognak reagálni, és így tovább. . . A folyamat végén előfordulhat, hogy az összes csomópont forgalma nullára fog csökkenni, egy aprócska módosítás miatt. Másfelől egy járműcsoport mozgása a megváltozó forgalomsűrűség (és a szomszédok erre való érzékenysége) miatt a párhuzamos útszakaszokon is kiválthat időzítésmódosításokat, ezzel pedig instabillá teheti a rendszert.

A probléma akár közgazdasági fogalmakkal is meghatározható, ahogyan azt a [7] mutatja. Sajnos az ebben a munkában vázolt megoldások vagy nem nyújtanak igazi online működést vagy nem jól skálázódnak az autók számával amiatt, hogy az ütemezést mindig újra- és újraszámolják, valahányszor egy újabb jármű érkezik. Így ez igen kommunikáció- és erőforrásigényes eljárássá tud válni, ha sok jármű van a rendszerben.

2. fejezet

Feltételezések

Mivel a dolgozatomban javasolt rendszer konkrét megvalósítása igen bonyolult, sőt részeiben még ma is kutatás tárgyát képezi [13], ezért néhány egyszerűsítő feltétellel élek.

Ilyen feltétel, hogy a forgalomban résztvevő járművek, *okosautók* azonos méretűek és azonos menettulajdonságokkal rendelkeznek, valamint teljes szintű önvezetésre (SAE-5-ös szint [21]) képesek. Továbbá tudnak egymással és az infrastruktúra bizonyos elemeivel, így például az intelligens forgalomirányító berendezésekkel, a *bírókkal* kommunikálni. Megértik a dolgozatban definiált protokollokat, és képesek a kapott üzeneteknek megfelelően viselkedni. Ezen kívül intelligens ágensként szabálykövetők, igazmondók és jóhiszeműek.

A bírók ugyancsak szabálykövetők, igazmondók és jóhiszeműek, az okosautókkal és akár egymással is képesek kommunikációra.

Ezek a kommunikációk egy ideális csatornán zajlanak, csomagvesztés nélkül, így feltételezzük, hogy az átvitel ideje, késleltetése és egyéb paraméterei megfelelnek ahhoz, hogy az üzeneteinket időben eljuttassuk a kommunikációs partnerekhez.

A rendszerben egyéb zavaró hatások, így például emberi sofőrök, gyalogosok, rolleresek, kerékpárosok stb. nincsenek.

Az okosautók funkcionalitását tekintve alapvetőnek tekintjük, hogy a korábbi munkámban (melyről rövid összefoglaló olvasható az A. függelékben) már ismertett képességekkel rendelkeznek, így a kereszteződések előtt csoportokba rendeződnek aszerint, hogy milyen trajektórián fognak mozogni a kereszteződésben. Illetve az így kialakult csoportok segítik egymás sávváltását is: a hátrébb lévő csoport lelassít vagy megáll annak érdekében, hogy az előrébb lévő csoport besorolhasson elé.

3. fejezet

Megvalósítás

3.1. Áttekintés

Alapvető célom egy explicit torlódásértesítésen, (*Explicit Congestion Notification, ECN*) alapuló jelzéskoordináció megvalósítása, annak érdekében, hogy megvizsgálhassuk, mennyire sikeresen vihető át az internetes hálózatoknak ez a védelmi mechanizmusa az – azért nem teljesen az információs csomagokkal analóg mozgású – okos autóágensek forgalmi helyzeteire [9]. Ehhez meg fogjuk keresni azokat a kereszteződéseket, amelyeket érdemes bevonni egy ilyen koordinációba.

Mivel az együttműködés miatt az egyes kereszteződésekhez tartozó forgalmi jelzések (melyik iránynak mikor szabad vagy tilos áthaladnia a kereszteződésen) állapottere óriási méretűre duzzad, ezért ezt már tervezési időben nem tudjuk kezelni, így szükségünk lesz egy olyan bírótípusra, amely képes valós időben készíteni jelzésterveket (amelyek csomópont szintjén mutatják, hogy melyik irány mikor fog szabadjelzést kapni).

Végül pedig, ha a bírók képesek figyelembe venni egymás jelzéseit, akkor a számítógépes hálózatoknál is ismert ECN algoritmust adaptálni fogjuk, hogy alkalmas legyen közlekedési rendszerek jelzéskoordinációjára.

3.2. Együtt vezérelhető kereszteződések

A városok úthálózatát tanulmányozva gyakran találkozhatunk azzal, hogy több kereszteződés koncentrálódik egy relatíve kicsiny geográfiai területre. Ilyen például Budapesten a BAH-csomópont, Moszkvában a Tanganszakaja kereszteződés vagy a bécsi Praterstern, lásd a 3.1. ábrán. Hasonló területen koncentrálódik ugyanakkor több kereszteződés Philadelphiában is, lásd a 3.2. ábrán, ám egy viszonylag szabálytalan, európai közlekedési hálózathoz képest egy négyzetrácsos, amerikai úthálózat topológiája alapvetően más. Mint látni fogjuk, ez befolyással van arra, miként kell kiválogatnunk az együtt vezérelt kereszteződéseket. Figyeljük meg ugyanis, hogy a felsorolt európai nagy közlekedési csomópontokban vannak olyan alcsomópontok,

amelyek kiesése akár azt is okozhatja, hogy a csomópont valamelyik iránya elérhetetlenné válik¹! Ugyanakkor ilyen probléma egy négyzetrácsos hálózatban nem képzelhető el, mivel a hálózatban – pontosan a rácsháló miatt – bármely két pont között legalább két, csomópontokat tekintve független út vezet.

Az ilyen összetett forgalmi csomópontok általában (alcsomópontokként) több forgalomirányító berendezéssel, rendőrlámpával szabályozottak. Klasszikus forgalomirányítás esetén is igyekeznek ezeket a lámpákat összehangolni, így a lámpák (korábbi anyagaimban *bírók*) intelligens együttműködésével érdemes lehet foglalkozni.

Egy ilyen együttműködést indokol az is, hogy egy-egy útvonalon való végighaladás során kötelezően és szigorú sorrendben érintünk kereszteződéseket egymás után. Egy intelligens közlekedési infrastruktúramenedzsment-rendszer ezt a tudást kihasználhatja a forgalmi folyamatok növelésének érdekében.

Ha egy komplex forgalmi csomópontra, mint egy hierarchikus rendszerre tekintünk, akkor megfogalmazhatunk még egy elvárást egy ilyen rendszer forgalomirányítójára. Ez pedig nem más, minthogy a járművek töltsenek minél kevesebb időt az alcsomópontok közötti útszakaszokon. Gondoljunk bele, hogy ezen szakaszok is – a tradicionális kereszteződésekhez hasonlóan – konfliktuszónák és közös használatú erőforrások.

3.2.1. A csomópont-konglomerátumok

Nevezzük *csomópont-konglomerátumnak* egyszerű kereszteződések tovább nem bővíthető² halmazát a következőképpen: Bármely, a konglomerátumot érintő útvonal kötelezően tartalmazza a konglomerátumot alkotó egyszerű kereszteződések lehető legnagyobb (nem üres) részhalmazát. Egyetlen egyszerű kereszteződés is csomópont-konglomerátumnak tekinthető.

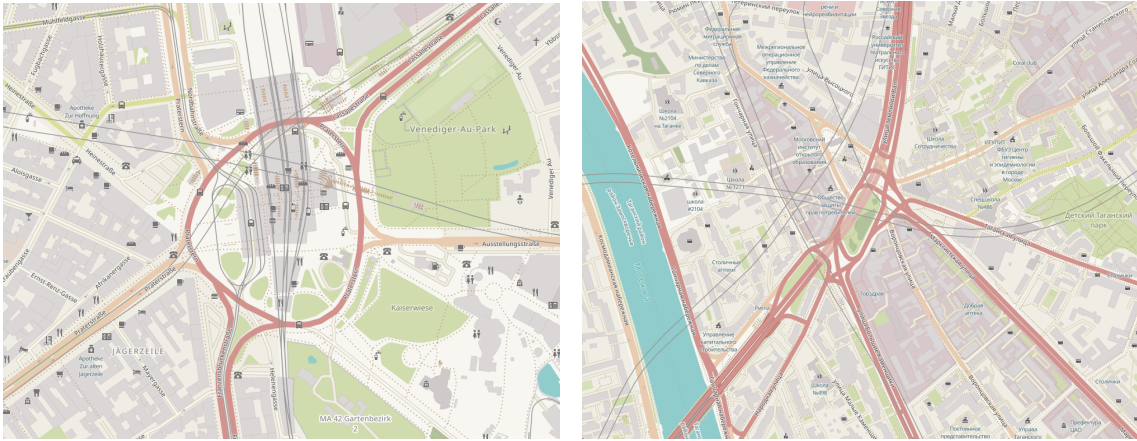
A konglomerátumot alkotó egyszerű kereszteződések az *alcsomópontok*. Válasszunk ki egy alcsomópontot a konglomerátumot alkotó halmazból. Ekkor azt mondjuk, hogy a kiválasztott csomópontnak a többi alcsomópont a *környezetében van*.

Ha az alcsomópontok összehangolt forgalomirányító berendezéssel vannak ellátva, akkor *menedzselt csomópont-konglomerátumról* beszélünk. Az alcsomópontok megoszthatnak egymással információt/parancsot annak érdekében, hogy a forgalmi folyamat növelni lehessen. Amennyiben ilyen megosztás történik egy konglomerátumban, abban az esetben *intelligens, menedzselt csomópont-konglomerátumról* beszélünk.

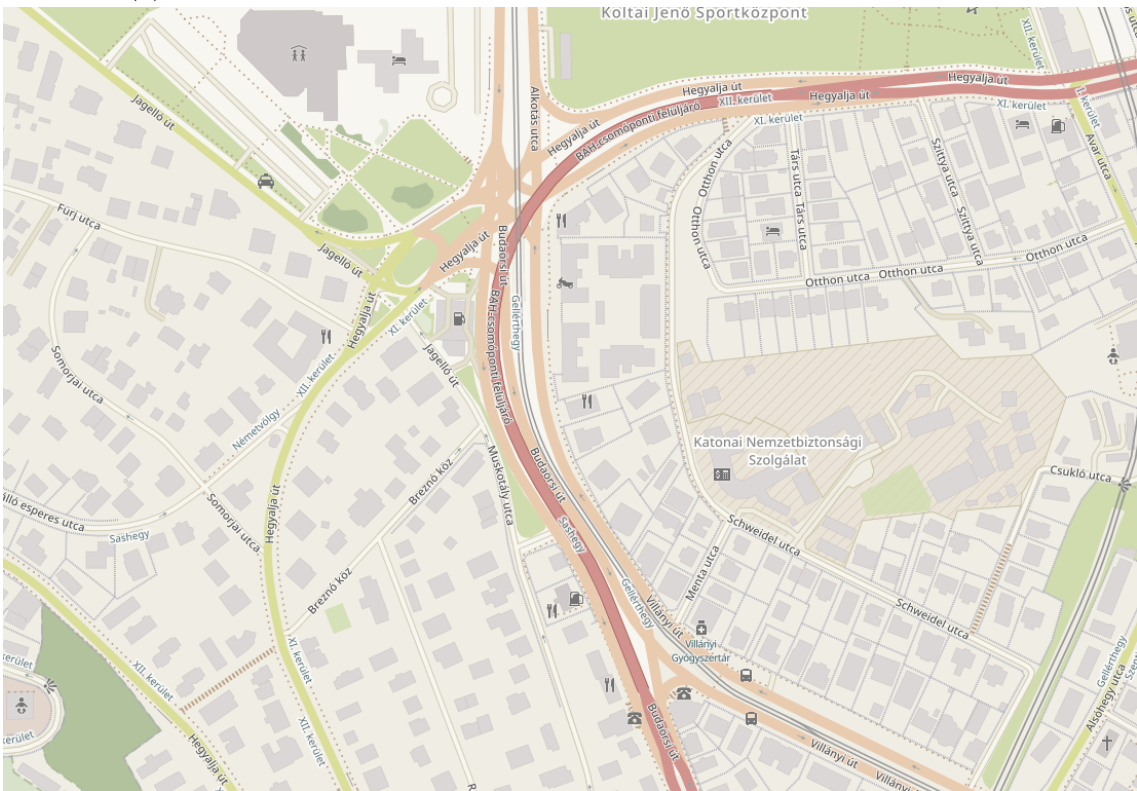
Az *alcsomópontok* természetes módon versenyeznek az alcsomópontokat össze-

¹Pl. a BAH-csomópont esetén, ha az Alkotás utca dél felé vezető oldalának végén történik egy baleset, akkor előfordulhat az, hogy a Villányi út felől érkező járművek nem tudnak a Jagelló és Hegyalja utak felé, nyugatra fordulni.

²Vagy a további bővítés értelme már megkérdőjelezhető lenne a nagy földrajzi távolság miatt.

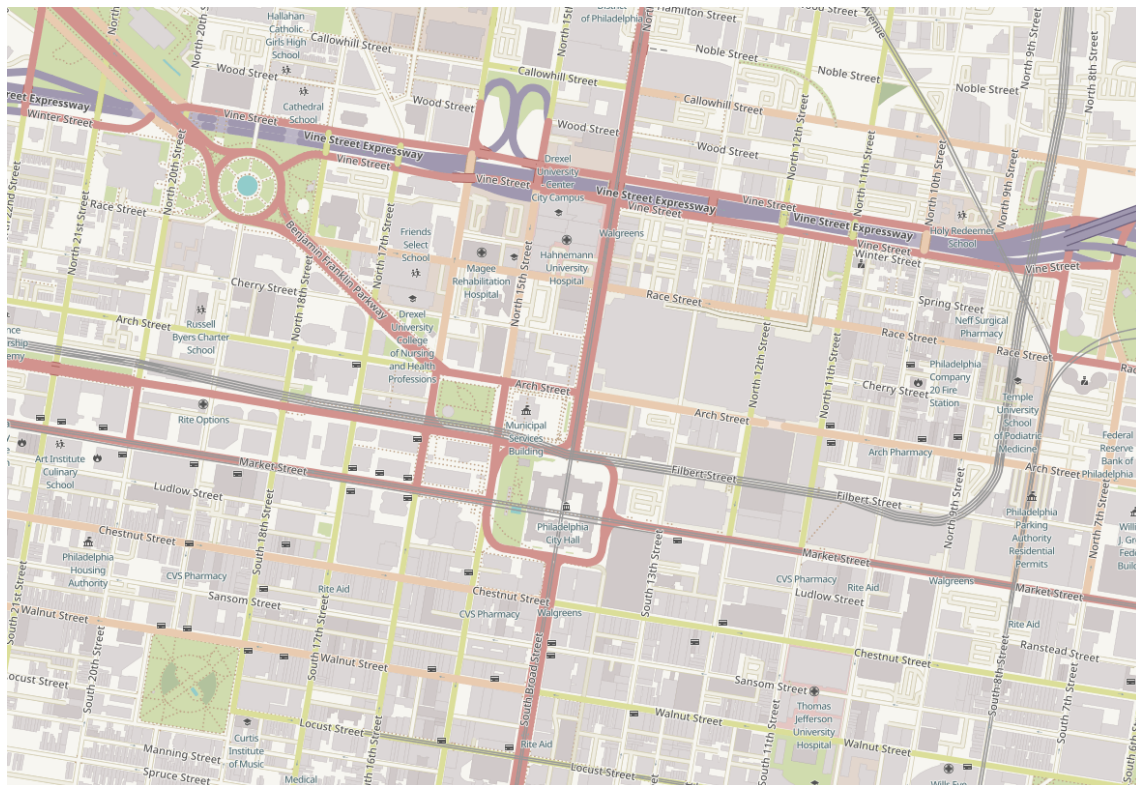


(a) Balra a bécsi Paterstern, jobbra a moszkvai Tanganskaja térképe



(b) A budapesti BAH-csomópont térképe

3.1. ábra. Példák csomópont-konglomerátumokra



3.2. ábra. Philadelphia kereszteződései

kötő (a konglomerátumon *áthaladó*) útszakaszokra adott idő alatt rábocsátható járművek számáért.

Egyszerű menedzselt konglomerátumoknak tekinthetőek a manapság elterjedt, a „zöldhullám” elvét követő időzítésen alapuló forgalomirányítás. Mi a továbbiakban az intelligens, menedzselt konglomerátumokról fogunk beszélni.

3.2.2. Algoritmus csomópont-konglomerátum keresésére

Az összetartozó alcsomópontok keresése ugyan „józan ésszel”, ad hoc módon is megoldható lenne, szükséges lehet egy szisztematikus módszer megadása is, amely megadja, hogy egy csomópont környezetében mely más csomópontok vannak. Azaz azt adja meg, hogy mely alcsomópontok alkotnak egy csomópont-konglomerátumot (X halmaz).

A következő algoritmus iteratíván növeli a kiindulási csomópont (c) környezetét mindig újabb alcsomópontok hozzávételével. Illetve tételezzük fel azt, hogy egy viszonylag kicsi területen (pl. egy L_{\max} sugarú körben) keresünk egy maximális méretű csomópont-konglomerátumot. Ismerjük még a konglomerátumon és a közelében haladó összes lehetséges útvonalat (R -et) is, amely a csomópontok sorrendjeként van megadva. A csomópont-konglomerátum méretére is megadunk egy korlátot, ezt

Algoritmus 1: Csomópont-konglomerátumot kereső algoritmus

Input: c, L_{\max}, R, \max **Output:** X

```
1 Eljárás
2    $X \leftarrow \{a\};$ 
3    $d \leftarrow X$ -be vezető úttal rendelkező csomópont, feltéve, hogy  $L_{\max}$ -nál
      közelebb van;
      /*  $K$  legyen az olyan  $R$ -beli útvonalak halmaza, amikben a  $d$ 
      csomópont és az  $X$  csomóponthalmaz (részhalmaza) ebben a
      sorrendben szerepel: */
4    $K \leftarrow \{r_i | r_i \in R \wedge (d \in r_i \wedge X \subset r_i), \text{ ebben a sorrendben}\};$ 
5   ha  $\forall r_i \in K$ -ra teljesül, hogy  $d$ -t az  $X$  (részhalmaza) közvetlenül követi
      akkor  $X \leftarrow X \cup \{d\};$ 
6   ha  $|X| = \max. \vee$  nincsen már  $d$ -nek jelölhető csomópont akkor
7     | return  $X;$ 
8   különben
9     | folytasd a 3. sornál;
10  | elágazás vége
11 Eljárás vége
```

\max -szal jelölöm. Az eljárást az 1. algoritmus mutatja be.

3.2.3. Példa az algoritmus futtatására

Tekintsünk meg egy példát az 1. algoritmus futtatására! Álljon a kereszteződés-rendszerünk 9 kereszteződésből, melyeket A, B, C, ..., I betűvel jelölünk, ahogyan azt a 3.3. ábra mutatja. A kereszteződések az ábra szerint vannak összekötve úttal, továbbá az A jelű csomópont felől érkező a B csomópontba nem kanyarodhatunk a C felé balra, míg F-ből érkező a G csomópontba jobbra nem kanyarodhatunk az E irányába.

Keressük arra a kérdésre a választ, hogy mely csomópontok tartoznak H csomópont környezetébe, azaz melyek alkotnak H-val egy csomópont-konglomerátumot. Arra jutunk, hogy a B csomópont a konglomerátumba tartozik, ugyanis a jobbra kanyarodás tiltása miatt minden olyan útvonal, amely G-t és H-t tartalmazza ebben a sorrendben, az G után szigorúan a H-t tartalmazza.

Nézzük ezután a G-vel szomszédos E-csomópontot! Mivel E-ből akár F-en keresztül is eljuthatunk G-be vagy akár közvetlenül is, ezért nem igaz az, hogy E-G szükségszerűen szigorúan követi egymást, így tehát E nem lesz tagja a konglomerátumnak. Ugyanez az indoklás igaz a G-vel szomszédos F-re is, ugyanis innen G-be mehetünk közvetlenül vagy az E-n keresztül is, tehát F sem lesz tagja a konglomerátumnak.

A hálózat szimmetriája miatt fogadjuk el részletesebb indoklás nélkül, hogy a B csomópont a G-hez hasonlóan a konglomerátum része lesz, C és A viszont nem. Már csak egy a konglomerátum tagjaival szomszédos kereszteződést kell megvizsgálunk, a D-t. Ez sem lesz tagja a konglomerátumnak, ugyanis a D-t és H-t tartalmazó útvonalakon D és H nem feltétlenül követi egymást ebben a szigorú sorrendben, mert közük C(A)B vagy E(F)G szakasz is beiktatható még.

3.2.4. Konfliktusforrások csomópont-konglomerátumokban

Vegyük észre, hogy érdekessé vált az alcsomópontok közötti *áthaladó útszakaszokról*, mint szűkös erőforrásról beszélnünk, amiért a járművek versenyeznek [16]. Ezt a versenyzést a bírónak érdemes csillapítania abból a célból, hogy ne alakuljanak ki torlódások ezeken a szakaszokon.

Ezt egy szemléletes analógiával is megfogalmazhatjuk: a kereszteződésben (ami szintén egy szűkös erőforrás, és versenyeznek érte a járművek) igyekszünk minél rövidebb időt eltölteni, és ehhez állítani be a jelzőlámpákat. Végző soron az áthaladó szakaszokon is erről van szó, így világos, hogy a járművek útvonaláról való (kommunikáció által nyert) többlettudást is hasznosíthatjuk azáltal, ha a korábbi bíróink ezt is figyelembe veszik áthaladásról való döntésük során. Bízhatunk tehát abban is, hogy ezáltal tovább növelhető a járműfolyam nagysága az úthálózatban.

3.3. Jelzésterv generálása

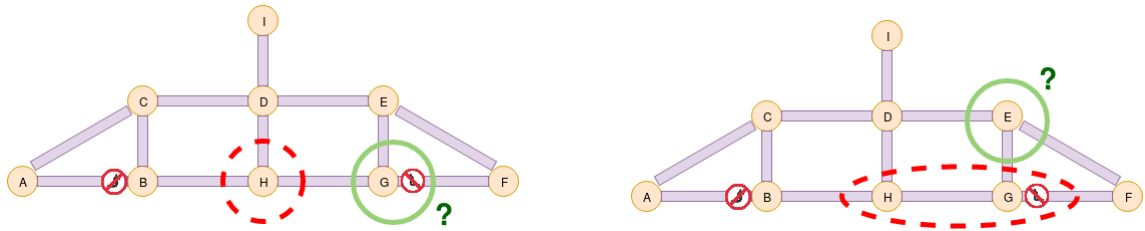
A hagyományos forgalomirányító berendezések jelzéstervei, azaz, hogy melyik időpontban mely irányok kapnak szabadjelzést, illetve mely irányoknak kell megállnia, többnyire előre elkészített, periodikus programot követnek. Ennél kicsivel modernebb megoldás, ha ez a program alkalmazkodik a pillanatnyi forgalomhoz, így például a hurokdetektorok adatai szerint néhány másodperccel módosíthatja az egyes programfázisok hosszát.

Vegyük észre azonban, hogy egy adaptív, kooperatív rendszer esetén nem tudjuk előre megmondani azt, hogy egy adott pillanatban kinek érdemes zöldjelzést mutatni. Sőt, egy periodikus programot sem tudunk meghatározni a klasszikus módon! Emiatt szükségünk van tehát arra, hogy minden pillanatban ki tudjuk számítani, hogy ki kaphat éppen szabadjelzést és kinek kell megállnia, hívjuk ezt a problémát a *pillanatnyi jelzések előállításának*.

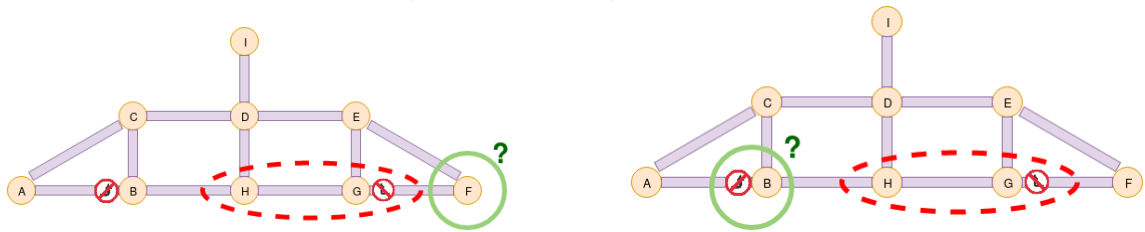
A másik megoldandó feladat az, hogy mikor kell újabb pillanatnyi jelzésekpet elő-, illetve beállítani. Erre a problémára pedig egyszerűen *időzítésként* fogok hivatkozni.

3.3.1. A pillanatnyi jelzések előállítása

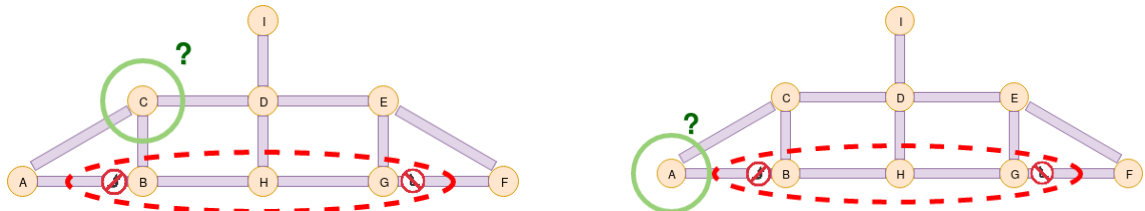
Hívjuk a kereszteződés *konfliktusmátrixának* azt a 0-ákból és 1-esekből álló (felső háromszög-) mátrixot, amelynek egy (i, j) , $i \neq j$ koordinátájú eleme pontosan



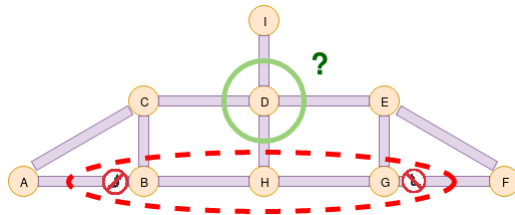
(a) A H jelű kereszteződésből indítjuk a keresést, először a G csomópontot vizsgáljuk, amely H környezetében van (*bal oldali ábra*). Az E-vel folytatjuk a keresést, de ez nem lesz tagja a H konglomerátumának (*jobb oldali ábra*).



(b) Az F-fel folytatjuk, mely nem lesz tagja a konglomerátumnak (*bal oldali ábra*). A B-t viszont bevesszük a konglomerátumba (*jobb oldali ábra*).



(c) Vizsgáljuk a C kereszteződését (*bal oldali ábra*). Ez nem kerül be a konglomerátumba. Megvizsgáljuk az A kereszteződését (*jobb oldali ábra*), ez sem vehető bele a konglomerátumba.



(d) Végül pedig megnézzük D-t, amely nem fog bekerülni a konglomerátumba.

3.3. ábra. Példa a csomópont-konglomerátumokat kereső algoritmus futtatására. A kereszteződéseket barack színű körök, az őket összekötő szakaszokat lila téglalapok jelölik. A felől B-ben C felé kanyarodni, illetve F felől G-ben E felé kanyarodni tilos. A végeredményként azonosított konglomerátum a 3.3d. alábbián látszik, piros szaggatott vonallal határolva.

akkor 1-es, ha a kereszteződés i trajektória és j trajektória metszi egymást. Ezen kívül tételezzük fel azt is, hogy az egyes trajektóriákra előírhatunk bizonyos tulajdonságokat, például, hogy egy trajektória biztosan kapjon zöldet, hívjuk ezeket

egyéb feltételeknek. Ezek az egyéb feltételek fogják biztosítani a forgalomirányítás logikáját.

Legyen $\mathbf{x} \in \{0, 1\}^N$ vektor x_i koordinátája 1, ha a kereszteződés i . trajektóriája mentén mozgó járművek szabadjelzést kapnak, 0 pedig akkor, ha nem. N jelölje azt, hogy hány különböző trajektória képzelhető el egy adott kereszteződésben.

Ekkor a C. függelékben részletesen definiált egészértékű programozási feladat \mathbf{x} megoldásai megadnak egy legnagyobb olyan trajektóriahalmazt, akik az adott kereszteződésben, a megadott feltételek mellett szabadjelzést kaphatnak az adott kereszteződésben.

3.3.2. Az időzítés

Azon kívül, hogy elő kell tudnunk állítani a pillanatnyi jelzéseképet, azt is tudnunk kell, hogy mikor kell újraszámolni a jelzéseképet, illetve tájékoztatni az okosautókat az új állapotról. Mondjuk azt, hogy azok a járművek, akik egyszerre áthaladhatnak a kereszteződésen, azok egy *konfliktusosztályt* alkotnak. Azok a járművek, akik éppen szabadjelzést kapnak, ők alkotják az *aktív konfliktusosztályt*.

A szimulátorban, annak érdekében, hogy mindig az aktuális forgalmi helyzetet vehessük figyelembe, minden lépésnél megállapítjuk, hogy mekkora T periódusidőnként kell újabb jelzéseképet számítanunk. Jelölje N_j az aktív konfliktusosztályban lévő járművek számát! Ekkor a T értéket a (3.1) képlet alapján számíthatjuk.

$$T = \begin{cases} N_j \cdot 1,5 \text{ s} + 5 \text{ s}, & \text{ha } N_j \leq 23 \\ 40 \text{ s} & \text{különben} \end{cases} \quad (3.1)$$

A (3.1) képlet *különben* ága biztosít egy felső korlátot a ciklusidőre nézve. A másik ág pedig úgy kalkulál, hogy egy általános kereszteződésben egy autó áthaladásához nagyjából 1,5 s-ra van szükség. Illetve az első jármű elindulását és az utolsó jármű biztonságos eltávolódását segíti a +5 s tag. Az $N_j \leq 23$ feltétel pedig azt segíti elő, hogy a *különben* ág által szabott maximális időkorláttal ne kerüljünk ellentmondásba, ugyanis $N_j = 23$ db jármű esetén a felső ág $T = 39,5 \text{ s}$ -ot ír elő, de $N_j = 24$ esetén ez a szám már $T = 41 \text{ s}$ lenne, ami ellentmondásba kerülne a felső korláttal.

Ha legutolsó újraszámítás óta már eltelt T időtartam, akkor újrageneráljuk az aktuális jelzéstervet a 3.3.1. fejezetben ismertetett módon. Amint az eredmény előállt, ezt a döntést közöljük is az okosautók felé.

3.4. Az ECN-bíró

Vezessünk most be egy új típusú kooperatív, intelligens bírót! Nevezzük ezt a bírót *ECN-bírónak*. Ez a bíró képes valós időben jelzéstervet generálni a 3.3. fejezetben leírtak szerint. Ezen kívül képes arra, hogy a saját forgalmi helyzete tekintetében

valamilyen szempontból optimális döntést hozzon a szabad- és tilosjelzések adásakor, illetve képes jelzéseit a környezetében lévő kereszteződések jelzéseire is igazítani.

Mivel korábbi munkám tapasztalata azt mutatta, hogy a körforgó, *Round-Robin*, *RR* alapú megoldás valamivel jobban teljesít az általam próbált *Minimal Destination Distance First*, *MDDF* algoritmushoz képest, ráadásul *fair* is az erőforrás nyújtása tekintetében az irányokkal, úgy döntöttem, hogy a lokális szabadjelzés-nyújtást *RR*-logika szerint fogom végezni. Ehhez persze a körforgó ütemezést meg kell fogalmaznunk úgy, hogy, azt a 3.3.1. fejezetben bemutatott egészértékű programozási feladat A mátrixába beírható legyen egyéb feltételként.

Hogy ezt megtehesük, nyilvántartunk egy listát arról, hogy melyik irány mikor kapott utoljára szabadjelzést. A jelzésterv generálásakor megnézzük, hogy melyik az az irány, ahol várakoznak autók, és a legrégebb óta nem kaptak szabadjelzést. Ennek az iránynak megfeleltethető a 3.3.1. fejezet \mathbf{x} vektorának egy eleme, jelölje ezt x_i . A programban tehát elő kell írunk egyéb feltételként, hogy ez az x_i érték kötött, azaz, hogy $x_i = 1$ legyen. Így elérjük azt, hogy a legrégebb óta várakozók továbbhaladhassanak, tehát ez így egyfajta körforgó ütemezést fog eredményezni.

Korábbi munkám során a bírók konfliktusosztályokba sorolták a beérkező járműcsoportokat. Ezt az *ECN*-bíró is megteszi, viszont itt a dinamikusán változó jelzésterv miatt a konfliktusosztályokba sorolás is dinamikus. Mivel ez csak egy adminisztratív jelölés a bíró oldalán, ez kommunikációs pluszterhelést nem fog okozni.

Két triviális konfliktusosztályunk lesz: az az osztály, amelynek a tagjai jelenlegi jelzés szerint áthaladhatnak a csomóponton; és az, akik éppen várakozásra kényszerülnek. Mindig, amikor egy újabb jelzések kiszámításra kerül, a járműveket az *ECN*-bíró újra beosztja egyik vagy másik konfliktusosztályba.

Kérdéses maradt viszont az, hogy miként érzük el, hogy az *ECN*-bírók képesek legyenek koordinálni egymás jelzését. Erre a kérdésre adok választ a következőkben.

3.4.1. Az explicit torlódásértesítés

A számítógépes hálózatoknál ismert az a probléma, hogy egy-egy csomóponthoz egy időintervallumban több kérés érkezik, mint amennyit az ki tud szolgálni. Ilyenkor a kérések egyrésze elvész, ami előidézi azt, hogy a kérés feladója újra elküldi kérés csomagját, így tetézve a rendszer túlterheltségét. Erre ad egy megoldást az explicit torlódásértesítés *Explicit Congestion Notification*, *ECN* [9]. Az eljárás lényege, hogy a fogadó hálózati csomópont érzékeli azt, hogy a bejövő buffere kezd megtelni, azaz egy torlódást mutató valószínűségi függvény átlép egy bizonyos felső küszöbszintet. Ilyenkor a csomópont jelzi a feladónak a torlódás tényét. A feladó kötelessége ilyenkor az, hogy csökkentse a fogadó irányába történő forgalmazás nagyságát. Amint a torlódás elmúlik, azaz a valószínűségi függvény egy bizonyos alsó küszöbszint alá esik, akkor fogadó jelzi a feladó számára, hogy a torlódás elmúlt, ismét adhat az eredeti rátával.

Vegyük észre, hogy ez a módszer forgalomirányítási rendszereknél is alkalmaz-

ható. A csomópontok itt a kereszteződések (egészen pontosan a bírók), a bemeneti buffer pedig a közlekedési csomópontokat összekötő útszakaszok. Azaz most arra fogunk törekedni, hogy ezeken a szakaszokon elkerülhessük a torlódásokat, bízva abban, hogy ez segíteni fog növelni a teljes hálózat áteresztőképességét is.

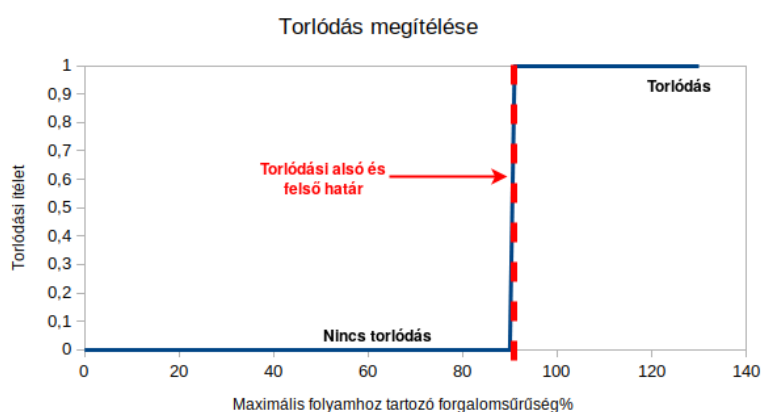
A jelzéskoordináció célja tehát az lesz, hogy egy csomópont-konglomerátum al-csomópontjai közötti szakaszokon ne alakuljanak ki torlódások. Vegyük észre továbbá azt is, hogy ha itt kezdene sűrűsödni a forgalom, akkor az al-csomópontok működhetnek tovább úgy, hogy ebbe az irányba kevesebb járműnek engednek szabadjelzést, ami ennél fogva azt is jelenti, hogy közben más irányok viszont hosszabb „zöldidőt” kaphatnak.

3.4.2. A torlódás valószínűsége

Az internetes környezetben viszonylag egyszerű torlódásvalószínűségi függvények (pl. a buffer 80%-nál telítettebb, akkor torlódás van egészen addig, míg ez az érték 40% alá nem csökken) is igen jó eredményeket adnak. A forgalomirányítási doménben ugyanakkor egy ilyen metrika kidolgozása sokkal bonyolultabb lehet.

Mivel a forgalmi torlódások felismerése messze nem triviális feladat [18], [23], ezért dolgozatomban nem volt célom, hogy egy optimális valószínűségi függvényt válasszak, ezért egy igen egyszerű megoldást alkalmaztam vizsgálataimhoz.

Szimulációval, a Simulation of Urban MObility adta hurokdetektorokkal kimértük az egyes útszakaszokon azt, hogy mekkora forgalomsűrűség, pontosabban sáv-foglaltság%³ tartozik a maximális járműfolyamhoz. Ha pillanatnyi sávfoglaltság meghaladja ennek az értéknek a 90%-át, akkor azt úgy ítéljük meg, hogy torlódás van az adott szakaszon, egyébként pedig nincsen, lásd a 3.4. ábrán.



3.4. ábra. Döntési küszöb, hogy mikor torlódik a forgalom

³Ez egy jobb mérték a forgalomsűrűséghez képest, ugyanis a sávfoglaltság jelentése az, hogy a sávfelület mekkora részét foglalják el járművek. Így nem kell foglalkoznunk elméleti szempontból azzal, hogy egységjárműveket, és egységjármű-egyenértékeket definiáljunk.

3.4.3. Bírók közti üzenetváltás

A bírók képesek számon tartani azt (az okosautókkal való kommunikációjuk alapján), hogy jelenleg hány jármű tartózkodik az adott sávon, így a feltételezett homogén járműpark segítségével pontosan tudja, hogy a kérdéses útszakasz foglaltsága jelenleg mekkora. Ezt az értéket a bírók a csomópont-konglomerátumon belül beléjük futó szakaszokon 15 másodpercenként megvizsgálják. Amennyiben átlépi a határ a küszöböt, akkor a bírók egy *broadcast* típusú, úgynevezett *ECN-üzenetben* közlik, hogy torlódás van valamelyik beléjük vezető olyan szakaszon, ami a csomópont-konglomerátumon belül fut, egyébként pedig ennek ellenkezőjéről tájékoztatják a többi ECN-bírót.

3.4.4. Torlódásjelzés figyelembe vétele jelzésterv-generálásnál

A kapott ECN-üzenetet természetesen figyelembe kell venni a jelzésterv generálásakor is. Ezt a 3.3.1. fejezetben definiált A mátrixba felvett egyéb feltételként tudjuk megtenni. A torlódást jelző ECN-üzenetben kapott iránynak megfeleltethető egy $\hat{\mathbf{x}} \subset \mathbf{x}$ irányhalmaz. A programban tehát elő kell írunk egyéb feltételként, hogy az $\hat{\mathbf{x}} = \mathbf{0}$, azaz jelenleg az egyik irányhalmazba nem szabad szabadjelzést adni.

Így egy csomópont torlódásjelzésére a környezetében lévő csomópont(ok) képesek nagyjából valós időben reagálni. A rendszerben a torlódás kialakulásához képest legfeljebb 55 másodperc telik el⁴ addig, amíg a környezet nem állítja át a jelzését a torlódást elkerülő állapotra.

3.5. Kommunikációs protokollok

3.5.1. Kommunikáció a bírók között

A bírók közötti kommunikáció meglehetősen egyszerű: minden ECN-bíró 15 másodpercenként *broadcast* módon elküldi a többi bírónak a következő információkat⁵:

⁴15 másodpercen belül detektálja az egyik alcsomópont ECN-bírója a torlódást, és jelzi ezt a többi bíró felé, akik legfeljebb 40 másodperc múlva (az új jelzések generálásának ideje) reagálnak erre az üzenetre.

⁵Dolgozatomban a periodikus küldés csupán egy egyszerűsítő eljárás az eseményvezérelt küldés helyett. Valós csatornák esetén viszont hibátűrés szempontjából előnyösebb tud lenni a periodikus küldés az eseményvezérelt küldéssel szemben: A periodikus broadcast küldés segít elérni azt, hogy a bírógensek tudásbázisa konzisztens legyen, míg egy eseményvezérelt küldésnél ezt külön eljárásokkal kellene biztosítani. A periodikus küldésnek továbbá megvan az az előnye is, hogy egyfajta életjelként is szolgálhat a hálózatban, azaz ha egy bíró már egy bizonyos ideje nem küld magáról státuszinformációt, akkor úgy tekinthetnénk, hogy az adott bíró jelenleg nem képes ellátni a funkcióját. Ezen előnyök mellett a periodikus küldés csatornaterhelése nem túl magas, mivel igen kis frekvenciájú és kis méretű üzeneteket használ. Ráadásul ez a csatorna a valóságban akár egy fizikai pont-pont összeköttetés is lehet két csomópont között, így nem a rádiós kapcsolat terhére működik.

- ECN-bíró azonosítója.
- ECN-bíróba a csomópont-konglomerátumon belül beérkező útszakasz, úgynevezett *port* azonosítója.
- A portok státusza (torlódik-e vagy sem).

Azon bírók, akik az üzenet vételekor kapcsolatban állnak a megadott azonosítójú ECN-bíró valamely portjával, eltárolják, hogy mit közölt erről a portról a küldő, azaz, hogy van-e éppen azon a porton torlódás vagy nincsen.⁶ Később ezt az eltárolt értéket fogja figyelembe venni az ECN-bíró a jelzésterv generálásakor.

3.5.2. Kommunikáció az okosautók és a bírók között

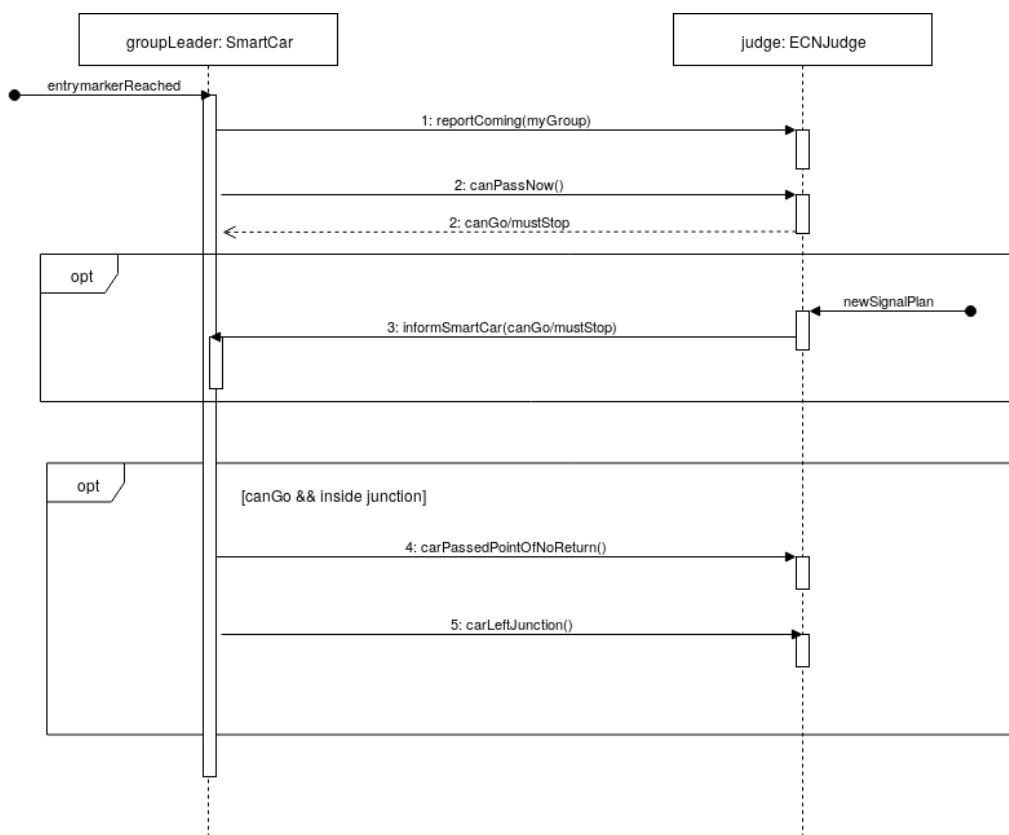
A korábban már ismertetésre került módon, ha az okosautók a csomópont adott irányához rendelt belépési markerre érnek, csoportokba rendeződnek és a csoport vezetője felveszi a kapcsolatot a csomópont bírójával. Bejelenti felé, hogy a csoport melyik irányba szeretne továbbhaladni a kereszteződésen át. A bírók ezt a tényt feljegyzik.

Ezután a korábbi munkámtól eltérő módon most nem folyamatos lekérdezést valósítok meg az okosautók által kezdeményezve, hanem a bírók üzennek szükség esetén az okosautóknak. Amennyiben egy konfliktusosztály aktívvá válik, a bíró a konfliktusosztályban lévő csoportvezetőket értesíti, hogy szabadjelzést kapnak, áthaladhatnak a kereszteződésen. Amikor pedig egy konfliktusosztály elveszíti az aktív státuszát, a bíró egy tilosjelzést küld a konfliktusosztály csoportvezetői felé.

Ha ezt jobban megnézzük, észre vehetjük, hogy a csoportvezetők a kereszteződést jelző pontra, *belépési markerre* lépéskor azt nem tudhatják, hogy éppen aktív-e az a konfliktusosztály, amibe tartoznak. Ezért ezt a kezdőstátuszt lekérdezi a bírótól közvetlenül a bíróhoz való bejelentkezése után.

A pontos protokoll leírását mutatja a 3.5. ábra. Ahogyan azt a [2] is bemutatja, az okosautók felismerik az úthálózat bizonyos pontjára elhelyezett belépési markert (*Entry Marker*), és ennek a pontnak az elérése indítja el a protokollt a csoportvezetők és a bírók között.

⁶Ez a tárolás lehetővé teszi, hogy a későbbiekben egy bonyolultabb számítás közlését is egyszerűen figyelembe vehessük. Például azt, ha a port státusza nem egy bináris állapotváltozó, hanem egy folytonos érték.



3.5. ábra. Az okosautók és egy ECN-bíró közötti kommunikáció protokollja

4. fejezet

Mérési eredmények és összehasonlításuk

Az elkészült rendszer képességeit a Simulation of Urban MObility nevű nyílt forráskódú mikroszkopikus forgalomszimulátor [17] segítségével teszteltem.

4.1. Mérési környezet

Méréseimet a BAH-csomópont egyszerűsített hálózatán végeztem, melyet a 4.1. ábra mutat. A betáplált forgalmi igény egy átlagos reggeli forgalmi igénynek felel meg. Méréseim során ezt a forgalomnagyságot szoroztam fel az eredeti konfiguráció egész számú többszöröseire.

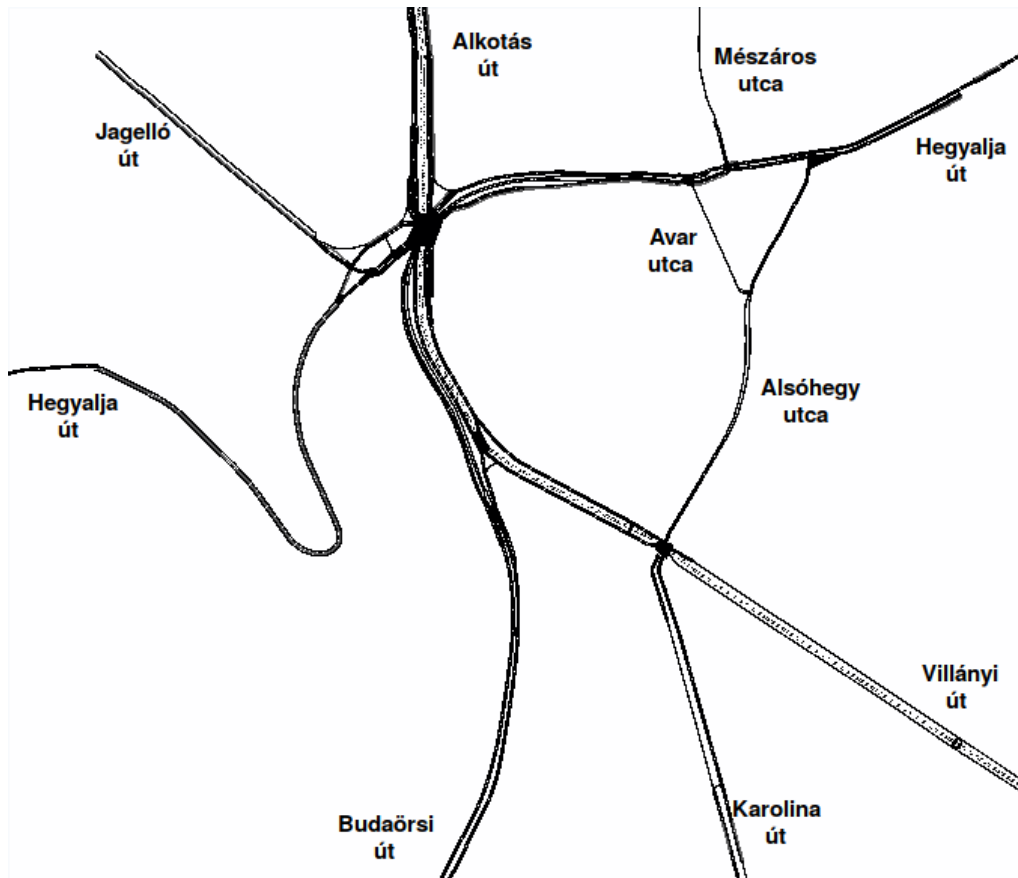
A szimulátor által adott összesített adatok alapján vizsgáltam a D [egység-jármű/km] egységben adott forgalomsűrűséghez a Q [egységjármű/óra] egységben adott járműfolyam-értéket. Ezek a jellemzők mutatják, hogyan viselkedik a forgalmi hálózat bizonyos forgalomnagyságok, forgalomsűrűségek mellett.

Az ECN-bírók beállításainál a B. függelékben meghatározott csomópontokat tartottam összetartozónak a Hegyalja, Villányi, illetve a Budaörsi utak megfelelő szakaszait tekintve a konglomerátumon belül futó, a kereszteződések összekötő szakaszoknak, ahogyan azt a 4.2. ábra mutatja.

Hogy összehasonlíthassuk az eredményeket, méréseket végeztem úgy is, hogy csak ECN-bírók vannak a rendszerben, illetve úgy is, hogy csak a konglomerátum tagjai ECN-bírók, a többi kereszteződés RR típusú intelligens bíróval van ellátva.

4.2. Mérési eredmények

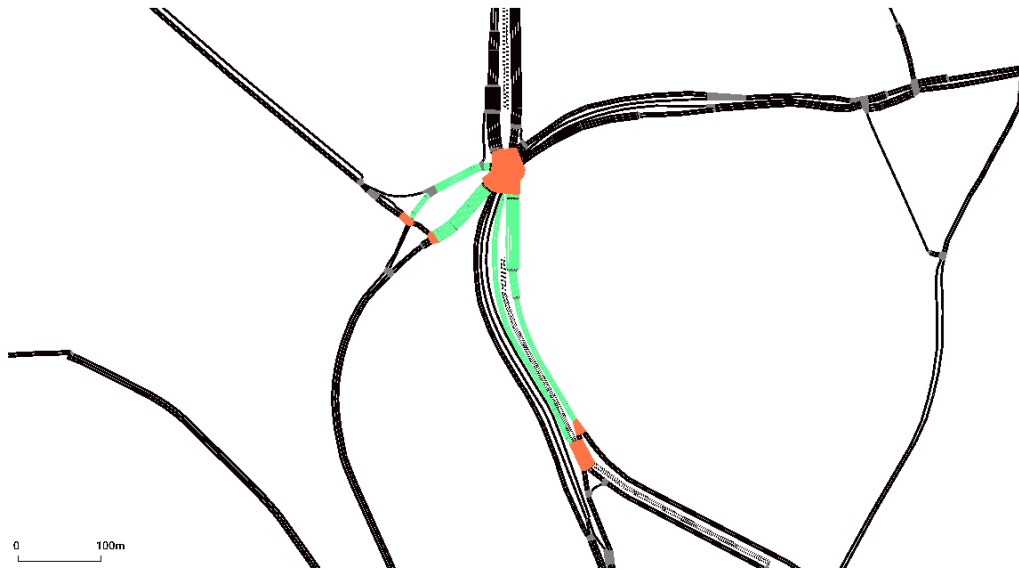
Mérési eredményeimből részleteket közlök a 4.1. táblázatban, a teljes mérés eredményét pedig a 4.3. ábrán grafikonként ábrázoltam, alkalmasnak tűnő trendvonal illesztésével.



4.1. ábra. A BAH-csomópont egyszerűsített hálózata

Megfigyelhetünk egy érdekességet a kapott eredményekben. Méghozzá azt, hogy az **ECN-bíró alkalmazásával** (főleg, ha homogén a rendszer), **nem képes akármilyen magas forgalomsűrűségi értéket felvenni a rendszer.** Ezt mutatja az összetorlódott mérési eredmények a homogén rendszer grafikonjában a $D = 60$ érték környékén, illetve a vegyes ECN-RR rendszer esetén a $D = 80$ környékén. Ez persze nem túl meglepő, ugyanis célul tűztük ki azt, hogy bizonyos útszakaszokon ne alakuljon ki torlódás, a torlódás kialakulását pedig a forgalomsűrűség bizonyos szintnél magasabbra emelkedésének tekintettük.

Megállapíthatjuk azt is, hogy alacsony terhelési szintek mellett a különböző konfigurációjú rendszerek nyújtotta eredmények alig különböznek egymástól, míg nagyjából a normálhoz képesti háromszoros terhelés környékétől kezdve az ECN-alapú rendszerek alulmaradnak a korábbi megoldásokhoz képest.



4.2. ábra. A BAH csomópont-konglomerátumának tagjai (narancssárgával jelölve) és az őket összekötő szakaszok (zölddel jelölve)

4.1. táblázat. Forgalm nagysághoz tartozó makroszkopikus jellemzők, részlet

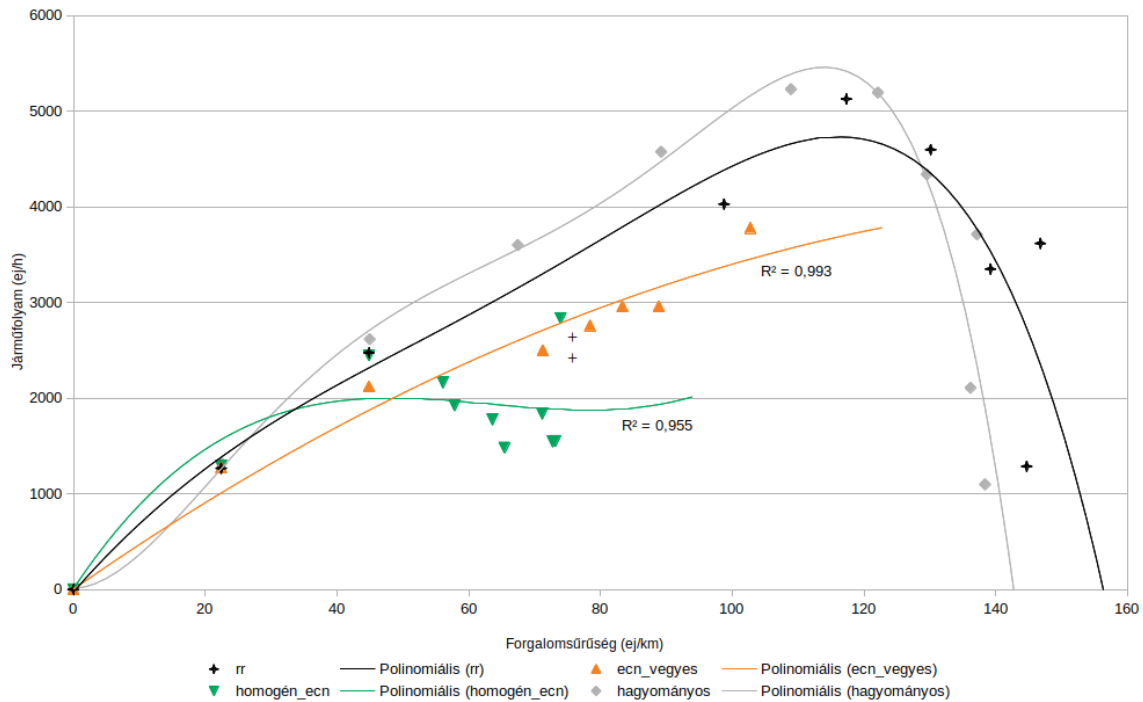
Bírók típusa	1x-es forgalom		2x-es forgalom		5x-ös forgalom	
	D	Q	D	Q	D	Q
hagyományos	22.492	1319.4	44.983	2617.2	108.930	5232.6
Intelligens RR	22.430	1267.2	44.86	2476.8	98.773	4030.2
Intelligens homogén ECN	22.430	1290.6	44.860	2448	65.450	1483.2
Intelligens vegyes RR és ECN	22.430	1274.4	44.860	2120.4	83.338	2957.4

4.3. A kapott eredmények értékelése

A kapott eredmények alapján arra a következtetésre juthatunk, hogy az ECN-bírók sikerrel kerülnek el a torlódásokat az általuk felügyelt szakaszon (lásd a 4.4. ábra), ám a csomópont-konglomerátum határainál jelentős torlódásokat képes okozni, lásd a 4.4b. ábrát!

Az ECN és RR-bírókat vegyesen tartalmazó rendszer eredményei alapján pedig azt mondhatjuk, hogy egy úthálózat áteresztőképességét azzal nem tudjuk növelni, ha bizonyos szakaszain arra törekszünk, hogy ne alakuljon ki torlódás. Azaz a forgalmi hálózat áteresztőképességét nem lehet néhány csomópontja összekapcsolásával a számítógépes hálózatokból ismert explicit torlódásértesítés alapján működő eljárással növelni.

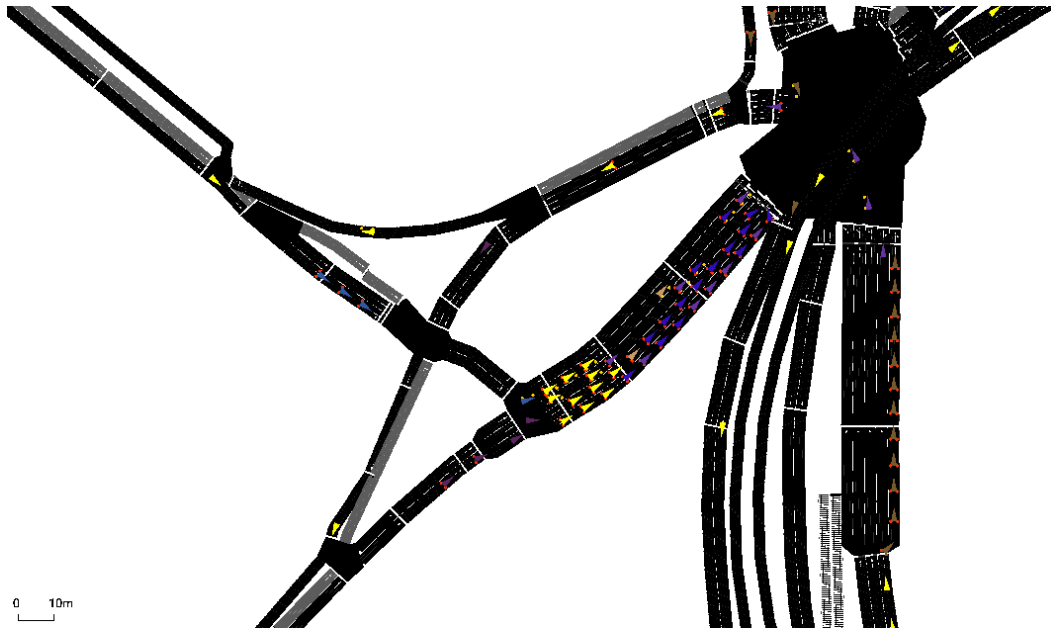
Előfordulhat viszont olyan eset, amikor pontosan az a célunk, hogy egy úthálózatnak egy kisebb részében ne alakulhasson ki torlódás. Ilyen eset például az, ha egy városban akármekkora torlódás is van, egy-egy lakó- vagy természetvédelmi övezet úthálózatára ez ne terjedjen át. A BAH-csomópont környékén például elérhető



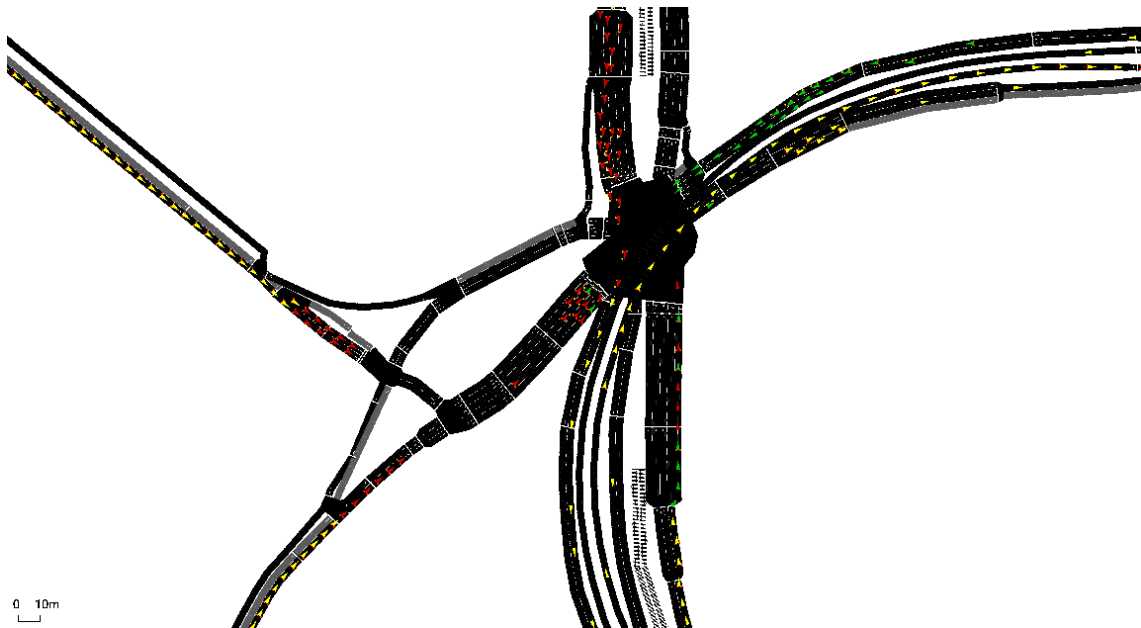
4.3. ábra. A kapott eredmények ábrázolása grafikonon. Az adatpontokra jól illeszkedő polinomiális (negyedfokú) trendvonalat illesztettem, melyet még 20 ponttal extrapoláltam. Megfigyelhető, hogy az ECN-t tartalmazó rendszerek görbéje ellaposodik, ez a rendszer működését mutatja: az ECN-bírók által védett szakaszokon nem lehet tetszőlegesen nagy a járműsűrűség, így a járműfolyam nagysága sem.

lehet, hogy a Gellért-hegyen lévő lakóövezet torlódásmentes maradjon a reggeli és esti csúcsforgalom idején. Vagy az is megoldható lehet, hogy a Sashegyre, ami egy szigorúan védett természetvédelmi övezet, ne hasson a BAH-csomópont környékén kialakult torlódás. Ilyenkor a szóbanforgó hálózat csomópontjait érdemes lehet egy konglomerátumba fogni, és így alkalmazni az ECN-bírókat. Sajnos ezen vizsgálatok meghaladják jelen dolgozat kereteit, viszont a jövőben érdemes lehet megvizsgálni, hogy valóban elérhető-e ez a sejtett hatás.

Az ECN-bírók másik lehetséges alkalmazási területe lehet, hogy egy olyan többszintű ütemezést alakítunk ki egy bíró számára, hogy kezdetben például az RR ütemezést alkalmazza, majd ha a forgalom nagysága elér egy bizonyos pontot, akkor átvált ECN-működésre, így biztosítva azt, hogy a rendszer bármekkora terhelést is kap, a forgalmi folyamat még extrém helyzetekben se csökkenjen 0-ra. Ezen vizsgálatok is elvégezhetőek lesznek a jövőben.



(a) RR-bíró: Torlódás(közel)i állapot a BAH-csomópont Hegyalja út nyugatról érkező oldalán.



(b) Torlódásmentes állapot a BAH-csomópont Hegyalja út nyugatról érkező oldalán. Cserébe a Jagelló és Hegyalja utak kereszteződésén kívül torlódik a forgalom.

4.4. ábra. ECN-bíró: Az RR típusú bíró és az ECN-típusú bíró által előidézett forgalmi helyzet azonos forgalomnagyságnál, azonos időpontban vizsgálva. A felső ábrán látható, hogy sok autó várakozik a nagyobbik kereszteződés bejáratánál, míg az alsó, némileg nagyobb léptékű térképen megfigyelhetjük, hogy sok autó a hálózat határán várakozik arra, hogy beljebb kerülhessen.

5. fejezet

Összefoglalás

A dolgozat keretein belül elkészült egy explicit torlódásértesítésen alapuló jelzésekordinációs rendszer, amely nem igényli artériák definícióját. Ehelyett egy formális eljárás segítségével vagyunk képesek meghatározni az összetartozó kereszteződéseket. Ezen kereszteződések pedig a hálózat topológiáján keresztül jelöl ki néhány „védett” útszakaszt, de ezek nem azonosak az artériákkal.

Az elkészült rendszer a pillanatnyi forgalomnak megfelelően koordinálja jelzéseit és generálja jelzéstervét. A beépített, nagyon egyszerű módszernek köszönhetően várakozási időt tekintve fair lesz, ugyanis legfeljebb a kereszteződésbe befutó utak számaszor 40 másodpercet kell várakoznia minden egyes irányból érkező járműnek két szabadjelzés között. Ez a rendszer a kereszteződések számával is jól skálázódik, ugyanis minden csomópont csak szomszédainak a befolyása alatt áll, így könnyen bővíthető újabb kereszteződéssel.

Ezen az explicit torlódásértesítésen alapuló rendszeren futtatott méréseim azt mutatják, hogy ez a megoldás nem alkalmas annak a problémának a megoldására, hogy növelje egy úthálózat kapacitását. Arra viszont alkalmasnak tűnik, hogy egy bizonyos kisebb területen elérje azt, hogy ne alakuljanak ki torlódások, és a forgalom nagyságától függetlenül garanciát tudjunk adni a forgalomsűrűséget és forgalmi folyamatot jellemző számokra. Ez sejtésünk szerint előnyös lehet egy nagyforgalmú utak szegélyezte lakó- vagy természetvédelmi övezet úthálózatának torlódásmentesen tartásához. A jövőben érdekes lehet megvizsgálni ennek a lehetőségét.

Érdekes lehet megvizsgálni a jövőben azt is, hogy ügyesebb időzítéssel a jelzéstervgenerálásnál, vagy egy bonyolultabb torlódásvalószínűségi függvénnyel javíthatóak lennének-e a rendszer nyújtotta eredmények.

Az elkészült ECN-bíró pedig kisebb módosításokkal alkalmassá tehető arra, hogy alapesetben szimulálja egy egyszerű Round-Robin alapú bíró működését, és csak azért lépjen működésbe az ECN funkció, hogy a rendszer végső vergődését megakadályozhassuk kellően nagy forgalomsűrűség esetén. Illetve az ECN-bíró alapján elkészíthető lehet akár egy olyan rendszer is, ami képes a vészvillogós járművek útvonalán egy torlódásmentes folyosót kialakítani szükség esetén.

5.1. Irodalomjegyzék

- [1] L. Alekszejenkó and T. P. Dobrowiecki, „SUMO Based Platform for Cooperative Intelligent Automotive Agents,” in EPiC Series in Computing, 2019, vol. 62, pp. 107–123.
- [2] L. Alekszejenkó and T. Dobrowiecki, „Intelligent Vehicles in Urban Traffic – Communication Based Cooperation,” in 2019 IEEE 17th World Symposium on Applied Machine Intelligence and Informatics (SAMi), 2019, pp. 299–304.
- [3] L. Alekszejenkó, „Intelligens gépkocsik városi forgalomban - Kommunikáción alapuló kooperáció vizsgálata”, TDK dolgozat, BME-VIK, November 2018.
- [4] M. Bani Younes and A. Boukerche, „An Intelligent Traffic Light scheduling algorithm through VANETs,” in 39th Annual IEEE Conference on Local Computer Networks Workshops, Edmonton, AB, Canada, 2014, pp. 637–642.
- [5] B. Cesme, „Self-organizing traffic signals for arterial control,” PhD. Dissertation, Northeastern Univ., Boston, 2013.
- [6] L. Chen, C. Englund, „Cooperative Intersection Management: A Survey,” IEEE Trans. on Intelligent Transportation Systems, Vol 17, Issue 2, Feb. 2016, pp. 570-586, DOI:10.1109/tits.2015.2471812
- [7] A. Deligkas, E. Karpas, R. Lavi, and R. Smorodinsky, „Traffic Light Scheduling, Value of Time, and Incentives,” in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 2018, pp. 4743–4749.
- [8] K. Dresner and P. Stone, „A Multiagent Approach to Autonomous Intersection Management,” JAIR, vol. 31, pp. 591–656, Mar. 2008.
- [9] S. Floyd, K. K. Ramakrishnan, and D. L. Black, „The Addition of Explicit Congestion Notification (ECN) to IP.” [Online]. Elérhető: <https://tools.ietf.org/html/rfc3168#section-1>. [Elérve: 2019. okt. 25.].
- [10] G. Fong, R. Cunard, and K. Slack, „Signalized Intersection Safety in Europe,” Office of International Programs, US. Dept. of Transportation, Washington DC, December 2003.
- [11] M. Girianna and R. F. Benekohal, „Dynamic Signal Coordination for Networks with Oversaturated Intersections,” Transportation Research Record, vol. 1811, no. 1, pp. 122–130, Jan. 2002.
- [12] R. Goldstein and S.F. Smith, „Expressive Real-Time Intersection Scheduling,” AAAI, vol. 33, pp. 9882–9883, Jul. 2019.

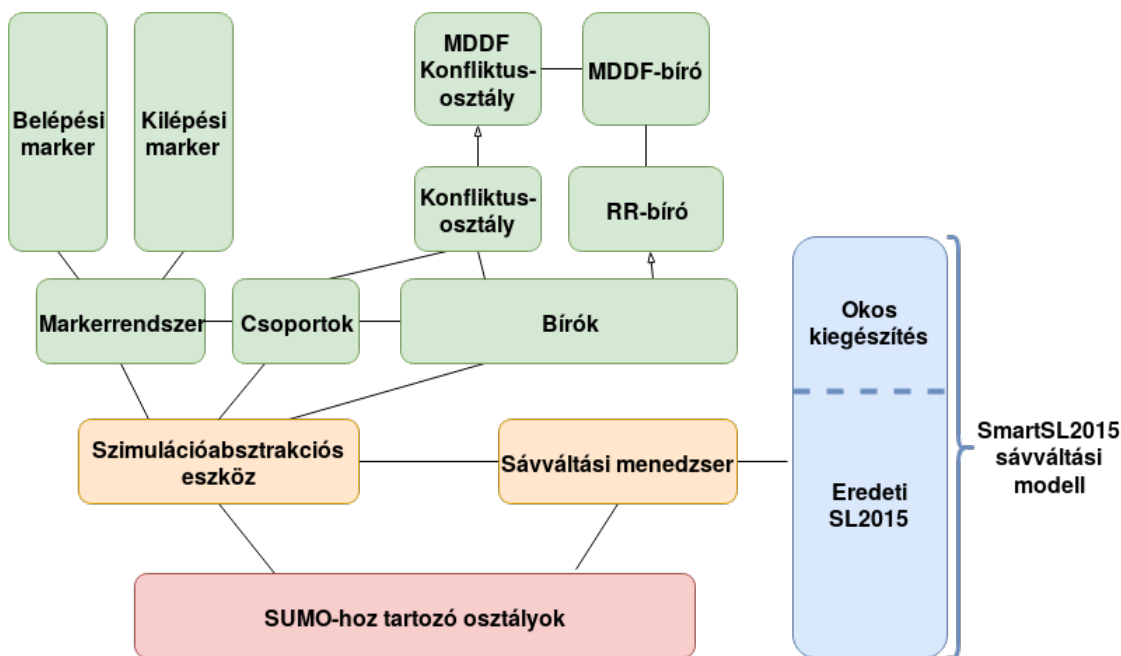
- [13] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, „Coordination of Cooperative Autonomous Vehicles: Toward safer and more efficient road transportation,” *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 74–84, Nov. 2016.
- [14] V. Jacobson et al., „Recommendations on Queue Management and Congestion Avoidance in the Internet.” [Online]. Elérhető: <https://tools.ietf.org/html/rfc2309>. [Elérve: 2019. okt. 25.].
- [15] L. Katkó, „Közúti közlekedési automatika,” *Budapesti Műszaki és Gazdaságtudományi Egyetem*, vol. 2, 2008.
- [16] A. Kesting, M. Treiber and D. Helbing, „Agents for Traffic Simulation”, Ch 11 (pp. 325-356) in „Multi-Agent Systems: Simulation and Applications” ed. by A. Uhrmacher and D. Weyns (2009), arXiv:0805.0300 [physics.soc-ph]
- [17] P. A. Lopez et al., „Microscopic Traffic Simulation using SUMO,” in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2575–2582.
- [18] K. Nagel, P. Wagner, and R. Woesler, „Still Flowing: Approaches to Traffic Flow and Traffic Jam Modeling,” *Operations Research*, vol. 51, no. 5, pp. 681–710, Oct. 2003.
- [19] S. Nigarnajagool and H. Dia, „A Multi-Agent Approach to Real-Time Traffic Signal Optimisation,” 29th Australasian Transport Research Forum, 2006.
- [20] D. I. Robertson, „Research on the TRANSYT and SCOOT Methods of Signal Coordination,” *ITE Journal*, Jan 1986 pp. 37-40.
- [21] S. O.-R. A. V. S. Committee et al., „Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems,” *SAE Standard J3016*, pp. 01–16, 2014.
- [22] T. Tettamanti, A. Mohammadi, H. Asadi, and I. Varga, „A two-level urban traffic control for autonomous vehicles to improve network-wide performance,” *Transportation Research Procedia*, vol. 27, pp. 913–920, 2017.
- [23] M. Treiber and A. Kesting, *Traffic Flow Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [24] B. Vizvári, *Egészértékű programozás*. Typotex Kft, Budapest, 2006.
- [25] P. Wagner, R. Alms, J. Erdmann, and Y.-P. Flötteröd, „Remarks on Traffic Signal Coordination,” in *EPiC Series in Computing*, 2019, vol. 62, pp. 244–255.

- [26] J. C. Williams, H. Mahmassani, and R. Herman, „Urban Traffic Network Flow Models”, *Transportation Research Record*, 1987., pp. 78-88.
- [27] J. Withanawasam and A. Karunananda, „Multi-agent based road traffic control optimization,” in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Yokohama, 2017, pp. 977–981.

A. függelék

A szimulációs keretrendszer

Korábbi munkám, illetve a 2018. évi Tudományos Diákköri Konferenciára beadott dolgozatom elkészítése során kialakítottam egy a Simulation of Urban MObility-n, SUMO-n [17] alapuló kooperatív intelligens szimulációs keretrendszert [1], [2], [3]. A keretrendszer a forgalom résztvevőit többágenses rendszerként (*Multi Agent System, MAS*) kezeli [27], [8], [16]. Az általam hozzáadott komponensek valósítják meg a résztvevő ágensek intelligenciáját és kommunikációs protokolljait. A keretrendszer két ágentípust használ: az *okosautókat* és a kereszteződésekhez kötött intelligens forgalomirányítókat, *bírókat*. A rendszer pontos blokkvázlata az A.1. ábrán látható.



A.1. ábra. A SUMO bővítésével korábban kialakított szimulációs keretrendszerünk blokkvázlata az [1], [3] nyomán. A kék színű komponensek részben a SUMO forráskódbázisából, részben saját fejlesztésű elemekből épülnek fel.

Ez a rendszer a kereszteződések előtt képes *csoportokat* kialakítani az egyforma pályán mozgó járművekből úgy, hogy a csoport élén haladó *csoportvezető* felelős a csoportjáért. Ő kommunikál a bírókkal, ő választja meg a haladási sebességet és dönt a sávváltásokról. A csoport többi tagja, a *csoporttagok* kötelesek az előttük haladó csoportvezetőt vagy csoporttagot követni meghatározott távolságon belül. Ha egy csoport sávot szeretne váltani, akkor a mögöttük haladó csoport ezt lassítással vagy akár megállással is lehetővé teszi.

Az intelligens forgalomszabályozók, a bírók az operációs rendszerek elméletéből kölcsönzött ütemező algoritmusok segítségével döntenek el, hogy kik kaphatnak szabadjelzést. Az egyszerűbb megoldás, a körforgó, *Round-Robin* elven működő bíró. A bonyolultabb az ismert és várakozási időt tekintve optimális *Shortest Job First* (legrövidebb futásidejűt előre) algoritmuson alapuló, ún. MDDF-bíró. Az MDDF a *Minimal Destination Distance First* (minimális hátralevő távolságút előre) rövidítéssel jelzett megoldás lényege, hogy azon járművek kaphatnak először szabadjelzést, akik a legközelebb vannak a céljukhoz [2], [3].

B. függelék

A BAH-csomópont konglomerátumának meghatározása

Futtassuk az 1. algoritmust a BAH-csomópont egyszerűsített helyszínrajzán, és vizsgáljuk meg, hogy az Alkotás utca – Hegyalja utca központi csomópont környezetében melyik másik csomópont van, azaz kikkel alkotnak csomópont-konglomerátumot, a lépéseket a B.1. ábrán szemléltetjük.

Induljunk tehát a már jelzett kereszteződésből, és nézzük meg, hogy a Jagelló és Hegyalja utak kereszteződése benne van-e a BAH-csomópont környezetében. Azt találjuk, hogy igen, mivel mind a két úton érkező járművek, ha érintik a BAH-csomópont eddigi kereszteződéseinek bármelyikét, akkor azt rögtön a jelenleg vizsgált Jagelló út – Hegyalja út kereszteződése után teszik, más útvonal ugyanis nem létezik. Vegyük bele a konglomerátumba ezt a kereszteződést!

Ezután vizsgáljuk meg az Avar utca – Hegyalja utak kereszteződését. Azt találjuk, hogy ez a csomópont viszont nincsen a BAH környezetében, ugyanis létezik olyan útvonal, ami érinti ugyan mindkét kereszteződést, de úgy, hogy a jármű közben elmegey az Avar, majd Alsóhegy utcán át a Villányi útig, és délről érkezik fel a BAH-csomópontba.

A Budaörsi út – Villányi út kereszteződés viszont tagja lesz a BAH konglomerátumának, mert ugyan létezik olyan útvonal, amely a Villányi út, Alsóhegy utca, Hegyalja út felől vezetne vissza a BAH-csomópontba, viszont a Villányi útról nem lehet balra, az Alsóhegy utcára kanyarodni.¹² Így vegyük fel ezt a csomópontot is a BAH konglomerátumába.

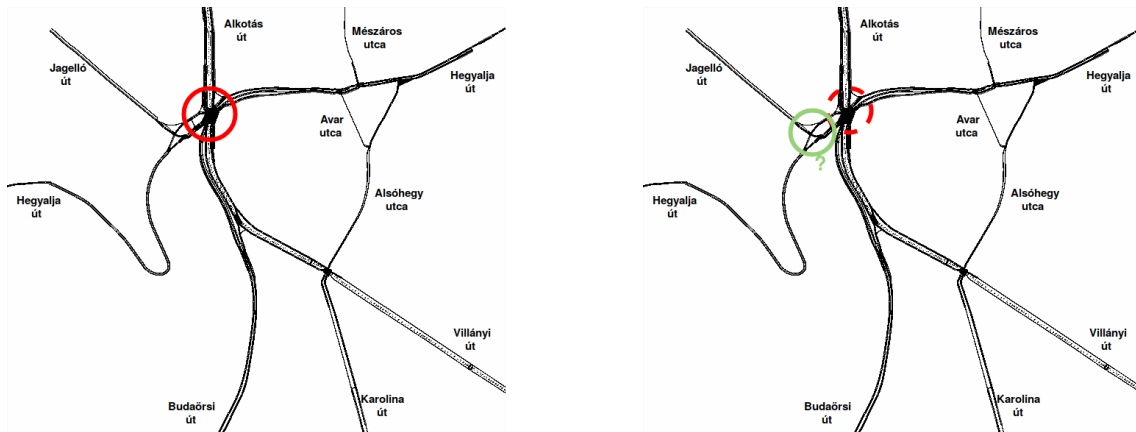
Már csak egyetlen kereszteződés vizsgálata maradt, ez a Villányi út – Alsóhegy

¹Sajnos ezt az ábra nem jelöli.

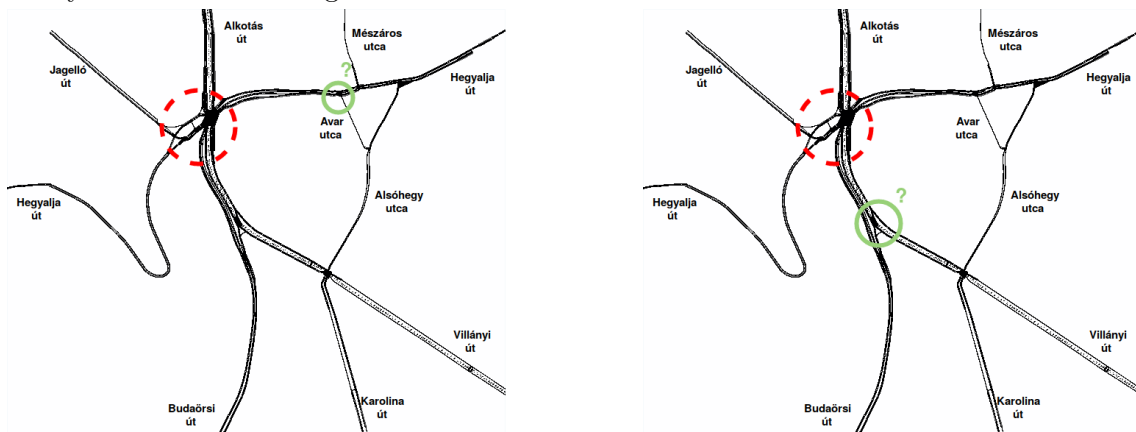
²Persze a valóságban el lehet jutni az Alsóhegy utcára egy visszafordulással a Villányi úton, de most tekintsük úgy, hogy erre nincsen lehetőség. A valóságban feltehetőleg ez nem volna egy túlságosan erős forgalmi igény, ezért úgy véljük, jelen dolgozat keretén túlmutat ennek a lehetőségnek a vizsgálata.

utca – Karolina út kereszteződése. Mivel pl. az Alsóhegy utcán északra haladva, majd a Hegyalja úton nyugat felé fordulva eljuthatunk a BAH-csomópontba nem közvetlen úton keresztül is, ez a kereszteződés nem lesz része a BAH konglomerátumának.

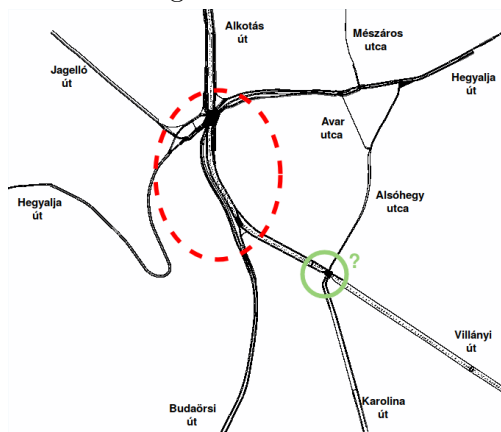
További kereszteződésnek már nincsen kapcsolata a BAH-konglomerátummal, így a Jagelló-Hegyalja, a Hegyalja-Alkotás-Budaörsi és a Villányi-Budaörsi utak kereszteződései alkotják a BAH-csomópont konglomerátumát.



(a) Az Alkotás út – Hegyalja út – Budaörsi út kereszteződésből indítjuk a keresést (*bal oldali ábra*). Megvizsgáljuk a Jagelló és Hegyalja utak kereszteződését (*jobb oldali ábra*), amely hozzávehető a konglomerátumhoz.



(b) Vizsgáljuk az Avar utca – Hegyalja út kereszteződését (*bal oldali ábra*). Ez nem kerül be a konglomerátumba. Megvizsgáljuk a Budaörsi út – Villányi út kereszteződését (*jobb oldali ábra*), amely hozzávehető a konglomerátumhoz.



(c) Vizsgáljuk a Villányi út – Karolina út – Alsóhegy utca kereszteződést. Ez nem kerül be a konglomerátumba. Több csomópont nincsen, az algoritmus leáll.

B.1. ábra. A csomópont-konglomerátumokat kereső algoritmus futtatása a BAH-csomóponton

C. függelék

A jelzésterv-generáláshoz használt egészértékű programozási feladat

A jelzésterv-generáláshoz a (C.1) feltételrendszerrel leírható egészértékű programozási feladatot oldjuk meg [24].

$$\begin{aligned} \mathbf{A}^{N+M} &= \begin{pmatrix} \mathbf{C} \\ \mathbf{F} \end{pmatrix} \\ \mathbf{b}^{N+M} &= \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \\ \mathbf{c}^\top &= (1, 1, \dots, 1) \in \mathbb{R}^N \end{aligned} \tag{C.1}$$

max : $\mathbf{c}^\top \mathbf{x}$, az $\mathbf{A}\mathbf{x} = \mathbf{b}$ feltételel mellett

Ahol:

- N : a kereszteződésen átfutó irányok száma
- M : a megadott kiegészítő feltételek száma, $M \leq N$
- \mathbf{C} : $N \times N$ méretű mátrix, a kereszteződés konfliktusmátrixa. $\mathbf{C}_{i,j} = 1$, ha az i . és j . irány metszi egymást, különben 0.
- \mathbf{F} : $M \times N$ méretű mátrix, az egyes irányok aktuális állapotát előíró egyéb feltételek mátrixa. Minden sorban legfeljebb egy darab 1-es van, $\mathbf{F}_{i,i} = 1$, ha az i . iránynak éppen szabadjelzést kell adni, különben 0-k vannak a táblázatban.
- \mathbf{x} : $1 \times N$ vektor, az irányok vektora. Az i . koordinátája 1, ha az i . irány szabadjelzést kaphat, különben 0.

- $\mathbf{b} : 1 \times (N + M)$ segédvektor, biztosítja, hogy a konfliktusmátrix kizárásai és a feltételek teljesüljenek, kihasználva az egyenlőséget és azt, hogy egy egészértékű programozási feladatot kell megoldani.
- $\mathbf{c} : 1 \times N$ segédvektor. Biztosítja, hogy a lehető legtöbb irány kaphasson egyszerre szabadjelzést.