

Megerősítéses Tanulás alkalmazása autópálya-forgalomirányításban véletlenszerű anomáliák kezelésére

Szerző:
Mitrenga Márk



BME

Közlekedésmérnöki és Járműmérnöki Kar
Közlekedés- és Járműirányítási Tanszék

Konzulens:
Kővári Bálint

TDK dolgozat

Tartalomjegyzék

1. Absztrakt	2
2. Bevezetés	4
2.1. Probléma feltárása, motiváció	4
2.2. Kapcsolódó kutatások	7
2.3. Hozzáadott érték	10
3. Algoritmusok	12
3.1. Integrált Forgalomirányító Algoritmusok	13
3.1.1. Autópálya-ellenőrző rendszer	13
3.1.2. Főáramú forgalomáramlás-szabályozás	13
3.2. Gépi Tanulás	14
3.3. Megerősítéses Tanulás	15
3.3.1. State	17
3.3.2. Action	17
3.3.3. Reward	17
3.4. Multi Ágens Megerősítéses Tanulás	18
3.5. Mélytanulás	19
3.6. Problémaspecifikus implementáció	23
3.6.1. State	23
3.6.2. Action	24
3.6.3. Reward	25
4. Környezet	26
4.1. SUMO környezet	28
5. Eredmények	32
6. Konklúzió	38

1. fejezet

Absztrakt

Az autópályák kapacitás csökkenése könnyen nagy forgalmi dugók kialakulásához vezethet. Ezek a torlódások nagyban hozzájárulnak a környezet szennyezéséhez, valamint a városközeli környezetben a kialakuló kémiai és fizikai szennyezések, nagyban károsítják a település élhetőségét és első sorban az emberek egészségét. TDK dolgozatomban a sávmegszűnéses problémakörben fellelhető számtalan nehézség közül az autópályákon megszűnő sávok miatt kialakuló torlódásokkal foglalkozom. Ennek a problémának a leküzdésére az úgynevezett változtatható sebességkorlátozású szabályozás nyújt segítséget, ami egy forgalomirányító rendszer, amely javítja az általános forgalomáramlási jellemzőket - az utazási idő, a várakozási idő és a sorok hossza - és csökkenti a kritikus fenntarthatósági mutatókat, mint például az üzemanyag-fogyasztás, valamint a CO₂- és NO_x-kibocsátás. Többször bebizonyosodott, hogy a Mélytanulás használata kiváló megoldás erre a problémára. Ebből kifolyólag kutatásom célja Megerősítéses Tanulás alkalmazásával kezelni a forgalomirányító rendszert és egy általános megoldást találni az autópályák kapacitás csökkenése miatt kialakuló torlódások kezelésére, annak érdekében, hogy a sávok számától függetlenül alkalmazható legyen a modell, javítva és felülmúlva ezzel a szakirodalomban eddig elért eredményeket több szempontból is.

Köszönetnyilvánítás

TDK dolgozatom megvalósulásáért köszönetet mondok kunzulensemnek, Kővári Bálintnak a témaválasztásért és a kutatás során nyújtott segítségéért.

A Megerősítéses Tanulás alkalmazása autópálya-forgalomirányításban projekt nevében köszönetet mondunk az HUN-REN Cloud (lásd: Héder et al. 2022; <https://science-cloud.hu/> [1]) használatáért, ami hozzájárult a publikált eredmények eléréséhez.



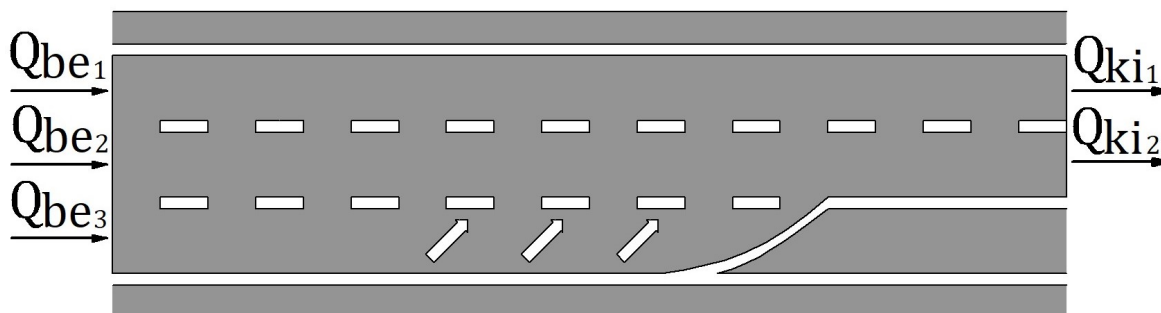
2. fejezet

Bevezetés

2.1. Probléma feltárása, motiváció

Az autópályák esetében gyakran előfordulnak olyan helyzetek, különösen a csúcs-időkben amikor a legsűrűbb a járműforgalom, hogy a forgalom áramlásának hatékony szabályozása nagy kihívást jelent, és ez a járművek problémamentes közlekedésének megzavarásához vezet. A járművek áramlására a folyadékok dinamikájának alapelveihez hasonló fizikai törvények igazak. Az egyik ilyen alapelv a folytonossági tétel, amely azt mondja ki, hogy a folyadék áramlási sebessége és a csővezeték keresztmetszetének szorzata állandó, ami nem más, mint a térfogatáram. Tehát ha a csővezeték átmérője csökken, akkor a folyadék nagyobb sebességgel mozog, míg, ha a csőátmérő növekszik, akkor a folyadék mozgása lelassul.

Jelen esetben a közlekedésben a [2]-ben leírtak alapján a következő képpen értelmezhető a folytonosság törvénye: a vizsgált szakasz bemenő és kimenő keresztmetszeteit a sávok száma adja, az áramló közeget a járművek mozgása reprezentálja. Ezek alapján a kijelölt útszakasz bemeneténél, valamint kimeneténél jelen lévő sávok aránya meg kell egyezzen a járművek bemenő és kimenő összesített sebességek arányával. Így teljesül a kontinuitás törvénye azaz az áramlás állandósága, ezt mutatja be a 2.1 ábra.



2.1. ábra. A folytonosság törvénye a járművek áramlásában

A folytonosság elvét matematikailag a következő egyenlettel 2.1 határozhatjuk meg,

$$\sum_{i=1}^n Q_{be_i} = \sum_{j=1}^m Q_{ki_j} \quad (2.1)$$

ahol n a következő útszakaszon lévő sávok számát, m pedig az aktuális cellában lévő forgalmi sávok számát jelöli.

Azonban a folytonosság törvénye magába foglalja azt is, hogy a csővezeték tömegárama állandó, amint azt a 2.2 egyenlet is leírja,

$$\dot{m}_{be} = \dot{m}_{ki} = \text{állandó} \quad (2.2)$$

ahol $\dot{m}$ a tömegáram.

A 2.3 egyenlet alapján pedig belátható, hogy a folyadék áramlása nem csak a sebesség és az adott cső vastagságának a függvénye, hanem a sűrűség is befolyásolja azt.

$$\dot{m} = v \cdot A \cdot \rho \quad (2.3)$$

Ebben az esetben, az egyenletben szereplő változók közül a v sebesség egyértelműen a járművek sebessége, az A keresztmetszet a már korábbiakban leírtak szerint jelenik meg, továbbá a ρ sűrűség a járművek számát írja le a vizsgált szakaszon.

A közlekedésben a járművek átáramlása nagyban függ a forgalom sűrűségétől. Elérve a kritikus pontot a járművek sűrűsödésével az áramlás csökkeni kezd. Jól reprezentálja a csúcspontot a 2.2 ábrán látható fundamentális diagram, melynek x tengelyén a sűrűség látható, y tengelyén pedig az áramlás mértéke. Tehát ebből adódóan belátható, hogy maga a sűrűség is fontos szerepet játszik az áramlásban, ezért a torlódások elkerülése érdekében elengedhetetlen, figyelembe venni a járművek sűrűségét és ügyelni arra, hogy az ne lépje túl a kritikus pontot. Ezzel együtt azonban ügyelni kell arra, hogy a kritikus ponton vagy a lehető legkisebb szórással a körül tudjuk tartani a sűrűséget. Annak érdekében, hogy ez teljesüljön és a forgalom áramlása folyamatos legyen és ne alakuljanak ki torlódások, a sebesség, mint egyetlen változtatható komponens megfelelő modifikálása kulcsfontosságú.

A szűk keresztmetszetek, amelyek olyan tényezők miatt alakulhatnak ki, mint például az építkezések, felújítások, valamint balesetek miatt lezárt sávok, képesek a közlekedési hálózatot a fizikai határai közé szorítani. Ilyen esetekben a reakcióidő okozta további késések miatt már egy kisebb korrigálás is jelentős ingadozásokhoz vezethet az egyes járművek sebességprofiljában. Ez a hatás különösen fokozódik autópályás környezetben a városi közlekedési helyzetekhez képest, mivel a magasabb alapértelmezett maximális sebesség miatt a járművek mozgásához választható sebességek szélesebb tar-

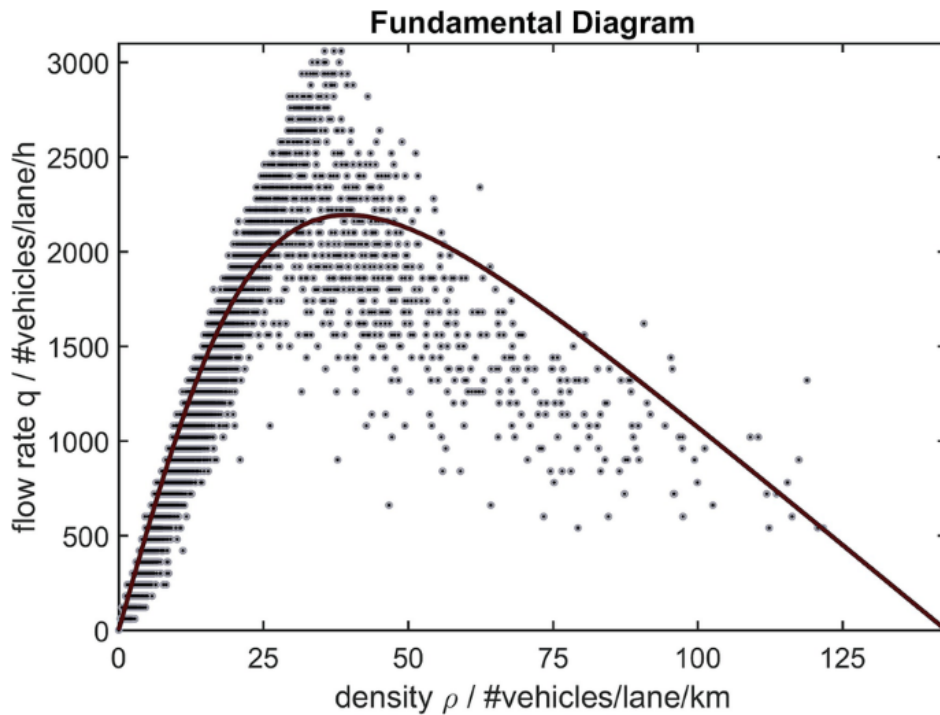
tománya áll rendelkezésre. Egyetlen jármű reakcióideje a következő képlettel 2.4 írható le:

$$\tau = \sum_{i=1}^n \tau_i \quad (2.4)$$

ahol τ a tényleges jármű reakcióidejét, n pedig a vizsgált járműre ható oszlop méretét jelöli.

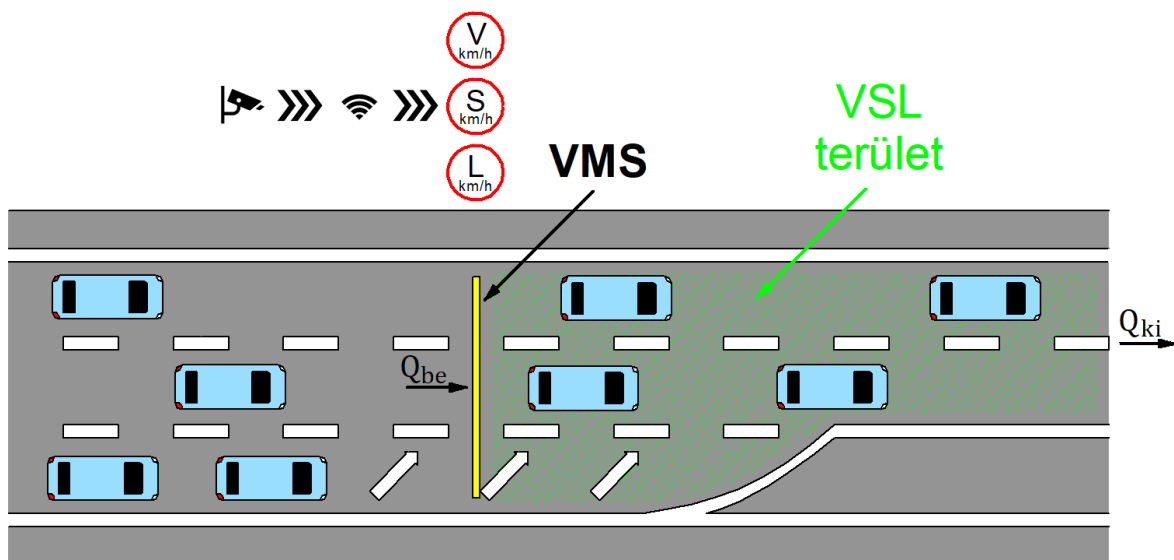
Fontos azonban felismerni, hogy az emberi fizikai korlátok, mint például a reakcióidő a járművekre - mint a forgalom áramlásának közegére - közvetve hatnak, és a környező térbeli objektumokra is kiterjednek. A fent említett reakcióidők miatt a járművek sebességét a balesetmentes körülményeket biztosító maximálisan elérhető sebességhez kell igazítani. Tehát mivel a sebességet nem lehet növelni - Magyarországon 130km/h fölé - az emberek biztonsága érdekében, így azt csökkenteni kell. A lassulást csak kis számú járműre indokolt korlátozni, így lehetővé téve a helyi feloldást a követési távolság biztonságos kereteken belül történő csökkentésével. Ennek a megoldása lehet az, ha a szakaszt több kisebb szegmensre osztjuk fel, a sebességváltoztatások megvalósítása érdekében.

Napjainkban a mesterséges intelligencia széles körben való alkalmazásának köszönhetően, az a forgalomirányításban is megjelent, így létrehozva egy biztonságosabb és kényelmesebb közlekedési környezetet. Egy ilyen rendkívül hasznos forgalomirányító rendszer a változtatható sebességkorlátozású szabályozás - Variable Speed Limit



2.2. ábra. Fundamentális diagram a járművek áramlása és sűrűsége függvényében [3]

Control (VSLC) - amely az autópályák kapacitás csökkenése okán kialakuló torlódások problémájának megszüntetése érdekében kiválóan használható. A VSLC rendszer a forgalom függvényében dinamikusan változtatja a megengedett maximális sebességet a szegmensek optimalizálására, figyelembe véve mind a statikus, mind a dinamikus objektumokat és azok környezeti hatásait a forgalomáramlásra [4], így segítve például az átsorolást a megszűnő sávból, vagy a besorolást az autópálya felhajtóknál. A cél elérése érdekében tehát szükséges még, hogy mind a sebesség, mind a járművek sűrűsége pontosan meghatározható legyen. A rendszer működését az 2.3 ábra mutatja be, amelyen látható a Variable Message Sign (VMS), ami a VSLC által vezérelt változtatható közlekedési jelzőtábla. A VSLC a vizsgált szakasz kimenő forgalmát veszi figyelembe és ennek alapján határozza meg a torlódás mentes áthaladás érdekében optimális megengedett legnagyobb sebességet a szakasz kezdeténél. Fontos kiemelni, hogy ennek a megoldásnak a célja nem a már kialakult forgalmi dugók gyorsabb megszüntetése, hanem a torlódások kialakulásának megelőzése.



2.3. ábra. VSLC működése autópálya kapacitás csökkenés esetén

Az algoritmus úgy működik, hogy bizonyos útszakaszok sávjait önállóan szabályozható sebességkorlátozással ruház fel, amelynek célja az aktív szakasz kiáramlásának befolyásolása. A következő cellába történő beáramlás meghatározását az [5]-ben is tárgyalt folytonossági törvény szabályozza.

2.2. Kapcsolódó kutatások

Az autópályákon gyakori probléma az, hogy csúcsidőben a kapacitás csökkenéssel rendelkező szakaszok közelében lévő területeken időszakos torlódások alakulnak ki.

Az ilyen forgatókönyv elkerülése érdekében lehetőség van az autópálya kapacitásának növelésére további forgalmi sávok építésével. Azonban költséghatékonyság és környezetvédelmi szempontból is elfogadhatóbb megoldás az intelligens közlekedési rendszerek - Intelligent Transportation Systems (ITS) - területéről származó szolgáltatások alkalmazása a forgalomirányításhoz.

A sejtátviteli forgalomáramlási modell - cell transmission traffic flow model (CTM) - jelentős érdeklődést váltott ki a közlekedés területén, köszönhetően egyszerűségének és a makroszkopikus forgalomáramlási jellemzők többségének megragadására való képességének. A CTM stabilitási tulajdonságai különböző keresleti és kapacitáskorlátozások mellett nem mindig egyértelműek. Ezen túlmenően a mikroszkopikus jelenségek, mint például a kapacitáscsökkenéshez vezető sávváltások hatását a szűk keresztmetszeteknél a CTM nem ragadja meg, ahogyan azt a [6]-ben is leírják.

Ezen problémák kiküszöbölésére a változtatható sebességhatárolású szabályozás (VSLC) remek megoldást nyújt. Ahogyan az már korábban is említésre került, ez az autópálya forgalmát úgy szabályozza, hogy a sebességhatárolást az aktuális forgalmi helyzetnek megfelelően változtatja. A cél az autópálya közlekedésbiztonságának növelése, a járműkibocsátás csökkentése és a forgalomáramlás homogenizálása. A szabványos VSLC-megközelítések online visszacsatolásos szabályozáson és a sebességértékekre leképezett alapvető áramlás-sűrűség összefüggésen alapulnak [7].

Sok cikk foglalkozik a VSLC rendszer feljavításával mint például a [8], ahol összekapcsolt járművek - Connected Vehicles (CV) - segítségével értek el jobb eredményeket. Egy másik tanulmány [9] a VSLC technológia azon kihívásával foglalkozik, miszerint bár makroszkopikus elemzésekben jelentős előnyöket mutat, a mikroszkopikus szimulációkban, különösen a forgalmi mobilitás szempontjából, csak csekély javulásokat eredményez. Az utazási idő javulásának hiánya számos baleseti esetben a szűk keresztmetszet közelében bekövetkező sávváltásoknak tudható be, amelyek súlyos kapacitáscsökkenéshez vezetnek, amit a legtöbb makroszkopikus modell nem képes megfelelően megragadni. Ennek a megoldására egy kombinált sávváltás- és VSL-szabályozási rendszert dolgoztak ki, amely mind a makroszkopikus, mind a mikroszkopikus modellekkel konzisztens javulást eredményez. A VSL-szabályozót a CTM-en alapuló visszacsatolt linearizációs megközelítéssel fejlesztették ki, és analitikusan kimutatták, hogy garantálja az exponenciális konvergenciát az optimális egyensúlyi ponthoz. Mikroszkopikus Monte Carlo-szimulációival demonstrálták, hogy ez a kombinált szabályozási stratégia képes konzisztens javulást generálni az utazási idő, a biztonság és a környezeti hatások tekintetében különböző forgalmi feltételek és eseményforgatókönyvek mellett.

A [10] cikk szerzői a VSLC-vel kapcsolatos egyik fő problémának megoldásával foglalkoznak, ami nem más, mint a járművezetők engedetlensége a feltüntetett sebességhatá-

látozásokkal szemben. Ez a probléma áthidalható a kooperatív megközelítésen alapuló alkalmazások fejlesztésével, amely lehetővé teszi az információcserét a forgalomirányító infrastruktúra és a fedélzeti egységekkel felszerelt járművek között. Az intelligens sebességalkalmazkodást a kooperatív technológiák alapján fejlesztették ki, amelynek fő célja, hogy javaslatot, segítséget vagy teljes ellenőrzést nyújtson az érvényes sebesség alatti jármű sebességére vonatkozóan. Ez a sebességkorlátozás az autópálya aktuális szakaszára érvényes, és az erről szóló információkat a közúti egységek továbbítják a fedélzeti egységgel felszerelt járművekhez. Ezzel egyidejűleg a járművek adatokat küldenek a közeli forgalmi környezetükből az út menti egységnek, amely ezeket az adatokat felhasználva pontos képet alkot az adott autópálya-szakasz aktuális forgalmi helyzetéről.

Azonban mivel a rendszer egy adott helyzethez/szituációhoz mérten választja meg a megfelelő sebességet, ebből adódik, hogy a Gépi Tanulási - Machine Learning (ML) - technikák, különösen a Megerősítéses Tanulási - Reinforcement Learning (RL) - módszerek ígéretes alternatívát jelentenek a VSLC beállításához - ahogyan azt a [11] [12] [13] cikkekben is írják - mivel az autópálya dinamikájának explicit modelljének ismerete nélkül képesek megtanulni és reagálni a különböző forgalmi helyzetekre. Tehát RL alkalmazásával, elkerüljük a bonyolult matematikai modellek felépítését, az ágens tanulási képességével. A [14] szerzői szintén RL-t használnak több ágenssel, és a Q-tanulási algoritmust alkalmazzák. A modell bemeneti változóként a főútvonalon a forgalmi áramlásokat és a forgalmi kihasználtságot, valamint az autópálya-behajtók forgalmi igényét vette. A kimeneti változók a sávban mért utazási sebességre, a teljes utazási időre és a teljes megállási időre vonatkoztak. A vezérlési policy a rendszer inherens jellemzőinek változására adott válaszként változott. Ismét beigazolódott, hogy a stratégia valóban adaptív és valós időben reagál a vizsgált szakasz forgalmi igényeire.

Könnyen belátható, hogy a bemenetként kapott aktuális helyzet a környezet által meghatározott state-nek, míg a kimenetként értelmezett sebességkorlát az action-nek tudható be, amelyek az RL fő alkotó részei, éppen ezért adódik a lehetőség, hogy RL-t alkalmazzunk ennek a problémának a megoldására. Ez önmagában csak lehetőséget kínál, viszont a metodológia szükségességét az bizonyítja, hogy tulajdonságai közé tartozik többek közt a skálázhatóság és az általánosító képesség, amelyek által kiemelkedő teljesítményt képes nyújtani az összetett szekvenciális döntéshozatali problémák területén, valamint képes az ágens generalizálni, az állapotot újra értékelni és azt követően gyorsan döntést hozni, így jól kihasználhatók az RL tulajdonságai.

Éppen ebből adódóan már többször bebizonyosodott, hogy egy Deep Q-Network (DQN) használata megfelelő megoldást nyújt ennek a problémának a megoldásához, amint azt a [15] is bizonyítja. Egy további cikk [16] szintén alátámasztja ezt az állí-

tást, mivel a VSLC, mint bármely típusú autópályán alkalmazható forgalomszabályozási módszere, valamint a Q-learningre, mint a tanuláson alapuló VSLC-algoritmusok tervezésének egyik általánosan használt módszerére összpontosít. Ez a tanulmány egy mély Q-hálózatot (DQN) vezet be a Q-függvény közelítésére, és egy új tanulási megközelítést mutat be a VSLC alkalmazáshoz, amely lehetővé teszi a járművek mikroszkopikus szintű követését. A javasolt jutalomfüggvény a tanulást a jutalom javítására és az egymást követő sebességhatárok közötti oszcilláció megakadályozására irányítja.

Azonban ahhoz, hogy a DQN felhasználható legyen be kell tanítani az ágenszt, amely időigényes és költséges feladat [17]. Valamint fontos azt is megemlíteni, hogy a megfelelő tanításhoz meg kell találni az adott problémához és környezethez tartozó optimális reward függvényt, aminek segítségével az ágens el tudja érni a kellő robusztussági szintet.

A [18] tanulmány célja a mély megerősítéses tanuláson alapuló, VSLC átvihetőségének javítás. A Double Deep Q-Network (DDQN) alapú VSLC stratégiát javasolják az autópályákon eltöltött teljes idő csökkentésére. A szimulációban egy valódi összefutó szűk keresztmetszet került kialakításra. Az elért eredmények azt mutatták, hogy a neurális hálózatok frissítésével a DDQN-alapú VSLC ágensben a transzfer-tanulás sikeresen átviszi az adott forrásforgatókönyvben tanult tudást más célforgatókönyvekre. Ezzel a módszerrel a tanítási idő nagyban csökkenthető, miközben a maximális jutalomszint hasonló marad, mint a teljes, nulláról való tanulással működő VSLC esetében. Azonban egy probléma ezzel a módszerrel az, hogy ha a feladat hasonlósága a forrásforgatókönyv és a célforgatókönyv között viszonylag alacsony, a transzfer tanulás helyi optimumhoz vezethet, és nem biztos, hogy eléri a globális optimális szabályozási hatást.

2.3. Hozzáadott érték

A kutatás célja egy olyan általános megoldást találni, amely az autópályák kapacitásának csökkenése esetén kialakuló torlódások csökkentésére, megszüntetésére, valamint elsődleges célként a megelőzésére irányul. Ennek érdekében egy Gépi Tanuláson alapuló megközelítést használva, a Megerősítéses Tanulás (Reinforcement Learning) nyújtotta lehetőségeket alkalmazom, multiágenses rendszerben - Multi Agent Reinforcement Learning (MARL) - amely hatékony eszköz a forgalmi helyzetek valós időben történő kezelésére.

Az RL alapú megközelítés lehetővé teszi az adaptációt különböző környezeti feltételekhez, amelyet más statikus modellek, mint például a CTM, nem képesek megvalósítani. Ezen felül az algoritmus képes arra, hogy a közlekedési helyzetekre vonatkozó

adatokat valós időben értékelje és tanuljon belőlük, anélkül, hogy előre meghatározott matematikai modellekre támaszkodna. Az RL előnye továbbá, hogy új forgalmi helyzetekhez gyorsan tud alkalmazkodni, így biztosítva a folyamatosan változó forgalmi viszonyokhoz való gyors reakciót.

A kutatás középpontjában a különböző kapacitású – sávszámmal rendelkező – autópályák közötti különbségek kezelése áll, és célja egy olyan forgalomirányítási rendszer fejlesztése, amely képes dinamikusan alkalmazkodni a változó forgalmi viszonyokhoz, minimalizálva a kapacitáscsökkenések és torlódások okozta negatív hatásokat, mindezt úgy, hogy a módszer univerzálisan felhasználható legyen.

Annak érdekében, hogy a hasonló, de mégis eltérő körülmények, mint például a sávok száma, esetén ne kelljen minden alkalommal új megoldásokat keresni, egy egyszerűsített megközelítés kidolgozása lenne a legmegfelelőbb, amely hatékonyan kezeli a problémát. Ebben az esetben nem szükséges módosítani az állapotrepresentációt, a beavatkozási téren vagy a jutalmazási függvényen.

A javasolt rendszer skálázható nagyobb közlekedési hálózatokra, valamint képes az általánosításra is, így alkalmas lehet különböző autópálya-szakaszokon történő alkalmazásra. A kutatás innovatív jellege abban rejlik, hogy integrálja a legmodernebb megerősítéses tanulási technikákat a forgalomirányításba, egy skálázható és adaptív, általánosítható megoldást kínálva ezzel, amely képes dinamikusan kezelni a torlódásokat és javítani az autópályák kapacitáskihasználtságát.

Annak érdekében, hogy a kifejlesztett eljárás általánosíthatósága megfelelően ellenőrizhető legyen az eredmények alapján, a betanult modellt három különböző környezetben fogom tesztelni, amelyek eltérő sávszámmal rendelkező, sávmegszűnéses autópálya-szakaszok. A három vizsgált környezet között 3, 5, és 7 sávós autópályák szerepelnek, lehetővé téve a módszer teljesítményének széleskörű kiértékelését.

A vizsgált környezetek szimulálására a Simulation of Urban MObility (SUMO) szimulációs rendszer nyújt kiváló megoldást. A SUMO könnyedén testre szabható a szimulálni kívánt helyzetek modellezéséhez, valamint kompatibilis a különböző algoritmusok tesztelésére. Ennek köszönhetően a SUMO lehetőséget biztosít arra, hogy a kifejlesztett megoldás hatékonyságát és általánosíthatóságát különböző forgalmi környezetekben is érvényesíteni lehessen, így hozzájárulva a forgalomirányítás és a torlódások megelőzése terén elért új eredményekhez.

3. fejezet

Algoritmusok

Az algoritmusok kulcsfontosságú szerepet játszanak a problémamegoldásban, és különböző típusokba sorolhatók, attól függően, hogy milyen módszereket és technikákat alkalmaznak. Három fő csoportot különböztethetünk meg:

- A Megerősítéses Tanulás (Reinforcement Learning - RL) alapú algoritmusok neurális hálózatokat használnak függvény-approximátorként. Ezek az algoritmusok a saját maguk által generált tanító minták feldolgozásával működnek, és a tanulási folyamat során finomhangolják a neurális hálózatokat az optimális döntések érdekében.
- A fa struktúrájú kereső algoritmusok jövőbeli kimeneteket szimulálnak és értékelnek az optimális döntések meghozatalához. A döntéshozatali fa mélyül és szélesedik új lépések és állapotok figyelembevételével.
- Az egyszerű aktuált vezérlőket, illetve a fix ciklusidejű vezérlést tartalmazó algoritmusok egyszerű szabályokon alapulnak. Például a SUMO szimulációs szoftver ilyen algoritmusokat használ a forgalom szabályozására.

Az algoritmusok tehát számos különböző formában jelennek meg, mindegyikük rendelkezik sajátos előnyökkel és specifikus alkalmazási területekkel. A megerősítéses tanuláson alapuló algoritmusok például a neurális hálózatok hatalmas számítási kapacitását és tanulási képességeit hasznosítják, így kiválóan alkalmasak bonyolult, dinamikusan változó környezetekben történő döntéshozatalra. Ezzel szemben a fa struktúrájú kereső algoritmusok az időbeli előre tekintésre építenek, ami különösen hatékony, ha nagy mennyiségű lehetséges állapot vagy lépés között kell megtalálni az optimális megoldást. Az egyszerű aktuált vezérlők és a fix ciklusidejű vezérlés rendszerei viszont gyors és hatékony megoldást nyújtanak a gyakorlati problémák széles skáláján. Ezek az algoritmusok tehát különböző helyzetekben kínálnak optimális megoldásokat, a konkrét feladat követelményeitől függően.

3.1. Integrált Forgalmirányító Algoritmusok

3.1.1. Autópálya-ellenőrző rendszer

Az autópálya-ellenőrző rendszer - Motorway Control System (MCS) - egy szabályalapú VSL-algoritmus, amely előre meghatározott sebességhatározási értékek közül - 110, 90 és 70 km/h - az adott körülményekhez mérten választja ki a megfelelőt a forgalom folytonos áramlása érdekében. A forgalmi viszonyokat egyszerű érzékelők segítségével kapja meg a rendszer, majd a 3.1 egyenletnek megfelelően állítja be a megengedett maximális sebességet a szegmenseken,

$$sebességhatár = \begin{cases} 130 \text{ km/h} & \text{ha } \text{átlagsebesség} \geq 40 \text{ km/h} \\ [110; 90; 70] \text{ km/h} & \text{ha } \text{átlagsebesség} < 40 \text{ km/h} \end{cases} \quad (3.1)$$

ahol, ha a vizsgált szegmens átlagsebessége meghaladja a 40 km/h-t, a maximális sebesség 130 km/h-ra lesz állítva, ellenkező esetben, három fokozatos sebességhatározást alkalmaz.

A rendszer alap felépítése szerint a leglassabb sáv, azaz amelyiknek az átlagsebessége a legalacsonyabb, határozza meg az összes sáv sebességhatározását, szegmensenként. Az algoritmus fokozatosan, szakaszonként halad végig az autópályán, kezdve az utolsó érzékelő által mért területtel. Ez azt biztosítja, hogy a sebességhatározások egymással összehangoltak legyenek, és az egész útszakaszon egyenletesen működjenek. Ezzel minimalizálják a sebességkülönbségeket és elősegítik a forgalom zökkenőmentes áramlását, miközben javítják a közlekedésbiztonságot és a hatékonyságot.

3.1.2. Főáramú forgalomáramlás-szabályozás

A modellalapú főáramú forgalomáramlás-szabályozás - Mainstream Traffic Flow Control (MTFC) - algoritmus [19] az összehasonlításként használt Integrált Forgalmirányító Algoritmusok közül a kiemelkedőbb eredményekkel rendelkezik. A sebességhatárok optimalizálásával az algoritmus célja, hogy szabályozza a forgalom áramlását a szűk keresztmetszeteknél, így biztosítva a területek hatékony kihasználását. Ez segít fenntartani az optimális forgalmi feltételeket, megakadályozva a torlódásokat és biztosítva a zökkenőmentes közlekedést. Ez az algoritmus a változtatható sebességhatározások értékét a t időpontban, az eredeti sebességhatár hányadosaként határozza meg a 3.2 egyenlet szerint,

$$v(t) = v(t-1) + K'_I \cdot e_0(t) \quad (3.2)$$

ahol $v(t)$ és $v(t-1)$ az új és az előző sebességhatárok, K_I' az integrálerősítést jelöli, $e_0(t)$ pedig a foglaltsági hiba, amelyet a kritikus $\hat{o}_{out}$ foglaltság és a szűk keresztmetszetenél mért $\tilde{o}_{out}$ foglaltság közötti különbségként határozzunk meg.

Az algoritmus paraméterei a megalapozott vizsgálatokból származnak, és a változó sebességhatárok $[20km/h; 130km/h]$ intervallum határain belül változnak. A rendszer négy, a szűk keresztmetszet körül elhelyezett érzékelőt integrál, és az ezekből az érzékelőkből származó maximális foglaltsági mérést használja fel. Ezután a kiszámított sebességhatárokat a 300m-es szakaszokra alkalmazzák.

3.2. Gépi Tanulás

A Gépi Tanulási technológia (ML – Machine Learning) a modern társadalom számos aspektusát hajtja: a webes kereséstől kezdve a közösségi hálózatokon történő tartalom-szűrésen át az e-kereskedelmi weboldalakon megjelenő ajánlásokig, és egyre inkább jelen van az olyan fogyasztói termékekben, mint a fényképezőgépek és az okostelefonok. Az ML rendszereket arra használják, hogy a képeken objektumokat azonosítsanak, a beszédet szöveggé írják át, a híreket, posztokat vagy termékeket a felhasználók érdeklődési körének megfelelővé tegyék, és kiválasszák a keresés releváns eredményeit.

A Gépi Tanulásnak három fajtáját különböztetjük meg. Ezek rendre a Megerősítéses Tanulás - Reinforcement Learning (RL), a Felügyelt Tanulás - Supervised Learning (SL) és a Felügyelet Nélküli Tanulás - Unsupervised Learning (USL).

Az SL meghatározó jellemzője a megjegyzésekkel ellátott tanítóadatok elérhetősége. A név egy „felügyelő” (supervisor) jelenlétére utal, amely arra utasítja a címkéken szereplő tanulási algoritmust, hogy társítsa a tanítási példákhoz. Ezek a címkék általában osztálycímkék az osztályozási problémákban. Az SL algoritmusok modelleket hoznak létre ezekből a tanítóadatokból, és ezek a modellek felhasználhatók más címkézetlen adatok osztályozására, amint azt a [20]-ban is olvashatjuk. Az SL során az ágens a kívánt $y_1, y_2, \dots$ kimenetek sorozatát is megkapja, és az ágens célja, hogy megtanulja a megfelelő kimenetet előállítani egy új bemenettel. Ez a kimenet lehet egy osztálycímké (az osztályozásban) vagy egy valós szám (regresszióban). Például jól használható különböző virágok osztályozására. Egy adott képkészletből bemenetként kapott képről a modell el tudja dönteni, hogy kimenetként melyik csoportba sorolja azt. Ez a Gépi Tanulási modell ebben az esetben az egyes képek tulajdonságait tanulja meg, például a szín, a virág alakja és egyéb képi jellemzők. Az előzetesen címkézett tanítóadatkészlet segítségével a modell képes lesz kategorizálni az ismeretlen virágképeket a megfelelő fajtájuk szerint. Tehát az SL csak előre meghatározott tanítóadatok segítségével offline módon képes tanulni.

A [21] a Felügyelet Nélküli Tanulás működésével foglalkozik. Az USL során az ágens csak $x_1, x_2, \dots$ bemeneteket kap, és sem felügyelt célkimenetekkel, sem jutalmak által nem lép interakcióba a környezettel. Lehetséges azonban egy formális keretrendszer kifejleszteni a Felügyelet Nélküli Tanuláshoz, azon az elgondoláson alapulva, hogy az ágens célja olyan reprezentációkat építeni a bemenetekből, amelyeket döntéshozatalhoz, jövőbeli bemenetek előrejelzéséhez, bemenetek hatékony átviteléhez egy másik gépnek stb. használhat. Bizonyos értelemben a Felügyelet Nélküli Tanulást úgy lehet elképzelni, mint olyan minták megtalálása az adatokban, amelyek a tisztán struktúrálatlan zajon felüliek. A Felügyelet Nélküli Tanulás két nagyon egyszerű klasszikus példája a klaszterezés és a dimenziócsökkentés.

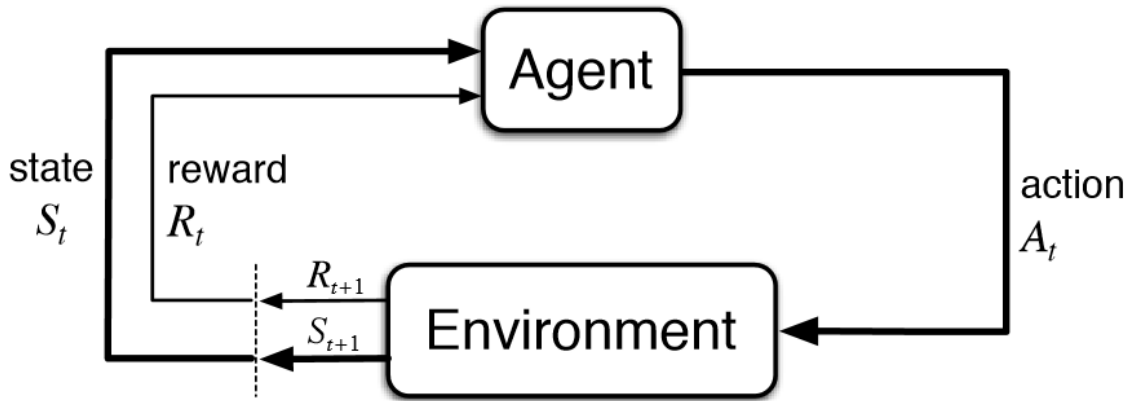
3.3. Megerősítéses Tanulás

A Megerősítéses Tanulás (RL - Reinforcement Learning) rendkívül széles problémakört fed le, mivel képes összetett döntéshozatali problémák megoldására, valamint ez a metodológia igen egyszerűen megérthető felépítésű, így üzemeltetésének a folyamatát könnyű elsajátítani. Ezeknek a tulajdonságoknak köszönhetően számos területen egyre népszerűbb az alkalmazása, amint azt a [22]-ben is leírják. Ezek közé tartoznak például a robotika, az önvezető autók fejlesztése, a közlekedési rendszerek irányítása, vagy akár az egészségügy és még a pénzügyi piacok is.

Az RL működése úgy írható le, hogy az ágens interakcióba lép a környezettel, amely az aktuális állapot alapján különféle beavatkozási lehetőségeket kínál fel. Az ágens által választott beavatkozást követően a környezet numerikus jutalommal (reward) értékeli az ágens döntését, amely lehet „díjazás” vagy „büntetés”, annak függvényében, hogy a reward pozitív vagy negatív értékű. A környezettel folytatott interakció során az ágens $a_1, a_2, \dots$ beavatkozásokat hajt végre, amelyek módosítják az aktuális állapotot, és ennek eredményeképpen az ágens egy skaláris jutalmat $r_1, r_2, \dots$ kap.

A 3.1 ábrán látható, hogy az ágens (agent) a környezettől (environment) megkapja az aktuális állapotát S_t (state), ami alapján egy véletlenszerű beavatkozást A_t (action) visz véghez. Ezt a környezet érzékeli és az ágens új állapotával S_{t+1} együtt most egy numerikus jutalmat R_t (reward) is küld az elvégzett beavatkozás és az állapotátmenet függvényében. A következő lépésben az ágens már a jutalom által meghatározott tapasztalat alapján fogja elvégezni a beavatkozást. És ez - az ágens tanítása - addig folytatódik amíg az ágens el nem éri az adott feladathoz teljesítéséhez szükséges robusztussági szintet.

Az RL diszkrét időlépésekben lép kölcsönhatásba a környezetével, amelyet jellemzően Markov-döntési folyamatként - Markov Decision Process (MDP) - fogalmazznak



3.1. ábra. Megerősítéses Tanulás sematikus ábrája [23]

meg, ami igen hasznos a dinamikus programozásban. Minden egyes időlépésben a folyamat valamilyen s állapotban van, és a döntéshozó - jelen esetben az ágens - bármely a beavatkozást választhat, amely az s állapotban elérhető. A folyamat a következő lépésben úgy reagál, hogy véletlenszerűen új s' állapotba kerül, és ennek megfelelő jutalmat ad a döntéshozónak $R_a(s, s')$. Az MDP négy absztrakciójának (S, A, R_a, P_a) a megfogalmazása, leírja az ágens kontrollproblémáját, és ezek támogatásával az ágens kialakíthatja a kívánt viselkedést.

Ezek az absztrakciók a következők:

- S az állapotok halmaza, amelyet állapottérnek nevezünk.
- A a beavatkozások halmaza, az úgynevezett beavatkozási tér.
- $R_a(s, s')$ az azonnali jutalom (vagy várható azonnali jutalom), amelyet az s állapotból az s' állapotba való átmenet után kapunk az a beavatkozás elvégzése végett.
- $P_a(s, s') = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ annak a valószínűsége, hogy az a beavatkozás s állapotban t időpontban az s' állapotba vezet a $t + 1$ időpontban.

Sajnos ezek megfogalmazása kizárólag a kutatók felelőssége, intuíciója és azon képessége, hogy megértsék a belső dinamikát, és ezt a tudást az absztrakciókra alkalmazzák. Következésképpen a kutatók megérzései a kontrollproblémával kapcsolatban korlátozhatják a végső teljesítményt. Ez egyrészt előnyös, mivel az RL-nél az elérhető teljesítményt nem korlátozza más eszköz, ellentétben például a felügyelt tanulással (SL), ahol a kezdeti teljesítmény korlátozott. Másrészt további létfontosságú kérdéseket vet fel az RL-hez kapcsolódó tudósok számára, például, hogy létezik-e optimális megoldás az egyes feladatokra, és ha igen, akkor hogyan határozható meg.

Az ágens célja, hogy megtanuljon úgy beavatkozni, hogy maximalizálja a jövőbeni jutalmakat (vagy minimalizálja a büntetéseket) a meghatározott időhorizonton.

3.3.1. State

Az állapot (state) a környezet adott pillanatbeli jellemzőinek együttes leírását jelenti a Megerősítéses Tanulás keretében, és azt reprezentálja, hogy az ágens milyen helyzetben van egy adott időpillanatban. Az állapot információkat tartalmazhat a környezet különböző elemeiről, például egy robot helyzetéről vagy egy játékos pontszámáról, és segíti az ágenst abban, hogy az adott feladatnak megfelelően meghatározza saját pozícióját vagy állapotát.

Az állapotok az állapotteren belül helyezkednek el - ugyan úgy mint az MDP-ben - és az állapottérben lévő állapotok közötti változás állapotátmenetek formájában megy végbe. Ezek az átmenetek az ágens döntései és cselekvései alapján mennek végbe, amelyek az adott állapothoz igazodnak, és céljuk a következő lépések optimalizálása. A részletes állapotleírás lehetőséget biztosít a finomhangolásra a tanítási eljárás során.

Az állapotok meghatározzák, hogy az ágens milyen információkat használ fel a döntéshozatalhoz és a környezet változtatásához.

3.3.2. Action

A beavatkozás (action) az ágens által meghozott döntés, amelyet az adott állapot alapján hajt végre annak érdekében, hogy a rábízott feladatot teljesítse vagy a problémát megoldja. Az ágens beavatkozásokkal módosítja a környezet állapotát, és a célja az, hogy megtanulja, mely cselekvések vezetnek a legjobb eredményhez. A lehetséges beavatkozásokat a beavatkozási tér (action space) tartalmazza - akárcsak az MDP-ben - amely meghatározza, hogy az ágens milyen beavatkozásokat hajthat végre az adott állapotban. Azt, hogy egy adott állapothoz képest milyen akciókat választhat az ágens, a környezet határozza meg, és a környezet visszajelzése (reward) függvényében értékeli az ágens, hogy az adott beavatkozás mennyire volt hatékony.

3.3.3. Reward

A jutalom (reward) az a visszajelzés, amelyet az ágens az egyes beavatkozásai után kap a környezettől, és számszerű formában értékeli az adott cselekvés hatékonyságát vagy hasznosságát a kitűzött cél elérése szempontjából. A jutalmazási függvény az RL algoritmus egyik legfontosabb eleme, mivel ez biztosítja a környezet által adott egyetlen skalárértéket, amely alapján az ágens megtudhatja, mennyire volt megfelelő az adott beavatkozás. A pozitív jutalmak a kívánatos cselekvéseket ösztönzik, míg a negatívak a

viselkedés módosítására készítetik az ágenszt. Ez a visszacsatolási mechanizmus nemcsak a beavatkozások minőségét számszerűsíti a feladat és a környezet függvényében, hanem irányítja az ágens tanulási folyamatát is, segítve a stratégiák finomítását és a döntések optimalizálását a kívánt célok elérése érdekében.

3.4. Multi Ágens Megerősítéses Tanulás

A multiágens rendszer autonóm, egymással kölcsönhatásban álló entitások csoportjaként definiálható, amelyek közös környezetet használnak, érzékelőkkel érzékelik azt, és beavatkozókcal hatnak rá.

Bár az előre leprogramozott, probléma-specifikus ágensek jó eredményeket tudnak elérni, nem képesek kezelni a dinamikusan változó környezeti hatásokat. Ennek megoldására a multiágens tanulással kapcsolatos kutatások jelentős része a Megerősítéses Tanulási technikákkal foglalkozik, ami megalapozza a Multi Agent Reinforcement Learning (MARL) területét. A MARL kibővíti a hagyományos, egyetlen ágensre épülő RL paradigmát több ágens bevonásával, amelyek egymással interakcióban tanulnak és döntenek saját céljaik optimalizálása érdekében egy közös környezetben. Ebben a környezetben az egyes ágensekhez tartozó egyéni policy-k összegződnek a közös policy halmazában, amelyet a 3.3 egyenlet szemléltet,

$$\pi = \{\pi_1, \pi_2, \dots, \pi_n\} \quad (3.3)$$

ahol π a közös policy, míg a halmazban lévő minden π_i az ágensekhez tartozó egyéni policy. A MARL keretében minden ágens arra törekszik, hogy saját várható kumulatív jutalmát maximalizálja. A cél az, hogy maximalizáljuk a várható cselekvési érték függvényt, ami egy nagyobb hozamot biztosító policy megtalálásához vezet.

A MARL területe gyorsan bővül, a legtöbb sikeres RL-alkalmazás, például a Go és a póker játékok, a robotika és az autonóm vezetés, valamint a forgalomiányítás több ágens részvételével zajlik, amelyek természetesen a MARL területéhez tartoznak. Az elmúlt évek során számos megközelítést javasoltak az előnyök kihasználására és a kihívások kezelésére [24]. A MARL robusztus megközelítést kínál az ágensek közötti versengő vagy együttműködő stratégiák kialakításához, ezáltal javítva a döntéshozatal összeteljesítményét. A szakirodalomban leírt MARL algoritmusok - explicit vagy implicit módon - e két cél valamelyikét vagy a kettő kombinációját célozzák meg, legyen szó teljesen kooperatív, kompetitív vagy általánosabb környezetekről. Ezek a megközelítések integrálják az egyágenses RL, a játékelmélet és az általánosabb, közvetlen szabálykeresési technikák fejlesztéseit.

3.5. Mélytanulás

A mélytanulás (Deep Learning) a Gépi Tanulás egy speciális ága, amely mesterséges neurális hálózatokra épül, és képes nagy mennyiségű nyers adatot feldolgozni anélkül, hogy előzetesen emberi tervezésre lenne szükség a jellemzők kivonásához. Annak érdekében, hogy olyan rendszereket hozzanak létre, amelyek az emberi tanuláshoz hasonlóan tanulnak, a mélytanulás alapstruktúráját az emberi agy felépítése ihlette. Emiatt a mélytanulás számos alapvető terminológiája visszavezethető a neurológiára. Hasonlóan ahhoz, ahogyan a neuronok alkotják az agy alapvető építőelemeit, a különböző csomópontok között áramlik az információ.

A hagyományos ML technikákkal ellentétben, amelyek csak korlátozottan képesek a nyers adatok kezelésére, a mélytanulás automatikusan felfedezi a feladat megoldásához szükséges reprezentációkat. Amint a [25] folyóiratban megjelent cikkben is leírták, évtizedeken keresztül a mérnökök gondosan tervezett jellemző-kivonatolókat alkottak, hogy a nyers adatokat (pl. képek pixelértékeit) olyan belső reprezentációvá alakítsák, amelyet a gépi tanulási rendszer, például egy osztályozó, képes volt feldolgozni. Ezzel szemben a mélytanulás több szintű reprezentációt használ, ahol az egyes rétegek - úgynevezett rejtett rétegek (hidden layer) - fokozatosan alakítják át a bemenetet egy magasabb szintű jellemzővé, ezáltal a modell képes önállóan absztrakt fogalmakat létrehozni a nyers adatokból.

A mélytanulás egyik alapvető jellemzője, hogy nagyszámú rejtett réteget tartalmaz, amelyek nem előre meghatározottak vagy emberi mérnökök által tervezettek, hanem az adatokból automatikusan tanulják meg a releváns jellemzőket. A neurális hálózatok rejtett rétegeinek száma meghatározatlan, és a rétegeken belüli csomópontok (neuronok) száma sincs előre rögzítve. Ezeket a paramétereket mindig az adott feladathoz kell igazítani az optimális teljesítmény érdekében, figyelembe véve a számítási erőforrások hatékony kihasználását.

A mélytanulásban az alsóbb rétegek egyszerű, de nemlineáris modulokkal dolgoznak, amelyek a nyers bemenetekből kiindulva egyre absztraktabb szintű reprezentációkat hoznak létre. Ez a képesség különösen hasznos az RL-ben, ahol az ágens folyamatosan interakcióba lép a környezetével, és ebből tanul. A mélytanulás lehetővé teszi az RL-algoritmusok számára, hogy komplex állapottereket és döntési stratégiákat kezeljenek, így az ágens képes lesz jobban felismerni a hosszú távú célokat, és hatékonyabban alkalmazkodni a dinamikus környezethez. A mély RL különösen alkalmas olyan feladatokban, ahol az állapot- és beavatkozási tér nagy, és a minták felismerése, illetve az optimális döntéshozatal kritikus szerepet játszik.

A neurális hálózatoknál a rejtett rétegeken kívül még két rétegtípust különböztünk meg. Ezek rendre a bemeneti és kimeneti rétegek. A bemeneti réteg a neurális

hálózat első rétege, ahol az adatok a hálózatba érkeznek. A réteg csomópontjainak száma megegyezik a feldolgozni kívánt adatok számával. Ezek lehetnek a különböző állapotok vagy például a képfeldolgozásnál a pixelek értékei. Az adatok fogadása után, azok továbbításra kerülnek az első rejtett réteghez, majd ennek segítségével szerez tudomást az ágens az adott állapotról. A kimeneti réteg a neurális hálózat utolsó rétege, amely a rejtett rétegekből érkező feldolgozott adatok alapján hozza meg a döntést a megfelelő beavatkozás elvégzéséhez. A réteg az adott környezetben lehetséges beavatkozásoknak megfelelő számú csomópontot tartalmaz. Ha például egy képosztályozó modell két lehetséges kategóriával dolgozik, akkor a kimeneti rétegnek két neuront kell tartalmaznia, amelyek az egyes kategóriákhoz kapcsolódnak. A feladattól függően a kimeneti réteg lehet bináris, több osztályos vagy folyamatos értékű is.

A bemeneti és kimeneti rétegek alapvető szerepet töltenek be, mivel ezek határozzák meg a hálózat interakcióját a külső adatokkal. Ezek a rétegek fontos szerepet játszanak az információ áramlásában az adatok és a válasz között, mivel ezek felelnek az adatok fogadásáért és az eredmény előállításáért.

Az információ továbbítása a bemenettől a kimenetig a rejtett rétegeken keresztül történik, ahol a csomópontok összekapcsolódása biztosítja a folyamatot. Két szomszédos réteg között úgy jön létre a kapcsolat, hogy az egyik réteg minden egyes neuronját összekötjük a másik réteg összes neuronával. Fontos kiemelni, hogy egy adott rétegen belül a neuronok között nincs közvetlen összeköttetés. A csomópontok közötti kapcsolatok súlyokként (W - Weight) definiálhatók, amelyeket súlymátrix formájában írhatunk fel, ez később elemzési előnyt nyújt. A két réteg közötti súlymátrix mérete a rétegek csomópontjainak számától függ, tehát ha az egyik réteg n , a másik pedig m csomóponttal rendelkezik, akkor a súlymátrix mérete $n \times m$ lesz.

A rejtett rétegek és a kimeneti réteg csomópontjainak értékei – amelyek bemenetként szolgálnak a következő réteg számára, vagy kimenetként a kimeneti réteg számára – nemcsak a súlyoktól, hanem az úgynevezett Bias-tényezőktől (B) is függenek. A Bias egy további paraméter, amely hozzáadódik az adott csomópont bemenetéhez, és fontos szerepet játszik a hálózat rugalmasságának növelésében. A Bias értékek, a súlyokhoz hasonlóan, általában kis számok, leggyakrabban 0 és 1 közötti tartományban, azonban ez változhat a felhasznált aktivációs függvényről és inicializálási módszertől függően. A Bias-tényező segít az aktivációs függvények eltolásában, így finomhangolva a hálózat teljesítményét. Utóbbit úgy lehet értelmezni, mint egy súlyt, ami független a többi csomóponttól, így tehát nem változik az értéke elvégezve a backpropagation-t, ami majd később kerül ismertetésre. A súlymátrix és a bemeneti értékek vektorának szorzata - ami egy vektor - a bias értékek vektorával kiegészülve egy összeg vektort alkot, amit egy aktivációs függvénnyel nemlinearizálva kapjuk a kimeneti értékek vektorát. Ez minden

egyes rétegben, minden csomópontnál így működik. Ez az úgynevezett forward pass (előreterjesztés) eljárás, amely által a bemenetből kimenetet kapunk a 3.4 egyenletnek megfelelően.

$$Y = f\left(\sum_{i=0}^n (W_i \cdot X_i) + B_i\right) \quad (3.4)$$

Az egyenletben szereplő W_i a súlymátrixokat jelöli, míg X_i a bementi adatoknak a vektorát. Ezek szorzata egy vektort eredményez, amelyhez hozzáadódik még a B_i bias értékek vektora is, ezáltal létrehozva egy lineáris kapcsolatot a bemenet és a kimenet között. Azonban a lineáris kapcsolat nem megfelelő minden problémára, ezért törekedni kell a nemlineáris kapcsolat elérésére. Ennek a megvalósításában van szerepe az $f()$ aktivációs függvénynek, és ezáltal megkapjuk a - már nem lineáris kapcsolatú - kimeneti vektort Y . Ezt a folyamatot rétegről rétegre haladva végzi el az algoritmus n alkalommal, ami a rétegek számánál eggyel kevesebb, mivel a rétegek közötti súlymátrixok számával megegyező.

Miután értelmeztük a mélytanulás modelljeit - az emberi agy működéséhez hasonlóan működő neurális hálózatok felépítését - megnézhetjük, hogyan is történik pontosan a tanítás. Amint az már korábban említésre került, a neurális hálózatban minden egyes súly megváltoztatható, és így a neurális hálózatnak számtalan konfigurációja lehet, ami kulcsfontosságú a tanítás szempontjából. A már betanított neurális hálózat olyan súlykonfigurációval rendelkezik, amely pontosan megjósolja a helyes kimeneteket bizonyos bemeneti adatokból.

A neurális hálózat a backpropagation (visszaterjesztés) nevű algoritmus használatával, az aktivációs függvények parciális differenciáltjának segítségével változtatja meg a súlyokat, annak érdekében, hogy elérje a kívánt súlykonfigurációt. Tehát nem csak a linearitás elkerülése érdekében, hanem a backpropagation megfelelő működéséhez is többféle aktiválási függvényt alkalmaznak a szakirodalomban. Ezek közül kettő igen gyakran használatos a **Sigmoid** és a **ReLU** függvények.

A **Sigmoid** aktivációs függvény már a 20. század óta gyakran használatos a mély neurális hálózatok nemlinearizálására [26]. A Sigmoid függvény kulcsfontosságú szerepet játszott a backpropagation kialakulása során, mivel deriváltja könnyen kezelhető, és alkalmas volt a folyamatos, nemlineáris összefüggések modellezésére a hálózat rétegei között. Ez alapozta meg a modern mélytanulás elterjedését. Ez a függvény bármilyen valós értékű bemenetet fel tud dolgozni, amelyeket aztán a kimeneti értékeként 0 és 1 közé skáláz. Ezt a műveletet a 3.5 egyenlet szerint definiált módon végzi.

$$y = \frac{1}{1 + e^{-x}} \quad (3.5)$$

Látható, hogy minél nagyobb a bemenet (pozitívabb), annál közelebb lesz a kimeneti

érték 1-hez, míg a kimenet annál közelebb lesz 0-hoz, minél kisebb a bemenet (negatívabb), valamint nullában 0.5 értéket vesz fel.

A **ReLU** (Rectified Linear Unit) egy újabb konstrukció a mélytanulásban [27], ami az egyik leggyakrabban használt aktivációs függvény. A ReLU aktivációs függvény létrehozására több ok miatt is szükség volt, mivel a Sigmoid aktivációval több probléma is felmerült a mély neurális hálózatok tanítása során. AZ egyik ilyen fő probléma a Sigmoid függvény karakterisztikájából adódik, mivel - ahogyan az már korábban említésre került - az értékei a $[0, 1]$ tartományba esnek, és az eredeti változójának extrém értékeinél (nagyon nagy vagy nagyon kicsi bemeneteknél) a deriváltja közel nullához tart. Ennek eredménye képpen a gradiens a backpropagation során nagyon kicsivé válhat a mélyebb rétegekben, ami megnehezíti a súlyok frissítését és lelassítja vagy akár meg is állítja a tanulást. Ez a probléma különösen mély hálózatok esetében jelentős. A ReLU függvényt, kétféle képpen adhatjuk meg: egy max függvény segítségével 3.6 vagy a 3.7 feltétellel.

$$y = \max(0, x) \quad (3.6)$$

$$y = \begin{cases} x & \text{ha } x \geq 0 \\ 0 & \text{ha } x < 0 \end{cases} \quad (3.7)$$

Ez az jelenti, hogy ha a bemenet negatív, akkor a kimenet 0 lesz, ellenkező esetben a kimenet lineáris, és a deriváltja 1, tehát a visszaterjesztett gradiens nem lesz túl kicsi. Ezáltal kiküszöbölhető a gradiensvesztési probléma, így hatékonyabb a tanulás, a különösen mély hálózatok esetén. Egyszerűségéből és számos kedvező tulajdonságából, mint például a gyorsabb számítás, gradiensvesztési probléma kiküszöbölése, hatékonyabb reprezentációk és a mély hálózatoknál elért jobb teljesítményből adódóan terjedt el nagyon széles körben.

Miután kiválasztottuk a megfelelő aktivációs függvényt a backpropagation a nevéből adódóan a kimeneti rétegtől visszafele haladva frissíti a súlyok értékét, az úgynevezett láncszabály segítségével, a hiba (E - Error) függvényében. A hibát a kimeneti érték y és a várható legjobb kimeneti érték $\hat{y}$ közötti különbség adja, ahogyan azt a 3.8 egyenlet is leírja.

$$E = y - \hat{y} \quad (3.8)$$

A cél elérésének érdekében a hiba minimalizálása a neurális hálózat elsődleges feladata. A jobb eredmény elérése érdekében a szakirodalomban gyakran használatos a hibának valamilyen függvénnyel módosított alakját használni. Egy ilyen lehetőség a Mean Squared Error (MSE) - magyarul átlagos négyzetes eltérés - használata, amely a 3.9 képlet alapján számítandó.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.9)$$

Egyetlen adatmintán használjuk a backpropagation algoritmust, és frissítjük a súlyokat. Ezt követően megismétljük a folyamatot a tanulási adathalmazban található összes adatmintán. Ezt az egész ciklust egy epochnak/epizódnak nevezzük. Általában a neurális hálózatok több epochot/epizódot igényelnek a tanuláshoz. A tanulási ciklusok száma nagyban függ a megoldandó problémától és attól, hogy a modellt mennyire szükséges finomhangolni a megfelelő működéshez.

Az utóbbi években a mélytanulás jelentős áttöréseket ért el - amint az a [28] - ben szerepel - és közel emberi pontossági szintet teljesítettek számos területen, mint a képfelismerés, a természetes nyelvfeldolgozás, az önvezető járművek, valamint az orvosi diagnosztika [29] és a gyógyszerkutatás. Az áttörések egyik kulcsa a számítási kapacitás növekedése, különösen a GPU-k fejlődése, amelyek lehetővé teszik a nagyobb és összetettebb modellek hatékony betanítását. A mélytanulás jelentős előre lépéseket hozott a mesterséges intelligencia számos területén, lehetővé téve olyan problémák megoldását, amelyek évekig nem voltak megoldhatók a hagyományos megközelítésekkel.

3.6. Problémaspecifikus implementáció

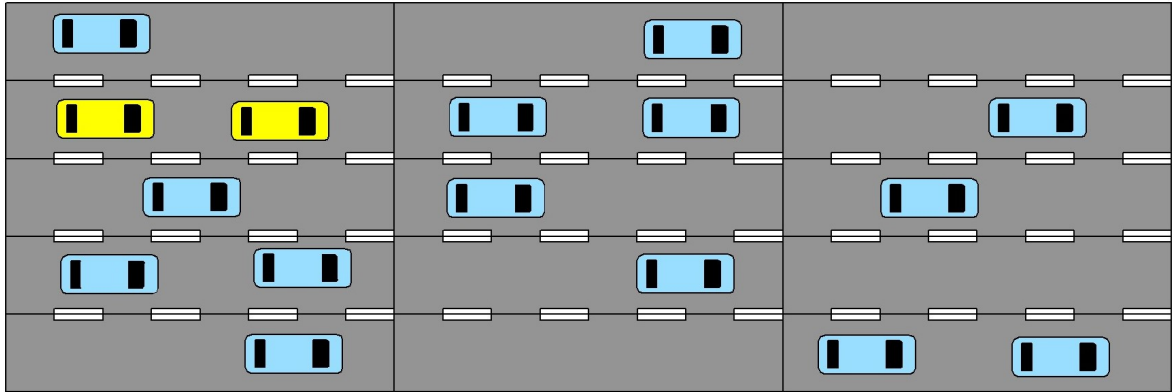
3.6.1. State

Az állapot reprezentáció magában foglalja a környezet azon kritikus elemeit, amelyek lehetővé teszik az ágens számára, hogy pontosan érzékelje és értelmezze a releváns információkat, melyek elengedhetetlenek a hatékony döntési policy-k elsajátításához. Az intelligens közlekedési rendszerek kapcsán a tervezőknek kiemelt figyelmet kell fordítaniuk az állapot reprezentációk gondos kiválasztására, amelyek maximalizálják az állapot, beavatkozás és jutalmazás (state, action, reward) szekvenciák információs tartalmát, miközben arra is ügyelnek, hogy az adatgyűjtés folyamata aránylag egyszerű és költséghatékony maradjon.

Ebben a tanulmányban az egyetlen megfigyelési szakasz (egy szegmens egy sávja) állapot reprezentációja kizárólag egy sávfoglaltsági metrikából áll, ahogyan azt a 3.10 egyenlet is leírja,

$$state_i = \rho_i = \frac{N_i}{l_i} \quad (3.10)$$

ahol a ρ_i foglaltsági metrikát az adott szakaszon lévő járművek számából N_i és a szakasz hosszából l_i számítjuk. Ez az érték hatékonyan mérhető akár egy út menti kameraegységgel, akár egy egyszerű, a VSLC zóna mindkét végén elhelyezett hurokdetektoros eszközzel, így egy praktikus, ugyanakkor kellően informatív állapotabsztrakciót biztosít az RL-alapú forgalomirányításhoz.



3.2. ábra. Az állapot reprezentáció mátrixának sematikus ábrája

Egy szegmens egy sávjának állapot reprezentációja egy 5×5 -ös méretű mátrixra írható le. Az elemzett állapot mindig a mátrix középpontjában helyezkedik el - amint azt a 3.2 ábrán a sárga színű autókkal telített szegmens egy sávjában láthatjuk - így minden irányban két egység marad fent az állapot leírására. Jól kivehető továbbá hogy a szegmensek (oszlopok) és a sávok (sorok) milyen formában bontják fel az autópálya szakaszt egy 5×5 -ös mátrixra.

A 3.2 ábra alapján felírható 5×5 -ös mátrix a 3.11 szerinti alakot veszi fel,

$$\begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & \rho_i & \rho_i & 0 \\ -1 & -1 & \rho_i & \rho_i & \rho_i \\ -1 & -1 & \rho_i & \rho_i & \rho_i \\ -1 & -1 & \rho_i & \rho_i & 0 \end{bmatrix} \quad (3.11)$$

ahol a telített cellák értékét ρ_i foglaltsági metrika adja, ebből adódóan és a 3.10 képlet szerint az esetleges üres cellák értéke 0. Az állapotreprezentáció mátrixának cellái akkor vesznek fel -1 -et, ha a vizsgált állapot középpontjához képest az adott távolságban már nem található vizsgálendő szegmens vagy sáv. Mivel a kijelölt középponttól csak kettő egységnyi távolságban vizsgáljuk a cellák telítettségét, így a 3.2 ábrán látható legalsó sáv értékei nem kerülnek be a mátrixba, valamint mivel ez az első vizsgált szegmens, a mátrix első két oszlopa szintén -1 értékkel van megadva.

3.6.2. Action

A beavatkozási tér az ágens által egy adott környezeti állapot esetén végrehajtható lehetséges beavatkozások halmazát jelenti, egyensúlyt teremtve a komplexitás és a kifejezőképesség között, annak érdekében, hogy lehetővé tegye a különféle stratégiák hatékony felfedezését és kiaknázását.

Számos tanulmány ezen a területen nagydimenziós vektorokat használ a megengedhető sebességkorlátozási értékek széles skálájának kezelésére ([30] [31]). Ezzel szemben ebben a kutatásban a beavatkozási teret egy csupán háromdimenziós vektorral le lehet írni, amely meghatározza a sebességhatár-változásokat. A 3.12 egyenlet mutatja be a beavatkozási tér vektorát,

$$action = \begin{bmatrix} +10km/h \\ 0km/h \\ -10km/h \end{bmatrix} \quad (3.12)$$

ahol jól látható, hogy a lehetséges beavatkozások a sebességkorlátozásokat negatív vagy pozitív irányban $10 - 10km/h$ -val történő megváltoztatása, vagy a sebességhatár megtartása.

3.6.3. Reward

A jutalomfüggvények reprezentációja a Megerősítéses Tanulás kritikus eleme, mivel ez ad tájékoztatást az ágensnek arról, hogy a beavatkozás, amit véghez vitt, milyen mértékben volt megfelelő az adott állapothoz mérten a feladat optimális elvégzésének függvényében. Éppen ezért, ez egy kritikus pont és sok kutatást és számítást igényel, mivel a megfelelő jutalmazási függvény konstruálásától függ a tanítás sikeressége.

Az ágensek jutalomfüggvénye úgy került kialakításra, hogy a 3.13 egyenletben látható módon minimalizálja a várakozási időt a teljes megfigyelt forgalmi hálózaton,

$$reward_{t+1} = \frac{1}{w_t + \epsilon} \quad (3.13)$$

ahol a jutalmat a $t + 1$ időlépésben a w_t várakozási idő adja az előző δt időintervallum alatt, egy kis ϵ konstans bevonásával, hogy megakadályozzuk a nullával való osztást.

4. fejezet

Környezet

Az ML-ben a környezet (environment) kifejezés a rendszer vagy algoritmus azon külső kontextusára utal, amelyben az ágens kölcsönhatásba lép, információt gyűjt, és döntéseket hoz. A környezet alapvető szerepet játszik különösen az RL-ben, ahol a tanulási folyamat során a környezet visszacsatolást ad az ágens számára.

A környezet számos formában megjelenhet a gépi tanulásban. Lehet determinisztikus vagy sztochasztikus, egyszerű vagy komplex. A fő funkciói a következők:

- **Információ biztosítása:** A környezet az ágens számára az állapotinformációkat biztosítja, amelyek alapján a döntéseket hozza meg.
- **Visszajelzés küldése:** A környezet visszajelzést nyújt az ágensnek a beavatkozások helyességéről jutalom formájában.
- **Rendszer dinamikájának meghatározása:** A környezet felelős azért, hogy megadja az ágens számára, hogyan változik az állapot a cselekvések hatására.

A különböző környezet típusokat érdemes valamilyen tulajdonságuk alapján osztályozni a könnyebb felhasználhatóság érdekében. Az ágens információelérési képessége alapján való osztályozásban léteznek teljesen megfigyelhető környezetek (fully observable environments), ahol az ágens hozzáfér a környezet minden releváns információjához, valamint részben megfigyelhető környezetek (partially observable environments), ahol az ágens csak részleges információkat kap az aktuális állapotról.

Egy másik lehetséges osztályozási fajta a dinamika alapján történik. A környezetek osztályozása az alapján, hogy milyen dinamika szerint működnek, segíthet megérteni, hogy milyen típusú problémákkal szembesül az ágens, és milyen módszerekkel érdemes megközelíteni azokat. Ebben az osztályban két alapvető kategóriát különböztetünk meg, a statikus és dinamikus környezeteket. A statikus környezetekben az ágens döntései nem befolyásolják a környezet állapotát, illetve maga a környezet is állandó marad

az idő előrehaladtával. Ezek a környezetek kevésbé komplexek, mivel nem szükséges figyelembe venni az állapot folyamatos változását. Az ilyen környezetek jellemzően determinista szabályrendszeren alapulnak, mint például egy sakkszimuláció, ahol a lépések fix szabályok alapján történnek, és a környezet csak az ágens döntései alapján változik, de minden lépés determinisztikus eredménnyel jár. Ezzel szemben a dinamikus környezetek sokkal komplexebbek, mivel nemcsak az ágens döntései változtatják meg a környezet állapotát, hanem maga a környezet is folyamatosan változik, akár az ágens beavatkozása nélkül is. Ilyen környezet az autópálya, ahol a járművek a forgalomirányító rendszer beavatkozásától függetlenül is sebességet és irányt válthatnak, valamint a közlekedési áramlatok és a véletlen események (például balesetek vagy torlódások) sztochasztikus jellege befolyásolja az ágens döntéshozatalát. A nem-determinista szabályrendszer értelmében a környezet sztochasztikus lehet, vagyis bizonyos helyzetekben ugyanaz a döntés más eredményt hozhat különböző időpontokban.

A környezet mivel a külvilág azon része, amely befolyásolja a modell működését, minél részletesebb leírása nagyban segíti a tanítás folyamatát, azonban kerülendő a túlhatározás, mivel fontos az, hogy a tanítás során, annak gyorsabb lefolyása érdekében minél kevesebb memóriát foglaljon a környezet adatainak halmaza. Ennek érdekében meg kell határozni azokat a legfontosabb tényezőket, amelyek már elegendőek ahhoz, hogy sikeres legyen az ágens tanítása, és a robusztussága elérje a feladat által megkívánt szintet.

Az ágens és a környezet közötti kölcsönhatásokat különböző módokon lehet kategorizálni, attól függően, hogy milyen gyakorisággal és milyen mélységben történik az információcsere. A kölcsönhatások típusa alapvetően meghatározza az ágens tanulási folyamatának dinamikáját és hatékonyságát. Az egyik kategóriában folytonosak az interakciók, míg a másokban diszkrét. A folytonos interakciók során az ágens és a környezet közötti kommunikáció folyamatos, valós időben történik, így az ágens folyamatosan tud döntéseket hozni a környezettől kapott információk alapján. Az ágens szüntelenül figyeli a környezet állapotát, és képes reagálni a legapróbb változásokra is. Ez különösen hasznos dinamikus környezetekben, például közlekedési szituációkban, ahol a járművek mozgása folyamatosan változik. Azonban a folytonos interakciók nagy számítási erőforrást igényelhetnek, mivel az ágensnek minden időpillanatban döntést kell hoznia. Ezzel ellentétben a diszkrét interakciók során az ágens csak bizonyos időintervallumokban kap visszajelzést a környezettől, és csak ekkor tud döntéseket hozni. Ezzel csökkenthető a számítási terhelés, mivel az ágens nem minden időpillanatban hoz döntést, hanem csak meghatározott lépésekben, azonban a diszkrét visszacsatolás miatt az ágens nem mindig reagálhat azonnal a környezet változásaira, ami csökkentheti a rendszer reaktivitását.

Jelen esetben a környezet az ágens felé a vizsgált szakasz adatait adja át az adott időpillanatnak megfelelően. Ezek az adatok sávonként a járművek átlagos sebessége a be- illetve kimenetnél, a forgalom sűrűsége valamint a be- és kimeneti keresztmetszetek, ami a sávok számával egyenértékű.

A VSLC által, a környezet felé közölt beavatkozás a közlekedési jelzőtáblán feltüntetett maximálisan megengedett sebesség dinamikus változtatása, ezzel közvetve változtatva a közlekedés áramlását és segítve a sávváltást és a biztonságos besorolást az autópályákon sávmegszűnés esetén.

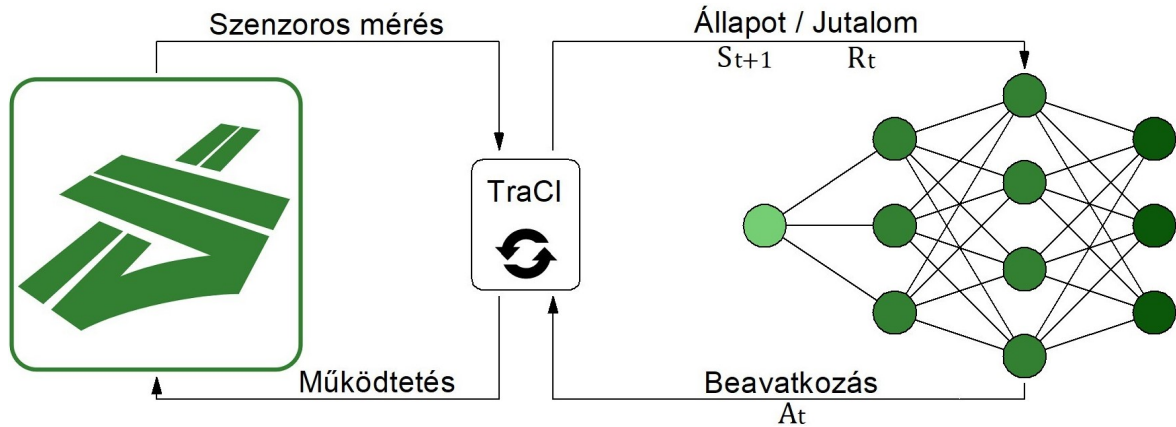
4.1. SUMO környezet

A tanítás, tesztelés és kiértékelés során a [32] által javasolt Simulation of Urban MObility (SUMO) [33] szoftvercsomag révén létrehozott környezetet használok. A SUMO a szakirodalomban széles körben elismert vezető közlekedési szimulátor, amely nyílt forráskódú és alkalmas nem csak makroszkopikus, hanem mikroszkopikus forgalmi szimulációk elvégzésére is, ami nagyon hasznosnak bizonyul a jelen környezet és probléma feloldásának tekintetében. Dinamikusan bővíthető, nagymértékben testre szabható és komplex hálózatok kezelésére tervezett, robusztus eszközkészletével támogatja az intermodális szimulációkat, beleértve a gyalogosforgalmat is, és számos eszközzel rendelkezik a különböző forgatókönyvek létrehozásához. A különféle szcenáriók létrehozása lehetővé teszi a környezet változatosságának modellezését, és lehetőséget nyújt arra, hogy az ágens különböző, valós világot utánzó helyzetekben tanulja meg az optimális döntéshozatalt. A SUMO segítségével modellezhető különböző forgatókönyvek például a forgalmi torlódás, a baleseti helyzetek, valamint az időjárási viszonyok befolyásolásának figyelembevétele.

A forgalmi torlódások modellezése olyan helyzetek szimulálása, amikor a nagy sűrűségű forgalom torlódásokat okoz, például csúcsforgalom idején, autópálya-szakaszokon, ahol szűk keresztmetszetek vannak. A kutatásban vizsgált probléma, ami a forgalom lassulása és hirtelen torlódások előrejelzése, valamint megelőzése komplex feladat, mivel a torlódások kialakulása sztochasztikus folyamat, amely sok változótól függ (pl. sávváltások, balesetek). Éppen ezért szükséges a SUMO használata, aminek ez a modellezhető képessége nagyban hozzájárul a forgalomirányításra betanítandó ágens sikeres tanításának elérésében.

A SUMO lehetőséget biztosít valós és mesterséges közlekedési hálózatok szimulálására, és részletes forgalmi áramlási metrikák numerikus nyomon követésére, mint például a sűrűség, az utazási és várakozási idők, valamint a fenntarthatósági paraméterek (például CO₂- és NO_x-kibocsátás, üzemanyag-fogyasztás). Ezen felül támogatja

a randomizálást, és a Traffic Control Interface (TraCI) által lehetővé teszi a környezet manipulálását különböző programozási nyelveken keresztül, így biztosítva a szimulációk sokoldalú testreszabását és elemzését. A SUMO környezet és a mély neurális hálózattal rendelkező ágens között, a TraCI által létrehozott kapcsolatot a 4.1 ábra szemlélteti.

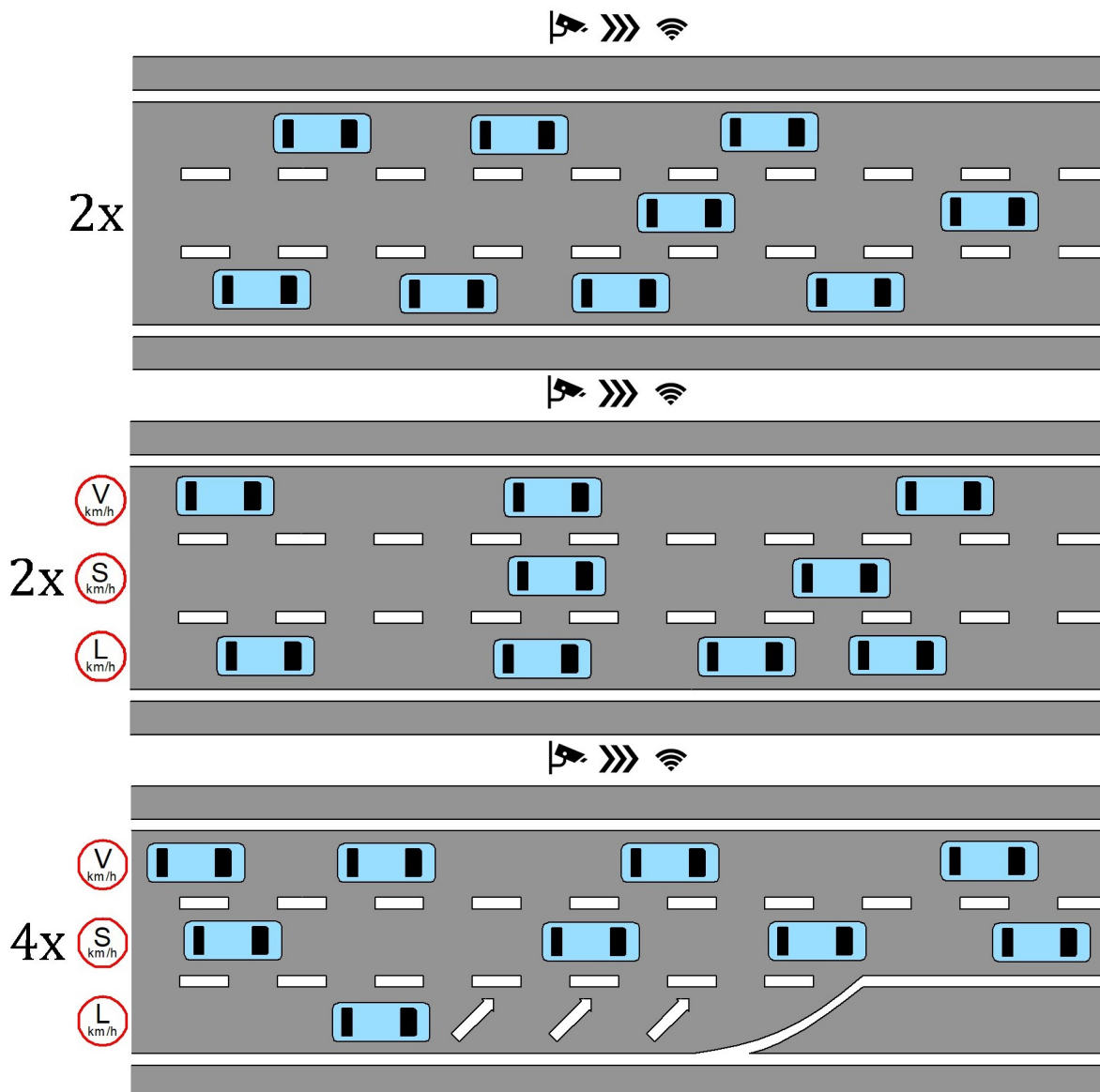


4.1. ábra. A szoftverkomponensek közötti kommunikáció sematikus ábrázolása

A SUMO és a TraCI interfész alapvető fontosságú a környezet szimulációjában és az ágens-környezet interakciók irányításában. A TraCI a SUMO egyik legfontosabb eleme, amely lehetővé teszi a szimulációs folyamat valós idejű irányítását. Az interfész segítségével az ágens döntései közvetlenül befolyásolhatják a szimuláció menetét. A TraCI API különféle funkciókat biztosít, például a járművek sebességének módosítását, a sávváltások vezérlését, vagy a forgalmi lámpák működésének beállítását. Az ágens ezen API-hívások segítségével valós időben avatkozhat be a forgalomirányításba. Az interfész biztosítja, hogy az ágens döntései közvetlenül érvényesüljenek a szimulációban, miközben valós idejű adatokat kap a járművek sebességéről, pozíciójáról és más környezeti tényezőkről.

A SUMO szimulációk során sok paramétert lehet konfigurálni, amely befolyásolja a szimuláció pontosságát és teljesítményét, mint például a sávkapacitás és sebességkorlátozások, szimulációs lépések gyakorisága, jelzőlámpák zöld és piros ideje vagy a forgalom áramlásának eloszlása. A sávkapacitás és sebességkorlátozások befolyásolják, hogy a járművek hogyan mozognak a szimulált környezetben. A megfelelő beállítások kritikusak a valósághű szimulációhoz. A szimulációs lépések gyakorisága a szimuláció felbontását befolyásolja, hogy milyen gyakran frissül a rendszer állapota. A kisebb időlépések finomabb szimulációt eredményeznek, de nagyobb számítási igénnyel járnak.

Egy valós forgalmi hálózat körülményeinek pontos modellezése érdekében számos tényezőt figyelembe kell venni. Ezek közé tartoznak a megengedett sávváltások, a döntéshozatalhoz szükséges minimális időintervallum, a sebességkorlátozások és a járművek áramlása. Valamint fontos a szimulációs környezet szakszerű kialakítása is. Jelen



4.2. ábra. A forgalomirányító hálózat geometriai kialakítása három sávos autópálya esetén

esetben a vizsgált környezetek és azok hálózatának geometriai kialakítását a 4.2 ábra szemlélteti.

A szimulációs környezet összesen nyolc egyenes útszakaszból áll, amelyek mindegyike egyirányú, $300m$ hosszú és állhat 3, 5 illetve 7 sávból. Minden egyes sáv $3,2m$ széles, így a teljes szegmens szélessége a sávok számának függvényében $9,6m$, $16m$ vagy $22,4m$. Az első szegmens elején minden egyes sávhoz kijelölt indítási pontok generálnak járműveket, amelyek a bevezetett randomizációs módszer alapján, a szimuláció minden egyes időlépésben véletlenszerűen és egymástól függetlenül juttathatók be a járművek a szimulációba. Ez a randomizálás lehetővé teszi, hogy a környezet változatos forgalmi forgatókönyveket generáljon, hozzájárulva az eredmények robusztusságához. A kezdeti

két szegmens, amely az első $600m$ -t fedi le, kizárólag megfigyelési célokat szolgál. A 3. és 4. szegmensek a változtatható sebességhatározási szabályozási zónáknak vannak kijelölve, de a hálózat ezen szakaszán nem generálnak rendellenességeket, lehetővé téve a VSLC rendszer számára, hogy proaktívan megelőzze és kezelje a későbbi helyeken esetlegesen megjelenő rendellenességeket. Az 5 – 8. szakaszok szintén VSLC-zónák, ahol véletlenszerű anomáliák jöhetnek létre, és így sávelhagyásos szűk keresztmetszetek alakulhatnak ki. Az eltérő sáv számmal rendelkező autópályák $6n$ VSLC-zónával rendelkeznek, ahol a 6-os konstans a beavatkozó szegmensek száma, míg az n változó a sávok számát jelöli. Mindegyik zóna önállóan szabályozható küszöbértékekkel rendelkezik, amelyek a minimum $20km/h$ -tól a maximum $130km/h$ -ig terjednek, és $10km/h$ -s lépésekben állíthatók be. A vezérlőrendszer feladata pedig az egyes zónák optimális sebességhatárának meghatározása. A szimulációban a sebességhatárokat csak valódi közlekedési táblákon lehet megjeleníteni, hogy a járművezetők könnyen betarthassák azokat, és a valósághű átmenetek biztosítása érdekében a sebességváltozások is legfeljebb $10km/h$ -ra korlátozódnak.

A megengedett sávváltások figyelembevétele érdekében a SUMO hálózati konfigurációs fájl lehetővé teszi a sávváltási viselkedésre vonatkozó szabályok meghatározását. Ezenkívül, mivel a környezetnek szerves részét képezi maga az emberi tényező, fontos az ezzel járó korlátokat is figyelembe venni. Éppen ezért minden útszakaszon a sebességhatározások megváltoztatásai között eltelt idő 15 másodperces minimális időintervallumra lett korlátozva. Ez az időkeret figyelembe veszi az emberi reakcióképességet, valamint azt a tényt is, hogy ha a sebességhatár megváltozásának értéke nagy mértékben módosul, akkor a lassításhoz, illetve a gyorsításhoz is több időre van szükség.

Egy másik fontos szempont a hálózaton belüli járműáramlás meghatározása. A valódi forgalmi viszonyokat tükröző környezet megteremtése érdekében a [4] cikk vált az összehasonlítás alapjául, amely szerint az áramlási sebesség értéke a következőre lett állítva: $1500jármű/óra/sáv$. Kutatásomban az egyenletes eloszláson alapuló randomizálást alkalmaztam.

5. fejezet

Eredmények

A forgalom folytonos és gördülésmentes áramlása, nagyon fontos környezetvédelmi, biztonsági és nem utolsó sorban egészségügyi szempontból is. Ebből adódóan a megfelelő forgalomirányítás elengedhetetlen és feltétlen szükséges a torlódások megelőzésének érdekében. Éppen ezért kulcsfontosságú lenne egy olyan rendszer kiépítése, amely a legjobb teljesítményt nyújtja a probléma feloldására és emellett általánosítható a különböző véletlenszerű anomáliák leküzdésére. Ennek megoldása érdekében 3, 5 és 7 sávval rendelkező autópályák esetén történik az ágensek betanítása. Ezeknek a betanult ágenseknek a teljesítménye kerül összevetésre az MTFC és az MCS algoritmusok által kihozott eredményekkel a szimulált környezetekben. A különböző technikák teljesítményének összehasonlításával - amelyben a SUMO által szolgáltatott adatok nyújtanak alapot - kiderül, hogy a Megerősítéses Tanulással megvalósított módszer valóban elég robusztus-e, a feladat általánosítására és a véletlenszerű anomáliák megoldására, amelyek a különböző sávkapacitású autópályákon keletkeznek.

Annak érdekében, hogy megtudjuk melyik megoldás a legmegfelelőbb a probléma univerzális megoldására a következő - szimulációk során kihozott - adatok felhasználásával kerülnek összehasonlításra az algoritmusok. Kettő csoportra lehet bontani a vizsgált szempontokat, melyek közül az egyik a forgalom áramlásával kapcsolatos adatokat foglalja magába, míg a másik csoport a környezetvédelemmel kapcsolatos elemeket vizsgálja. A forgalom áramlásának vizsgálatában a menetidő (A járművek átlagos áthaladási ideje a vizsgált szakaszon), várakozási idő (A járművek átlagos várakozással töltött ideje a szakaszon) és az álló járművek (A sávban álló járművek átlagos százalékos aránya) játszanak szerepet. A környezetvédelem szempontjából a szimuláció során teljes CO_2 és NO_x kibocsátás a vizsgált szakaszon és az összes üzemanyag mennyiség másodpercre lebontva.

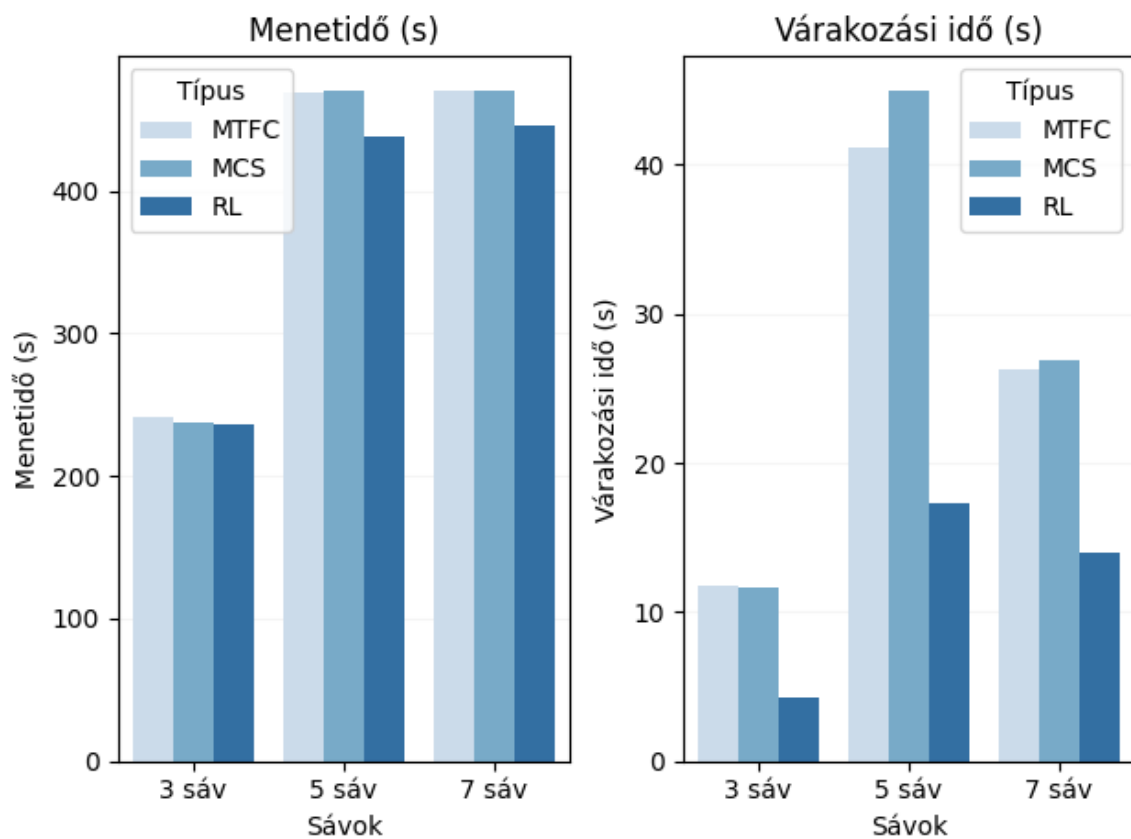
Az 5.1 táblázat pontos adatai alapján egyértelműen látható, hogy az általam javasolt technikával irányított forgalom minden vizsgált szempontból kedvezőbb ered-

ményeket hozott a forgalom folytonos áramlásának biztosításában. Ez nemcsak a menetidő és a várakozási idő csökkentését eredményezte, hanem környezetvédelmi szempontból is figyelemre méltó sikereket ért el, hiszen jelentős mértékben csökkentette a CO_2 és NO_x -kibocsátásokat, valamint az üzemanyag-fogyasztást is. Ezek a tényezők együttesen bizonyítják, hogy a bevezetett megoldás nemcsak költséghatékony és energiatakarékos, de fenntartható közlekedési rendszerek kialakításához is hozzájárul. Ennek köszönhetően megoldható a forgalom optimális áramlásának biztosítása, plusz sávok építése nélkül. Az RL-alapú módszer robusztussága különösen kiemelkedőnek bizonyult a forgalmi ingadozások és anomáliák kezelésében, amelyek az áteresztőképesség javulásához, a torlódások enyhítéséhez és a forgalmi környezet stabilizálásához vezettek, így egy hosszú távon is hatékony megoldást kínálva a közlekedési hálózatok optimalizálására.

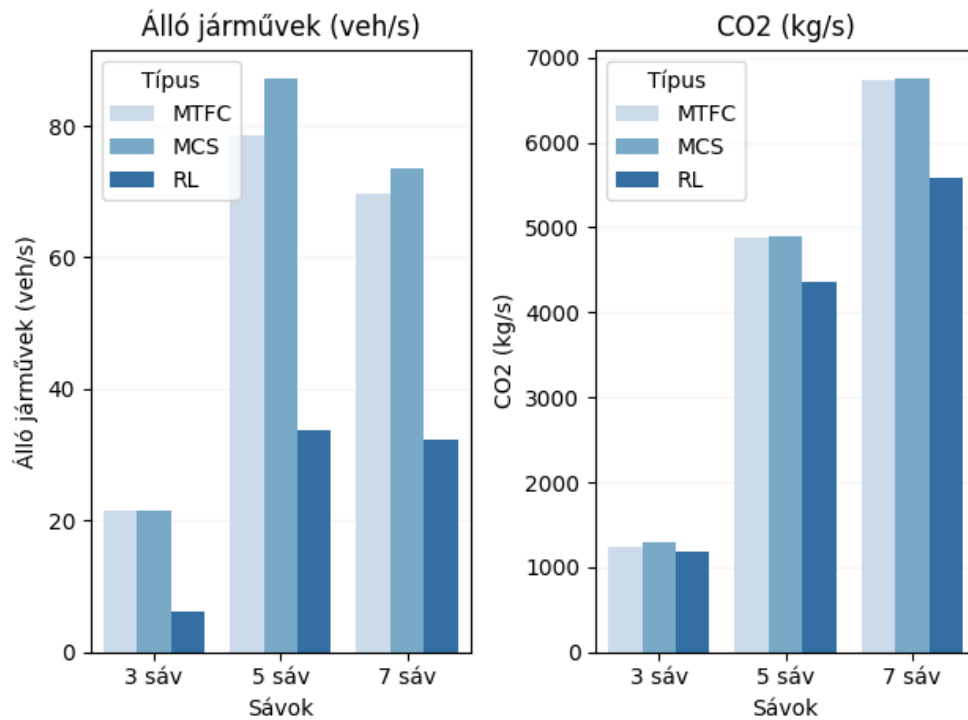
3 sáv			
	RL	MTFC	MCS
Menetidő (s):	235,9	241,1	237,3
Várakozási idő (s):	4,3	11,8	11,6
Álló járművek (veh/s):	6,1	21,5	21,6
CO_2 (kg/s):	1186,4	1242,7	1294,7
NO_x (g/s):	495,4	522,1	546,6
Üzemanyag (kg/s):	378,4	396,4	413
5 sáv			
Menetidő (s):	437,5	469	470,7
Várakozási idő (s):	17,3	41,2	45
Álló járművek (veh/s):	33,7	78,6	87,2
CO_2 (kg/s):	4366,3	4883,8	4891,9
NO_x (g/s):	1869,5	2110,8	2114,3
Üzemanyag (kg/s):	1392,7	1557,8	1560,3
7 sáv			
Menetidő (s):	445,3	469,8	470,2
Várakozási idő (s):	14	26,3	26,9
Álló járművek (veh/s):	32,4	69,8	73,6
CO_2 (kg/s):	5589,4	6738,5	6754,2
NO_x (g/s):	2374,6	2909,1	2917
Üzemanyag (kg/s):	1782,8	2149,3	2154,3

5.1. táblázat. A vizsgált algoritmusok által irányított forgalom adatai különböző sávok esetére

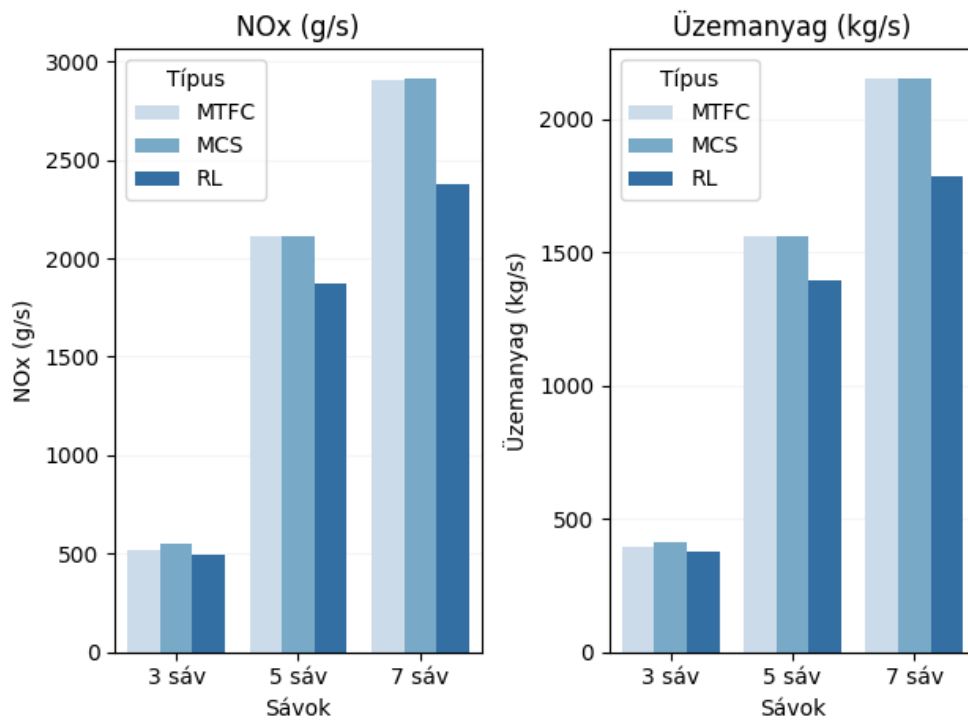
A táblázat adatai alapján készült diagramok 5.1, 5.2, 5.3 remekül vizualizálják az RL robusztusságát a szakirodalomban használatos MTFC és MCS algoritmusokkal szemben a random anomáliák által okozott problémák megoldására. Jól látható, hogy az általam javasolt módszer mind a 3 szimulációs környezetben - 3, 5 és 7 sávós autópályák - figyelemre méltó sikereket ért el, ugyanis csökkentette a forgalomáramlás kritikus paramétereit, az eddig használt módszerekkel szemben. Az eredmények nemcsak azt mutatják, hogy az RL hatékonyan kezeli a váratlan zavarokat, hanem azt is, hogy képes adaptálódni különböző úthálózati struktúrákhoz, különböző sávszámok mellett is. Ez különösen fontos, mivel az algoritmusoknak a valós forgalmi környezetben is alkalmazkodniuk kell a dinamikus változásokhoz, mint például a hirtelen fellépő torlódásokhoz vagy balesetekhez.



5.1. ábra. A vizsgált algoritmusok által irányított forgalmak menetidőt és várakozási időt bemutató diagramjai különböző sávkapacitások esetére



5.2. ábra. A vizsgált algoritmusok által irányított forgalmak álló járműveit és CO_2 -kibocsátást bemutató diagramjai különböző sávkapacitások esetére



5.3. ábra. A vizsgált algoritmusok által irányított forgalmak NO_x -kibocsátást és üzemanyag-fogyasztást bemutató diagramjai különböző sávkapacitások esetére

A kutatás eredményességét hivatott bemutatni az 5.2 táblázat, amely a forgalom-irányító rendszer szabályozását Megerősítéses Tanulás alkalmazásával valósítja meg. Az RL által elért eredményeket az MTFC és MCS algoritmusokéval összevetve kiderül, hogy a várakozási időt és a sávban megálló járművek számát nagy mértékben sikerült csökkenteni a korábbiakhoz képest (50 – 60%), így a forgalom áramlása dinamikus marad, mivel a járművek folyamatosan tudnak haladni anélkül, hogy megállásra kényszerülnének. Ezzel együtt a hirtelen elindulások és megállások okozta veszélyek mellett a károsanyag kibocsátást és az üzemanyag fogyasztást is sikerült figyelemre méltó százalékkal csökkenteni (11 – 12%), ezzel óvva a környezetet. Továbbá nem elhanyagolható,

3 sáv			
	MTFC%	MCS%	Átlagosan%
Menetidő (s):	2,15	0,57	1,36
Várakozási idő (s):	63,56	62,88	63,22
Álló járművek ($\frac{veh}{s}$):	71,74	71,86	71,80
CO_2 ($\frac{kg}{s}$):	4,53	8,37	6,45
NO_x ($\frac{g}{s}$):	5,13	9,37	7,25
Üzemanyag ($\frac{kg}{s}$):	4,53	8,37	6,45
5 sáv			
Menetidő (s):	6,71	7,05	6,88
Várakozási idő (s):	58,00	61,60	59,80
Álló járművek ($\frac{veh}{s}$):	57,17	61,39	59,28
CO_2 ($\frac{kg}{s}$):	10,60	10,74	10,67
NO_x ($\frac{g}{s}$):	11,43	11,58	11,50
Üzemanyag ($\frac{kg}{s}$):	10,60	10,74	10,67
7 sáv			
Menetidő (s):	5,22	5,30	5,26
Várakozási idő (s):	46,98	48,17	47,58
Álló járművek ($\frac{veh}{s}$):	53,58	56,04	54,81
CO_2 ($\frac{kg}{s}$):	17,05	17,24	17,15
NO_x ($\frac{g}{s}$):	18,37	18,59	18,48
Üzemanyag ($\frac{kg}{s}$):	17,05	17,24	17,15
Általánosítva elért javulás %			
Menetidő (s):	4,5		
Várakozási idő (s):	56,87		
Álló járművek ($\frac{veh}{s}$):	61,96		
CO_2 ($\frac{kg}{s}$):	11,42		
NO_x ($\frac{g}{s}$):	12,41		
Üzemanyag ($\frac{kg}{s}$):	11,42		

5.2. táblázat. Az RL által elért javulás százalékos értéke a többi algoritmushoz képest a vizsgált adatokban

hogy általánosságban véve az utazási időt is sikerült csökkenteni 4 – 5%-al.

Ezzel bebizonyosodott, hogy ez a megoldás általánosan használható az autópályák forgalmának optimalizálására, függetlenül az autópálya kapacitásától – legyen szó X sávós útszakaszokról. A módszer rugalmassága lehetővé teszi, hogy különböző forgalmi terhelés esetén is hatékonyan alkalmazzuk, így segítve a forgalmi torlódások csökkentését, a járművek áthaladási idejének rövidítését, valamint az üzemanyag-fogyasztás és a környezeti terhelés mérséklését.

6. fejezet

Konklúzió

Az autópályák forgalmának hatékony irányítása komoly kihívás, különösen a véletlenszerű forgalmi anomáliák, mint például hirtelen fékezések vagy az ennek következtéből kialakuló balesetek miatt fellépő kapacitáscsökkenés esetén. A szinte mindennapos csúcsidőszakokban fellépő kifejezetten nagy mértékű torlódások nem csak a menetidőt növelik, hanem súlyos környezeti problémákat is okoznak, hiszen jelentősen megnövekedik a károsanyag-kibocsátás (CO_2 , NO_x) és az üzemanyag-fogyasztás, amely városközeli környezetben rontja az életminőséget és az élhetőséget. A hagyományos forgalomirányítási megoldások gyakran nem képesek dinamikusan kezelni a váratlan helyzeteket, és nem alkalmasak a folyamatosan változó forgalmi környezethez való alkalmazkodásra. Annak érdekében, hogy a forgalom előreláthatatlan hirtelen változásai lekövethetők legyenek és a forgalomáramlást optimalizálni lehessen, valamint csökkenti a környezeti terhelést, olyan intelligens forgalomirányító rendszer fejlesztése a cél, amely dinamikusan képes változtatni a sebességhatárokat a forgalom növekedésének érdekében a random anomáliákkal szemben.

A dolgozatban már ismertetett Megerősítési Tanuláson (RL) alapuló algoritmus bevezetésével, a forgalomirányító rendszer dinamikusan változtatja a megengedett sebességhatárokat a vizsgált autópályaszakasz minden egyes szegmensén, sávokra lebontva, az adott szituációhoz igazodva, figyelembe véve a forgalom sűrűségét, a járművek átlagsebességét és a kapacitáskorlátokat. Ez annak köszönhető, hogy az RL valós időben tud beavatkozni az adott állapothoz mérten, így képes alkalmazkodni az autópályákon fellépő különböző anomáliákhoz és változó forgalmi körülményekhez. Az algoritmus a forgalmi torlódások megelőzésére fókuszál, és hatékonyan kezeli a szűk keresztmetszetek okozta problémákat is, mindezt anélkül, hogy bonyolult matematikai modellekre lenne szükség.

Az eredmények fényében belátható, hogy az RL-alapú forgalomirányítási megoldás valóban általánosan alkalmazható különböző sávkapacitással rendelkező autópályákon,

és jobb eredményeket ér el a hagyományos algoritmusokkal szemben a véletlenszerű forgalmi anomáliák kezelésére, mind a forgalom folyamatos áramlásának biztosításában, mind pedig környezetvédelmi szempontból. Az algoritmus sikeresen csökkentette a menetidőt és a várakozási időt, miközben jelentős mértékben mérsékelte a károsanyag-kibocsátást (CO_2 , NO_x) és az üzemanyag-fogyasztást.

A kutatás során elért eredmények alapján, adódik a lehetőség, hogy a bemutatott algoritmus segítségével más közlekedésirányítási problémák, mint például a városi közlekedésben vagy a regionális utak forgalomszabályozásában fellelhető nehézségek is megoldásra kerüljenek. Jövőbeli kutatási irány lehet a valóság pontosabb leírásának érdekében a különböző időjárási körülmények, valamint az emberi szabálysértő magatartás beépítése a modellekbe, továbbá az autonóm járművek és az intelligens infrastruktúra integrációjának vizsgálata a forgalom optimalizálása érdekében.

Ábrák jegyzéke

2.1. A folytonosság törvénye a járművek áramlásában	4
2.2. Fundamentális diagram a járművek áramlása és sűrűsége függvényében [3]	6
2.3. VSLC működése autópálya kapacitás csökkenés esetén	7
3.1. Megerősítéses Tanulás sematikus ábrája [23]	16
3.2. Az állapot reprezentáció mátrixának sematikus ábrája	24
4.1. A szoftverkomponensek közötti kommunikáció sematikus ábrázolása	29
4.2. A forgalomirányító hálózat geometriai kialakítása három sávos autópályán	30
5.1. A vizsgált algoritmusok által irányított forgalmak menetidőt és várakozási időt bemutató diagramjai különböző sávkapacitások esetére	34
5.2. A vizsgált algoritmusok által irányított forgalmak álló járműveit és CO_2 -kibocsátást bemutató diagramjai különböző sávkapacitások esetére	35
5.3. A vizsgált algoritmusok által irányított forgalmak NO_x -kibocsátást és üzemanyag-fogyasztást bemutató diagramjai különböző sávkapacitások esetére	35

Táblázatok jegyzéke

5.1. A vizsgált algoritmusok által irányított forgalom adatai különböző sávok esetére	33
5.2. Az RL által elért javulás százalékos értéke a többi algoritmushoz képest a vizsgált adatokban	36

Rövidítések

CTM cell transmission traffic flow model.

CV Connected Vehicles.

DDQN Double Deep Q-Network.

DQN Deep Q-Network.

ITS Intelligent Transportation Systems.

MARL Multi Agent Reinforcement Learning.

MCS Motorway Control System.

MDP Markov Decision Process.

ML Machine Learning.

MTFC Mainstream Traffic Flow Control.

RL Reinforcement Learning.

SL Supervised Learning.

SUMO Simulation of Urban MObility.

TraCI Traffic Control Interface.

USL Unsupervised Learning.

VMS Variable Message Sign.

VSLC Variable Speed Limit Control.

Irodalomjegyzék

- [1] Mihály Héder, Ernő Rigó, Dorottya Medgyesi, Róbert Lovas, Szabolcs Tenczer, Ferenc Török, Attila Farkas, Márk Emődi, József Kadlecsek, György Mező, Ádám Pintér, and Péter Kacsuk. The past, present and future of the ELKH cloud. *Információs Társadalom*, 22(2):128, aug 2022.
- [2] Martin Treiber and Arne Kesting. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation*, Springer-Verlag Berlin Heidelberg, pages 983–1000, 2013.
- [3] Nour Khoudari and Benjamin Seibold. Multiscale properties of traffic flow: the macroscopic impact of traffic waves. In *Multiplicity of Time Scales in Complex Systems: Challenges for Sciences and Communication I*, pages 267–296. Springer, 2022.
- [4] Ellen F Grumert, Andreas Tapani, and Xiaoliang Ma. Characteristics of variable speed limit systems. *European transport research review*, 10:1–12, 2018.
- [5] Bidoura Khondaker and Lina Kattan. Variable speed limit: an overview. *Transportation Letters*, 7(5):264–278, 2015.
- [6] Yihang Zhang and Petros A Ioannou. Stability analysis and variable speed limit control of a traffic flow model. *Transportation Research Part B: Methodological*, 118:31–65, 2018.
- [7] Krešimir Kušić, Nino Korent, Martin Gregurić, and Edouard Ivanjko. Comparison of two controllers for variable speed limit control. In *2016 International Symposium ELMAR*, pages 101–106. IEEE, 2016.
- [8] Youngjun Han, Danjue Chen, and Soyoung Ahn. Variable speed limit control at fixed freeway bottlenecks using connected vehicles. *Transportation Research Part B: Methodological*, 98:113–134, 2017.

- [9] Yihang Zhang and Petros A Ioannou. Combined variable speed limit and lane change control for highway traffic. *IEEE Transactions on Intelligent Transportation Systems*, 18(7):1812–1823, 2016.
- [10] Martin Gregurić, Sadko Mandžuka, and Marko Šoštaric. The application of co-operative variable speed limit control for reduction of crash potential. In *2019 International Symposium ELMAR*, pages 1–4. IEEE, 2019.
- [11] Krešimir Kušić, Edouard Ivanjko, Martin Gregurić, and Mladen Miletić. An overview of reinforcement learning methods for variable speed limit control. *Applied Sciences*, 10(14):4917, 2020.
- [12] Feng Zhu and Satish V Ukkusuri. Accounting for dynamic speed limit control in a stochastic traffic environment: A reinforcement learning approach. *Transportation research part C: emerging technologies*, 41:30–47, 2014.
- [13] Krešimir Kušić, Edouard Ivanjko, and Martin Gregurić. A comparison of different state representations for reinforcement learning based variable speed limit control. In *2018 26th Mediterranean Conference on Control and Automation (MED)*, pages 1–6. IEEE, 2018.
- [14] Kostandina Veljanovska, Kristi M Bombol, and Tomaž Maher. Reinforcement learning technique in multiple motorway access control strategy design. *PROMET-Traffic&Transportation*, 22(2):117–123, 2010.
- [15] Bálint Kővári, István Knáb, and Tamás Bécsi. Variable speed limit control for highway scenarios a multi-agent reinforcement learning based appraoch. Technical report, EasyChair, 2024.
- [16] Martin Gregurić, Krešimir Kušić, Filip Vrbanić, and Edouard Ivanjko. Variable speed limit control based on deep reinforcement learning: A possible implementation. In *2020 International Symposium ELMAR*, pages 67–72. IEEE, 2020.
- [17] Bálint Kővári, Bálint Pelenczei, István Gellért Knáb, and Tamás Bécsi. Beyond trial and error: Lane keeping with monte carlo tree search-driven optimization of reinforcement learning. *Electronics*, 13(11):2058, 2024.
- [18] Zemian Ke, Zhibin Li, Zehong Cao, and Pan Liu. Enhancing transferability of deep reinforcement learning-based variable speed limit control using transfer learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(7):4684–4695, 2020.

- [19] Eduardo R Müller, Rodrigo C Carlson, Werner Kraus, and Markos Papageorgiou. Microscopic simulation analysis of mainstream traffic flow control with variable speed limits. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 998–1003. IEEE, 2013.
- [20] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.
- [21] Zoubin Ghahramani. Unsupervised learning. In *Summer school on machine learning*, pages 72–112. Springer, 2003.
- [22] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.
- [23] Shweta Bhatt. Reinforcement learning 101. *Towards Data Science*, 2018.
- [24] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.
- [25] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [26] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation, parallel distributed processing, explorations in the microstructure of cognition, ed. de rumelhart and j. mcclelland. vol. 1. 1986. *Biometrika*, 71(599-607):6, 1986.
- [27] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.
- [28] Matiur Rahman Minar and Jibon Naher. Recent advances in deep learning: An overview. *arXiv preprint arXiv:1807.08169*, 2018.
- [29] Hafiz M Ahmad, Muhammad Jaleed Khan, Adeel Yousaf, Sajid Ghuffar, and Khurram Khurshid. Deep learning: a breakthrough in medical imaging. *Current Medical Imaging*, 16(8):946–956, 2020.

- [30] Zhibin Li, Chengcheng Xu, Yanyong Guo, Pan Liu, and Ziyuan Pu. Reinforcement learning-based variable speed limits control to reduce crash risks near traffic oscillations on freeways. *IEEE Intelligent Transportation Systems Magazine*, 13(4):64–70, 2020.
- [31] Krešimir Kušić, Ivana Dusparic, Maxime Guériau, Martin Gregurić, and Edouard Ivanjko. Extended variable speed limit control using multi-agent reinforcement learning. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.
- [32] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 2575–2582. IEEE, 2018.
- [33] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo—simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind, 2011.