Adaptív jelzőlámpa irányítás Megerősítéses Tanulással SUMO környezetben

Kővári Bálint* Bécsi Tamás*

*Közlekedés és- Járműirányítási Tanszék, Budapesti Műszaki és Gazdaságtudományi Egyetem
(email: kovari.balint@kjk.bme.hu, becsi.tamas@kjk.bme.hu).

Kivonat: A Traffic Signal Control (TSC) probléma régóta nagy figyelmet kap a kutatók részéről, aminek köszönhetően a feladat megoldására sokféle módszert alkalmaztak már az egyszerű szabályalapú algoritmusoktól egészen a gépi tanulásig bezárólag. Az irányítási feladat megoldása során különböző optimalizációs célok választhatók, amelyek segítségével meghatározzuk a döntéshozó viselkedését. Ebben a munkában a kibocsájtás minimalizálására fektetjük a hangsúlyt. Megoldásunk lényege egy új, rewarding koncepció kifejlesztése megerősítéses tanulás alapú algoritmusok számára. Annak ellenére, hogy a rewarding koncepció nem használja az eddig a szakirodalomban jól megszokott metrikákat, képes felülmúlni a szimulátorként használt SUMO szoftver beépített algoritmusait mind klasszikus teljesítmény indikátorok, mind pedig fentarthatósági indikátorok szempontjából, mint az üzemanyag fogyasztás és a NO_x kibocsájtás.

1. BEVEZETÉS

Nemzetközi Energia Ügynökség szerint a közlekedési ágazat felelős a közvetlen szén-dioxid-kibocsátás 24%-ért, amelynek háromnegyede 2020-ban a közúti szállításból származik. A CO_2 -kibocsátás csökkentése jelentős kihívássá nőtte ki magát, mivel a CO_2 nagymértékben hozzájárul az éghajlatváltozáshoz (Al-Ghussain, 2019). A közúti közlekedés kibocsátását többféle módon is csökkenteni lehet, például kevésbé szennyező üzemanyagok felhasználásával, mint a hidrogén, energiatakarékosabb járművek (Hannappel, 2017), vagy az Intelligens Közlekedési Rendszerek alkalmazásával. Az Intelligens Közlekedési Rendszerek képesek arra, hogy mérsékeljék a mindennapokban kialakuló torlódásokat, amelyek komoly hatással vannak életünk több területére is. Ilyen például a termelékeny órák száma, kevesebb kibocsátás, biztonságosabb ingázás stb. Ebből adódóan ez a cikk a jelzőlámpás kereszteződések hatékonyabb működtetésére összpontosít, mivel ezek a városi közlekedési környezet legfőbb gyengepontjai a torlódások kialakulásának és kezelésének szempontjából. A jelzőlámpás kereszteződések hatékonyabban irányíthatók, ha a fix periódussal működő jelzőlámpa programokat felváltjuk a közlekedés aktuális állapotát figyelembe vevő algoritmusokkal, amelyek adaptívan reagálnak a folyamatosan változó forgalmi terhelésekre. A TSC problémát számos különböző megközelítéssel oldották meg az elmúlt időben. A legkorábbi megoldások szabályalapú rendszereket alkalmaztak (Ekeila, et al., 2009). Egy másik széleskörben kutatott megoldás a Genetikus algoritmusokhoz köthető (Wang, et al., 2019) amelyek gyorsan Genetikus Algoritmus-alapú szimulációs programokká fejlődtek (Zhang & Chen, 2013). További módszer még a szimuláción alapuló algoritmusok, amelyek forgalomáramlási modelleket alkalmaznak az optimális lámpaprogramok kialakításához (McKenney & White, 2013). Rengeteg publikáció foglalkozik továbbá Dinamikus Programozás alapú megoldásokkal, elsősorban azért, mert forgalmi szituációk széleskörében alkalmazható, illetve sokféle teljesítmény indikátor használatát teszi lehetővé (Cai, et al., 2009). A Multi-Ágens Rendszerek szintén elterjedt megközelítésnek tekinthetők a TSC-probléma kezelésére (Jin & Ma, 2017), ugyanez vonatkozik a Játékelmélet alapú módszerekre is (Villalobos, et al., 2008). A legújabb trendekben azonban Megerősítéses Tanulás alapú algoritmusokat alkalmaztak a TSC probléma megoldására. Megerősítéses Tanulás alapú algoritmusok több kimagasló eredményhez vezettek ezen a területen (Guo & Harmati, 2020) és számos más közlekedéssel kapcsolatos alkalmazásban is (Fehér, et al., 2020). Megerősítéses Tanulást mind egy (Wan & Hwang, 2018), (Genders & Razavi, 2016) mind több kereszteződés (Tan, et al., 2019) (Van der Pol & Oliehoek, 2016), esetére alkalmazták már a szakirodalomban. A Megerősítéses Tanulás TSC probléma megoldására való felhasználásának fő oka, hogy ez az algoritmus család rengetek olyan tulajdonsággal rendelkezik, amelyek segítik a komplex szekvenciális döntéshozatali problémák megoldását. Ezek a skálázhatóság az egyszerű kereszteződésekről a komplexebb kereszteződésekre. További előny, a függvény approximátorok alkalmazása, amelyek lehetővé teszik a generalizálást és a valós idejű alkalmazhatóságot. Az említett előnyök a fő okai annak, hogy a Merősítéses Tanulás alapú megoldásoknak hatalmas szakirodalma van ezen a területen.

1.1 A publikáció célja

A publikációban bemutatott munka két fő újdonságot tartalmaz. Először egy új rewarding koncepciót mutatunk be Megerősítéses Tanulás alapú algoritmusok tanításához TSC probléma megoldására., amely a korábbi megközelítésekhez képest nem használ forgalmi metrikákat. Másodszor, betanítunk egy Policy Gradient és egy Deep Q-Network algoritmust, amelyek teljesítményét összehasonlítjuk a SUMO szoftver beépített algoritmusával. Ezen túlmenően a

publikációban bemutatott összehasonlítás nem csak a klasszikus metrikákra terjed ki, mint például a várakozási idő, utazási idő, és a torlódás hossza, hanem összehasonlítja a fenntarthatósági indikátorokat is a különböző megoldások esetére.

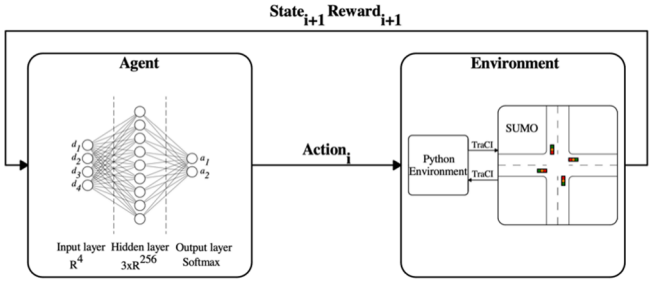
2. ALGORITMUSOK

2.1 Megerősítéses Tanulás

Számos területen nincs lehetőség elegendő tanító minta előállítására, amely lehetővé tenné felügyelt tanulás alapú algoritmusok használatát. Ezekben az esetekben a Megerősítéses Tanulás használható, mivel Megerősítéses Tanulás során az ágens önmaga számára generálja le a tanításhoz szükséges adatokat. Az optimális vagy preferált viselkedést pedig az alkalmazott rewarding koncepció segítségével juttathatjuk érvényre. Ugyanis a tanítási folyamat során a környezet skalár visszacsatolása az egyetlen faktor, ami képes a döntéshozó viselkedésének formálására. Ennek megfelelően a döntéshozó célja, hogy maximalizálja a környezettel való interakciók során összegyűjthető jutalom mennyiségét:

$$G = \sum_{t=1}^{T} \gamma^t r_t \qquad (1)$$

ahol γ^t az úgynevezett discount faktor, amely meghatározza, hogy a jelen döntései hogyan befolyásolják a jövő jutalmát, r_t a t időpontban kapott jutalom. Az 1.ábra a Megerősítéses Tanulás két fő entitása közötti interakciók egymásutánját mutatja be. A folyamat következő képpen zajlik.



1.Ábra: Interakció a környezet és az ágens között.

Az ágens megkapja a környezet aktuális állapotát kódoló reprezentációt $s_t \in \mathcal{S}$, majd ez alapján választ egy döntést $a_t \in \mathcal{A}$, amit továbbít a környezetnek. A környezet végrehajtja a választott beavatkozást, ami egy állapot változást eredményez a környezetben $P(s_t, a_t | s_{t+1})$. Végül a környezet elküldi a döntéshozó számára az új állapot reprezentációt $s_{t+1} \in \mathcal{S}$, illetve a beavatkozás minőségét értékelő jutalmat $r_t \in \mathcal{R}$.

2.2 Policy Gradient algoritmus

A policy-alapú Megerősítéses Tanulásos módszerek több alkalmazási területen értek el kimagasló eredményt az elmúlt időben, aminek köszönhetően komoly érdeklődés mutatkozik a kutatók részéről a technológia iránt. Ezeknél a módszereknél arra hangoljuk be a Neurális Hálózatok súlyait, hogy minél pontosabban becsüljék meg az egyes beavatkozások választási valószínűséget. Következésképpen a Policy Gradient algoritmus dinamikus heurisztikaként működik, mivel nem mutatja be egy adott beavatkozás hosszú távú hatását, hiszen kizárólag az adott állapotban szükséges viselkedést tükrözi a választási valószínűségeken keresztül. A PG algoritmus kimenete tehát egy valószínűségi eloszlás, amelyet a Neurális Hálózat θ paraméterei határoznak meg. Ezért az optimalizálási feladat célja, hogy a Neurális Hálózat súlyait úgy hangoljuk, hogy az maximalizálja az ágens célfüggvényét $J(\theta) = J(\pi_\theta)$. Ez a függvény a következőképpen alakul epizodikus tanító környezetben:

$$J_{\pi_\theta} = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t r_t\right] \qquad (2)$$

A (2) egyenlet alapján látható, hogy az algoritmus lényege, hogy megtalálja a lokális szélsőértéket a kumulált visszacsatolás legnagyobb növekedése irányában. A közelítő függvény θ paramétereinek frissítési szabálya a (Sutton, et al., 2000), (Williams, 1992) alapján határozható meg, amely a következőképpen alakul:

$$\theta \leftarrow \theta + \alpha \nabla \log \pi_\theta(s_t, a_t) \sum_{t=1}^{T} \gamma^t r_t \qquad (3)$$

és a következő képpen működik:

1. Először inicializálja a Neurális Hálózat súlyait, majd a tanulási folyamat elindul a kezdeti s_1 állapotból.
2. Az ágens és a környezet interakcióba lépnek egészen addig amíg egy termináló esemény be nem következik.
3. Kiszámítjuk a kumulált discountol-t jutalom értékét, amit az interakciók során elmentettünk.
4. Kiszámítjuk az egyes állapotokhoz tartozó gradienseket, amelyeket hozzá adunk a buffer-ben kumulált gradiensekhez, majd a megfelelő epizódban frissítjük a gradiensekkel a Neurális Hálózat súlyait.

Ahol $\pi_\theta(s_t, a_t)$ az a_t beavatkozás választási valószínűsége az s_t állapotban. α pedig a tanulási ráta, amely az egyik legfontosabb hiperparaméter hiszen befolyásolja a Neurális Hálózatok súlyain végrehajtott változás mértékét.

2.3 Deep Q-Network

A Megerősítéses Tanulás első jelentős eredménye a Deep Q-Network algoritmushoz kapcsolódik, ugyanis ennek a módszernek a segítségével sikerült először bemutatni, hogy a Megerősítéses Tanulás hatékonyan kombinálható nemlineáris függvény approximátorokkal. Ezzel az algoritmussal arra

hangoljuk be a Neurális Hálózat súlyait, hogy minél pontosabban becsüljék meg az egyes állapotokban a beavatkozások végrehajtásával elnyerhető kumulált visszacsatolás mennyiségét a teljes epizódra nézve. Mindehhez a Bellman egyenletet használjuk fel a tanítás során:

$$Q(s_t, a_t; \theta_t) = r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t; Q_t^-) \qquad (4)$$

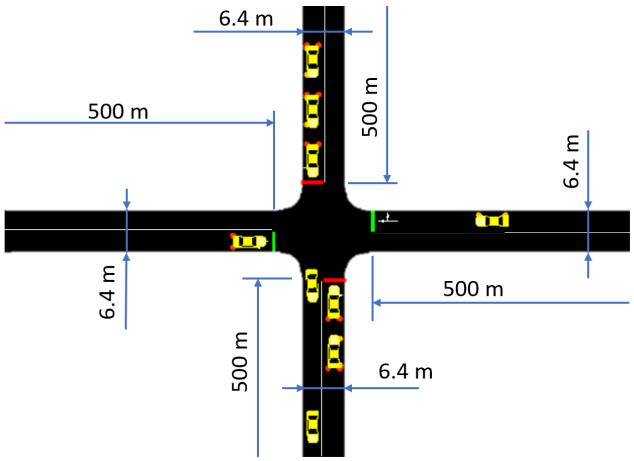
A (4) egyenletben a $Q(s_t, a_t)$ megmutatja, hogy mekkora mennyiségű jutalom nyerhető el az epizód végéig, ha a döntéshozó az s_t állapotban az a_t beavatkozást választja.
A DQN algoritmust közvetett módszernek tekinthetjük, mivel nem tesz közvetlen ajánlásokat a beavatkozások kiválasztására vonatkozóan. A DQN által prediktált értékek inkább egy helyzetértékeléshez hasonlítanak, amely feltárja az egyes beavatkozások hosszútávú megtérülését. Tehát ahhoz, hogy az ágens viselkedést hozhasson létre, valamilyen stratégia segítségével felkel használni a prediktált értékeket. Ez a működési mód azt mutatja, hogy a DQN prediktált értékei inkább globális heurisztikához hasonlítanak, mivel minden értéknek abszolút jelentése van a teljes folyamatra nézve.

3. KÖRNYEZET

Amint azt korábban említettük, a Megerősítéses Tanulás keretrendszerének szüksége van egy környezetre a tanításhoz szükséges adatok legenerálásához. Esetünkben a Simulation of Urban MObility (SUMO) szoftver segítségével hozzuk létre az említett környezetet. A SUMO egy nyílt forráskódú, mikroszkopikus és makroszkopikus forgalomszimulátor, amely dinamikusan bővíthető és könnyen átalakítható az adott felhasználási igények szerint. A SUMO szimulátor lehetővé teszi mind mesterséges mind valós úthálózatok felhasználását a szimulációkban. Ahogy az 1.Ábra mutatja, infrastruktúránk a következőképpen néz ki: A PG és a DQN ágensek Python nyelven kerültek implementálásra a Pytorch Deep Learning könyvtár használatával. A szimulációs környezet pedig két fő részből áll, egy Python-ban implementált komponensből, illetve a SUMO-ban létrehozott kereszteződésből, amely kereszteződést a 2.Ábra szemlélteti. A Python komponens felelős a SUMO környezet irányításáért az ágens beavatkozásai alapján, valamint összegyűjti a megfelelő információkat az állapot reprezentáció létrehozásához, a rewarding koncepció, valamint a kiértékelésben használt metrikák számításához. További előnye a SUMO környezetnek, hogy könnyen létrehozható véletlenszerű terhelés a kereszteződés számára, ami alapvető fontosságú Megerősítéses Tanulás szempontjából, hiszen ez biztosítja a tanítóminta diverzitását.

Környezetünkben négy útszakasz található, amelyek egy kereszteződésben találkoznak, ahogy ez a 2.Ábrán is látható. Mindegyik útszakasz 500 [m] hosszú, és kétirányú sávokat tartalmaz, melyek mindegyike 3,2 [m] széles. A járművek az epizódok során a sávok elején lépnek be a hálózatba. Minden epizód elindítása előtt sávonként meghatározzuk a járművek belépési gyakoriságát, aminek köszönhetően véletlenszerű terhelés fogja érni az adott epizódban a kereszteződést.

A kereszteződésből a ball kanyarodás lehetőségét kivettük, annak érdekében, hogy az egymással ellentétes irányba közlekedő járművek ne okozhassanak többlet torlódást azzal, hogy várakoznak egymásra ballra kanyarodás esetén. Az egysávból érkező járművek között pedig egyenletesen oszlik meg azok aránya, akik egyenesen tovább haladnak, illetve a kereszteződésben jobbra fordulnak.



2.Ábra: A kereszteződés geometriai paraméterei.

A kereszteződést egy jelzőlámpa irányítja, amely vagy a horizontális agy a vertikális sávoknak biztosít zöld időt. Ebből adódóan az ágens célja az, hogy az aktuális forgalomnak megfelelően biztosítsa a zöldidőt az egyes irányoknak.

Értékelés céljából a SUMO szimulátor beépített funkcióit használjuk, amelyek segítenek mérni a forgalom környezetre gyakorolt hatását. A SUMO szimulátor segítésével mérhetjük a CO_2, CO, NO_x, PM_x emissziók kibocsájtását az üzemanyag fogyasztás mellett. Ennek eredményeként a Traffic Signal Control probléma különböző megoldásai összehasonlíthatók azok környezetre gyakorolt hatása szempontjából is a standard utazási idő, várakozási idő és sorhossz mellett.

3.1 Állapot reprezentáció

A Megerősítéses Tanulás keretrendszerében a fejlesztőknek két kulcsfontosságú feladata van. Ezek az állapot reprezentáció, illetve a rewarding koncepció meghatározása. Ugyanis ezek az absztrakciók alapvetően befolyásolják az elérhető eredmény minőségét. Az állapot reprezentáció azért kulcsfontosságú, mert az ágens kizárólag ennek segítségével értheti meg az egyes beavatkozásainak a környezetre gyakorolt hatását és ezen keresztül a környezet működését. A reprezentáció szempontjából tehát nagyon fontos, hogy minden a környezet működését befolyásoló mennyiséget tartalmazzon, azonban ezeknek a meghatározása kizárólag a fejlesztő intuícióitól függ. Ebben a munkában minden egyes sávot, amelyeken a járművek érkeznek a kereszteződés felé, egyetlen értékkel reprezentálunk, ami a sáv telítettsége lesz a [0,1] intervallumra skálázva. A skálázás célja, hogy a minél inkább csökkentsük a tanulás során gyakran előforduló numerikus problémák kialakulásának lehetőségét.

3.2 Beavatkozások

A TSC probléma esetén kétféle beavatkozást típust használnak a szakirodalomban. Az első esetben tetszőlegesen meghatározható a zöld idő az egyes jelzőlámpák esetén egy előre definiált intervallumban. A második esetben, az ágens meghatározhatja, hogy az egyes jelzőlámpák fix ideig milyen program szerint működjenek, majd, ha az lejárt újra lehetőséget kapnak a döntéshozatalra. Vannak hibrid megközelítések is, amelyek a két korábbi megoldást ötvözik. Ebben a munkában a második megközelítést alkalmazzuk.

Két beavatkozás pedig a következő:

1. East-West Green (EWG) zöld jelzést ad a vertikális sávoknak, pirosat pedig a horizontális sávoknak.
2. North-South Green (NSG), zöld jelzést ad a horizontális sávoknak, pirosat a vertikális sávoknak.

Ezekkel a beavatkozásokkal az ágens meghosszabbíthatja az egyes irányok zöld idejét. Az egyes beavatkozások pedig 30 másodpercig tartanak és ennek letelte után változtathat az ágens az aktuális programon. Érdemes megemlíteni, hogy minden fázisváltást sárga, sárga-zöld fázis választ el.

3.3 Rewarding koncepció

Ahogy már említettük az állapot reprezentáció mellett, a rewarding koncepció kialakítása a Megerősítéses Tanulás másik kulcsfontosságú komponense. Ugyanis kizárólag a skalár visszacsatolások segítik az ágenst az optimális viselkedés kialakításában. A szakirodalomban a szerzők többsége várakozási időn, átlagsebességen vagy sor hosszon alapú rewarding koncepciókat használnak. Ezekhez a megközelítésekhez képest új stratégiát mutatunk be. Ennek lényege, hogy az állapot reprezentáció esetén már említett az egyes sávokhoz tartozó telitettségi értékeket egy eloszlásként értelmezzük és az aktuális rewardot az eloszlás szórása alapján számítjuk. Az ágens célja pedig, hogy csökkentse a telítettségekből létrehozott eloszlás szórását. Az egyes beavatkozások (NSG, EWG) ennek köszönhetően képesek befolyásolni a telitettségekből alkotott eloszlás átlagát és szórását azáltal, hogy több zöld időt biztosítanak az egyik vagy másik sávnak. A szórás alapján számított rewardokat pedig a [-1,1] intervallumra skálázva csatolja vissza a környezet az ágens számára.

A szórást azért választottuk, mert a jellemző természetéből adódóan egyensúlyozza ki az egyes sávoknak biztosított zöld időt az aktuális forgalmi szituáció alapján.

A javasolt rewarding koncepció formálisan a következőképpen alakul:

$$R = R_{max} + \frac{(\sigma - \sigma_{min} * (R_{min} - R_{max}))}{(\sigma_{max} - \sigma_{min})} \qquad (5)$$

A (5) egyenletben látható paraméterek a σ szórástól függnek, amit a sávok telítettségéből számítunk ki. A szélsőértékeket pedig az 1.Táblázatban láthatók.

1.Táblázat A reward stratégia paraméterei

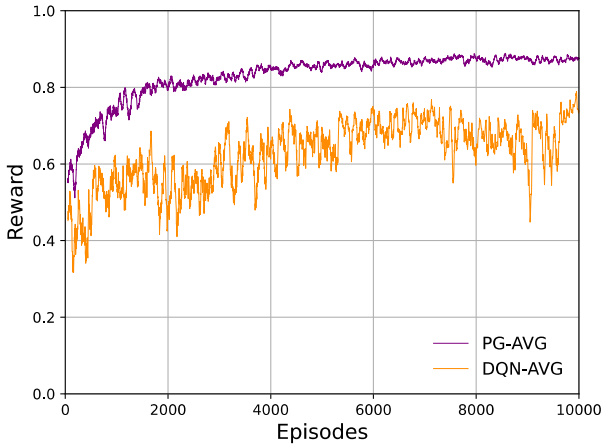
Paraméter	$\sigma < 0{,}1$	$\sigma > 0{,}1$
R_{max}	1	0
R_{min}	0	-1
σ_{max}	0,1	0,5
σ_{min}	0	0,1

A rewarding koncepciók létrehozása során különös figyelmet kell fordítani a szélsőséges esetekre. A mi esetünkben előfordulhat olyan közlekedési helyzet, amikor egyes sávok telítettsége nagyon alacsony míg más sávokban nem is tartózkodnak járművek. Ilyenkor az eloszlás szórása alacsony, vagyis az ágens pozitív rewardban részesül még akkor is, ha annak a sávnak biztosít zöld időt, amiben nem tartózkodnak járművek. Ez egyértelműen káros hatással van az optimális viselkedés kialakítására, hiszen megerősíti az ágenst a hibás működésben. Ezért ezekben az esetekben elvonatkoztatunk a szórás alapú rewarding-tól és -1 es büntetést kap az ágens, viszont nem terminájuk az epizódot annak érdekében, hogy megfelelő mennyiségű adatot gyűjthessen az algoritmus ezekről a szélsőséges esetekről, aminek köszönhetően gyorsabban megtanulhatja a helyes viselkedést.

3.4 A tanítás folyamata

A tanítási folyamat során az egyik legfontosabb szempont, a változatos közlekedési helyzetek generálása. Ugyanis ez teszi lehetővé a robusztus eredmény elérést. A mi esetünkben ez a kereszteződések véletlenszerűen meghatározott terhelésének segítségével kerül megvalósításra. Így a döntéshozónak minden egyes epizódban egy új közlekedési helyzetet kell megoldania.

A tanítási folyamat során az epizódok elején elkezdjük feltölteni a hálózatot járművekkel addig a pontig amig a sávok összesített telítettsége el nem éri a 10%-t. Ez alatt az idő alatt a kereszteződésben az összes lámpa pirosra van állítva. Ha ez a szakasz véget ér az ágens elkezdheti az egyes lámpák irányítását. Egy tanuló epizód addig tart, amíg az összes jármű át nem halad a kereszteződésen. Ha az ügynök nem képes olyan beavatkozásokat választani, amelyek lehetővé teszik az összes jármű áthaladását a kereszteződésen akkor a tanulás folyamat a SUMO szimulátorban számított 2000 [s] után megáll és új epizód kezdődik. A két Megerősítéses Tanulás alapú algoritmus konvergenciája a 3.Ábrán látható.



3.Ábra: A PG és DQN algoritmusok konvergenciája.

A konvergencia karakterisztikák az egyes epizódokban összegyűjtött átlagos reward mennyiségét mutatják. Ezek alapján látható, hogy a PG algoritmus stabilabb és gyorsabban konvergál a DQN algoritmusnál. Ennek oka az lehet, hogy a DQN algoritmus gyakran túlbecsüli a Q-értékeket, aminek köszönhetően a beavatkozás kiválasztása során a prediktált Q-értékek jelentősen eltérhetnek a valóságtól. Ennek ellenére azt láthatjuk az ábrán, hogy mindkét algoritmus magabiztosan konvergál a maximálisan elérhető 1 értékhez. Ezen ábra alapján azt várjuk, hogy a PG algoritmus minden kiértékelési metrika alapján felülmúlja majd a DQN algoritmust.

4. EREDMÉNYEK

Napjainkban kiemelt figyelmet kap az üvegházhatású gázok kibocsátásának csökkentése, amiből egyértelműen adódik, hogy a közúti közlekedésben olyan irányító algoritmusokat fejlesszünk, amelyek figyelembe veszik ezeket a célokat is. Ezt szemelőt tartva a kiértékelések során az egyes megoldásokat emissziós metrikák alapján is összehasonlítottuk.

Annak érdekében, hogy a kiértékelések reprezentatívak legyenek 1000 véletlenszerűen generált epizódon elért átlag eredményeit hasonlítjuk össze a különböző megoldásoknak, de természetesen mindegyik algoritmusnak ugyan azt az ezer epizódot kell megoldania. Viszont a kiértékelés során az egyes epizódok nem addig tartanak, amíg az utolsó jármű is áthalad a kereszteződésen, hanem addig amíg az utolsó jármű is elhagyja a hálózatot.

4.1 Összehasonlítás fenntarthatósági metrikák alapján

A 2.Táblázat és a 3.Táblázat mutatja a TSC probléma megoldására alkalmazott algoritmusoknak a teljesítményét a fentarthatósági metrikák alapján. A táblázatokban láthatók az egyes algoritmusok átlag teljesítményei, illetve azok szórása. Ezeknek a mennyiségeknek a számításához a The Handbook Emission Factors for Road Transport (HBEFA) modellt használja fel a SUMO szoftver.

2.Táblázat Összehasonlítás fenntarthatóság szerint-I

Ágens	CO_2 $[kg]$	CO [kg]	NO_x $[g]$
PG	66,7	2,2	29,1
DQN	70,9	2,3	31,1
SUMO	73,8	2,6	32,3

A táblázatokban bemutatott eredmények alapján látható, hogy a PG algoritmus szinte minden kibocsátással kapcsolatos metrikában felülmúlja a többi megoldást.

3.Táblázat Összehasonlítás fenntarthatóság szerint-II

Ágens	PM_x $[g]$	HC $[g]$	Üzemanyag [l]
PG	1,4	11,4	28,7
DQN	1,5	12,6	30,5
SUMO	1,6	13,7	31,7

Az eredmények alapján a szintén az új reward stratégiával tanított DQN algoritmus is csökkentette a káros anyag kibocsájtást a SUMO beépített algoritmusához képest azonban nem olyan mértékben, mint a PG algoritmus. Az bemutatott eredmények alapján továbbá látható, hogy az éghajlat változás szempontjából legkritikusabb CO_2 kibocsájtást a PG algoritmus 9%-al csökkentette hasonlóan az üzemanyag fogyasztáshoz ami szintén nagyon jelentős környezet kímélő hatással bír.

4.2 Összehasonlítás klasszikus metrikák alapján

Ebben az alfejezetben klasszikus metrikák alapján hasonlítjuk össze a különböző algoritmusok teljesítményét.

Az utazási idő alatt azt az időtartamot értjük, amely a jármű indulásától a céljába való beérkezése alatt telik el. Ezt summázzuk az összes jármű esetére majd pedig leosztjuk a hálózatba belépett járművek számával, így adódik az átlagos utazási idő.

Formálisan a következőképpen alakul:

$$ATT = \frac{1}{N_{veh}} \sum_{j=0}^{N_{veh}} t_{j,start} - t_{j,end} \qquad (6)$$

A várakozási idő az összesített állásidő az indulástól a kereszteződésen való áthaladásig. Egy jármű pedig akkor van várakozó állapotban, ha sebessége kisebb, mint 0,1 [m/s].

Ezt a hálózatba belépő összes járműre számítjuk ki majd pedig summázuk végül pedig leosztjuk a hálózatba belépett járművek számával.

Formálisan a következő képpen alakul:

$$AWT = \frac{1}{N_{veh}} \sum_{j=0}^{N_{veh}} WT_j \quad (7)$$

A sor hossza az adott sávban várakozó járművek számaként kerül számításra.

Az egyszerűség kedvéért az összes sáv sorhosszát egyetlen értékbe foglaljuk össze.

Formálisan a következő képpen alakul:

$$QL = \sum_{l=0}^{L} q_l \quad (8)$$

A 4.Táblázatban bemutatott eredmények azt mutatják, hogy a PG ágens, mind átlagos várakozási idő, mind sor hossz szempontjából felülmúlja a DQN módszert, illetve a SUMO beépített algoritmusát. Az átlagos utazási idő szempontjából a SUMO beépített algoritmusa felülmúlja a DQN ágenst és közel azonos eredményt ér el, mint a PG algoritmus.

A különbség a SUMO beépített algoritmusa és a PG ágens között az eredmények szórása alapján már szignifikánsabb. Ez azt mutatja, hogy a PG ágens sokkal kiegyensúlyozottabban irányítja a kereszteződést a SUMO algoritmusához képest.

4.Táblázat Összehasonlítás standard metrikák szerint

Ágens	Utazási idő [s]	Várakozási idő [s]	Sor hossz [db]
PG	141,3±42	29,4±21	10,8±8
DQN	150,7±37	37,8±22	13,2±8
SUMO	147,7±45	46,8 ±49	16,1±16

A különbségek a megoldások között abból adódhatnak, hogy a SUMO beépített algoritmusa sokkal hosszabb zöld időkkel dolgozik a PG és DQN ágensekhez képest. Ebből adódik, hogy ezek az algoritmusok nem csupán a bemutatott metrikák alapján, hanem a járművezetők elégedettsége alapján is jobb eredményt érnek el, hiszen az egyes gépkocsivezetők kevesebb időt töltenek az aktuálisan kialakult torlódásban.

5. KONKLÚZIÓ

Ebben a munkában egy új rewarding stratégiát dolgoztunk ki Megerősítéses Tanulás alapú algoritmusok esetére. Melynek lényege, hogy folyamatosan kiegyenlítse az egyes sávoknak biztosított zöld időt azok telítettsége alapján. Az új stratégiával egy PG és egy DQN algoritmust tanítottunk be a TSC probléma megoldására, majd ezeket összehasonlítottunk a SUMO szoftver beépített algoritmusával. Az egyes megoldások teljesítményét pedig mind klasszikus metrikák, mint például átlagos utazási idő, átlagos várakozási idő, sorhossz, mind fentarthatósági metrikák alapján összehasonlítottuk. Az eredmények azt mutatták, hogy az új stratégiával tanított PG és DQN ágensek minden emissziós metrikában felülmúlták a SUMO beépített algoritmusát. A legjobb eredményt pedig a PG ágens érte el. A statisztikai kiértékeléseken túlmenően ezek a Megerősítéses Tanulás alapú algoritmusok a közlekedésben résztvevők nagyobb elégedettségét képesek elérni, mivel a várakozási idő csökken az egyes gépkocsivezetők esetében. A kutatási feladat folytatását az irányítási feladat komplexitásának növelésében látjuk. Ezt egy több összekapcsolt kereszteződésből álló hálózat létrehozásával tervezzük megvalósítani, valamint a felhasznált algoritmusok tovább fejlesztésével, amit Multi-Ágens Megerősítéses Tanulás alkalmazásával kívánunk elérni. A választásunk azért esett erre a módszerre mert, itt lehetőségünk van az egyes kereszteződésekhez önálló ágenseket rendelni és így feloldani a beavatkozások exponenciális növekedésének problémáját, amit a kereszteződések számának növekedése okoz.

KÖSZÖNÖETNYILVÁNÍTÁS

EFOP-3.6.3-VEKOP-16-2017-00001: Tehetséggondozás és kutatói utánpótlás fejlesztése autonóm járműirányítási technológiák területén - A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

HIVATKOZÁSOK

Tan, T. és mtsai., 2019. Cooperative deep reinforcement learning for large-scale traffic grid signal control. *IEEE transactions on cybernetics*, pp. 50(6), 2687-2700..

Cai, C., Wong, C. K. & Heydecker, B. G., 2009. *Adaptive traffic signal control using approximate dynamic programming.* hely nélk.:Transportation Research Part C: Emerging Technologies, 17(5), 456-474..

Genders, W. & Razavi, S., 2016. Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint*, p. arXiv:1611.01142..

Zhang, L. Y. Y. & Chen, S., 2013. *Robust signal timing optimization with environmental concerns.* hely nélk.:Transportation Research Part C: Emerging Technologies 29, 55-71..

Sutton, R. S., McAllester, D. A., Singh, S. P. & Mansour, Y., 2000. Policy gradient methods for reinforcement learning with function approximation. *In Advances in neural information processing systems*, pp. (pp. 1057-1063)..

Guo, J. & Harmati, I., 2020. Comparison of Game Theoretical Strategy and Reinforcement Learning in Traffic Light Control. *Periodica Polytechnica Transportation Engineering*, pp. 48(4), 313-319..

Al-Ghussain, L., 2019. *Global warming: review on driving forces and mitigation.* hely nélk.:Environmental Progress & Sustainable Energy,38(1), 13-21.

Ekeila, W., Sayed, T. & Esawey, M. E., 2009. *Development of dynamic transit signal priority strategy..* hely nélk.:Transportation research record, 2111(1), 1-9..

Fehér, Á., Aradi, S. & Bécsi, T., 2020. Fast Prototype Framework for Deep Reinforcement Learning-based Trajectory Planner. *Periodica Polytechnica Transportation Engineering*, pp. 48(4), 307-312..

Hannappel, R. (. A. T. i. o. g. w. o. t. a. i. I. A. C. P. (. 1. N. 1. p. 0., 2017. *The impact of global warming on the automotive industry. In AIP Conference Proceedings.* hely nélk.:AIP Publishing LLC..

Jin, J. & Ma, X., 2017. A group-based traffic signal control with adaptive learning ability. *Engineering applications of artificial intelligence*, pp. 65, 282-293..

McKenney, D. & White, T., 2013. *Distributed and adaptive traffic signal control within a realistic traffic simulation.* hely nélk.:Engineering Applications of Artificial Intelligence, 26(1), 574-583..

Van der Pol, E. & Oliehoek, F. A., 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems*, p. (at NIPS 2016)..

Villalobos, I. A., Poznyak, A. S. & Tamayo, A. M., 2008. Urban traffic control problem: a game theory approach. *IFAC Proceedings Volumes*, pp. 41(2), 7154-7159.

Wang, F., Tang, K., Li, K. L. Z. & Zhu, L., 2019. *A group-based signal timing optimization model considering safety for signalized intersections with mixed traffic flows..* hely nélk.:Journal of advanced transportation.

Wan, C. H. & Hwang, M. C., 2018. Value-based deep reinforcement learning for adaptive isolated intersection signal control.. *IET Intelligent Transport Systems*, pp. 12(9), 1005-1010..

Williams, R. J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, pp. 8(3), 229-256..