



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
KÖZLEKEDÉSMÉRNÖKI ÉS JÁRMŰMÉRNÖKI KAR
KÖZLEKEDÉS- ÉS JÁRMŰIRÁNYÍTÁSI TANSZÉK

TUDOMÁNYOS DIÁKKÖRI KONFERENCIA

Adaptív jelzőlámpa-irányítás Megerősítéssel Tanulás és
MCTS alkalmazásával

Pelenczei Bálint

Konzulens: Kővári Bálint

2021

Köszönetnyilvánítás

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.

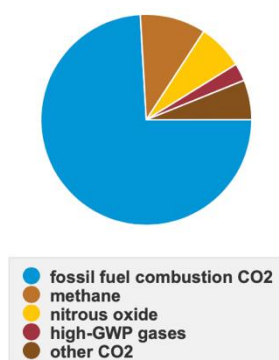
Tartalomjegyzék

Köszönetnyilvánítás.....	1
Tartalomjegyzék.....	2
1 Bevezetés.....	3
1.1 Motiváció	3
1.2 Kapcsolódó kutatások, eredmények	4
1.3 Kontribúció.....	9
2 Fejlesztési környezet	11
2.1 Állapot-reprezentáció	12
2.2 Beavatkozás típusok	13
2.3 Rewarding koncepció	14
2.4 Tanítás	15
3 Algoritmusok	17
3.1 Megerősítéses tanulás	17
3.1.1 Policy Gradient algoritmus bemutatása	19
3.2 Fa típusú keresés alapú megoldások.....	20
3.2.1 Monte Carlo Tree Search algoritmus bemutatása	21
3.3 Baseline kontrollerek	23
4 Eredmények.....	24
4.1 Fenntarthatósági paraméterek összehasonlítása.....	24
4.2 Klasszikus mérőszámok összehasonlítása.....	26
5 Konklúzió.....	31
Táblázatjegyzék	33
Ábrajegyzék	34
Rövidítések.....	35
Hivatkozások.....	36

1 Bevezetés

1.1 Motiváció

A 21. század egyik legmeghatározóbb kihívásaként számontartott jelenség a globális felmelegedés. Ez a fogalom nem más, mint a Föld klímarendszerének hosszútávú megváltozása az emberi beavatkozások következtében. Ezen beavatkozások alatt elsősorban fosszilis energiahordozók elégetését értjük, amelyek még inkább az üvegház hatású gázok szintjét a Föld atmoszférájában, ezzel felgyorsítva az öngerjesztő folyamatot. A különböző üvegház hatású gázok eloszlását mutatja az 1. ábra, amelyen világos barna színnel van jelölve a metán, sárgával a nitrogén-oxidok, kézzel a fosszilis energiahordozók elégetéséből származó szén-dioxid, sötét barnával az egyéb módon kibocsátott szén-dioxid, illetve pirossal az egyéb üvegház hatású gázok. Fontos kiemelni, hogy a teljes mennyiség mintegy 80%-át egyedül a szén-dioxid fedi le, ezzel az egyik legnagyobb hozzájárulója a klímaváltozásnak [2].



1. ábra: Az üvegház hatású gázok eloszlása [1]

A Nemzetközi Energia Ügynökség 2020-as felmérése alapján [3] ennek a szén-dioxid kibocsájtásnak körülbelül 24%-a származik közvetlenül a közlekedési szektorból, amelyből körülbelül háromnegyed arányt tesz ki a közúti forgalom. Az Európai Bizottság 2019-es kimutatásai szerint [4] a szigorú jogi szabályozások ellenére a közlekedés részaránya az üvegházhatású gázok kibocsájtásában 0,9%-kal növekedett az azt megelőző évhez képest a technológiai fejlődés ellenére is.

Egy másik fontos probléma – habár szoros összefüggésben állnak egymással – maga a légszennyezés, amelyben hasonló részarányt képvisel a közúti közlekedésben résztvevő járművek összkibocsátása [5]. A légszennyezést okozó anyagok [6] – kén-oxidok, nitrogén-oxidok, szén-monoxid, szénhidrogének, szilárd anyagi részecskék – mind végtermékei a belső égésű motoroknak, amelyeknek nem csak a környezetre, hanem az emberi szervezetre

gyakorolt hatásai is jelentősek. Ezen kívül beszélhetünk a környezeti hatásokról is: szmog (füstköd), savas eső, eutrofizáció, ózonréteg elvékonyodása, illetve globális felmelegedés. Az egyszerű felsorolás ellenére mindegyik elem külön-külön is óriási terhet jelent egyaránt a bolygónkra, illetve ránk, emberekre nézve is.

Számos megközelítés, megoldás létezik már a felsorolt problémák mérséklésére: ilyenek például az alternatív energiaforrással működő járművek (elektromos energia [7], hidrogén [8], illetve földgáz alapú meghajtások [9]), a magasabb hatásfokkal üzemeltethető környezetkímélőbb motorok használata [10], a közösségi autómegosztó szolgáltatások igénybevétele [11], az alternatív közlekedési eszközök használata [12], illetve Intelligens Közlekedési Rendszerek (Intelligent Transportation System - ITS) alkalmazása.

Az imént említett káros kibocsátások azonban jóval inkább a városi környezetekben fejtik ki hatásukat [13], hiszen a – nagyobb népsűrűség okán – jóval kisebb kapacitású közlekedési infrastruktúra miatt könnyen telítődik a közúthálózat. A telített és szabad áramlású útszakaszok forgalmi szituációi közötti differencia nem csak utazási időben, várakozási időben, átlagsebességben, vagy a sorok hosszával mérve drasztikus, hanem a kibocsátott károsanyagok mennyiségében is számottevő az akár 60%-os növekedés [14].

Mivel sok tényezőt (szén-dioxid-, nitrogén-oxidok-, szilárd anyagi részecskék emissziója, illetve maguk a fogyasztási paraméterek) legnagyobb mértékben a gyorsulás, illetve annak mértéke befolyásolja [15], célszerű azokon a helyszíneken beavatkozni, ahol ezen kinematikai paraméter relatíve nagy fluktuációval rendelkezik – a kereszteződésekben. A jelenleg használatos fix ciklusidejű jelzőlámpák nem tudnak alkalmazkodni a forgalom dinamikájához, viszont különböző adaptív forgalomirányítási módszerekkel fejleszthetők a bejövő forgalom függvényében.

Ezek a módszerek képesek a torlódások csökkentése révén nem csak az életünket, hanem a környezetünket is pozitív irányban befolyásolni számos területen. Ilyenek például a kisebb ökológiai lábnyom generálása, biztonságosabb közlekedés, hozzájárulás csökkentése a globális felmelegedéshez, hatékonyabb időmenedzsment, az emberek egészségügyi állapotának javítása, illetve az őket körülvevő állat- és növényvilág védelme.

1.2 Kapcsolódó kutatások, eredmények

A probléma megoldására az eddigi eredmények alapján tapasztalt pozitív tulajdonságai miatt – amik a későbbiekben kifejtésre kerülnek – Gépi Tanulást (Machine Learning - ML) alkalmazunk, ezen belül pedig Megerősítési Tanulást (Reinforcement Learning - RL), illetve fa alapú kereső algoritmust. Ebben a fejezetben egy rövid áttekintést végzünk a

szakirodalomban eddig használt eljárásokról, megoldásokról, amelyek az általunk használt módszerekkel szorosan összefüggnek.

Gépi tanulás alkalmazásával számos más közlekedés-irányítási feladat kapcsán lehet találkozni, de kimagasló eredmények születtek már a konkrét probléma megoldására is, amelyre a későbbiekben a TSC angol rövidítéssel hivatkozunk (Traffic Signal Control).

A TSC problémát több különböző megközelítéssel oldották meg az idők folyamán. A legkorábbi megoldások szabályalapú rendszereket alkalmaztak. Ezek az algoritmusok a kulcsállapotokhoz egyedi beavatkozásokat rendeltek (ezáltal állapot-beavatkozás párokat létrehozva), amelyek felhasználhatók az elvárt teljesítménykritérium eléréséhez. [16] -ban bemutattak egy új, REALBAND elnevezésű rendszert, amely a közlekedési hálózatban csoportokat azonosít, majd a forgalmi adatok fúziója és szűrése segítségével predikciót ad azok várható érkezési idejére a kereszteződéshez. Emellett a cikk megoldásokat kínál az egymásnak ellentmondó csoportok érkezésére való predikció esetére is. [17] szerzői kifejlesztettek egy dinamikus forgalomirányító-jel elsőbbségi rendszert, amely képes kezelni a valós idejű forgalmat és a tranzitfeltételeket.

Egy másik alaposan kutatott megoldás az úgynevezett Dinamikus Programozás (Dynamic Programming - DP), elsősorban annak rugalmassága, finomhangolhatósága és annak lehetősége miatt, hogy a forgalmi viszonyok széles körében alkalmazható lehet. Ezek mellett előnye a teljesítménymutatók széles körének felhasználási lehetősége. [18] -ban bemutatásra kerül a RHODES, amely szétbontja és átszervezi a TSC problémát hierarchikus alproblémákká. A tanulmány azt is kimutatta, hogy a RHODES késés tekintetében felülmúlja az általuk vizsgált félig-aktuált vezérlőket. [19] -ben a szerzők egy dinamikus programozáson alapuló megközelítést alkalmaztak, amely csökkentette a járművek kumulált késedelmi idejét, és az értékfüggvény approximálásával mérsékelte a szükséges számítási kapacitást.

A szimuláción alapuló megközelítések is széles körben alkalmazottak a problémára. [20] a CRONOS algoritmus viselkedését elemzi, aminél kimutatták, hogy annak felhasználásával a késedelmi időt csökkenteni lehet mind lokális, mind pedig központosított irányítási módszerekhez képest. [21] -ben a szerzők olyan módszert mutatnak be, amely lokális kommunikációt használ az érzékelők és a közlekedési lámpák között, és képes forgalom-reszponzív módon szabályozni őket komplex, valós közlekedési hálózatokban.

Sok kutató alkalmazott Genetikus Algoritmusokat (Genetic Algorithm - GA) a TSC probléma megoldásához. [22] -ben a szerzők egy csoportalapú optimalizációs modellt vizsgáltak, amely vegyes és kiegyensúlyozatlan sűrűségű forgalom esetén oldja meg a kötött fázissorenden alapuló jelvezérlés biztonsági és hatékonysági problémáit. [23] -ban egy

szimuláción alapuló genetikus algoritmust használnak fel egy többcélú optimalizálási modellel, amely már figyelembe veszi a késedelmet és a kibocsátási paramétereket is.

A Multi-Ágens Rendszerek (Multi-Agent System - MAS) használata szintén általános, elterjedt megközelítésnek számít a TSC-probléma kezelésének tekintetében. [24] egy csoportalapú jelvezérlési módszert mutat be, ahol a jelcsoportokat külön ágensekként modellezik, és ezek az ágensek a kereszteződés szintjén döntenek az adott forgalmi szituációnak megfelelően.

Egy másik jól ismert módszer a játékelmélet alkalmazása. [25] szerzői a TSC problémát egy nem-kooperatív játékként fogalmazták meg, ahol az ágensek célja a sorok hosszának minimalizálása.

A legújabb trend azonban a TSC probléma megoldására a Megerősítéses Tanulás alapú algoritmusok alkalmazása. Az RL kiemelkedő eredményeket mutatott ezen a területen és számos más közlekedéssel kapcsolatos alkalmazásban is. [26] -ban a szerzők nagy felbontású eseményalapú adatok felhasználását mutatták be az állapot-reprezentációhoz, és ezzel felülmúlták mind a fix, mind az aktuált-vezérlőket a kialakult sorok hosszában és a teljes késedelmi idő tekintetében. A szerzők [27] -ben dinamikusan változó discount-faktort alkalmaztak a Bellman-egyenlethez, amely a DQN algoritmus célértékeinek generálásáért felelős. Összehasonlították eredményeiket a fix-ciklusidejű vezérlővel és az eredeti DQN-ágenssel, ami arra világított rá, hogy a továbbfejlesztett ágens jobb eredményeket ért el kumulált késedelmi időben és átlagos áteresztőképességben, mint bármely más megoldás. [28] új állapot-reprezentációt mutatott be, amelyet Diszkrét Forgalomállapot-Kódolásnak neveznek. Összehasonlítva ugyanazzal az algoritmussal, amely feature-based vektor állapot-reprezentációt tartalmaz, az új reprezentáció jobb teljesítményt ér el az átlagos kumulált késedelmi idő és az átlagos sorhossz tekintetében. [29] összehasonlította a játékelmélet- és az RL alapú megoldásokat fix ciklusidejű-vezérléssel egyetlen kereszteződés esetén, és az eredmények azt mutatták, hogy mindkét módszer kiemelkedő teljesítményt nyújt sorok hosszában a fix ciklusidejű vezérléshez képest. A szerzők [30] -ban kifejlesztettek egy hibrid beavatkozási-térrel rendelkező ágenszt, amely egyesíti a diszkrét és a folytonos beavatkozási-tereket, és összehasonlították megoldásukat a Mély Megerősítéses Tanulási (Deep Reinforcement Learning - DRL) algoritmusokkal, diszkrét- és folytonos beavatkozási-terekkel és fix ciklusidejű vezérléssel. Az eredmények azt mutatták, hogy a bemutatott módszer felülmulta az összes többi az átlagos utazási idő, a sorok hossza és az átlagos várakozási idő tekintetében. [31] -ben összehasonlításra kerültek hagyományos Megerősítéses Tanulási algoritmusok a DQN algoritmussal az egyszerű kereszteződések TSC problémája mentén, és

az eredmények azt mutatták, hogy a DQN algoritmus nagyobb mértékben képes a kumulált késedelmi idő csökkentésére.

Az RL-t alkalmazták már több kereszteződésű TSC probléma kezelésére is. [32] szerzői új rewarding koncepciót és a DQN algoritmust mutatták be, amelyet egy koordinációs algoritmussal kombinálva több kereszteződés egyidejű vezérlésére használtak fel. [33] -ban a Multi-Ágens A2C algoritmus kerül bemutatásra a több kereszteződéses TSC probléma megoldására, amelynek eredményei alapján összehasonlításra kerül más DRL algoritmusokkal mind szintetikus forgalmi hálózaton, mind pedig egy valós forgalmi hálózaton. [34] egy új, kooperatív DRL keretrendszert mutat be Coder néven, amely az eredeti RL problémát részproblémákká bontja. A tanulmányban összehasonlítják a saját módszerüket más RL megközelítésekkel, valamint szabályalapú módszerekkel, mint például a fix ciklusidejű forgalomirányítás. A TSC-probléma további, tisztán RL-alapú megoldásait mutatják be: [35].

Ezek a kutatások bizonyítják, hogy az RL sikeresen alkalmazható ezen a területen. Az RL TSC-problémában való gyakori alkalmazásának oka és alapja pedig a megerősítéses tanulás nyújtotta rendkívüli sokoldalúság a szekvenciális döntéshozatali problémák területén, amelyet maga az RL-keretrendszer hordoz magában. Ezek egyike a valósidejűség, amely az előre behangolt függvény-approximátorok (neurális hálózatok) gyakorlatban elhanyagolható mértékű állapotfelmérési késedelmének köszönhető. Emellett fontos tulajdonság a skálázhatóság, aminek következtében (elméletileg) szinte tetszőleges számú és komplexitású közlekedési csomópontra alkalmazhatóak, illetve a magas szintű generalizáló képesség, amely a kezdetektől fogva szintén ismert és kutatott téma.

Az említett előnyök miatt sokan kutatják a témát, így napjainkra az RL-alapú megoldásoknak hatalmas szakirodalma van ezen a területen. A TSC-problémára nézve megfelelő, rendelkezésre álló algoritmusok sora az RL fejlődésével együtt bővült. Egyszerű érték-alapú és policy-alapuló megoldásokból indult ki, mint például a DQN és a Policy Gradient (PG), majd az Actor-Critic megközelítések következtek. Ugyanez történt az állapot-reprezentációk esetében is. A feature-based értékvektorok mellett a képszerű ábrázolások is kezdtek elterjedni a Konvolúciós Neurális Hálózatok (Convolutional Neural Network - CNN) kimagasló sikerének köszönhetően.

A megerősítéses tanulás TSC problémára való alkalmazásának lehetőségével – ahogy az fentebb kifejtésre került – relatíve régóta, mélyebben foglalkozik a tudomány, azonban ez nem mondható el a kereső algoritmusok családjáról, és specifikusan a Monte Carlo Tree Search (MCTS) algoritmusról sem. Ennek megválaszolására a későbbiekben kerül majd sor, ám előbb következzenek egy rövid áttekintés a témáról.

Az RL-hez hasonlóan itt is megjelennek játékelmélet-alapú megoldások, ilyen például [36], amelyben a szerzők egy CoSIGN-nak elnevezett iteratív, kereső algoritmust mutattak be 75 közlekedési lámpa jelének együttes optimalizálására, majd alkalmazták azt valódi környezetben. Az alkotók a TSC-t egy diszkrét optimalizációs problémának fogalmazták meg, amelyben a beavatkozások jelfázisok. Így tehát az algoritmus iteratív módon megkeresi az egymással egyidejűleg legjobban együttműködő jelfázisokat, és az így kialakult egyedi fázistervekkel irányítja a közlekedési hálózat egészét. Az algoritmus alkalmazhatóságának korlátját a szükséges számítási kapacitás okozza: elméletileg 1000 darab párhuzamosan működő CPU-val a cikkben vizsgált 20 iteráció nagyjából 10 percet venne igénybe, ami nagyjából 20%-os átlagos utazási idő csökkenést eredményezne.

[37] egy raj-intelligencia alapú, Hangya Kolónia elnevezésű optimalizációs algoritmust alkalmaz sikeresen a TSC probléma megoldására. Ez egy metaheurisztikus algoritmus, amelyet nagy számítási kapacitást igénylő kombinatorikus optimalizációs problémák megoldására fejlesztettek ki. Lényege, hogy az első iteráció folyamán a „mesterséges hangyák” (ágensek) cselekedetei véletlenszerűek, míg végül a lépések végi tapasztalat-frissítések hatására, egymás tapasztalatait felhasználva az optimális döntés felé konvergálnak.

Egy kereső algoritmust mutatnak be [38] -ban is, ahol a jármű-infrastruktúra kommunikáció révén létrejött prioritási kérelmeket kell feldolgoznia a forgalomirányító berendezésnek. Az algoritmus először feldolgozza a prioritási kérelmeket, majd egy iteráció sorozat által kiválasztja az optimális fázissorrendet úgy, hogy közben minimalizálja a teljes késedelmi időt.

Korábban az RL kapcsán esett már szó a dinamikus programozásról, [39] -ben azonban azt egy fa alapú keresőalgoritmussal kombinálják, amely kisebb állapottér által (többlépéses iterációk, egyes állapotok kiiktatása) képes csökkenteni a számítási kapacitás igényt, ezáltal megvalósítva a közel valós idejű futást.

[40] -ben egy Monte Carlo Tree Search alapú jelo optimalizációs modellt mutatnak be, amely a csomópontokba befutó újonnan létrejött (tehát relatíve rövid) sávok túltelítődését, ezáltal a többi sáv szabad mozgását hivatott elősegíteni. Ezen útszakaszok sorhossz paramétereiben, illetve az átlagos utazási idők és kumulált késedelmi idő tekintetében szintén felülmúlták az általuk használt szimulációs szoftver (Synchro) beépített jelo optimalizációs algoritmusát. Fő hátránya, hogy a számítási kapacitást figyelembe véve csak időzített vezérlésre alkalmas.

A Monte Carlo Tree Search nevezetű algoritmussal foglalkoztak tovább a szerzők [41] -ben, amelyben főútvonali kereszteződésben vizsgálták az ember vezette járművek, és az

autonóm, infrastruktúrával együttműködő járművek alkotta vegyes forgalmat irányító jelzőlámpákat, és elérték az algoritmus valós idejű futtathatóságát. Ezt az általuk kifejlesztett exploration-exploitation dilemmát kiegyensúlyozó kalibrációs modell felhasználásával, illetve a fa terjeszkedését determináló modullal érték el, miközben 6,5%-os üzemanyag megtakarításhoz és 3,5% átlagos utazási idő csökkenéshez jutottak.

Ezek a kutatások, illetve a kutatások tendenciái is jól mutatják, hogy habár az MCTS TSC problémára való alkalmazásában nagy potenciál található, annak kihasználása erőforrás igényes feladat, amely az idő előre haladtával bekövetkező hardveres fejlődés hatására egyre elérhetőbbé válik.

Teljesítmény tekintetében megkérdőjelezhetetlen tény azonban, hogy az ágensek által alkalmazott rewarding koncepció tekinthető mind az RL, mint pedig az MCTS egyik leglényegesebb komponensének. Számos szakirodalmat tanulmányozva azt tapasztaltuk, hogy a leggyakrabban alkalmazott rewarding koncepciók a kezdetektől fogva nem változtak. Ma is elsősorban a forgalom áramlásával kapcsolatos mérőszámokon alapulnak, mint például a várakozási idő, a sorok hossza, az áteresztőképesség és a fázisok frekvenciája, ezért érdemesnek tartottuk ebből a szempontból is megvizsgálni és elemezni a problémát.

Célunk tehát az, hogy az így létrejött felvetésünket felhasználva adaptív forgalomirányítási technológiákkal megvizsgáljuk és egy új rewarding koncepcióval előrehaladást érjünk el a kereszteződések forgalomirányítási problémájában, miközben kiemelt hangsúlyt fektetünk a TSC-probléma környezeti vonatkozásaira is.

1.3 Kontribúció

A dolgozat kontribúciója több részből tevődik össze. Először egy, a problémára nézve új megközelítésű rewarding koncepció [42] továbbfejlesztését mutatjuk be a TSC problémára, amely egy érdekes betekintést mutat ebbe a bizonyos irányítási feladatba.

Másodszor, egy Policy Gradient ágens kerül betanításra ezzel az új rewarding koncepcióval, majd bemutatjuk annak teljesítményét egy szintén az új koncepciót alkalmazó MCTS ágenssel együtt. Ezen ágensek három másik rewarding módszer segítségével kerülnek kiértékelésre és összehasonlításra, amelyek a következő mérőszámokon alapulnak: sorhossz, átlagsebesség, és sávkihasználtság szórása. Ezek közül az első a szakirodalom által leggyakrabban használt rewarding stratégia, amely már bizonyította sikerességét ebben a feladatban [43]. A második egy szintén sokat alkalmazott módszer, míg legutóbbi az eredeti szórás alapú koncepció, amely továbbfejlesztésre került. Ezeken felül megoldásainkat további módszerekkel hasonlítjuk össze: egy idővesztésen alapuló aktuált vezérlő algoritmussal, és

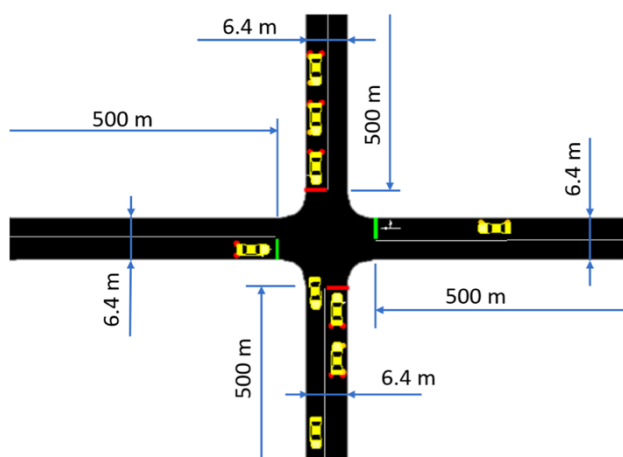
a fix ciklusidejű jelzőlámpa vezérléssel. Ezen túlmenően, az eredmények értékelése nem csak a klasszikus metrikákra fókuszál, hanem kitér a különböző megoldások fenntarthatósági lehetőségére is, mint például a CO₂-, NO_x-, és CO-emissziók. Az értékelés ilyen formája a fenntartható közlekedés nevében már kényszerűnek is nevezhető, mivel a közlekedési- és szállítási szektorok kibocsátásának csökkentése világszerte komoly aggodalmakat kelt manapság.

Harmadszor, a Monte Carlo fakeső algoritmus egy érdekes, de még nem ismert tulajdonságára hívjuk fel a figyelmet.

A dolgozat a következőképpen szerveződik: a 2. fejezetben bemutatjuk a tanító környezetet, valamint annak legfontosabb összetevőit, például az állapot-reprezentációt, a beavatkozási lehetőségeket és az új rewarding koncepciót. A 3. fejezetben részletesen ismertetjük az alkalmazott stratégiákat, végül a 4. fejezet összehasonlítja a különböző módszerek teljesítményeit. Az 5. rész tartalmazza az eredményekből levont következtetéseket és a továbblépési, továbbfejlesztési lehetőségeket veti fel a téma kapcsán.

2 Fejlesztési környezet

Amint azt korábban tárgyaltuk, mind az RL keretrendszernek, mint pedig az MCTS tervező algoritmusnak szüksége van egy környezetre az interakciók lebonyolításához. Esetünkben ezt a feladatot a Simulation of Urban MObility (SUMO) [44] elnevezésű szimulációs szoftverben létrehozott kereszteződés látja el. A SUMO egy nyílt forráskódú, egyaránt mikroszkopikus, és makroszkopikus forgalmi szimulátor, amely dinamikusan bővíthető és nagymértékben testre szabható. A közösség által fejlesztett API-kon keresztül különféle közlekedési hálózatokat lehet egyedileg építeni, vagy akár teljes, létező hálózatokat is betölthetünk a szoftverbe. Amint azt a 3. ábra mutatja, infrastruktúránk a következőképpen néz ki: a PG és MCTS ágensek pythonban vannak implementálva, előbbi a Pytorch Deep Learning könyvtár használatával. Emellett maga a környezet két fő részből tevődik össze: egy python komponensből és a SUMO-ban létrehozott közlekedési csomópontból, lásd 2. ábra. A python komponens felelős a SUMO környezetbe történő beavatkozásokért (az ágens döntései alapján), amit a TraCI interfész segítségével tud megvalósítani. Emellett feladata még a SUMO-környezetből származó információk begyűjtése az állapot-reprezentáció, a rewarding formula, illetve a teljesítményértékelések számára. Érdeemes megjegyezni ezen felül, hogy a szimulációk könnyen véletlenszerűsíthetők, ezzel egy optimális választást nyújtva az RL számára, mivel annak robusztus kimeneteléhez sok különböző féle tapasztalatra van szüksége a tanítási folyamat során.



2. ábra: A SUMO kereszteződés felépítése [42]

Az általunk használt hálózat a következőképpen néz ki: négy útszakasz van, amelyek egy kereszteződésben találkoznak (2. ábra). Minden szegmens 500 méter hosszú és kétirányú, egyenként 3,2 méter széles sávokat tartalmaz. Mindegyik szegmens végén található egy-egy

kiindulási (spawn) pont a szimulációba belépő járművek számára, amik minden egyes lépésnél véletlenszerűen generálódnak. Minden epizódban, belépési pontonként megadjuk annak a valószínűségét, hogy az adott belépési pontra adott időlépésben generálódjon-e új jármű. Ez azt jelenti, hogy az 1-es valószínűség esetében minden szimulációs időlépésben ($dt = 1 s$) egy járművet indítunk el az adott belépési pontról. Az elindított járművek úticélja szintén az egyenletes eloszlást követi, ami azt jelenti, hogy közel egyenlő számú jobbra kanyarodó és egyenesen a kereszteződésben keresztül haladó autót generálunk. A balra kanyarodás lehetőségét kihagyjuk a hálózathoz annak érdekében, hogy a prioritási sorrend felállításával járó idővesztés ne generáljon további torlódásokat. Ennek köszönhetően az ágensek képesek a saját döntéseik környezetben megnyilvánuló hatásainak hatékonyabb megfigyelésére, ami tovább gyorsítja és pontosítja a tanulási folyamatot, illetve az MCTS esetében segíti a konvergenciát. A kereszteződést jelzőlámpával irányítjuk, melynek optimális vezérlése lesz az ágensek feladata. Értékelési célokra a SUMO szimulátor néhány beépített funkcióját használjuk, amelyek segítségével monitorozhatjuk a közúti forgalom környezetre gyakorolt hatását. Ezzel az eszközzel olyan metrikákat nyerünk ki, mint a CO₂-, CO-, NO_x-, HC-, PM_x-kibocsátás és az üzemanyag-fogyasztás. Ennek eredményeként a TSC probléma különböző megoldásai nem csak a klasszikus minőségjelzőkkel, hanem (a 4. fejezetben részletezett) fenntarthatósági metrikák alapján is összehasonlíthatók lesznek.

2.1 Állapot-reprezentáció

A Megerősítéses Tanulás keretrendszerében két feladatot kell megoldanunk: az állapot-reprezentáció-, illetve a rewarding koncepció megfogalmazását. Az állapot-reprezentáció definiálása lényegében az irányítási probléma leírásának egy absztrakciója, amely tartalmazza az adott probléma lényeges tulajdonságait. Ebből eredően csupán a kutató intuícióján múlik, hogy ágens ki tudja-e használni az RL-ben rejlő potenciált, vagy sem. A TSC-feladat megoldása kapcsán két fő megközelítést alkalmaznak az állapot-reprezentációhoz. Az első a nyers pixeladatok kinyerése a kereszteződésről készült pillanatképek formájában, amely viszonylag nagy számítási kapacitást igényel a CNN-ek használata miatt [45]. A második a feature-based értékvektorok használata, amelyek az egyes sávokról egyszerű detektorok által kinyert információkat tartalmaznak [46]. Ez az információ lehet például a járművek átlagos sebessége, az egyes sorhosszak vagy a várakozási idő. A valódi közlekedési hálózatokra való alkalmazhatóság lehetőségének megőrzése szempontjából szignifikáns kérdés az állapot-reprezentáció összetétele, hiszen az csak olyan jellemzőket tartalmazhat, amelyek könnyen elérhetők. Már napjainkban is viszonylag sok érzékelőt használnak a TSC-

problémával kapcsolatos mérőszámok megfigyelésére, kiértékelésére, például kamerákat és hurokdetektorokat, ebben a dolgozatban azonban az állapot-reprezentációk kapcsán csak az utóbbi eszköz által kinyerhető mérőszámokat használjuk, mivel azok már sok helyen ma is rendelkezésre állnak, viszonylag megfizethető és emellett megbízható eszközöknek számítanak. Az általunk bemutatott kutatásban a második megközelítést alkalmazzuk, és minden sávot egyetlen értékkel reprezentálunk, amely az adott 500 m hosszú útszakaszon álló járművek száma. Álló jármű alatt azt a járművet értjük, amelynek aktuális sebessége nem éri el a 0,1 m/s sebességet. Ezeket az értékeket $[0,1]$ intervallummá alakítjuk, hogy elkerüljük a visszaterjedési módszer numerikus problémáit.

2.2 Beavatkozás típusok

A TSC-problémára két féle beavatkozást különböztetünk meg. Az első esetben, ami ebben a dolgozatban is alkalmazásra kerül, az előző fázis lejárta után megváltoztatjuk a közlekedési lámpák fázisainak sorrendjét [47]. A második esetben folyamatosan lehet változtatni a fázisok időtartamát egy előre meghatározott intervallumban. Léteznek hibrid megközelítések is, amelyek a két korábbi módszert egyesítik [30]. A mi esetünkben két fő fázis van:

- Az első, az úgynevezett észak-déli zöld, amely zöld jelzést ad a függőleges sávpárok számára, és pirosat a vízszintes sávok számára.
- A második fázis, az úgynevezett nyugat-keleti zöld, amely lehetővé teszi a vízszintes sávok áthaladását, és piros jelzést ad a függőleges sávoknak.

E két fázis beállításával az ágens meghosszabbíthatja a sávok aktuális zöld idejét, aminek időtartama legalább 30 másodperc. Érdemes megemlíteni, hogy minden fázisváltást sárga, illetve sárga-zöld fázis választ el. Ez a megközelítés biztosítja, hogy az ágens megértse tettei következményeit, és lehetővé teszi, hogy a járművek alkalmazkodjanak a környezet új állapotához. Fontos megjegyezni azt is, hogy a szimulációkat véletlenszerűen generáljuk, meghatározott véletlenszerű járműsűrűséggel minden epizódban, ami lehetőséget ad az ágensnek arra, hogy különböző helyzetekkel találkozzon, így kialakítva az általánosító képességét. Mivel a generálás különböző forgalmi terheléseket eredményez az egyes útszakaszokon, az ágenseknek figyelnie kell a lámpák aktuális állapotára, és meg kell találniuk az optimális váltási pontokat, hogy megközelítsék a kereszteződés korlátait jelentő áteresztőképességet.

2.3 Rewarding koncepció

A rewarding – más néven jutalmazási – koncepció egyaránt az RL és az MCTS egyik legfontosabb része, mivel az ágensek egyedül ezeket a visszacsatolási (skalár) értékeket használják fel a folyamat belső dinamikájának megértéséhez, ami a tanulási folyamat magját képezi. Ennek köszönhetően a rewarding koncepció megfelelő megválasztása javíthatja a tanítás, illetve a tervezés sikerességét és ezáltal az ágens teljesítményét. A szakirodalom által leggyakrabban használt koncepció a sorok hosszát veszi alapul, de emellett sokan vannak, akik teljesen más metrikát vizsgálnak, mint például az átlagsebességet. Ezekhez a rewarding módszerekhez képest egy újszerű stratégia továbbfejlesztése kerül bemutatásra, majd összehasonlításra.

A szerzők [42] -ben a kereszteződésbe bejövő sávok kihasználtságát vették alapul, amelyeket, mint eloszlás értelmeztek. Az általuk bevezetett beavatkozásokkal (amik megegyeznek az általunk használtakkal) szabályozhatóvá vált a sávok kihasználtságából kialakított eloszlás átlaga és szórása oly módon, hogy több zöld időt biztosítanak egyik vagy másik bejövő sávpárnak. Ez után az egyes lépések reward-ját a kihasználtsági eloszlásból számítják ki, mint annak $[-1; 1]$ intervallumra normalizált szórása. Azért a szórás lett kiválasztva, mivel az a kereszteződésbe bejövő aszimmetrikus terhelés függvényében igyekszik kiegyenlíteni az egyes sávpároknak megadott zöldidőt.

Habár az általuk használt rewarding formulával is kimagasló eredmények születtek baseline kontrollerek (egyszerűbb vezérlési technikák) ellen, az elgondolásnak vannak teljesítmény-béli korlátai. Ez pedig a forgalom tehetetlenségével magyarázható: az egyszerű sávkihasználtság alkalmazása esetén nincsen információnk arról, hogy az egyes sávokban a járművek milyen kinematikai paraméterekkel jellemezhetők, így ez a módszer magában hordozza a forgalom tehetetlenségét is. Ha például két sáv kihasználtsága megegyezik, ugyanolyan mennyiségű reward-ot kapnak az adott időlépésben attól függetlenül, hogy az adott sávokban található járművek állnak, vagy éppen mozgást végeznek.

Ezen problémát kiküszöbölendő az általunk használt metrika nem a sávok kihasználtsága, hanem az egyes sávokban feltorlódott járművek száma (sorok hossza), amelyet szintén, mint eloszlás értelmezzünk. Tehát ebben az esetben az egyes lépések reward-ját a feltorlódott járművek eloszlásából számítjuk ki, mint annak $[-1; 1]$ intervallumra normalizált szórása, ami ezáltal már nem tartalmazza a forgalom tehetetlenségéből származó hibát. Az így kialakított reward-dal időlépésenként adunk visszajelzést az ágenseknek, amelyeknek célja így a kumulatív visszacsatolás legnagyobb növekedésének irányát megtalálni, és azt követni: így

frissíteni RL kapcsán a neurális hálózatot, illetve MCTS kapcsán a tervezési fát felépítve az optimális utat megtalálni. A három absztrakció között (állapot-reprezentáció, beavatkozások, rewarding) ebben az esetben nagyon szoros kapcsolat alakul ki, hiszen az állapot-reprezentáció a sávokban feltorlódott járművek eloszlása, a beavatkozásokon keresztül manipulálhatjuk ennek az eloszlásnak az átlagát és szórását, a rewarding eljárásunk lényege pedig a szórás minimalizálása. Ezért egy, az RL ágens számára könnyen értelmezhető optimalizációs feladatot mutatunk be neki, ami várhatóan gyors és stabil konvergenciát és széleskörű generalizálást eredményez.

RL esetében a sikeres rewarding koncepció kialakításakor az egyik legfontosabb feladat lehet a tanulható hibás viselkedésformák felismerése és kiküszöbölése, amik akár teljes mértékben leronthatják a betanított modellünk teljesítményét. Esetünkben előfordulhat olyan szituáció, hogy az ágens zöld időt biztosít a forgalomtól mentes iránynak, mivel az egyes sávokban feltorlódó járművek számának szórása relatíve kicsi. Emiatt az ágens viszonylag magas reward értékeket kapna a környezettől annak ellenére, hogy rossz sávoknak biztosít zöld időt. Természetesen ez lehetetlenné tenné az irányítási feladatunk megoldását, ezért a felmerülő problémát úgy hatástalanítjuk, hogy az ágenst -1 értékű skalár visszacsatolási értékkel büntetjük, amikor zöld időt biztosít a forgalom nélküli iránynak. Ettől függetlenül ebben az esetben az epizódot nem termináljuk, mivel ezáltal az ágens több tapasztalathoz tud hozzáférni, ezzel elősegítve a tanulást és általánosító képességének kialakulását.

2.4 Tanítás

Az RL modellünk sikerességét tekintve létfontosságú szerepet tölt be a tanítás folyamata. Ahhoz, hogy a robusztusság kellő szintjét, így a lehető legjobb teljesítményt és általánosító képességet érjük el, az ágensnek minél diverzebb tanító mintát kell biztosítanunk. A tanítási folyamatot a már bemutatott SUMO környezettel valósítjuk meg, amellyel lehetőségünk van a forgalom minden epizódra történő véletlenszerűsítésére. Ez azt jelenti, hogy a fent említett kiindulási pontokból különböző epizódokban különböző valószínűséggel indulnak járművek véletlenszerű sebességprofilokkal és útvonali célokkal, ezáltal minden epizódban előre meghatározott, mégis véletlen számú jármű lép be a szimulációba. Ennek a véletlenszerűsítésnek köszönhetően az RL ágensnek lehetősége van sokféle közlekedési szituáció megoldására.

Minden tanító epizód egy bemelegítési fázissal kezdődik, ahol a járművek elkezdenek belépni a hálózatba. Ez addig tart, amíg a négy sáv összesített kihasználtsága el nem éri a 10%-ot. A bemelegítés időtartama alatt minden közlekedési lámpa pirosra vált. Miután a

bemelegítési szakasz véget ért, az ágens belátása szerint, a fent bemutatott két féle beavatkozással vezérli a közlekedési lámpákat. Egy tanító epizód addig tart, amíg az összes jármű át nem halad a kereszteződésen. Ha az ágens nem képes az összes szimulációban résztvevő járművet átjuttatni a csomóponton, az epizód a SUMO szimulációs szoftver által mért 1800 lépés (másodperc) után leáll.

3 Algoritmusok

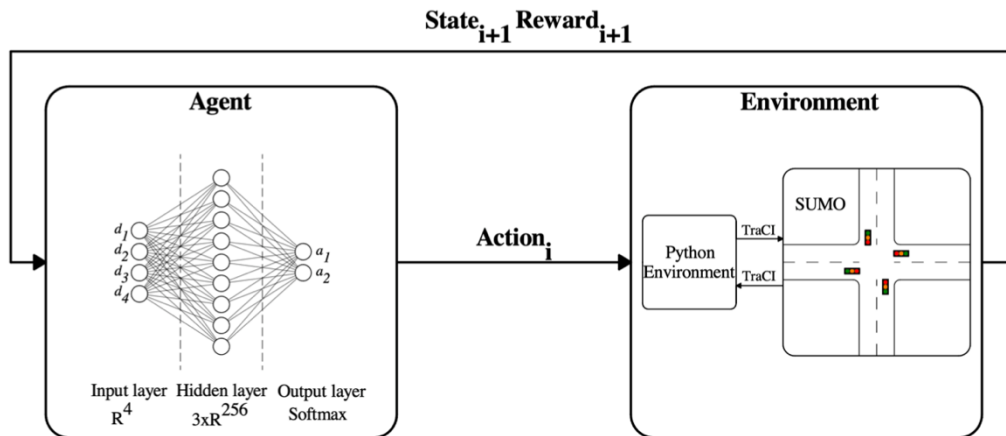
A probléma megoldásához felhasznált algoritmusokat alapvetően három különböző csoportba oszthatjuk. Az egyik a megerősítéses tanulás alapú, amely neurális hálózatot, mint függvény-approximátort használ fel a saját maga által generált tanító minták feldolgozásához, kiértékeléséhez, majd ennek eredményeként a hálózat finomhangolásához. A másik úgynevezett fa struktúrájú kereső algoritmus, amely gyakorlatilag az időtengelyen való előre tekintéssel választja ki az adott időpontban (jelenben) meghozandó optimális döntést. A harmadik csoport az egyszerű aktuált vezérlőt, illetve a fix ciklusidejű vezérlést tartalmazza, utóbbi a SUMO szimulációs szoftver beépített algoritmusa.

3.1 Megerősítéses tanulás

A számítástechnika és a hardver erőforrások egyre fokozódó fejlődése lehetővé tették a Mesterséges Intelligencia (Artificial Intelligence - AI) és a Gépi Tanulás robbanásszerű fejlődését. Ezen kívül a Mély Tanulás (Deep Learning - DL) lehetőséget adott az ML megújulására is: a legújabb eredmények azt mutatják, hogy a DL és RL kombinációja (DRL) képes az emberi teljesítőképességet is túlszárnyalni, akár bizonyos komplexebb feladatokban is [48]. Ezek a megdöbbentő eredmények számos kutatást ösztönöztek az alkalmazások széles spektrumában, mint például a robotvezérlésben, az autonóm járműirányításban vagy a társasjátékoknál.

A Megerősítés Tanulás értelmezhető egy olyan folyamatként, amely során a Mesterséges Neurális Hálózato(ka)t (Artificial Neural Network - ANN) hangoljuk. Ellentétben a Felügyelt Tanulással (Supervised Learning - SL) – amely szintén az ML egyik fajtája –, az RL-nek nincs szüksége nagy mennyiségű címkézett tanító adatra, amelyet csak nehezen, vagy egyáltalán nem lehet összegyűjteni. Az RL tanító mintáit az ágens és a környezet közötti interakciók révén állítjuk elő. Ezen interakciók közben az ágens egy reward-nak nevezett skalár visszacsatolást kap a környezettől, amelynek maximalizálásával próbálja elérni az optimális viselkedést. Ezek a reward értékek implicit módon jellemzik az adott irányítási feladat belső dinamikáját, hiszen számszerűsítik a környezeten végrehajtott beavatkozások hatásait.

A környezet szerepe tehát kettős a folyamat során: egyrészt elő kell állítania azokat az információkat, amelyekből az ágens képes lesz azok feldolgozása után döntést hozni, ezt nevezzük a környezet állapotának. Másrészt a környezet állapotának megváltozását kell jellemeznie valamilyen módon. Ilyenkor az ágens döntésének következménye, illetve magát a döntést minősítő jelző kerül kialakításra a környezet által egyetlen skalár formájában, ez a reward.



3. ábra: Az RL ágens és a környezet közötti interakció körfolyamata [42]

Az RL ágens és a környezet közötti kapcsolatot szemlélteti a 3. ábra. Egy interakció a következőképpen zajlik le: az ágens döntése meghatároz egy beavatkozást, amely állapotváltozást eredményez a környezetben. Ez után az ágens megkapja a környezet által visszacsatolt reward értéket (amely az állapotváltozásból származik). Ebben a konkrét esetben a környezet egy olyan vezérelhető szimulátor, ahol a kereszteződés általunk tervezett, lásd 2. fejezet.

A Megerősítéses Tanulás matematikai keretrendszerét a Markov Döntési Folyamatok (Markov Decision Process - MDP) biztosítják, amelyeket általánosságban a szekvenciális döntéshozatali problémákhoz használunk. Egy MDP esetén azt feltételezzük, hogy a vizsgált rendszer Markov tulajdonsággal rendelkezik. Ez azt jelenti, hogy egy véletlenszerű folyamat következő állapotának feltételes valószínűség-eloszlása egyedül az aktuális állapottól függ, feltéve, hogy az összes előző állapot ismert [49]. Ezért az állapotoknak tartalmazniuk kell minden olyan információt, amelyre szükségünk van ennek a valószínűségnek a megajzolásához, és így az aktuálisan megfigyelhető állapot alapján megalapozott döntéseket tudjunk hozni.

Egy MDP kapcsán 4 elem definiálható, ahol:

- S az állapottér és $s_i \in S$ az állapot az i -edik időlépésben,
- A a beavatkozási tér és $a_i \in A$ az i -edik időlépésben végrehajtott cselekvést jelenti,
- $P_{s,a}$ az $s \in S$ állapotból $a \in A$ beavatkozás általi állapotváltozás valószínűsége, amelyet általában a következőképpen jelölünk: $P(s'|s, a)$.
- R a reward függvényt jelöli, amelyben $r_i \in R$ jelöli azt az azonnali reward értéket, amely az a beavatkozás eredményeként az s állapot s' állapotba való változásához vezetett.

Az RL felfogható egy epizodikus folyamatként is, ahol az ágens minden diszkrét időlépésben, valamely végállapot eléréséig lép interakcióba a környezettel a fent bemutatott keretrendszer alapján. Megfigyeljük a környezet $s \in S$ állapotát minden $t \in \{1; 2; 3; 4; \dots\}$ időpontban, és kiválasztunk egy $a \in A$ beavatkozást, amely maximalizálja a várható $r \in R$ reward-ot. Minden lépésnél a környezet s állapotból s' állapotba kerül, ezért hosszú távon az ágens arra törekszik, hogy maximalizálja a kumulált reward-ot az adott reward függvény alapján. Ezáltal megfogalmazhatjuk az RL ágens célját: találjon egy $\pi: S \rightarrow A$ viselkedést, amely minden állapot-beavatkozás pár értékét maximalizálja. Ennek a célnak matematikai leírása az (1) egyenlet alapján fogalmazható meg:

$$Q^\pi(s, a) \equiv \mathbb{E} \left[\sum_{i=0}^{\infty} \gamma^i r(S_{t+i}; A_{t+i}; S_{t+i+1}) \mid S_t = s; A_t = a \right] \quad (1)$$

ahol $\mathbb{E}^\pi$ a π viselkedés általi elvárásunkat jelöli, r a reward függvény, amely az aktuális és soron következő állapottól, valamint az átmenetet okozó beavatkozástól függ. Továbbá, $\gamma \in [0; 1)$ a discount faktor, amellyel meghatározható, hogy a múlt beavatkozásai milyen mértékben befolyásolják a jövő állapotát.

3.1.1 Policy Gradient algoritmus bemutatása

A Policy Gradient algoritmus az RL-ben alkalmazott egyszerűbb, policy-alapú algoritmusok közé sorolandó, amellyel a közelmúltban több különböző területen érték el jelentős sikereket [50]. Továbbá a PG algoritmusnak van egy lényeges előnye az értékalapú módszerekkel szemben, nevezetesen, hogy konvergenciája garantált legalább egy lokális optimumhoz [51]. Általánosságban policy-alapú algoritmusok esetén a neurális hálózatok

súlyait úgy hangoljuk, hogy azok minden egyes állapothoz az állapotból végrehajtható beavatkozások közötti valószínűségi eloszlást approximálják. Emiatt tehát a hálózat által prediktált értékek egy dinamikus heurisztikaként jelennek meg, nem tükrözve az egyes választások hosszú távú következményeit. Ebben a dolgozatban az optimális felfedezés támogatása érdekében a valószínűségi kimeneti értékek alapján egy kategorikus függvényt hozunk létre, és mintavételezéssel határozzuk meg, hogy az ágens milyen végső beavatkozást hajtson végre.

PG esetén a súlyok frissítését a (2) egyenlettel végezzük el:

$$\theta \leftarrow \theta + \alpha \nabla \log \pi_{\theta}(s_t, a_t) \sum_{t=1}^T \gamma^t r_t \quad (2)$$

ahol θ jelöli a hálózat súlymátrixát, α a tanulási ráta, $\sum_{t=1}^T \gamma^t r_t$ a kumulált, discount faktoral súlyozott reward a T -edik időlépésig, míg $\pi_{\theta}(s_t, a_t)$ az a_t beavatkozás választási valószínűsége az s_t állapotból.

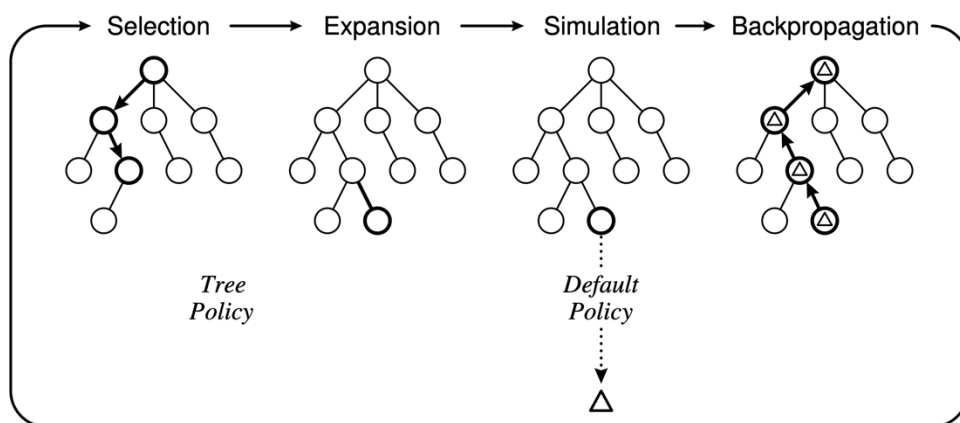
3.2 Fa típusú keresés alapú megoldások

A különböző keresés alapú, fa struktúrájú algoritmusok működésének alapja, hogy a mintavételezések során a mintákat leíró komplex állapotból, mint gyökérből (root) kiindulva, csúcsról csúcsra felépítenek egy fát a lehetséges döntésekből, mint elágazásokból. Így tehát egy elágazásból annyi felé ágazik a fa, amennyi a lehetséges beavatkozások száma. A fa terjedelme – ami az ágens előre tekintésének mértékével egyenértékű – az iterációk számától függ, tetszőlegesen megválasztható az elérhető számítási kapacitás, idő és memória függvényében. A fa így kialakult csúcsai (elágazások) lehetséges jövőbeli állapotokként értelmezhetők. Ezen csúcsok mindegyikéhez társítunk egy értéket, amely később részletezésre kerül. A csúcsok értékeit a fában felfelé (a gyökér felé) propagálva kiszámítható, hogy az aktuális állapot és előre tekintés mértékének függvényében melyik beavatkozás lehet az optimális. Ezen algoritmusok legfőbb problémája, hogy a tervezési folyamat során a fa mélységével exponenciálisan növekszik az abban megtalálható elágazások száma, ezáltal a számítási kapacitás-igény, tervezési idő és a szükséges memória nagysága egyaránt drasztikusan megnövekedhet.

3.2.1 Monte Carlo Tree Search algoritmus bemutatása

A Monte Carlo Tree Search egy olyan tervező-kereső algoritmus, amely egyesíti a fakesés pontosságát a véletlenszerű mintavételezésből származó általánosító képességgel. A kutatók nagy csoportjának érdeklődését akkor keltette fel, amikor a Go elnevezésű (számítástechnikailag) rendkívül komplex játékban kimagasló sikereket ért el, ezért gyakran az ehhez hasonló, táblajátékokban megjelenő tervezési problémákkal asszociálják. Ettől függetlenül az algoritmus alkalmazási köre ennél sokszínűbb, használták már például valós idejű vasúti átszervezési probléma, autonóm járműirányítás, illetve területi szkennelés problémájának megoldására is.

Egyik pozitív tulajdonsága, hogy az MCTS a fakeső algoritmusok fent ismertetett, általánosabb érvényű problémáját mérsékelni képes. Az algoritmus célja az aktuális állapotban meghozható optimális döntés meghozatala, amelyet véletlenszerű szimulációk és fakesés segítségével ér el. Az algoritmus az általános fa struktúrát követve futása során felépít egy keresési fát, amely közben elágazásról elágazásra véletlenszerű szimulációk segítségével jut.



4.ábra: Az MCTS algoritmus működését leíró lépések [52]

Az MCTS működésének folyamatát mutatja be a 4. ábra, amelynek lépései sorra a következők:

- **Megfelelő csúcs kiválasztása:** A környezet aktuális állapotából kiindulva rekurzív módon, az elágazások mentén lefelé haladva a fában, kiválasztjuk a legnagyobb értékű beavatkozáshoz tartozó ágat. Addig hajtjuk végre ezt a műveletet, ameddig el nem érünk egy olyan csúcsához, amiből már nincs elágazás. Ez két dolgot jelenthet. Vagy olyan állapotot értünk el, amelyet újonnan fedeztünk fel, és még nincsenek „ágai” vagy végállapotról beszélünk (esetünkben ez akkor valósul meg, ha a kereszteződés teljesen kiürült). Utóbbi esetben nem hajtjuk végre a következő lépést, és kiválasztjuk ezt a csúcsot.

- Expanzió: Abban az esetben, ha a csúcs nem végállapot, létrehozuk annak „ágait” a hozzá tartozó lehetséges beavatkozások számának megfelelően, és kiválasztjuk a kiindulási csúcsot.
- Random szimuláció: Az eredeti algoritmus szerint a választott csúcsból vagy véletlenszerű beavatkozások egymásutánja által, vagy azonnal elérünk egy végállapotig, és az így szerzett visszacsatolt skalár értéket (reward) a csúcs értékének választjuk. A TSC probléma kapcsán ezt a lépést azzal gyorsítjuk fel, hogy a random szimulációk helyett a kiválasztott csúcs aktuális állapotának megfelelő reward-ot választjuk a csúcs értékének. Így tervezési időt takaríthatunk meg, ezzel elősegítve a valós idejű futás lehetőségét.
- Backpropagation (visszaterjesztés): Az előző lépésben meghatározott értéket a kiválasztott csúcsból a fa gyökerébe vezető útvonalon végig vezetjük, és az összes így érintett csúcshoz hozzáadjuk annak értékét.

Az MCTS legfontosabb eleme azonban a csúcsok kiválasztásához használt úgynevezett Upper Confidence Bound for Trees (UCT) algoritmus, amelyet magyar kutatók fejlesztettek ki először (Kocsis Levente és Szepesvári Csaba [53]). Alapvetően ez az algoritmus határozza meg a fa terjedésének irányát, ezáltal igyekszik megoldani az úgynevezett exploration-exploitation dilemmát, amely a gépi tanulás egyik fontos problémaköre. Az ilyen módon kialakított fa szerkezete aszimmetrikus, ezáltal a már korábban említett számítási kapacitás, idő, és memóriaszükséglet nagyban lecsökken más keresési algoritmusokhoz képest. A formulát, amelyet a megfelelő csúcs kiválasztására, ezáltal a fa aszimmetrikus terjedésének elősegítésére használunk, a (3) egyenletben látható.

$$\bar{X}_i + 2 * C * \sqrt{\frac{2 * \ln N}{n_i}} \quad (3)$$

Ahol $\bar{X}_i$ a vizsgált csúcs átlagos reward-ja, n_i a csúcs vizitjeinek száma, N a vizsgált csúcs feletti csúcs vizitjeinek száma és C a felfedezés mértékét szabályozó konstans. Ez a kifejezés lényegében egy állandóval módosítható kompromisszum kialakítását teszi lehetővé az exploration és az exploitation között, aminek köszönhetően képes lehet az említett erőforrás szükségletek lecsökkentésére.

3.3 Baseline kontrollerek

A legtöbb kutató különböző RL-megoldásokat, aktuált vezérlőket és fix ciklusidejű irányítási módszereket használ fel kiindulási alapként (úgynevezett baseline-ként), hogy saját megoldásainak teljesítményét többféleképpen felmérhesse. Ebben a dolgozatban összehasonlításra kerül két, az új rewarding koncepciót (lásd 2.3) alkalmazó ágens (egy Policy Gradient és egy MCTS), illetve mellettük baseline kontrollerként időveszteség-alapú aktuált- és fix ciklusidejű vezérlést használunk. Az alkalmazott aktuált vezérlés a SUMO szimulációs szoftver beépített algoritmusa, amely a sávban lévő járművek időveszteségét tekinti viselkedése alapjának.

4 Eredmények

Napjainkban bolygónk egészségének megőrzése egy rendkívül fontos téma, ezáltal egy kiemelt célt tűz ki a kutatók szeme elé is, akik következőképpen nagy figyelmet fordítanak a közlekedés környezeti lábnyomának minimalizálására. Az új forgalomirányító algoritmusok kiértékelésénél magától értetődőnek tűnik, hogy azok teljesítményét nem csak a hagyományos mérőszámokkal jellemezzük, mint például az utazási idő, a várakozási idő és a sorhossz, hanem azokat kiegészítve, a fenntarthatósági metrikákkal együtt jelenítsük meg.

Az összehasonlíthatóság érdekében az összes megoldást 1000 epizódon értékeljük, ahol a bejövő sávok forgalmát véletlenszerűen generáljuk. Mivel véletlenszerűség alatt itt pszeudorandom számokat értünk, viszonylag egyszerű az összes ágens esetében ugyanazokat a beállítási értékeket alkalmazni. Az RL esetében – ellentétben a tanító epizódokkal – a tesztepizódok addig tartanak, amíg az összes jármű el nem hagyja a teljes hálózatot. Ez utóbbi módszer vonatkozik természetesen az MCTS ágensre, a fix ciklusidejű vezérlésre, illetve a SUMO beépített algoritmusára is, ezáltal a kiértékelés fázisa azonos feltételek mellett történik.

4.1 Fenntarthatósági paraméterek összehasonlítása

1. táblázat: Fenntarthatósági mérőszámok statisztikai összehasonlítása 1000 tesztepizód alatt

Ágens	CO ₂ emisszió [kg]	CO emisszió [kg]	NO _x emisszió [g]
MCTS – sorhossz	109,9	3,4	47,9
MCTS –sorhossz szórása	108,7	3,4	47,3
MCTS - átlagsebesség	131,1	4,9	57,7
MCTS – kihasználtság szórása	113,2	3,7	49,4
PG – sorhossz	110,1	3,5	48,0
PG – sorhossz szórása	108,5	3,3	47,2
PG – átlagsebesség	118,7	4,1	52,0
PG – kihasználtság szórása	110,4	3,5	48,1
SUMO – aktuált	123,3	4,4	54,1
Fix ciklusidejű	229,5	10,8	102,5

2. táblázat: Fenntarthatósági mérőszámok statisztikai összehasonlítása 1000 tesztepizód alatt

Ágens	PM_x emisszió [g]	HC emisszió [g]	Üzemanyag fogyasztás [l]
MCTS – sorhossz	2,3	18,0	47,3
MCTS –sorhossz szórása	2,3	17,9	46,7
MCTS - átlagsebesség	2,9	25,2	56,4
MCTS – kihasználtság szórása	2,4	19,4	48,7
PG – sorhossz	2,4	18,7	47,3
PG – sorhossz szórása	2,3	17,7	46,6
PG – átlagsebesség	2,6	21,3	51,0
PG – kihasználtság szórása	2,4	18,7	47,4
SUMO – aktuált	2,7	23,0	53,0
Fix ciklusidejű	5,2	54,6	98,7

Az 1. és 2. táblázat a TSC-probléma különböző megoldásainak teljesítményét mutatja be sok különböző kibocsátási paraméter összehasonlítása által, ilyenek például a szén-dioxid, szén-monoxid, nitrogén-oxidok és az üzemanyag-fogyasztás. A különböző értékeket a SUMO szimulációs szoftver kalkulálja ki időlépésenként, mi pedig az 1000 epizód alatt kinyert adathalmazt feldolgozva az értékek átlagát jelenítettük meg az 1. és 2. táblázatban. A SUMO által használt matematikai modell a kibocsátások pontos számításához a The Handbook Emission Factors for Road Transport, amely a járművek kibocsátását a járművek sebességéből, és gyorsulásából származó polinomok alapján számítja ki. Ez a megközelítés különbséget tesz a járműtípusok és a kibocsátási kategóriák között is, további információkért lásd: [54].

Minél alacsonyabb az adott érték, annál jobban teljesít a módszer. Tisztán látható, hogy az általunk bemutatott rewarding-ot alkalmazó ágensek minden kibocsátási paraméterben felülmúlják nem csak a fix ciklusidejű vezérlést és a SUMO beépített időveszteségen alapuló algoritmusát, hanem a szakirodalom által leggyakrabban használt, legnépszerűbb rewarding technikát, amely a sorok hosszát veszi az egyes reward értékek alapjául. Mint ahogy azt korábban már említettük, a globális felmelegedés szempontjából az egyik leglényeges kibocsátási metrika a szén-dioxid. A továbbfejlesztett rewarding koncepciónknak köszönhetően az MCTS esetében 5% CO₂-kibocsátást takaríthatunk meg az eredeti kihasználtság-szórás alapú módszerhez képest, 12%-ot a SUMO algoritmusához képest, és majdnem 1,2% -ot a legelterjedtebb módszerhez képest, ami azt bizonyítja, hogy az általunk bemutatott rewarding koncepció hatékonyan vezérli a kereszteződés jelzőlámpáit,

fenntarthatósági szempontból. Emellett drasztikus, majdnem 53%-kal ér el alacsonyabb kibocsátást a fix ciklusidejű vezérléshez képest, amely a mai napig a legszélesebb körben alkalmazott irányítási technika a világon.

Egy másik kritikus mérőszám az üzemanyag-fogyasztás alakulása a különböző kiértékelési epizódok közben. Megoldásunkkal az üzemanyag felhasználást is sikerült tovább mérsékelni: a PG ágens ezzel a módszerrel majdnem két további százalékot javított az eredeti szórás alapú rewarding-gal elért eredményhez képest, és megközelítőleg 1,5%-kal jobb eredményt ért el, mint a legnépszerűbb rewardingot alkalmazva. Habár ez a mennyiség nem feltétlenül tűnik soknak, az Amerikai Energetikai Ügynökség becslései alapján [55] 2020-ban globálisan 14,5 milliárd liter üzemanyagot használt fel az emberiség naponta. Ebből a számadatból már érzékelhető, hogy ez a másfél százalék akár több millió liter megtakarítást is jelenthet 24 óra alatt, amely hosszú távon jelentős környezeti hatással járhat.

Érdemes továbbá megfigyelni, hogy fenntarthatósági paraméterek esetében az MCTS által alkalmazott rewarding módszereket összehasonlítva és azokat kiértékelve, a környezetszennyezés szempontjából teljesítményük alapján sorrendbe téve ugyanazt a sorrendet kapjuk, mint ha ezt a műveletet megismétljük a PG ágens esetében. Mindkettő ágensnél azt tapasztaljuk, hogy az általunk bemutatott rewarding eljárás után második helyen a sorhossz alapú megoldás szerepel, harmadikként végez az eredeti, sávok kihasználtságának szórását figyelembe vevő módszer, míg mindkét esetben utolsó, ezáltal a legrosszabb eredményt éri el fenntarthatóság szempontjából az átlagsebesség alapú stratégia. Ez a kijelentés tehát alapvetően azt jelenti, hogy az MCTS tervező-kereső algoritmus alkalmazható különböző reward stratégiák összehasonlítására és keresésére például RL ágensek számára. Ez a gyakorlatban nagy előnyt jelenthet, hiszen egyes esetekben a tanítási folyamat óriási pénzbefektetéssel és időfelhasználással jár, amelyeket ezzel a módszerrel mérsékelni lehet.

Tekintettel arra, hogy az általunk bemutatott rewarding koncepció semmilyen fenntarthatósággal kapcsolatos mérőszámot nem használ fel, az ágens teljesítménye valószínűsíthetően a kereszteződésben kialakuló kisebb torlódások és az optimális forgalom-áramlás következménye lehet. Hipotézisünk alátámasztására a következőkben bemutatásra kerül, hogyan alakult az átlagos utazási idő, az átlagos várakozási idő és a sorok hossza a vizsgált 1000 epizód alatt.

4.2 Klasszikus mérőszámok összehasonlítása

Utazási idő alatt azt az időmennyiséget értjük, amelyet a jármű az elindulásának pillanatától a célállomására való megérkezéséig felhasznál. Az átlagos utazási idő pedig nem

más, mint a szimulációba belépett járművek teljes utazási idejének összege elosztva a járművek számával. Ezt formálisan a (4) egyenlet írja le,

$$\bar{t}_t = \frac{1}{N} \sum_{i=0}^N (t_{i,vég} - t_{i,start}) \quad (4)$$

ahol $\bar{t}_t$ az átlagos utazási idő, és N a járművek száma. A $t_{i,vég}$ az i -edik jármű érkezési idejét jelöli, $t_{i,start}$ pedig az i -edik jármű indulási idejét.

A várakozási idő definíció szerint nem más, mint az összesített állással eltöltött idő az indulástól a kereszteződésen való áthaladásig. Egy jármű akkor tekinthető állónak, ha sebessége kisebb, mint 0,1 [m/s]. Ezt az értéket a szoftver minden, a hálózatba belépő járműre kiszámítja. Az átlagos várakozási időt az összes jármű összesített várakozási idejének összegével számítjuk ki, amelyet elosztunk a járművek számával. Ennek matematikai jelentését lásd (5) egyenlet,

$$\bar{t}_w = \frac{1}{N} \sum_{i=0}^N t_{i,w} \quad (5)$$

ahol $\bar{t}_w$ az átlagos várakozási idő, $t_{i,w}$ pedig az i -edik jármű kumulált várakozási ideje a kereszteződés elhagyásának pillanatában.

A sorhossz alatt tulajdonképpen az adott sávban várakozó járművek számát kell érteni. Az egyszerűség kedvéért az összes sáv sorhosszát egy értékbe vonjuk össze, amelyet aztán elosztunk a sávok számával. Ez egyetlen skalár metrikát eredményez, amely a feltorlódás mértékét reprezentálja a hálózatban, kiszámítása a (6) egyenlet alapján történik:

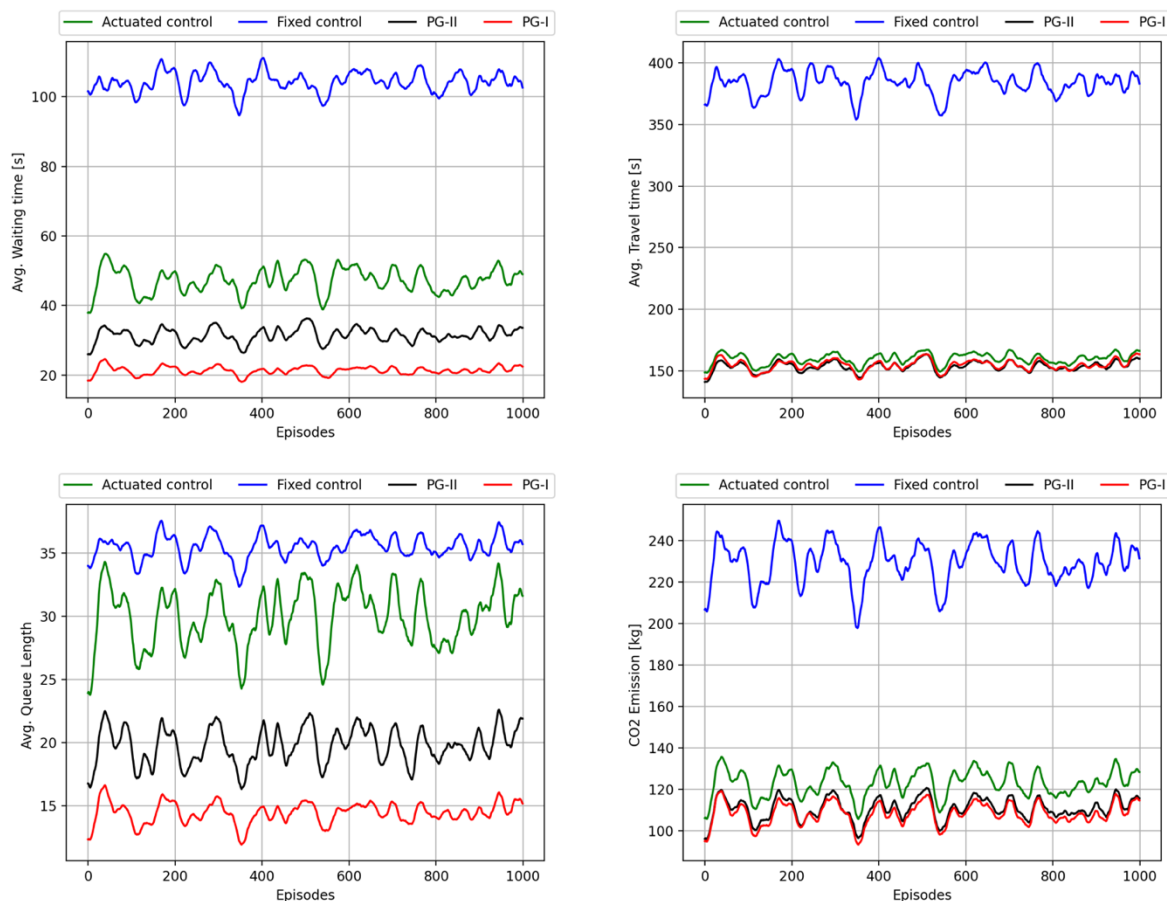
$$q_l = \frac{\sum_{l=0}^L q_{l,i}}{L} \quad (6)$$

ahol q_l a sávok átlagos sorhossza (tehát az egyes sávokban feltorlódott járművek számának átlaga), $q_{l,i}$ pedig az i -edik sávban lévő sorhossz.

3. táblázat: Klasszikus teljesítménymutatók összehasonlítása 1000 tesztpizód alatt

Ágens	Utazási idő [s]	Várakozási idő [s]	Sorhossz [darab jármű]
MCTS – sorhossz	151,3 ± 26	21,3 ± 6	3,53 ± 4
MCTS – sorhossz szórása	153,7 ± 25	23,3 ± 7	3,77 ± 5
MCTS - átlagsebesség	170,1 ± 26	53,5 ± 24	6,37 ± 13
MCTS – kihasználtság szórása	158,4 ± 24	34,6 ± 12	5,17 ± 7
PG – sorhossz	153,6 ± 23	31,5 ± 12	4,95 ± 9
PG – sorhossz szórása	154,7 ± 23	21,3 ± 7	3,61 ± 5
PG – átlagsebesség	161,7 ± 36	38,4 ± 23	5,62 ± 9
PG – kihasználtság szórása	155,1 ± 23	27,8 ± 9	4,34 ± 6
SUMO – aktuált	160,0 ± 25	47,3 ± 21	7,51 ± 14
Fix ciklusidejű	384,4 ± 69	104,3 ± 22	8,88 ± 7

A 3. táblázatban látható eredmények azt mutatják, hogy az általunk bemutatott eljárással a PG ágens teljesítménye abszolút felülmúlja várakozási idő, utazási idő és sorhossz tekintetében a SUMO beépített algoritmusát, illetve az összes többi általunk tesztelt rewarding koncepciót (kivével ez alól az utazási idő értéke a sorhossz alapú rewarding-nál, amely elenyésző, 0,8%-kal jobb eredményt ért el). Ha figyelembe vesszük az értékek szórását is, az látszik, hogy szintén az ágens teljesítménye abban az esetben a legstabilabb, amikor a sávok sorhosszainak szórását választjuk a rewarding függvény alapjául. Az MCTS esetében ugyan megváltozik a sorrend – a sorok hossza alapján működő rewarding teljesít jobban –, de emellett közel azonos teljesítményt nyújt a továbbfejlesztett módszerünk. A SUMO beépített algoritmusát, illetve a fix ciklusidejű vezérlést minden esetben képes felülmúlni az új, szórás alapú rewarding, és ez a különbség az aktuált vezérlés és az MCTS, PG algoritmusok között határozott, ha figyelembe vesszük még az eredmények szórását is. Azok alapján kijelenthető, hogy az MCTS és PG ágensek jóval kiegyensúlyozottabban végzik a kereszteződés forgalmának irányítást abban az esetben, amikor rewarding eljárásként a sorhossz szórását választjuk. A különböző megoldások működésének nyomon követése során azt figyeltük meg, hogy az algoritmusok közötti fő különbség az, hogy az idővesztés-alapú vezérlés hosszabb zöld fázisokat használ, amelyek meghatározzák a várakozási idő és a sorhossz, valamint az utazási idő szórásának eltérését.



5. ábra: Különböző megoldások teljesítményei 1000 tesztepizód alatt

Az 5. ábra néhány kulcsfontosságú mérőszám értékeinek alakulását mutatja (átlagos várakozási idő, átlagos utazási idő, sorok hossza, és kumulált CO₂-kibocsátás) az 1000 epizódos kiértékelési folyamat során. Piros színnel van jelölve az a PG ágens, amely az általunk továbbfejlesztett rewarding stratégiát alkalmazza, feketével azt a PG ágenst jelöltük, amely a – szakirodalomban leggyakrabban használt – sorhosszak alapján határozza meg a reward értékeit. Zöld színnel jelöltük a SUMO szimulációs szoftver beépített algoritmusát, illetve kékkel a legegyszerűbb, fix ciklusidejű vezérlést, amely gyakorlatban a legtöbb helyen alkalmazva van. Látható, hogy míg utazási időben a szórás alapú megoldás néhol kis mértékben elmarad a legelterjedtebb módszertől, mind a három másik bemutatott paraméter már nagyobb differenciát mutat a mi rewarding eljárásunk javára.

Ezek az adatok alátámasztják korábbi megállapításainkat a különböző megoldások viselkedéséről és teljesítményéről. Megjegyezzük, hogy a bemutatott RL és MCTS megoldások szinte minden epizódban felülmúlják a SUMO beépített megoldását, kivétel ez alól az átlagsebesség alapú reward függvény alkalmazása esetén az utazási idő, illetve a várakozási idő értékeket.

A fent ismertetett eredmények azt mutatják, hogy az új, sorok hosszának szórásán alapuló, rewarding koncepcióval tanított ágensek jobb teljesítményt értek el mind a fenntarthatósági metrikák, mind pedig a klasszikus minőségjelzők tekintetében. Ez arra utal,

hogy egy érdekes betekintést találtunk a TSC-probléma belső dinamikájába, amely kimagasló teljesítményt eredményez. Emellett a kiértékelés során egy hasznos potenciális tulajdonságát neveztük meg a Monte Carlo Tree Search elnevezésű fakeső algoritmusnak, nevezetesen, hogy az akár a különböző rewarding technikák összehasonlítására és keresésére is alkalmas lehet, ami további kutatásokra ad motivációt.

5 Konklúzió

Ebben a dolgozatban egy új, továbbfejlesztett rewarding koncepció került bemutatásra az egyszerű egykereszteződéses közlekedési jelzőlámpák irányításának problémájára, miszerint az ágensek célja, hogy megtalálják azt a fázissorend kombinációt, amely fenntarthatósági szempontból az optimális szabályozást eredményezi, és egyúttal csökkenti az egyes járművek várakozási idejét és utazási idejét, valamint a kereszteződésben lévő sorok hosszát.

A probléma megoldására egy PG ágens lett betanítva négy különböző rewarding koncepció alkalmazásával, illetve ugyanaz a négy reward függvény lett alkalmazva egy másik, MCTS ágens által a bemutatott irányítási feladat megoldására. Ezek a módszerek a következő metrikák alapján határozzák meg a reward értékeket: sorok hossza, sorhosszak szórása, átlagsebesség és sávkihasználtság szórása. Mindkét ágens teljesítményét az átlagos utazási idő, az átlagos várakozási idő és a sorok hosszára vonatkozó klasszikus teljesítménymérők alapján jellemezzük. Ezen túlmenően az ágenseket fenntarthatósági paraméterek alapján is értékeljük, mint például a CO₂-kibocsátás, az üzemanyag-fogyasztás, és az NO_x-kibocsátás.

Az eredmények azt mutatták, hogy a továbbfejlesztett rewarding koncepcióval mindkét ágens minden fenntarthatósági paraméterben felülmúlta a SUMO beépített, idővesztés-alapú aktuált vezérlőjét, az eredeti eljárást-, illetve a szakirodalom által leggyakrabban alkalmazott rewarding formulát alkalmazó ágenseket.

Megközelítésünk fő előnye, hogy a kereszteződésbe bejövő sávok forgalmának egyensúlyát tulajdonságai miatt igyekszik fenntartani, a forgalom tehetetlenségének mellőzése mellett. A statisztikai adatokon túlmenően a közlekedésben résztvevők körében is nagyobb elégedettséget érhetünk így el, mivel várakozási idő tekintetében is előrehaladást tudunk bemutatni. Egy másik fontos szempont, hogy ez a megközelítés képes azon kereszteződéseket is kezelni, amelyekbe beérkező sávok prioritásai nem egyeznek meg. Ez a differenciálhatóság könnyen megoldható az egyes sávok konstansokkal való skálázásával, aminek eredményeként megoldásunknak számos előnye lehet a valós környezetekben történő alkalmazásokban.

A közlekedési jelzőlámpákat vezérlő fejlett algoritmusok által ténylegesen csökkenthető a városi területek kibocsátása, és mérsékelhető a forgalmi dugókban eltöltött idő mennyisége is. Emellett az ágensek általunk bemutatott állapot-reprezentációjához szükséges infrastruktúra (hurokdetektorok) a legtöbb nagyvárosban már napjainkban is megtalálhatók, de azok könnyen telepíthetősége, olcsósága és megbízhatósága hozzásegít a megoldás széleskörű alkalmazhatóságához.

A fejlesztett szórás-alapú rewarding koncepcióban rejlő további potenciál és a TSC-problémára vonatkozó kutatásunkból nyert eredmények alapján indokoltnak tűnik a kutatás folytatása, erre több lehetőség is rendelkezésre áll. Először is, a bemutatott TSC vezérlési probléma komplexitása kiterjeszthető lehet az egymással összefüggő több kereszteződést magába foglaló közlekedési hálózattal. Másodszor, az általunk bemutatott rewarding koncepció úgynevezett Multi-Ágens Megerősítéses Tanulásban (Multi-Agent Reinforcement Learning - MARL) történő implementálására is sor kerülhet. Az MARL keretrendszer olyan alapvető jellemzőkkel rendelkezik, mint az ágensek közötti kommunikáció, sőt, tárgyalásról is beszélhetünk ennek kapcsán, amelyek kétségtelenül segítik a komplex, valós problémák megoldását.

Táblázatjegyzék

1. táblázat: Fenntarthatósági mérőszámok statisztikai összehasonlítása 1000 tesztepizód alatt	24
2. táblázat: Fenntarthatósági mérőszámok statisztikai összehasonlítása 1000 tesztepizód alatt	25
3. táblázat: Klasszikus teljesítménymutatók összehasonlítása 1000 tesztepizód alatt.....	28

Ábrajegyzék

1. ábra: Az üvegház hatású gázok eloszlása [1].....	3
2. ábra: A SUMO kereszteződés felépítése [42].....	11
3. ábra: Az RL ágens és a környezet közötti interakció körfolyamata [42].....	18
4. ábra: Az MCTS algoritmus működését leíró lépések [52].....	21
5. ábra: Különböző megoldások teljesítményei 1000 tesztépizód alatt	29

Rövidítések

AI	-	Mesterséges Intelligencia
ANN	-	Mesterséges Neurális Hálózat
CNN	-	Konvolúciós Neurális Hálózatok
DL	-	Mély Tanulás
DP	-	Dinamikus Programozás
DRL	-	Mély Megerősítéses Tanulás
GA	-	Genetikus Algoritmus
ITS	-	Intelligens Közlekedési Rendszerek
MARL	-	Multi-Ágens Megerősítéses Tanulás
MAS	-	Multi-Ágens Rendszerek
MCTS	-	Monte Carlo Tree Search
MDP	-	Markov Döntési Folyamat
ML	-	Gépi Tanulás
PG	-	Policy Gradient
RL	-	Megerősítéses Tanulás
SL	-	Felügyelt Tanulás
SUMO	-	Simulation of Urban MObility
TSC	-	Jelzőlámpa-Irányítás
UCT	-	Upper Confidence Bound for Trees

Hivatkozások

- [1] <https://www.eia.gov/energyexplained/energy-and-the-environment/where-greenhouse-gases-come-from.php> (utolsó letöltés: 2021.10.26.)
- [2] Al-Ghussain, L. (2019). Global warming: review on driving forces and mitigation. *Environmental Progress & Sustainable Energy*, 38(1), 13-21.
- [3] <https://www.iea.org/reports/tracking-transport-2020> (utolsó letöltés: 2021.10.26.)
- [4] <https://www.eea.europa.eu/data-and-maps/indicators/transport-emissions-of-greenhouse-gases-7/assessment> (utolsó letöltés: 2021.10.26.)
- [5] Campbell-Lendrum, D., & Prüss-Ustün, A. (2019). Climate change, air pollution and noncommunicable diseases. *Bulletin of the World Health Organization*, 97(2), 160.
- [6] Leung, D. Y. (2015). Outdoor-indoor air pollution in urban environment: challenges and opportunity. *Frontiers in Environmental Science*, 2, 69.
- [7] Hannappel, R. (2017, August). The impact of global warming on the automotive industry. In *AIP Conference Proceedings* (Vol. 1871, No. 1, p. 060001). AIP Publishing LLC.
- [8] Acar, C., & Dincer, I. (2020). The potential role of hydrogen as a sustainable transportation fuel to combat global warming. *International Journal of Hydrogen Energy*, 45(5), 3396-3406.
- [9] Abdullah, N. N., & Anwar, G. (2021). An Empirical Analysis of Natural Gas as an Alternative Fuel for Internal Transportation. *International Journal of English Literature and Social Sciences*, 6(1).
- [10] Johnson, T., & Joshi, A. (2018). Review of vehicle engine efficiency and emissions. *SAE International Journal of Engines*, 11(6), 1307-1330.
- [11] Jung, J., & Koo, Y. (2018). Analyzing the effects of car sharing services on the reduction of greenhouse gas (GHG) emissions. *Sustainability*, 10(2), 539.
- [12] Zhang, Y., & Mi, Z. (2018). Environmental benefits of bike sharing: A big data-based analysis. *Applied Energy*, 220, 296-301.
- [13] Pandian, S., Gokhale, S., & Ghoshal, A. K. (2009). Evaluating effects of traffic and vehicle characteristics on vehicular emissions near traffic intersections. *Transportation Research Part D: Transport and Environment*, 14(3), 180-196.
- [14] Samaras, C., Tsokolis, D., Toffolo, S., Magra, G., Ntziachristos, L., & Samaras, Z. (2018). Improving fuel consumption and CO2 emissions calculations in urban areas by coupling a dynamic micro traffic model with an instantaneous emissions model. *Transportation Research Part D: Transport and Environment*, 65, 772-783.
- [15] Panis, L. I., Broekx, S., & Liu, R. (2006). Modelling instantaneous traffic emission and the influence of traffic speed limits. *Science of the total environment*, 371(1-3), 270-285.
- [16] DELL, P., & MIRCHANDANI, B. (1995). REALBAND: An approach for real-time coordination of traffic flows on networks.
- [17] Ekeila, W., Sayed, T., & Esawey, M. E. (2009). Development of dynamic transit signal priority strategy. *Transportation research record*, 2111(1), 1-9.
- [18] Mirchandani, P., & Head, L. (2001). A real-time traffic signal control system: architecture, algorithms, and analysis. *Transportation Research Part C: Emerging Technologies*, 9(6), 415-432.
- [19] Cai, C., Wong, C. K., & Heydecker, B. G. (2009). Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C: Emerging Technologies*, 17(5), 456-474.
- [20] Boillot, F., Midenet, S., & Pierrelee, J. C. (2006). The real-time urban traffic control system CRONOS: Algorithm and experiments. *Transportation Research Part C: Emerging Technologies*, 14(1), 18-38.

- [21]McKenney, D., & White, T. (2013). Distributed and adaptive traffic signal control within a realistic traffic simulation. *Engineering Applications of Artificial Intelligence*, 26(1), 574-583.
- [22]Wang, F., Tang, K., Li, K., Liu, Z., & Zhu, L. (2019). A group-based signal timing optimization model considering safety for signalized intersections with mixed traffic flows. *Journal of advanced transportation*, 2019.
- [23]Zhang, L., Yin, Y., & Chen, S. (2013). Robust signal timing optimization with environmental concerns. *Transportation Research Part C: Emerging Technologies*, 29, 55-71.
- [24]Jin, J., & Ma, X. (2017). A group-based traffic signal control with adaptive learning ability. *Engineering applications of artificial intelligence*, 65, 282-293.
- [25]Villalobos, I. A., Poznyak, A. S., & Tamayo, A. M. (2008). Urban traffic control problem: a game theory approach. *IFAC Proceedings Volumes*, 41(2), 7154-7159.
- [26]Wang, S., Xie, X., Huang, K., Zeng, J., & Cai, Z. (2019). Deep reinforcement learning-based traffic signal control using high-resolution event-based data. *Entropy*, 21(8), 744.
- [27]Wan, C. H., & Hwang, M. C. (2018). Value-based deep reinforcement learning for adaptive isolated intersection signal control. *IET Intelligent Transport Systems*, 12(9), 1005-1010.
- [28]Genders, W., & Razavi, S. (2016). Using a deep reinforcement learning agent for traffic signal control. *arXiv preprint arXiv:1611.01142*.
- [29]Guo, J., & Harmati, I. (2020). Comparison of Game Theoretical Strategy and Reinforcement Learning in Traffic Light Control. *Periodica Polytechnica Transportation Engineering*, 48(4), 313-319.
- [30]Bouktif, S., Cheniki, A., & Ouni, A. (2021). Traffic signal control using hybrid action space deep reinforcement learning. *Sensors*, 21(7), 2302.
- [31]Li, L., Lv, Y., & Wang, F. Y. (2016). Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3), 247-254.
- [32]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 (at NIPS 2016)*.
- [33]Chu, T., Wang, J., Codecà, L., & Li, Z. (2019). Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3), 1086-1095.
- [34]Tan, T., Bao, F., Deng, Y., Jin, A., Dai, Q., & Wang, J. (2019). Cooperative deep reinforcement learning for large-scale traffic grid signal control. *IEEE transactions on cybernetics*, 50(6), 2687-2700.
- [35]Wei, H., Zheng, G., Gayah, V., & Li, Z. (2021). Recent advances in reinforcement learning for traffic signal control: A survey of models and evaluation. *ACM SIGKDD Explorations Newsletter*, 22(2), 12-18.
- [36]Cheng, S. F., Epelman, M. A., & Smith, R. L. (2006). CoSIGN: A parallel algorithm for coordinated traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 7(4), 551-564.
- [37]Renfrew, D., & Yu, X. H. (2009, August). Traffic signal control with swarm intelligence. In *2009 Fifth International Conference on Natural Computation* (Vol. 3, pp. 79-83). IEEE.
- [38]He, Q., Head, K. L., & Ding, J. (2011). Heuristic algorithm for priority traffic signal control. *Transportation Research Record*, 2259(1), 1-7.
- [39]Yin, B., Dridi, M., & El Moudni, A. (2015). Forward search algorithm based on dynamic programming for real-time adaptive traffic signal control. *IET intelligent transport systems*, 9(7), 754-764.
- [40]Qi, H., & Hu, X. (2019). Monte Carlo Tree Search-based intersection signal optimization model with channelized section spillover. *Transportation Research Part C: Emerging Technologies*, 106, 281-302.

- [41] Cheng, Y., Hu, X., Tang, Q., Qi, H., & Yang, H. (2020). Monte Carlo tree search-based mixed traffic flow control algorithm for arterial intersections. *Transportation research record*, 2674(8), 167-178.
- [42] Kővári, B., Szőke, L., Bécsi, T., Aradi, S., & Gáspár, P. (2021). Traffic Signal Control via Reinforcement Learning for Reducing Global Vehicle Emission. *Sustainability*, 13(20), 11254.
- [43] Zheng, G., Zang, X., Xu, N., Wei, H., Yu, Z., Gayah, V., ... & Li, Z. (2019). Diagnosing reinforcement learning for traffic signal control. *arXiv preprint arXiv:1905.04716*.
- [44] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y. P., Hilbrich, R., ... & Wießner, E. (2018, November). Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2575-2582). IEEE.
- [45] Mousavi, S. S., Schukat, M., & Howley, E. (2017). Traffic light control using deep policy-gradient and value-function-based reinforcement learning. *IET Intelligent Transport Systems*, 11(7), 417-423.
- [46] Liang, X., Du, X., Wang, G., & Han, Z. (2019). A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology*, 68(2), 1243-1253.
- [47] Guo, M., Wang, P., Chan, C. Y., & Askary, S. (2019, October). A reinforcement learning approach for intelligent traffic signal control at urban intersections. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 4242-4247). IEEE.
- [48] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540), 529-533.
- [49] Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 135). Cambridge: MIT press.
- [50] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., ... & Kavukcuoglu, K. (2016, June). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning* (pp. 1928-1937). PMLR.
- [51] 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. 1057-1063).
- [52] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., ... & Colton, S. (2012). A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1), 1-43.
- [53] Kocsis, L., & Szepesvári, C. (2006, September). Bandit based monte-carlo planning. In *European conference on machine learning* (pp. 282-293). Springer, Berlin, Heidelberg.
- [54] Notter, B., Keller, M., Althaus, H. J., Cox, B., Knörr, W., Heidt, C., ... & Jamet, M. (2019). HBEFA 4.1. *INFRAS*.
- [55] <https://www.eia.gov/todayinenergy/detail.php?id=46596> (utolsó letöltés 2021. 10. 26.)