Mély megerősítéses tanulás alkalmazása vasúti forgalomirányításban

Szerző:
Balogh Csanád Levente

BME

Közlekedésmérnöki és Járműmérnöki Kar

Konzulens:
Kővári Bálint

TDK dolgozat

Contents

Chapter 1

Bevezető

1.1 Motiváció

A vasúti közlekedés hosszú ideje minden ország infrastruktúrájának fontos és elengedhetetlen része. Különösen igaz ez az utóbbi években, a környezettudatos és fenntarthatóságra törekvő szemléletmód egyre szélesebb körben való elterjedésével. A klímavédelem érdekében kitűzött célok eléréséhez hozzájárul a vasúti forgalom és szállítmányozás mértékének és sebességének növelése. Az Európai Unió főbb célkitűzései közt szerepel a szállítmányozás és közlekedés "zöldebbé" tétele a vasúti forgalom 2050-re való megduplázása által, illetve új innovatív adat és mesterséges intelligencia alapú megoldások felhasználásával a vasúti szektorban [4]. Ilyen jellegű megoldásokról a [11] részletesebben ír. Az egyik fontos mesterséges intelligencia alapú módszer a megerősítéses tanulás, mely megoldást nyújthat számos kihívásra a szektorban, valamint egy hasznos eszköz a folyamatok optimalizálásában [7]. Az első nehézség a vasúti infrastruktúra kapacitásának növelése. Különösképpen a városi és sűrűn lakott területeken, a sínek száma korlátozott és elfogadható költséghatárokon belül nem bővíthető. A kapacitás növelésének másik módja, a járatok számának növelése. Ez már végrehajtható, de ugyan úgy rejt veszélyeket, hiszem a megemelkedett járatszám együtt járhat a késések számának növekedésével, okozhat hosszabb várakozási időt. Ezen kívül lassíthatja a szerelvények átlagos sebességét, amennyiben valamilyen váratlan, zavaró esemény történik, mely beavatkozást és átütemezést igényel. Ez azonban kiküszöbölhető, amennyiben rendelkezésre áll hatékony módszer az automatikus átütemezésre. Az emberi hibák minimalizálása érdekében használható a költséges automatic train controll (ATC) vagy automatic train opertion (ATO). Ez segít az emberi operátoroknak a döntéshozatalban. Váratlan szituáció esetén újratervezi a feladatokat a vasúti biztonsági előírásoknak megfelelően. Az eredeti menetrendtől való minden fajta eltérés késésekhez vezethet, melyek másodlagos késéseket is okozhatnak az érintett vonatoknál.

Számos konfliktushelyzet kezelhető a menetrend átütemezésével. Ilyenkor egy lokálisan konfliktus mentes helyzetet alakítunk ki, ez azonban eredményezhet késést egyéb szerelvényeknél, különösen a limitált infrastruktúrájú (például csak egy sínnel rendelkező) szakaszokon. A hatékonyság függ a kontrollált terület nagyságától. Minél nagyobb területen, minél több szerelvényt veszünk figyelembe az újratervezés során, annál jobb eséllyel kerüljük ez a másodlagos késések megjelenését. A komplexitása miatt, ez sok esetben időigényes lehet, azonban egy nagy komplexitású átütemezési probléma is kevés idő alatt megoldható válhat a mesterséges intelligencia alapú módszerek segítségével.

Az újratervezési algoritmusok a futhatnak az irányítóközpontokban (OCC: Operation Controll Center), mivel itt történik nagyobb területek távoli megfigyelése és irányítása. Az algoritmus hatékonysága a megvalósíthatóságán és a futási idején múlik. Megvalósíthatóság alatt azt értjük, hogy képes figyelembe venni a vasúti biztonsági követelményeket, megtartja a közlekedés minőségét pontos, és mindemellett a késéseket minimalizálja. A döntéshozatalnak komoly hatása van a szerelvények útvonalára és sebességére, így a döntési idő kulcskérdés lehet konfliktushelyzetek esetén. A döntés végrehajtása függ a vonatok gyorsítási és lassítási képességétől, az irányítórendszerek korlátaitól (ATP, ATC, ATO), és a döntéshozatal időpontjától. A konvencionális diszpécser rendszerekben a vonatok pontos helye nem ismert, mivel az interlocking rendszer track vacancy rendszer segítségével biztosítja a biztonsági előírások betartását. Egy adott szerelvény pontos helyét csak becsülni tudjuk, mivel egy adott foglaltság jelző szekció (sínáramkör) hossza állandó és jellemzően nagyobb, mint a szerelvény hossza. Bináris módon működik, a sínáramkör jelezhet szabad vagy foglalt státuszt. Ezen státusz akkor változik, ha egy új vonat legelső tengelyével belép a sínáramkör által figyelt területre, vagy az legutolsó tengelyével kilép onnan. A modernebb diszpécser rendszereknél (melyek együttműködnek ETCS L2 vagy L3 vonatirányító rendszerekkel) ez a probléma nem jelenik meg, mivel a GSM-R hálózaton és az RBC egységen keresztül elérhető a vonat pontos helyzete, egy konfidencia intervallumon belül. Ezen tanulmány konvencionális diszpécser rendszert feltételezve próbál egy megoldást találni, mivel ezek jóval elterjedtebbek és a modernizáció igen lassú ezen rendszerek esetén.

Az első lépés egy optimális megoldáshoz a deadlock elkerülése. Ez azt jelenti, hogy minden vonatnak találnia kell egy szabad útvonalat a célállomásig, melyben nem keresztezi az útját más vonatoknak, illetve nem zárják el egymás útját kölcsönösen a szerelvények. A valóságban a vasúti hálózatot felügyelő biztonsági rendszerek megakadályozzák az olyan helyzeteket, amelyek balesethez vezetnének. A hálózat reprezentálására használhatunk egy irányított gráfot, melyet a váltók határoznak meg.

1.2 Kapcsolódó irodalom

Egy már létező algoritmus a tabu search, mely az optimális vonat szekvenciák kiszámításával minimalizálja a késéseket [5]. Az alternate graph módszer szintén használható a probléma megoldására [16]. Ez egy konfliktus felismerő és megoldó módszer, mely valós idejű forgalomirányítási rendszerekben is működőképes (ROMA: Railway trafic Optimization by means of Alternate graphs) [6]. További megoldásokat találhatunk mixed linear integer programming módszer felhasználásával. [8]. Adott szerelvények megállóinak csökkentéséve, ezen megoldás akár az energiafogyasztás optimalizálására is törekedhet. Az implementáció történhet OpenTrack környezetben [15]. Az OpenTrack egy valós-idejű vasúti szimulátor, melyben különböző közlekedési helyzetek megvalósíthatóak. Külön tárolja a vasúti infrastruktúrát, a vonat állományt és a menetrendet. Az infrastruktúra reprezentálására szolgáló gráf élei a sínáramköröket, csomópontjai pedig a váltókat, jeladókat képviselik. A valós-idejű szimuláció közben instrukciókat is lehet adni a programnak az application programming interface-en (API) keresztül. Például tetszőleges vonatot lehet gyorsítani, lassítani vagy akár megállítani.

A metró rendszer meghibásodások esetén elvégzendő átütemezési feladat megoldására ajánl egy neighbourhood search algoritmust a [3]-as cikk. A módszer valós hálózaton is működőképesnek bizonyult [2]. A megoldások közt található, offline eljárás is mely mikroszkopikus és sztochasztikus szimulációt használ a zavarhelyzet kezelésére, az utazásra való kereslet és a szolgáltatás végrehajtásának figyelembevételével. Az eljárást sikeresen alkalmazták a Nápolyi metróvonalak által nyújtott szolgáltatások javítására.

A fentebb említett módszereken és eljárásokon túl, a mesterséges intelligencia alapú megközelítés is alkalmas lehet komplex problémák megoldására. Az átütemezése probléma esetén különösen igaz, hogy az ilyesfajta megoldások hatékony megoldást tudnak biztosítani. Alkalmazhatunk például Monte Carlo fakeresési eljárást, mely rövid idő alatt talál konfliktus mentes útvonalat [9]. A valóságos vasúti infrastruktúra magas komplexitására (alapvető jellemzők, limitált és különböző mennyiségű sínpár, biztonsági előírások, szerelvény modellek) való tekintettel, célszerű egyszerűbb modell alkotása. Ilyen például a Flatland környezet [13], mely egy grid modell alapján tanítható multiágens módszerekkel. Egy gráf reprezentáció és mély Q-hálózat segítségével (DQN: deep Q-network) is megoldást találhatunk a problémára [14], [10]. Egyes felhasználásokban, mint például egy zárt metró hálózat, a Q-learning akár energia-optimális megoldást is nyújthat.

1.3 Kontribúció

Ezen dolgozatban egy multiágensű mély megerősítéses tanuláson alapú megoldást mutatunk be a valós idejű vasút átütemezési probléma megoldására. Az első, kép jellegű állapotreprezentáció feldolgozását és értelmezését egy konvolúciós neurális háló végzi. Képfeldolgozási problémák esetén a konvolúciós hálók a legelterjedtebbek, így ez összhangban van az általunk alkotott reprezentációval. A második bemutatásre kerülő reprezentáció egy vektorba fűzve tartalmaz minden információt. Ehhez lineáris hálót alkalmazunk. Tekintve, hogy a valós problémakör igen komplex, számos különböző célt kitűzhetünk a megoldás során. Mind a reprezentáció, mind a jutalmazási rendszer az alapján kerül kidolgozásra, hogy milyen szempontból szeretnénk optimális megoldásra jutni. A bemutatott megoldás fő célja a deadlock, tehát a szimpla útvonalváltoztatással nem megoldható torlódás megakadályozása. Ez akkor jelentkezik, ha egymással szemben haladó szerelvények egy időben ugyan azon sínszakaszt használják, oly módon, hogy egyikük sem tud egy váltópontnál másik útvonalat választani, tehát valamely szerelvények visszafele kellene haladni, hogy a közlekedés folytatódhasson. Emellett azt is szeretnénk elérni, hogy a vonatok minél hatékonyabban, minél kevesebb idő alatt érjék el a célpozíciójukat. Összefoglalva a probléma továbbá torlódás nélküli és minimális menetidővel rendelkező megoldására törekszünk. A legfontosabb kontribúció azonban a virtuális ágensek bevezetése, mivel a dolgozat egy másik célkitűzése a generalizáció minél magasabb fokú támogatása. A neurális hálók egy fontos tulajdonsága a generalizáció, tehát hogy egy adott probléma megoldása után, ugyan az a háló hasonló, de komplexebb problémát is képes megoldani. A következőkben bevezetjük a virtuális ágensek fogalmát, és megmutatjuk, hogy a segítségükkel hogyan lehet a neurális hálót egy állomás használatával tanítani, majd az eredményeket több egymás utáni állomás esetén felhasználni.

Chapter 2

Algoritmusok

2.1 Megerősítéses tanulás

A Megerősítéses tanulás (RL: reinforcement learning) a gépi tanulás egyik válfaja. A felügyelt és felügyeletlen módszerekkel ellentétben, itt nem egy előre meghatározott adathalmazból tanul az algoritmus, hanem kísérletezés és tapasztalan alapján. A megközelítés lényege, hogy előre elérhető adatok helyett a tanító adatok működés közben generálódnak a környezettel való interakció során. A hálózat a megfelelő viselkedést egy jutalom (reward) érték alapján sajátítja el, melyet szintén a környezet biztosít, az alapján, hogy a háló által alkalmazott megoldás, vagy tett lépés mennyivel viszi közelebb a folyamatot a céljához. A keretrendszer Markov döntési folyamatként (MDP: Markov Decision Process) van megfogalmazva. Ahogy minden fajta gépi tanulás, ezen metodológiát is a természet ihlette. Az RL a "felfedező" vagy "kalandor" típusú tanulási folyamatot valósítja meg, mint ahogy akár az állatvilágban a különböző egyedek, akár az embereket nézve a gyerekek a saját tapasztalataikon keresztül ismerik meg a körülöttük lévő világot. A megerősítéses tanulás a neurális hálózatok, illetve a kontroll elmélet keresztmerszedében helyezkedik el. Az ágens olyan kontroll stratégiát (policy) keres melynek segítségével a kíván módon tud interakcióba lépne egy komplex környezettel. Mindezt a környezettől kapott pozitív vagy negatív jutalom, illetve a környezet aktuális állapota (state) függvényében. Mély megerősítéses tanulás esetén a stratégiát egy neurális háló tartalmazza, és a stratégia keresése, illetve fejlesztése egyet jelent a háló súlyainak hangolásával.

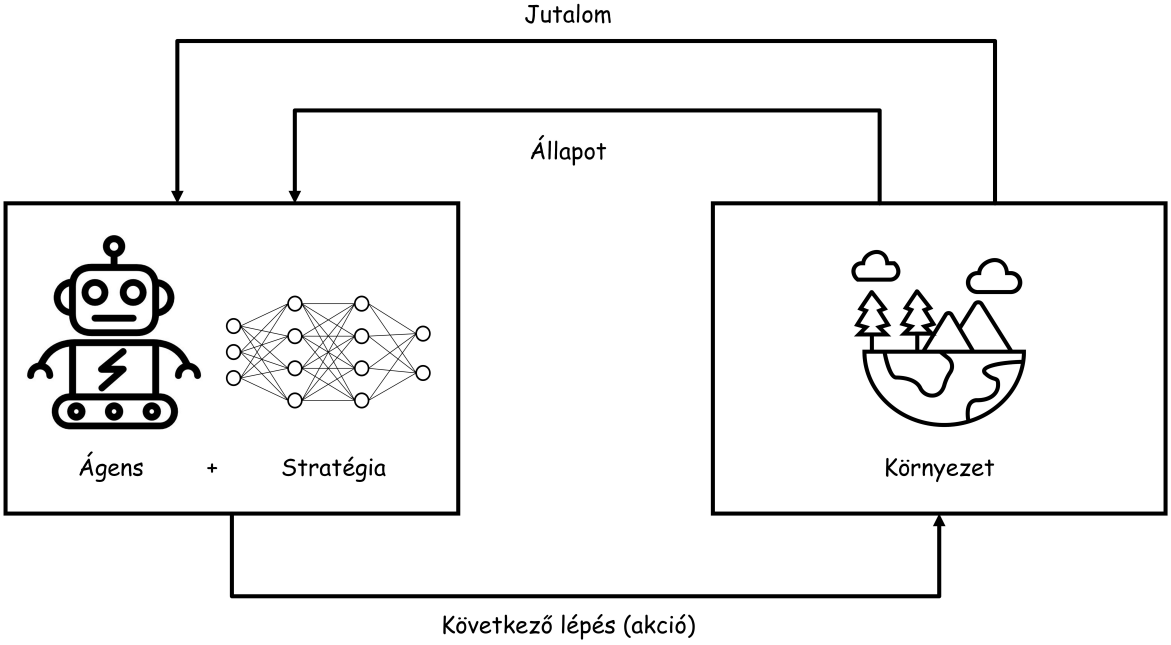

Figure 2.1: Megerősítéses tanulás menete

2.1.1 Deep learning

A gépi tanulás a mesterséges intelligencia egyik változata, a deep learning pedig a gépi tanulás egyik változata. Míg az egyszerű gépi tanulás kevésbé komplex korrelációk felismerésére és problémák megoldására alkalmas, jellemzően kevesebb adat felhasználásával, a deep learning az emberi neurális működés példájából kiindulva, képes bonyolultabb összefüggések felismerésére. Itt a tanulási folyamat során emberi beavatkozásra nincs szükség, a haló magától épül. Ezt szemlélteti a 2.2 ábra. A mély neurális hálók sikerét követően megjelent a mély megerősítéses tanulás (DRL: deep reinforcement learning). A DRL jelenleg egy intenzíven kutatott téma, de a mély neurális hálót használó RL alkalmazások már rengeteg sikert tudnak felmutatni. Komoly kihívást jelentett például olyan program létrehozása, mely az embereknél jobban tud teljesíteni az Atari játékokban. A [12]-ben leírtak szerint sikeresen létrehoztak olyan hálózatot, mely számos Atari játékot magas szinten tud megoldani, akár olyan taktikák felhasználásával, melyre az emberi játékosok nagy része sem jön rá. Egy másik példa az AlphaGo néven elhíresült hálózat, mely képes volt legyőzni profi GO játékosokat, köztük a világ legjobbjának tartott Ko Csie is verséget szenvedett 2017-ben, annak ellenére, hogy a GO a világ legkomplexebb és összetetted stratégiai játékainak egyike.

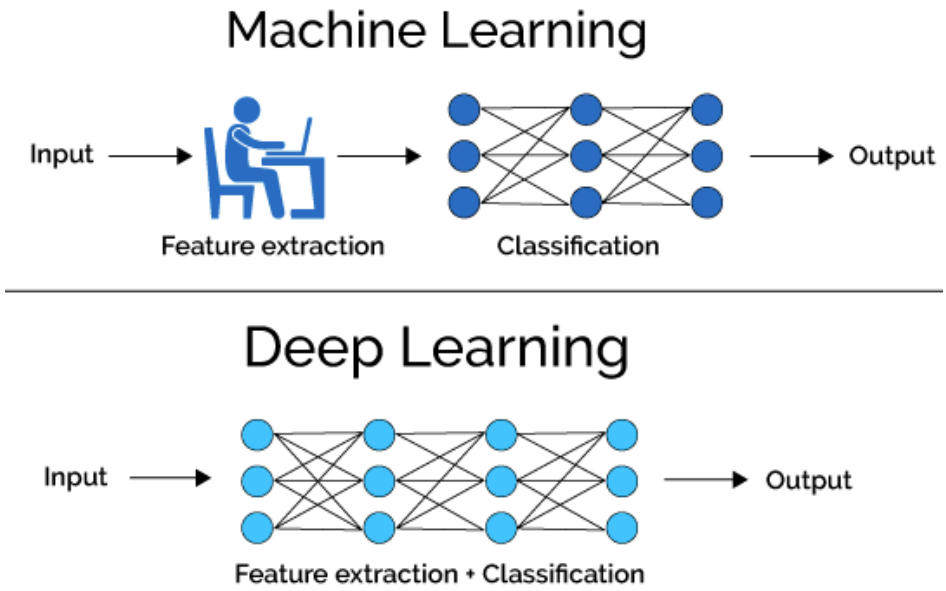

Figure 2.2: Machine learning és deep learning közti különbség [1]

2.1.2 Multi-ágens koncepció

Multi-ágens DRL során több játékos egyszerre játssza a játékot, azaz több ágens párhuzamosan hoz döntéseket, dolgozik valamilyen probléma megoldásán. Ilyenkor az ágensek interaktálnak a környezettel, valamint sok esetben egymással is (centralizált kontroll). Ez a megnövekedett dimenzionalitás illetve a skaláris jutalmazási rendszer miatt valamelyest megnehezíti a tanulási folyamatot is. Az ezen dolgozatban használt koncepció lényege, hogy az ágensek külön-külön mennek végig a tanulási folyamaton, azáltal, hogy a közös függvény becslő minden esetben egyszerre csak egy ágens számára végzi el a számításokat. Éppen emiatt a reprezentáció úgy van kialakítva, hogy az aktuálisan vizsgált ágens nem tesz különbséget a környezet és a többi ágens között, a környezet részének tekinti azokat. Ezzel kikerüljük a dimenzionalitás problémáját, azaz hogy amennyiben minden ágens tud egymásról, a felhasznált függvény becslő függene az ágensek számától. Így minden esetben mikor változtatjuk az ágensek számát, a használt aproximátor is változni fog. Ez egy kevésbé robosztus megoldást eredményez. Jelen helyzetben az akciótér (amely a lehetséges döntéseket tárolja) minden ágens számára állandó és változatlan. Ez szintén nem lenne igaz centralizált kontroll esetén, hiszen a különböző döntések kombinációi válnának az akciótér elemeivé. Ismét a dimenzionalitás és az ágens számosság függés problémájával szembesülünk, melyet az általunk használ megközelítés kiküszöböl. Ez általában hatékonyabb konvergenciát is eredményez.

2.1.3 Felhasznált neurális háló

Ahogy minden hálózat esetén, itt is a neurális háló súlyait (azaz a neuronok közti összeköttetést) hangoljuk. A végső cél, hogy a bemenetként érkező állapot alapján a háló ki tudja választani a megfelelő lépést egy adott ágens számára. A háló súlyai tartalmazzák a kialakított stratégiát. Mivel a probléma sok esetben sztochasztikus vagy nem lineáris, az elsajátított stratégia is probabilisztikus jellegű. Az outputként kapott Q értékek, melyek eldöntik a következő lépést, nem egyértelmű választ adnak, hanem gyakorlatilag egy valószínűséget arra nézve, hogy melyik lépés vezet leggyorsabban, illetve legbiztosabban a leginkább pozitív reward érték irányába. A tanítás során az első izgalmas kérdés minden lépés előtt az exploration/exploitation tradeoff, azaz hogy az ágens egy véletlenszerű lépést tegyen, ezzel új, eddig nem látott állapotokat felfedezve, vagy a már látott állapotok segítségével hangolja tovább a súlyok értékét. Itt az epsilon-greedy stratégiát használva minden esetben generálunk egy véletlen számot uniform eloszlással. Amennyiben ez a szám egy előre beállított epszilon érték alatt van, a lépés véletlenszerű lesz, amennyiben fölötte, a legmagasabb Q értékhez tartozó lépést választjuk. Az epszilon értéke 1-ről indul, majd a tanulási epizódok számától függően valamekkora ütemben csökken, így a folyamat elején nagy eséllyel fedez fel újabb útvonalakat, a végén pedig a meglévők javítására koncentrál. Az aktuális lépés hasznosságát eldöntő veszteségfüggvény számításához szükséges érték a Bellman-egyenletből adódik:

$$Q(s_t, a_t; \theta_t) = r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t; \theta_t^-) \tag{2.1}$$

ahol s_t az aktuális állapotot jelöli t pillanatban, a_t a választott lépés szintén a t pillanatban. γ a discount factor, ami azt jelzi, hogy az ágens milyen mértékben akar tanulni a távolabbi jövőből, mekkora hangsúlyt fektet a jövőbeli jutalom lehetőségére. Amennyiben a γ érték 0, az ágens nem törődik a jövőbeli potenciállal, és csak azon lépések kapnak jó értéket, melyek azonnali jó eredményt produkálnak. Amennyiben a γ értékkel közelítjük az 1-et, az ágens egyre nagyobb hangsúlyt fektet a jövőbeli jutalom értékekre. Ez azért fontos, mert a stratégiának szem előtt kell tartani a lépések távolabbi következményét, de képesnek kell lennie felhasználni a legújabb tapasztalatokat. θ a főháló súlyainak értékeit tartalmazza, míg θ^- a célhálóét, melyet nem tanítunk, csak bizonyos időközönként szinkronizáljuk a főhálóval. Erre azért van szükség, mert állandóan változó bemenetet próbálunk leképezni egy állandóan változó kimenetre. Így ez a probléma nem áll fent. Végül r_{t+1} a környezettől kapott reward érték a $t+1$ időpillanatban. Mint az látszik, az új érték a jelenlegi jutalom értékétől függ, illetve a jövőbeli jutalomtól a discount factor mértékében. mivel az aktuális jutalom mindig nagy hangsúlyt kap, a háló folyamatosan a megfelelő kimenet irányába halad

a jelenlegi és a múltbéli tapasztalatai kombinálásával.

Konvolúciós háló

A képfeldolgozás területén leggyakrabban használt neurális hálótípus a konvolúciós háló. Ezt azért nevezik így, mert konvolúciós rétegek sorba fűzéséből adódik. Egy konvolúciós réteg a teljes bemeneti mátrixon végigcsúsztatott kernel ablak által közrefogott elemek konvolúcióját kiszámolva állítja be a kimeneti mátrix elemeit. Képfeldolgozás esetén a dimenzió redukció mellett, a kép jellemzőinek felismerését szolgálja. Az első réteg azegyszerűbb jellemzőket képes felismerni (sarkok, élek), és ahogy egyre több réteget kötünk egymás után, annál komplexebb dolgokat képes a háló felismerni, az egyszerű jellemzők kombinálásával. Ezt általánosabban úgy is meg lehetne fogalmazni, hogy az egyszerű mintázatokból alkot bonyolultabb mintázatokat. Mivel a 2.2.1 bekezdés kétdimenziós reprezentációra vonatkozó részében leírtak szerint, egy kép jellegű reprezentációt használunk, a konvolúciós háló összhangban van a reprezentációval. Esetünkben a kép jellemzői helyett a közlekedés szabályait, pontosabban az egyes állapotok jelentését és jelentőségét a közlekedés szabályaira vonatkozóan akarunk megértetni a hálózattal.

Lineáris háló

Egy másik ismert neurális háló típus a fully connected network amely több fully connected layer-ből áll. Ennek lényege, hogy egy réteg minden neuronja összeköttetésben van a következő és elő réteg minden neuronjával. A 2.2.1 bekezdés egydimenziós reprezentációjához egy ilyen, teljes összeköttetésben álló lineáris rétegekből felépülő hálózatot alkalmaztunk. Míg a konvolúciós háló egy kernelablakon konvolválja az ott található értékeket, a lineáris háló esetén ilyen köztes művelet nem történik.

2.2 Környezet

A környezet modellezése szintén egy fontos feladat. Mivel az ágens a környezet állapota alapján tanulja meg a megfelelő lépések, a környezet illetve a környezet, változásának mintázatait figyelve sajátítja el a célravezető viselkedést, lényeges lépcsőfok a tanításban a megfelelő környezet megtervezése. Ebbe nem csupán az tartozik bele, hogy minden fontos információt tartalmaznia kell, hogy a hálónak legyen esélye megtanulni a szabályokat, de az is, hogy lehetőleg fölösleges adatok ne jelenjenek meg, amelyek összezavarhatják az ágenst. Érdemes a minimális mennyiségű adatot kiválasztani mellyel a tanítás még hatékony. A vasúti infrastruktúra esetén például befolyásolja a közlekedést az egyes szakaszok hossza, a kanyarodó részek sugarának

hossza, a szakasz emelkedése vagy süllyedése, a különböző típusú vonatokra kiszabott sebességkorlátozás és egyéb vasúti szabályozások. A valóságban a vonatok lassítási és gyorsítási képessége is fontos faktor lehet. Azonban addig amíg az ágensek nem tanulták meg az egyes pozíciókban tehető lépéseket, valamint egymást elkerülve eljutni a céljukhoz, további szabályozások fölöslegesen nehezítik a tanulást és a megfelelő minták felismerését az ágensek számára. Az általunk használt egyik környezetmodell egy kép jellegű reprezentáció a környezetről valamint a saját és egyéb ágensek pozíciójáról. Ezen kívül bemutatásre kerül egy másik, egydimenziós reprezentáció is, amely szintén tartalmazza az összes ágens pozícióját, illetve végcélt, csak kompaktabb módon. Az elsődleges célunk a deadlock elkerülése, aza minden esetben olyan útvonal felderítése és végigjárása minden szerelvény számára, amely további fennakadással nem jár. Ezen kívül cél a minél rövidebb útvonal és utazási idő elérése minden ágens számára. Az általunk vizsgált állomás modell két végpozíciót tartalmaz, egyet a pozitív irányba (jobbra) haladó, egyet a negatív irányba (balra) haladó szerelvények számára. Az ágensek által választható útvonal irányhoz kötött. A vonatok nem haladhatnak visszafele, így a kereszteződéseket elérve, csak a megfelelő irányba haladó ágens választhatja a kétfelé ágazó útvonalak (vágányok) egyikét. Kereszteződésnek számít az, ahol több mint kér útvonal találkozik. Itt a megfelelő irányba haladó ágensek dönthetnek, hogy melyik irányba akarnak tovább haladni, vagy várakoznak. Mikor egy ágens éppen nem ért kereszteződéshez a két opció, hogy tovább halad, vagy pedig várakozik, ami szintén egy fontos elsajátítandó stratégia, például olyan esetekben, ahol a továbbhaladás deadloc-ot eredményezne. Tanítás során azonban választhatók olyan irányok is, melyek addot pozícióban nem bejárhatók, mivel a háló ezen keresztül tudja megtanulni a valid lépéseket a különböző pozíciókban.

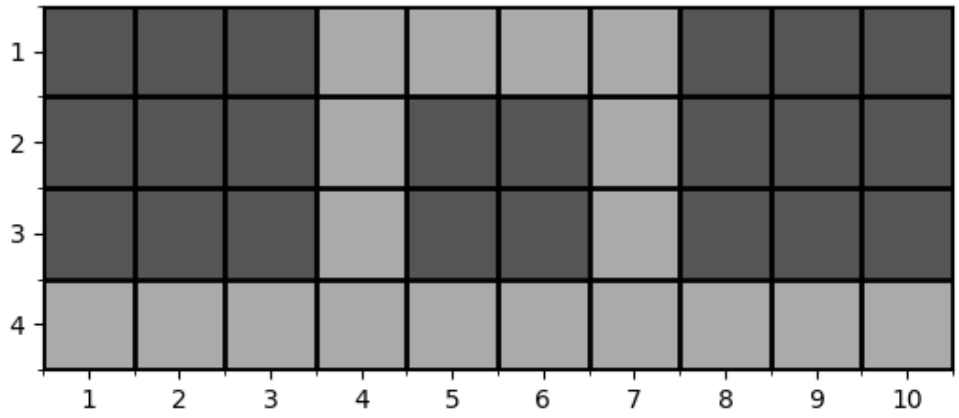

Figure 2.3: Az állomás alaprajza

A 2.3 ábrán az útvonalak világos, a falak sötétszürkével vannak jelölve. A cellák az egy-egy sínáramkör által határolt szakaszt jelzik. Minden szerelvényről annyi információ áll rendelkezésünkre, hogy mely sínszakaszon belül tartózkodik éppen, azaz mely cellát foglalja el. Egy szakaszt egyszerre egy vonat foglalhat csak. Az alaprajz két pontján található váltó, azaz olyan cella, melyből a megfelelő irányba haladó szerelvény két irányban is folytathatja az útját.

2.2.1 Állapotreprezentáció

Az állapotreprezentáció kidolgozása a tanítás egyik kulcsfontosságú lépése. Az ágens egyedül ez alapján tudja megérteni a környezet működését, így minden fontos információt tartalmaznia kell a kontroll probléma leírásához. Átfogó megoldás vagy keretrendszer nem létezik, a környezetet létrehozó kutató, intuiciójára és elképzeléseire támaszkodva alakítja ki.

Kétdimenziós reprezentáció

A reprezentáció létrehozásánál érdemes a reward rendszer, a konvergencia és a végső teljesítmény támogatását is szem előtt tartani amennyiben lehetséges. Törekedtünk továbbá laza reprezentáció létrehozására. Ez annyit jelent, hogy az adatokat minél szeparáltabban tároljuk, annak érdekében, hogy könnyeb legyen szétválasztani az állapotteret és észrevenni benne a mintázatokat. Tapasztalatok szerint, ez növeli a konvergenciát. Jelen esetben például, minden ágens minden lépésben létrehoz egy saját reprezentációt a környezet aktuális állapotáról. Ezen reprezentáció öt különböző

csatornából áll:

- 1. csatorna: Az első csatorna az magának az állomásnak a felépítését mutatja. Az alaprajzot jelképező mátrixban a szabad útvonalakat 0 jelöli, a falakat pedig 1-es. Annak jelzésére, hogy a vonat hátrafele nem haladhat, azon pozíciókba melyek visszalépést jelentenének rakunk egy "falat" azáltal, hogy az ottani étéket az aktuális lépés során 1-esbe állítjuk. Mivel az egész reprezentáció felfogható, mint falakkal határolt ösvények, a helyes irány ilyen fajta jelzése nemcsak az irányfüggést segít megtanítani, de a generalizációt is támogatja.

- 2. csatorna: Ezen csatorna az ego, tehát az aktuálisan vizsgált ágens pozícióját tartalmazza. a pozíció értéke1, még a többi celláé 0.

- 3. csatorna: A célpozíciót mutató csatorna, azon pozíció melyet az ágensnek el kell érni 1-es értéket kap, a többi 0.

- 4. csatorna: Ez a csatorna az azonos irányba haladó vonatokat mutatja, tehát azon ágenseket, melyeknek célpozíciója megegyezik az aktuálisan vizsgált ágensével, és így ugyanazon irányba tudnak lépni. Ismét 1 az értéke az ágenseknek, és 0 az összes többi cella.

- 5. csatorna: Az utolsó csatorna 1-essel jelzi az ellentétes irányba haladó ágenseket, és 0-val tölti fel a mátrix többi részét.

Ez megkönnyíti a hálózat dolgát az adatok értelmezésében, például a különböző irányba haladó ágensek felismerésében. Azáltal valósul meg a laza reprezentáció, hogy a különböző jelentőséggel bíró adatok külön csatornákon jelennek meg, a logikailag összetartozóak pedig ugyan azon.

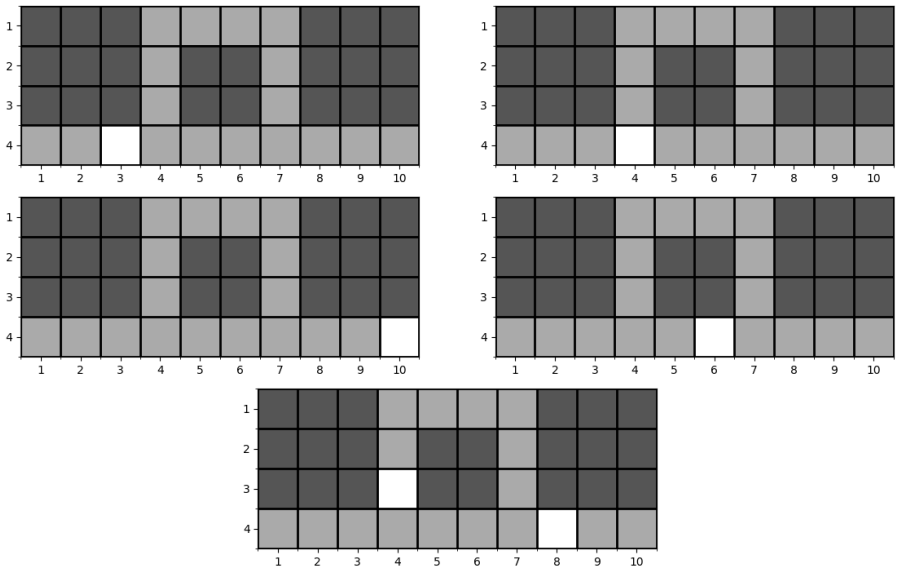

Figure 2.4: Egy ágens 2D állapotreprezentációja

A 2.4 ábra egy példa az állapotreprezentációra, ahol az eredeti állomás tervrajzán az utak világos, a falak sötétszürkével vannak jelölve, az állapotreprezentáció által érintett pozíciók pedig fehérrel. A bal felső ábrán az alaprajzon a fehérrel jelölt cella is fallá alakul (azaz egyes értéket vesz föl) mivel a vizsgált ágens pozitív irányba halad, és ezzel jelezzük neki, hogy nem léphet hátrafelé. A többi ábrán az alaprajz csak a szemléletesség kedvéért szerepel. A jobb felső ábra a vizsgált ágens pozíciója. Mivel az ágens pozitív irányba halad, a bal középső ábra jobb oldali célpozíciója van kiemelve. A jobb középső ábra megmutatja, hogy ezen kívül még egy ágens ugyan azon irányba halad, valamint az alsó ábráról tudhatjuk, a két ellentétes irányba haladó ágens aktuális pozícióját. A különböző szerepet betöltő ágensek a reprezentációban is szét vannak választva.

1D reprezentáció

A második esetben a teljes állapotteret egyetlen vektorral jellemezzük egy csatornán. Enne az előnye, hogy egy csatorna egy dimenzióban egy jóval kevésbé komplex állapottérhez vezet, mint a kétdimenziós változat. Ezt segíti az is, hogy a vektor nem tartalmaz minden adatot, amelyet a kétdimenziós reprezentáció. Egyedül a sínáramkörök által meghatározott részek, azaz az előző esetben a szabad útvonalakat jelző csempék alkotják a vektort, mely három részre bontható. Mindegyik rész 18 elemből áll, tekintve hogy 18 bejárható csempe van, amely nem falat reprezentál. Ez tulajdonképpen

egy occupancy map, ahol az 1-es a foglalt érték, a 0 pedig a szabad.

- 1. rész: Tartalmazza az ágens saját pozícióját, valamint a célpozíciót. Így minden esetben két elem értéke lesz 1-es, a többi 0.
- 2. rész: Ez az ellentétes irányba közlekedő szerelvényeket mutatja. Ezek pozíciója 1-es értékű, a többi 0.
- 3. rész: Az előzőhöz hasonlóan a többi ágens helyzetéről ad információt. Az azonos irányba közlekedő ágensek pozíciója 1-es értéket vesz fel, a szabad helyek ismét 0-t.

Ezen részek összefűzésével kapjuk meg a teljes állapotreprezentációt. Így minden ágens pozíciója és a végpozíció is ismert, és ez alapján tudja az adott ágens jellemezni a környezetet.

Figure 2.5: Egy ágens 1D reprezentációja

A 2.5 ábra az egydimenziós reprezentáció három részletét mutatja, melyek később összefűzésre kerülnek. Ez lesz az egydimenziós reprezentációja ugyan annak a forgalmi helyzetnek, amelyet a 2.4 ábrán is láthatunk. Felül a saját és végpozíció, középen a két ellentétes irányba haladó ágens majd alul egy azonos irányba haladó ágens.

2.2.2 Virtuális ágensek

A problémakörhöz való hozzájárulásunk fontos része a virtuális ágensek bevezetése. A munkánk egyik fő célja a generalizáció támogatása, azaz hogy egy egyszerűbb problémán végzett tanítás után a háló képes legyen magasapp komplexitású feladat megoldására is. Jelen esetben a célunk az volt, hogy egy háló, mely egy állomás használatával tanul, képes legyen megoldani az átütemezési problémát számottevő arányban több összekapcsolt állomás esetén is. Az alapvető probléma az állomások egymás után kapcsolásával, hogy az ágensek alapvetően a saját állomásukra látnak rá, illetve amennyiben

más állomásokról is tudomásuk kell hogy legyen, az nagyban megnehezíti a reprezentáció általánosítását és az állapottér is sokkal összetettebbé válik. Így az állapottér fontos jellemzőinek szétválasztása is nehezebb, a tanulási folyamat jelentősen kisebb eséllyel fog megfelelő stratégiához vezetni. Ennek feloldására vezetjük be a virtuális ágenseket. Mint az fentebb már szerepelt, a reprezentációban érdemes a minimális mennyiségű hasznos információt szerepeltetni. Amennyiben egy ágens a saját állomására lát csak rá, veszélyes lehet átlépni a másik állomásra, mert előfordulhat, hogy ez deadlock helyzethez vezet. Egy ágens számára elég tehát csupán azt tudni, hogy a számára következő állomáson foglalt-e olyan pozíció, vagy pozíció kombináció ellentétes irányba tartó ágensek által, mely deadlock-ot eredményezhet.

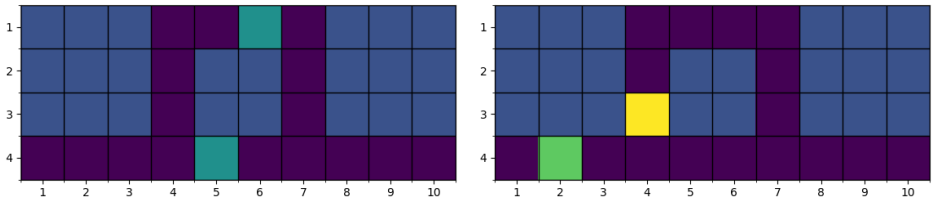

Figure 2.6: Virtuális ágens potenciális deadlock helyzetben

A 2.6 ábra két egymást követő állomást ábrázol. A bal oldali állomás jobb alsó cellájának elérésével az ágens átkerül a jobb oldali állomásra, és a jobb oldali állomás bal alsó cellája pedig a bal oldali állomásra vezet át. Az itt látható esetben például a kékkel jelölt pozitív irányba haladó ágensek oly módon állják el az utat, hogy a balra haladó sárgával jelölt ágens deadlock helyzetet eredményezne az útja folytatásával. Az azonban nem fontos információ számára, hogy pontosan miért, milyen formációban vezetne az előrelépés deadlock-hoz. Ahelyett, hogy az ágens reprezentációjában szerepelne a következő állomás, csupán a zölddel jelölt virtuális ágens szerepel. Ez igazából nincsen ott, jelen esetben is csak a szemléltetés kedvéért jelenik meg a közös reprezentációban. A jobbra haladó ágensek nem látja, csak a sárga ágens állapotterében jelenik meg, mint pozitív irányba haladó szerelvény. Az útját akkor sem folytathatná, ha a zöld cella helyén valóban ott lenne az ágens. Így egyéb állomásoktól függetlenül képes felismerni a torlódáshoz vezető veszélyhelyzetet. Továbbá a reprezentációban nincs különbség virtuális és valós ágensek között, így az elsajátított viselkedésmintát ez nem fogja befolyásolni. Ezzel tehát elértük, hogy kizárólag a saját állomás aktuális állapotától függ az ágens által elsajátított viselkedés, azáltal, hogy a saját reprezentációjába szükség esetén felveszünk virtuális ágenseket.

2.2.3 Akciótér

Megerősítéses tanulás során az ágens számára rendelkezésre álló cselekvések vagy választások halmazát, mely a környezettel való interakciót biztosítja, akciótérnek nevezzük. Az akciótérnek rendelkeznie kell két fontos tulajdonsággal: completeness és validity. A completeness (teljesség) akkor teljesül, ha az adott akciók segítségével az ágens képes elérni a célját. Előfordulhat olyan eset, hogy egy olyan akció hiányzik, amely szükséges lenne olyan trajektória bejárásához, amely a feladat sikeres teljesítéséhez vezet. A validity (érvényesség) pedig azt jelenti, hogy az akciótér csak olyan lépéseket tartalmaz melyez az ágens valóbaban meg tud tenni. Például egy vonat nem tud két cella között átlósan közlekedni, vagy egy másik szerelvényt átugrani. Az általunk használt kétdimenziós reprezentációhoz tartozó diszkrét állapottér 5 különböző lépést tartalmaz. Az ágens választhat a négy irány között (jobbra, balra, fel, le) illetve dönthet úgy hogy várakozik. Ezen utolsó opció fontossága azon esetekben jeletkezik, melyekben a deadlock csak akkor elkerülhető, ha az egyik szerelvény vár más szerelvények továbbhaladására. Ilyen módon definiált akciótér esetén lehetséges, hogy az ágens olyan lépést választ, amely adott cellában nem elérhető, ami elsőre lehet hogy nem tűnik hatékonynak, ugyanakkor ezáltal elérjünk, hogy az ágens megtanulja az általában lehetséges lépések közül a megfelelőt adott pozícióban. Ezene felül így az adott lépések ugyan azon jelentést képviselik az aktuális pozíciótól függetlenül, ami támogatja a generalizációt. Így tehát az akcióterünk aktuális állapota a többitől független és konzisztens a reprezentációval. Az egydimenziós reprezentáció hasonló módon működik. Azonban az nem tartalmaz falakat, így ott az ágens választhat hogy tovább halad (kereszteződésben a két irány egyikébe) vagy mint a kétdimenziós esetben, várakozik. A két reprezentáció közül az utóbbinak nagyobb az információsűrűsége.

2.2.4 Jutalmazási stratégia

A tanítás sikerének egy másik kulcsa, a reward stratégia. Ez szintén a mérnöki intuíción alapszik. Az ágensek a reward érték segítségével kapnak visszajelzést hogy adott állapotban egy adott lépés mennyire minősül jónak, mennyivel juttat közelebb a cél eléréséhez. Míg az állapotreprezentációból tudják, hogy néz ki a körülöttök lévő világ, a jutalom értéke jelzi, hogy jó irányba haladnak-e. Amennyiben pozitív jutalmat (megerősítést) kapnak, az adott lépést nagyobb eséllyel fogják választani, amennyiben negatívat, kevésbé valószínű, hogy következő alakalommal is azt lépik.

Teljes versegő stratégia

A legegyszerűbb stratégia tehát az lenne, hogy amennyiben egy ágens befejezte a játékot, kap egy pozitív jutalmat, amennyiben dedlock jön létre vagy nem sikerül megtalálni a kiutat, kap egy negatív jutalmat. Mivel számunkra nem csak az a cél, hogy deadlock nélkül megoldják a problémát, de az is, hogy minél hatékonyabban, tehát minél rövidebb úton és minél kevesebb lépésből érjék el az úticéljukat, ezt is valamilyen módon számításba kell venni a jutalom számításánál. Ezt úgy tesszük meg, hogy minden egyes megtett lépés után kapnak egy negatív jutalmat, így minél több lépést tesznek annál több negatív pontot gyűjtenek, majd végül az ágens mely elérte a célját, kap egy nagy pozitív jutalmat, melyből az addigi lépések után összegyűlt negatív összeg levonódik. Gyakorlatilag a lépésszámmal skálázzuk a végső megszerezhető jutalom nagyságát. Mindemellett, amennyiben egy ágens deadlock helyzetet teremt, egy nagy negatív jutalmat kap, hogy ezt mindenképpen elkerülje.

$$\text{jutalom} = \begin{cases} r_d, & \text{deadlock esetén} \\ r_c - r_s, & \text{sikeres teljesítés esetén} \\ -r_s, & \text{maximális lépésszám meghaladása teljesítés nélkül} \end{cases}$$

ahol r_d egy nagy negatív jutalom, r_c egy nagy pozitív jutalom, r_s pedig a lépések függvényében meghatározott jutalom, mely levonódik a sikeres teljesítésért járó jutalomból, vagy amennyiben az ágensek kifutnak a maximális lépésszámból, ezen lépésszámnak megfelelő negatív jutalmat kapnak. Fontos, hogy ezen jutalom minden ágens számára egyénileg számolódik. Ezzel a full competitive learning koncepcióját valósítjuk meg, azaz minden ágens egymástól függetlenül kap jutalmat, és mindegyik a saját célját akarja elérni, anélkül hogy egymást figyelembe vennék.

Kooperatív stratégia

Habár a kezdeti sikereket ezen koncepcióval értük el, sikerült javítanunk a jutalmazási stratégián azáltal, hogy figyelembe vettük a közös célt is. Amellett, hogy minden ágens célja, hogy ő maga eljusson a végcéljáig, közös cél, hogy fennakadás nélkül, mindenki eljusson a kijelölt célig. Amennyiben egy ágens eljut saját céljáig, de más ágensek képtelenek sikerrel zárni az epizódot, esetleg deadlock-ra futnak, a közös neurális hálónak és döntéshozatalnak része lesz a mintázat, melyben egy ágens pozitív jutalomban részesült, annak ellenére, hogy más ágensek egymás útját elállva lehetettlenné tették a sikeres befejezést. Ennek elkerülése végett bevezettünk egy negyedik jutalom fajtát is, mely nem bünteti a játékon sikeresen befejezett ágenst, ugyan akkor nem is jutalmazza,

hiszen a közös cél nem teljesült.

$$\text{jutalom} = \begin{cases} r_d, & \text{deadlock esetén} \\ r_c - r_s, & \text{sikeres teljesítés minden ágens számára} \\ r_n, & \text{sikeres teljesítés adott ágens számára, de a közös cél nem teljesül} \\ -r_s, & \text{maximális lépésszám meghaladása teljesítés nélkül minden ágens számára} \end{cases}$$

ahol a jelölés változatlan, viszont bevezetésre kerül a r_n semleges jutalom.

Chapter 3

Eredmények

Ezen fejezetben az eddigiekben leírt módszereket és koncepciókat használó tanítás és kiértékelés eredményeit fogjuk bemutatni. Szemléltetjük a koncepció fejlődését és ezzel együtt a sikeres tantások és a generalizációra való képesség javulását. Először bemutatásra kerül az első olyan tanítás, mellyel ígéretes eredményeket értünk el, majd kifejtjük hogy miben változtattunk az eredeti hozzáálláson és ez hogyan befolyásolta a konvergenciát, a generalizációra való potenciált és a feladat megoldásának sikerességét. A sikeres megoldáson kívül a lépésszám kérdésével is foglalkozunk, mivel ahogy azt az eredeti célkitűzésben is hangsúlyoztuk, törekszünk a minimális útidő, azaz a minimális lépésszám elérésére. A tanítás minden esetben egy állomáson történik, a cél pedig, hogy az egy állomáson tanult háló két egymás után kapcsolt állomás esetén is meg tudja oldani az újraütemezési problémát.

3.1 Kezdeti koncepció

Első alkalommal minden epizód azzal kezdődik, hogy véletlenszerűen kiválasztunk két ágens pozíciót illetve a haladás irányát, szintén véletlenszerűen. Az egyetlen megkötés, hogy a kezdeti helyzet nem lehet alapvetően deadlock, mivel ebben az esetben az ágenseknek esélyt sem adunk arra, hogy megoldják a problémát. Ez minden tanításban közös. Az ágensek véletlenszerűen de kezdeti deadlock helyzet nélkül kerülnek elhelyezésre a hálózatban. Ezen kívül meg van szabva, hogy maximum hány lépésen belül kell teljesíteniük a feladatot. Minden alkalommal mikor az ágensek lépnek, amennyiben a lépés valid az adott pozícióban, ez végrehalytásra kerül. Ha egy ágens olyan lépést választ mely után a falba ütközne, a lépés nem halytódik végre, de a tanítási epizód nem áll le. A maximális megengedett lépésszámot több tanítás alatt gyűjtött tapasztalatok alapján választottuk úgy, hogy az alatt bármilyen kezdeti torlódás mentes pozícióból lehetséges legyen megoldani a feladatot, de ne legyen fölösleges

sok lépési lehetőség. A tanítás során a 2.2.4 fejezetben leírt első jutalmazási stratégiát alkalmaztuk. Mivel 2 ágens esetén a deadlock azt jelenti, hogy mindkettő elakad és képtelen folytatni az útját, nem volt szükséges, hogy számításba vegyük a közös célt, hiszem amennyiben a deadlock miatt a két ágens célja meghiúsul, a közös cél is automatikusan teljesíthetetlenné válik, és nem marad ágens aki ettől függetlenül a saját célját (hogy ő egyedül elérje a végpozícióját) meg tudná valósítani.

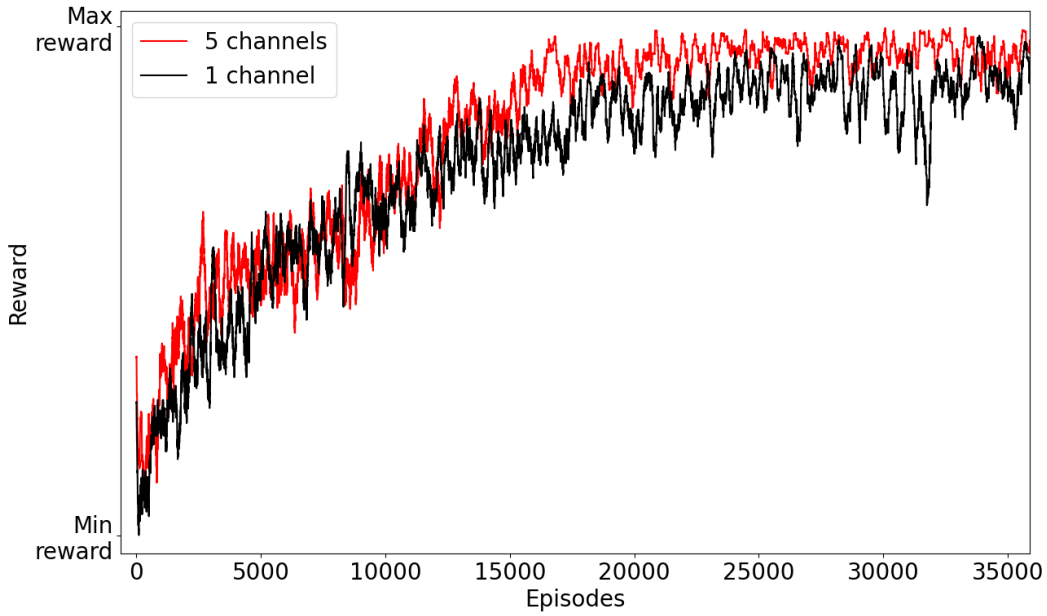

Figure 3.1: 2 ágens tanítás konvergeciája különböző csatornaszám esetén

A 3.1 ábra az első tanítás konvergenciáját mutatja. Érdekességképpen került csak ábrázolásra egy olyan tanítás melyben minden tanítási paraméter változatlan, viszont 5 helyett 1 csatornán ábrázoljuk az összes információt. Mint látszik, a végén ezen esetben is megközelíti a tanítás a maximális jutalmat, a betanult háló ugyan úgy képes megoldani a problémát, viszont a konvergencia egy kissé lassabb. A végére mindkét tanítás a maximális jutalom körül ingadozik. Nem konstans a maximális értéket veszi fel, mivel a reward érték függ a lépésszámtól, a lépésszám pedig függ a az ágensek kezdeti pozíciójától. Ugyanezen tanítást elvéheztük az egydimenziós reprezentáció segítségével is. Ezen reprezentáció esetén a konvergecia gyorsabb volt, ahogy az a 3.2 ábrán is látszik.

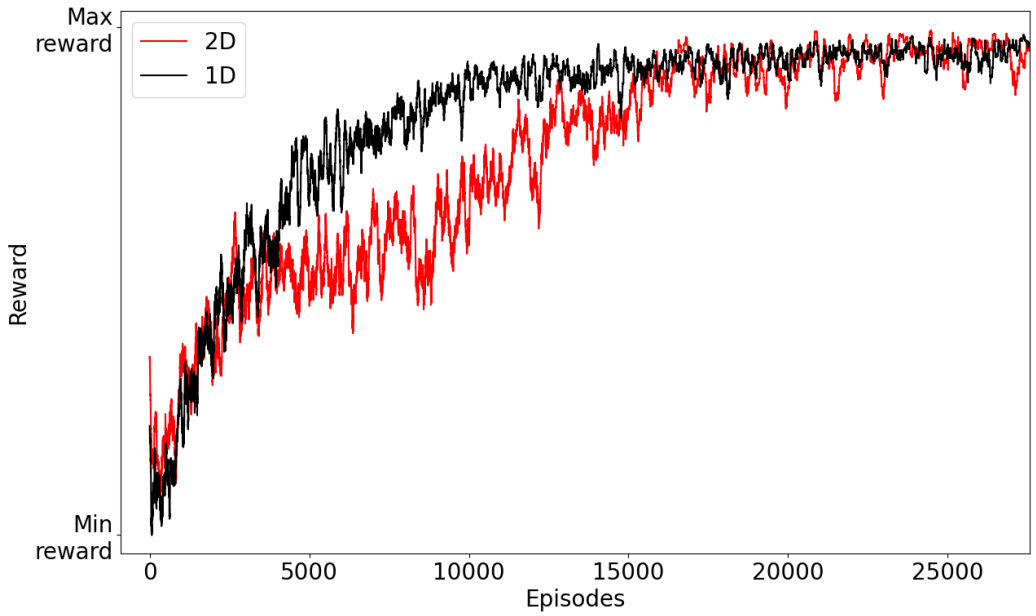

Figure 3.2: 2 ágens tanítás konvergenciája

Habár a konvergencia különbözik, mindkét reprezentáció esetén a végső betanított hálózat képes volt az eredeti problémát (1 állomás, 2 ágens) 100%-os hatékonysággal megoldani. Kiértékelés során minden esetben más random seed-et használtunk, mint tanítás során.

Sikerráta		
Reprezentáció	2 ágens, 1 állomás	2 ágens, 2 állomás
2D	100%	63.42%
1D	100%	70.26%

Table 3.1: Sikeres teljesítés aránya 2 ágenss tanítás után

Minthogy a generalizációt akarjuk vizsgálni, a cél, hogy a betanított hálózat 2 állomáson is jó eredményeket érjen el, azaz magas sikerrátával meg tudja oldani az átütemezési feladatot. Ahogy az a fenti táblázatból látszik, az ágensek gond nélkül tudták teljesíteni az eredeti feladatot, és két állomás esetén is sokkal nagyobb arányban találnak megfelelő megoldást, mint egy véletlenszerűen működő ágens, habár ennél magasabb sikerrátát vártunk. Erről a következő bekezdésben lesz bővebben szó. Az is látszik, hogy az egydimenziós reprezentáció valamivel hatékonyabban tudja megoldani a feladatot. Ez feltehetőleg az egyszerűbb állapottérnek köszönhető. Az egy vektorban tárolt információkat könnyebb lehet feldolgozni, illetve az akciótér is kevesebb elemből áll. Ezek mellett a reprezentáció nem tartalmaz falakat, így valamivel letisztultabb

módon találhatók meg benne a közlekedési szabályok megismeréséhez szükséges adatok. Mivel a menetidőre szeretnénk optimalizálni, érdekes számunkra a feladat teljesítéséhez szükséges lépésszám. Ezt mutatja a 3.3 ábra.

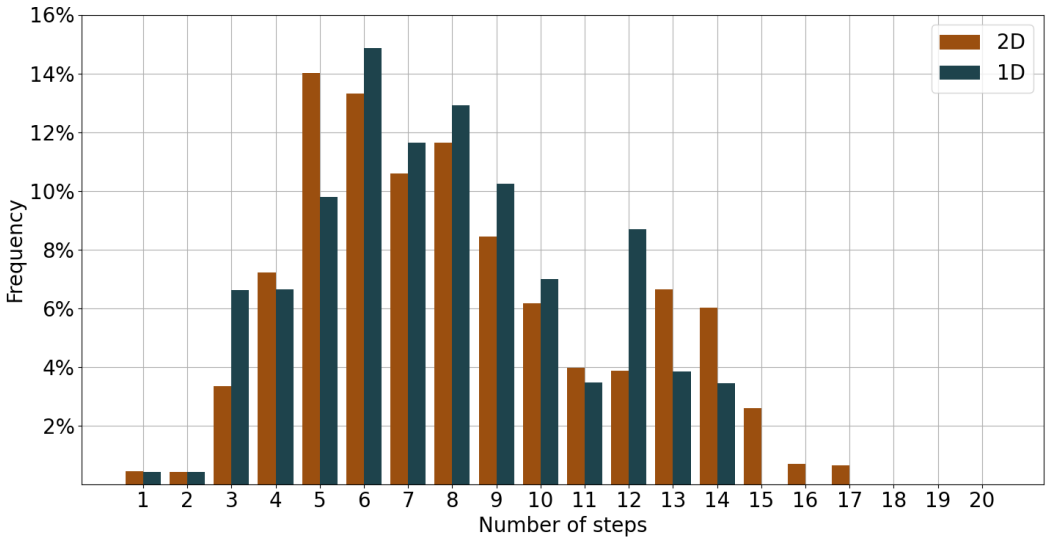

Figure 3.3: Lépésszámok összehasonlítása 1 állomáson, 2 ágens tanítás esetén

Elmondható, hogy az egydimenziós reprezentáció általában valamivel kevesebb lépésből oldja meg a feladatot, de igazán jelentős különbség nem tapasztalható. Szintén igaz viszont, hogy az alacsony lépésszám jellemző. Az ágensek szinte minden esetben megoldják a feladatot 15 lépésen belül (1D esetén mindig), és az esetek nagy részében 10 lépésen belül. Ez szinkronban van az egyes esetek szúrópróba szerű tanulmányozásával, mely során azt tapasztaltuk, hogy az ágensek nagyon ritkán várakoznak fölösleges, és akkor is maximum 1-2 lépést. Mikor egy ágens megáll, az szinte kivétel nélkül azért történik, hogy elkerülje a deadlock-ot, egy másik ágens elengedése útján. Ez két állomás esetén is így történik, azon esetekben, ahol az ágensek végül találnak megoldást. Az természetes nem elvárható, hogy 3-4 lépésből megoldódjon minden szituáció. Ez csak bizonyos kezdő pozíciók esetén működik. Viszont, ha az egyik ágens az állomás egyik oldaláról indul, és a másikba kell eljutni, úgy hogy a másik ágenst kikerüli, tehát a hosszabb utat választja, az a legrosszabb esetben 14 lépést vesz igénybe állandó haladást feltételezve, és mint azt megállapítottuk, ezt az ágensek az esetek kevesebb mint 4%-ában lépik túl az egyik, 0%-ában a másik esetben. Szintén a lépésszámot, csak a két állomáson sikeresen megoldott esetek lépésszámát ábrázolja a 3.4 ábra.

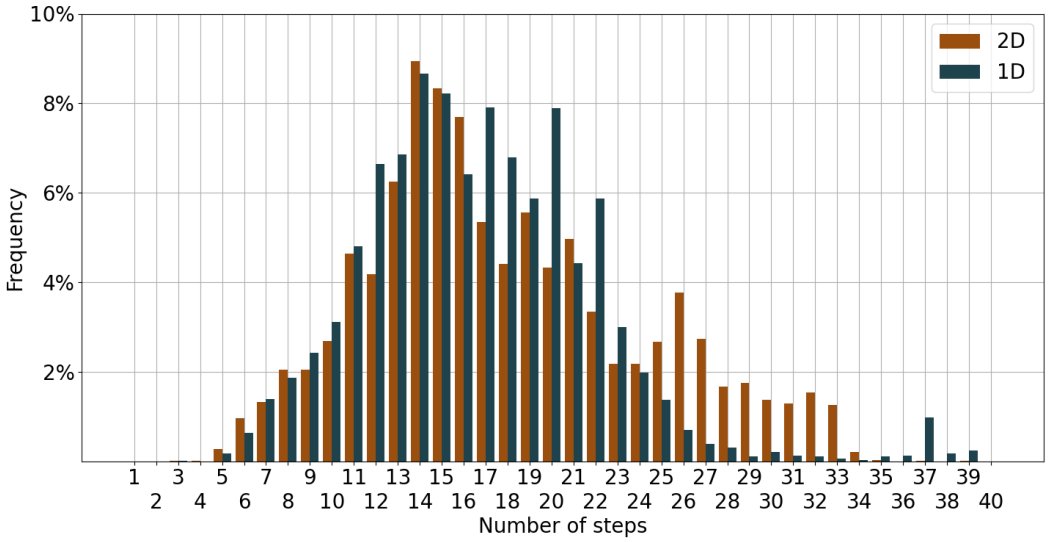

Figure 3.4: Lépésszámok összehasonlítása 2 állomáson, 2 ágens tanítás esetén

Az ebben az esetben is, az egydimenziós reprezentáció úgy tűnik jobban teljesít, de nem kiemelkedő módon. Mivel összesen 4 ágens van játékban és a két állomás kapcsolódási pontja egyfajta bottleneck, több várakozás várható. Ennek fényében minden esetben 40 alatti, és szinte minden esetben 30 alatti lépésszám jó eredménynek tűnik.

3.2 Az állapottér bővítése

Ahogy az az előző bekezdésben említésre került, mivel a környezet úgy lett kialakítva, hogy minél jobban támogassa a generalizációt, 70%-nál magasabb arányú sikeres teljesítést szeretnénk elérni. A sikertelenül végződő epizódokat tanulmányozva fontos problémának tűnik, hogy habár a kezdetben minden állomáson két ágens van, ez a szám könnyen megnőhet az epizód során. Ez azért problémás, mert előidéz számos olyan helyzetet melyben az állapottér olyan mintázatokat vesz fel, melyre az ágensek, melyek csak egy másik ágenst láttak magukon kívül azelőtt, nincsenek megfelelően felkészülve. Ilyen helyzetben is van, hogy találnak megfelelő megoldást, de jelentősen megnő a kudarc esélye. Előfordulhat például, hogy olyan viselkedést kéne tanúsítaniuk, melyre tanítás során sosem volt szükség. Egy ilyen helyzetet mutat például a 3.5 ábra.

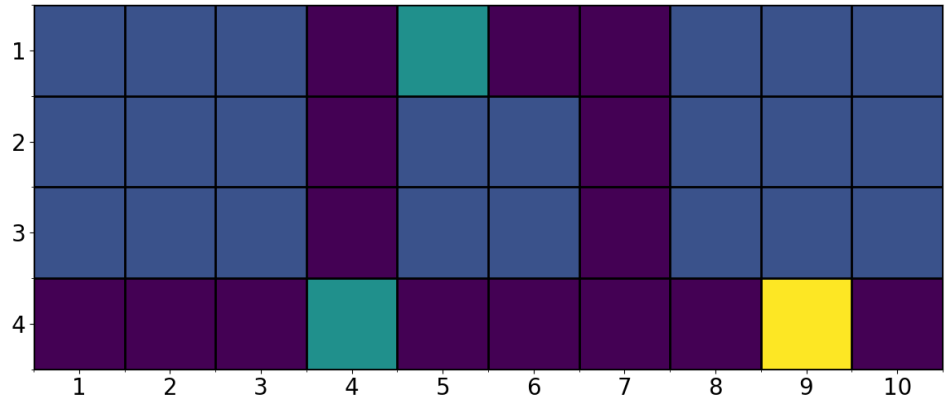

Figure 3.5: Példa a két ágensel való tanítás alatt nem megjelenő konfliktushelyzetre

Ebben a helyzetben a bal alsó, pozitív irányba haladó ágens találkozik eddig nem látott problémával. Mikor két ágens van egy állomás két átellenes mezőjén, az egyikük választ egy útvonalat, a másik dolga szimplán az, hogy a másik útvonalat válassza és ezáltal elkerüljék egymást. Ebben a konkrét helyzetben viszont a bal alsó ágensnek arra kéne rájönnie, hogy a helyes lépés a várakozás, mivel az egyik útvonal már blokkolva van, így a jobb alsó, negatív irányba háló ágensnek muszáj az alsó utat választani. Amennyiben mindkét útvonalat elállják pozitív irányba haladó ágensek, biztos deadlockot eredményez, ha van a másik irányba haladó ágens az állomás pozitív végpozíciójánál. Csak két ágens esetén viszont bármikor elfoglalhatják mindkét útvonalat. Ez csak egy példa, de számos ehhez hasonló szituáció van, melyre a tanítás alatt látott állapottér mintázatok nem készítik fel az ágenseket. Éppen ezért, következő lépésként kipróbáltuk kibővíteni az állapotteret azzal, hogy ágens segítségével tanítjuk be a hálót, majd megnézzük hogyan képes megoldani ugyan azt a problémát, tehát hogy 2 állomáson 2-2 ágens kerül elhelyezésre. Természetesen ismét mindkét reprezentációt felhasználva. Fontos különbség azonban a két tanítás között, hogy míg az első esetben a nincs szükség a közös cél és az egyes ágensek célja közti distinkcióra, jelen esetben előfordulhat hogy két ágens deadlock-ra fut, így a közös cél már nem teljesülhet, de a harmadik képes kikerülni őket, és saját célját véghezvinni a célpozíció elérvén. Ez szükségessé tette az új jutalmazási stratégia bevezetését, mely a 2.2.4 fejezet második részében olvasható. Az ágensek egyszerre tartják szem előtt a közös célt és a saját

céljukat.

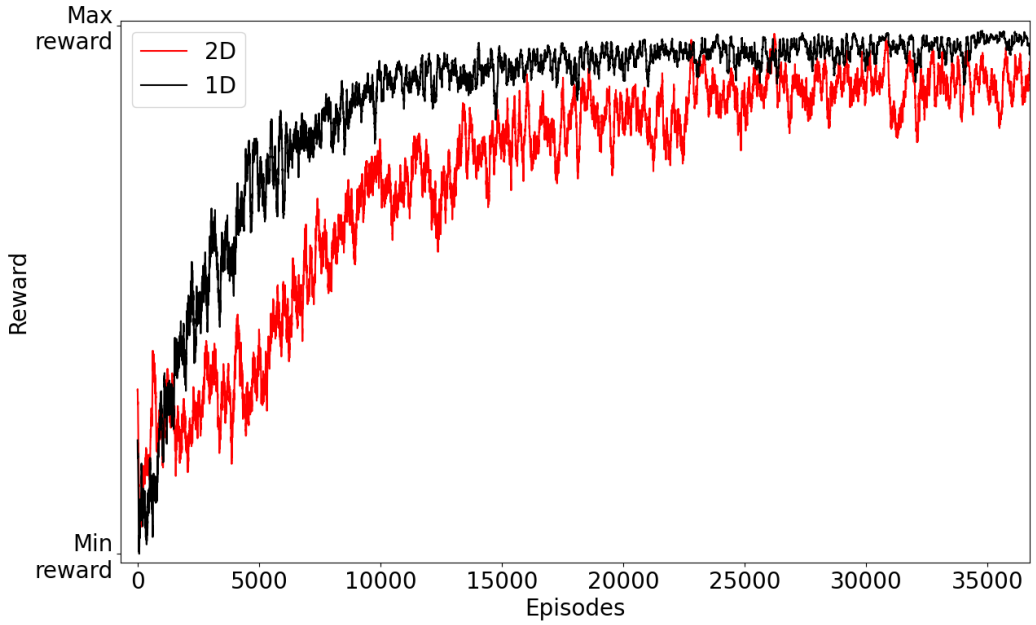

Figure 3.6: A két reprezentáció konvergenciája, 3 ágens tanítás esetén

Ismét gyorsabb konvergenciát tapasztalunk az egydimenziós reprezentációs használatánál, viszont az előző esettől eltérően, itt a végső szint sem pontosan egyezik. Az egydimenziós reprezentáció láthatóan felülmúlja a kétdimenziós változatot, bár nem sokkal. Érdekes azonban, hogy teljesen 100% sikeres teljesítést egyik tanítás sem ért el 1 állomás és 3 ágens esetén. Erről a későbbiekben lesz szó.

Sikerráta			
Reprezentáció	3 ágens, 1 állomás	2 ágens, 1 állomás	2 ágens, 2 állomás
2D	94.87%	98.61%	85.56%
1D	99.18%	99.87%	86.44%

Table 3.2: Sikeres teljesítés aránya 3 ágenssel való tanítás után

Egyértelműen látszik, hogy három ágens használata komoly javulást eredményez. A kétdimenziós reprezentáció esetén több mint 22%-al, egydimenziós esetben 16%-al javult a sikeres teljesítés aránya. Ez mindenképpen azt jelenti, hogy az állapottér ilyen fajta bővítése célravezető. Érdekességképpen szerepel a sikerráta 1 állomás és 2 ágens esetén. Látszik, hogy az egyszerűbb problémát nagyon magas hatékonysággal teljesíti. A 85% feletti siker már mindenképpen magas generalizálási potenciált jelez. A 3.7 ábrán ismét a lépésszámok összehasonlítása látható, 3 ágens és egy állomás esetén.

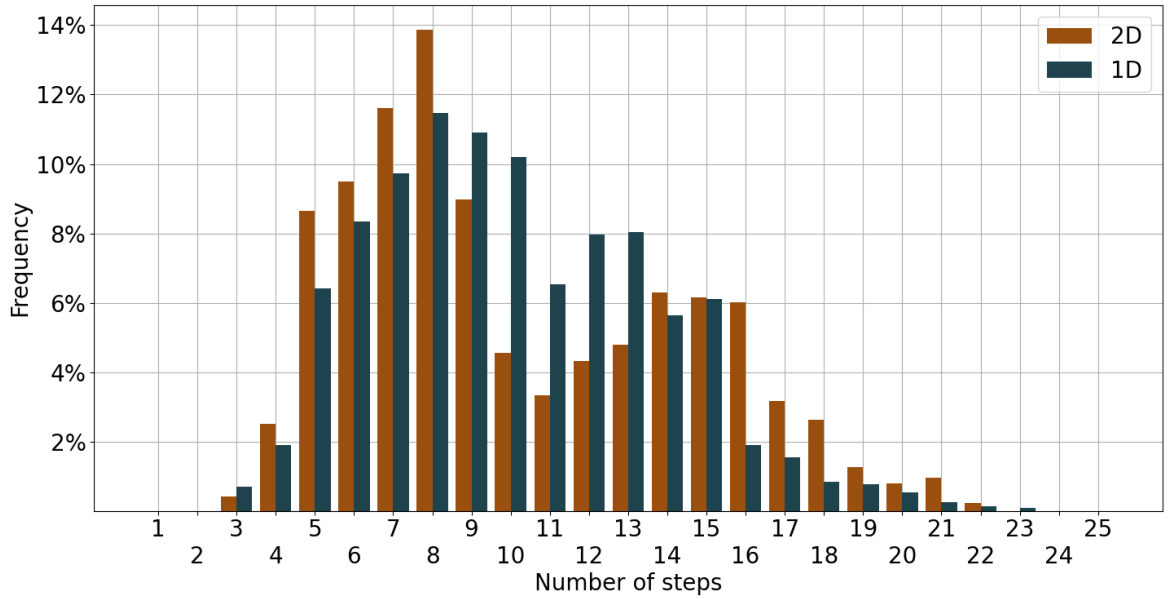

Figure 3.7: Lépésszámok összehasonlítása 1 állomáson, 3 ágens tanítás esetén

Érdekes módon lépésszámok mindkét esetben kettéválnak, és két haranggörbe látszik kirajzolódni. Ennek a magyarázata az lehet, hogy egyszerűbb mintázatok esetén akár megszakítás és várakozás nélkül haladhat végig minden ágens, olyan esetekben viszont amikor deadloc-ban végződhetne a helyzet, van hogy 1-1 ágens huzamosabb ideig kell hogy várakozzon, és ez hirtelen nagymértékben megnöveli a lépésszámot. Az is látszik, hogy a kétdimenziós reprezentáció valamivel végletesebb. Vagy kifejezetten hamar végez, vagy elnyújtja a probléma megoldását. Az egydimenziós változat eloszlása jóval egyenletesebb. Fegyelembe véve, hogy az előző esetben a lépésszám 15-20 lépés alá korlátozódott, egyel több ágens esetén a 20-25 lépés mint felső határ kifejezetten reálisnak tűnik. Mivel itt lesznek olyan helyzetek, ahol mindenképpen várakozásra lesz szükség, kijelenthető, hogy a minimális lépésszám adott helyzethez most is teljesül és a jutalmazási stratégia, illetve az az állapottér változása nem volt rossz hatással ezen célunk elérésére. A két állomáson elért eredményeket szemlélteti a 3.8 ábra. Itt is megfigyelhető (csak enyhén) a két haranggörbe, valamint az egydimenziós reprezentáció laposabb alakja. Az eddigi 35-40-es lépésszám azonban csak kicsit tolódik fel, olyan 37-42 köré. Mivel itt sem az állomások, sem az ágensek száma nem változott ez várható volt. De vajon akkor miért nőtt meg a lépésszám egyáltalán? Erre a válasz, a megoldott problémák jellegében keresendő. Azon konfliktushelyzetek, melyeket ezen neurális hálók meg tudtak oldani, de az előzőek nem, jogosan feltételezhető, hogy nagyobb komplexitásúak, komolyabb torlódással járó problémák. Ezen helyzetek deadlockot nélkülöző megoldása nagyobb lépésszámot igényel, hiszen a legtöbb esetben lesz olyan ágens melynek várakozni kell. Alapvetőn valószínűbb ilyen probléma felbukkanása,

amennyiben az ágenseknek nagyobb utat kell bejárni (tehát a célpozíciójuktól távolabb indulnak), hiszen ilyenkor több potenciális konfliktushelyzet adódik a többi ágenssel.

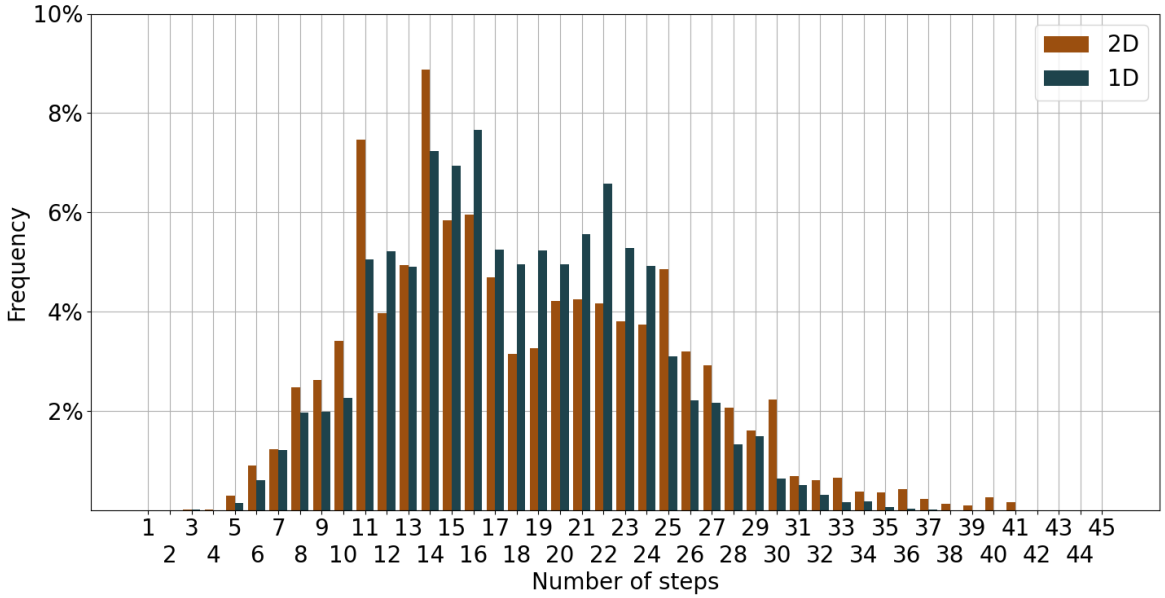

Figure 3.8: Lépésszámok összehasonlítása 2 állomáson, 3 ágens tanítás esetén

Nehéz volna megállapítani, hogy melyik reprezentáció teljesített jobban lépésszám szempontjából. Kétdimenziós esetben az átlagos lépésszám 18.23, egydimenziós esetben 17.87. A sikeres megoldások halmazán nézve hasonló hatékonysággal dolgozik a két reprezentáció.

3.3 Az állapottér diverzifikálása

Mivel az előzőekből látszik, hogy az állapottér bővítések sikeresebb végrehajtáshoz vezet, következőnek mégtovább bővítettük. A következő tanítás továbbra is egy állomáson történik. Annak érdekében, hogy a hálózat minél általánosabb módon tanulja meg a közlekedés szabályait, a következőkben az ágensek számát is véletlenszerűen határoztuk meg. Minden epizód elején egyenlő eséllyel egy, kettő, három vagy négy ágens kerül inicializálásra, majd az eddigiekhez hasonlóan mindegyik pozíciója szintén véletlenül kerül meghatározásra. Így a tanítás során előforduló állapotok jóval sokszínűbbek is mint ezelőtt. A különböző reprezentációs konvergenciája jelen tanítás során a 3.9 ábrán látható.

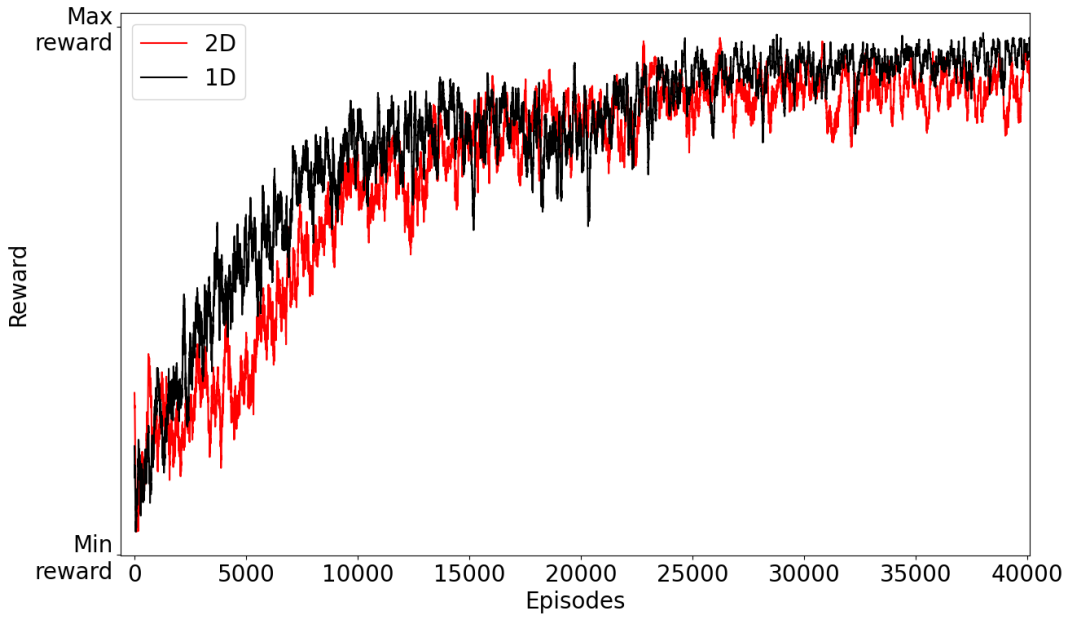

Figure 3.9: Konvergencia váltózó számú ágensnél

Ismét az egydimenziós reprezentáció konvergál hamarabb, habár kisebb különbséggel, mint ezelőtt. Mindkét esetben látszik egy beesés, egy enyhe fluktuáció, ami azért lehet, mert a véletlenszámú ágens, habár sokszínűbbé teszi az állapotteret, komolyabb kihívás elé állítja a neurális hálót. Végül ismét eléri a közel maximális értéket mindkét reprezentáció. Habár a konvergencia lassabbnak tűnik, a sikeres teljesítés aránya ismét látványosan nőtt, ahogy az a 3.3 táblázatból látszik. Egyfelől három ágens egy állomáson nagyjából ugyan olyan jól teljesít, mint amikor kifejezetten erre lett tanítva a háló, két állomás esetén sikerült elérni a 90% körüli, egydimenziós reprezentáció esetén a 90% fölötti értéket, amely mindenképpen erős generalizációs képességet jelez.

Sikerráta		
Reprezentáció	3 ágens, 1 állomás	2 ágens, 2 állomás
2D	97.33%	89.23%
1D	98.82%	92.93%

Table 3.3: Sikeres teljesítés aránya változó számú ágenssel való tanítás után

Az előző tanítással való összehasonlításhoz ábrázoltuk a lépésszámok arányát úgy, hogy a kiértékelés során három ágenst helyeztünk egy 1 állomáson, akárcsak az előző esetben. (3.10)

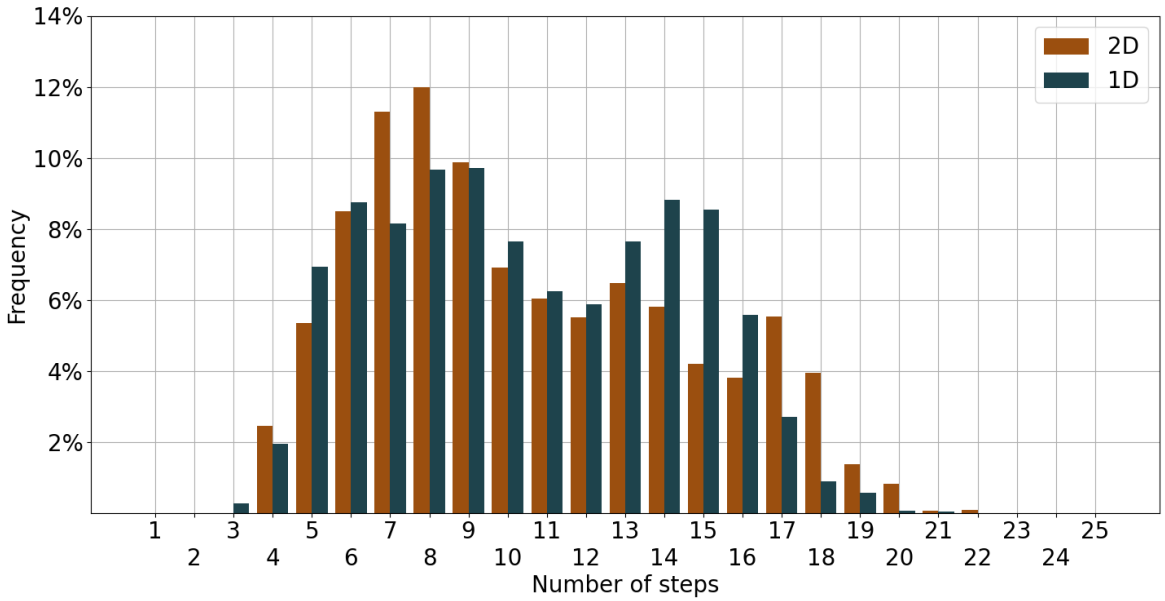

Figure 3.10: Lépésszámok összehasonlyítása 1 állomáson 3 ágenssel, változó ágens-számú tanítás esetén

A két reprezentáció szinte ugyan olyan hatékonysággal oldja meg a feladatot, és az előző tanításhoz képesti eltérés is elhanyagolható. Ugyan úgy 20-25 lépés a maximum, ugyan úgy 10 körüli az átlagos lépésszám. Egy állomáson 3 ágens tehát nagyjából megegyező lépésszám mellett képes elérni a célt.

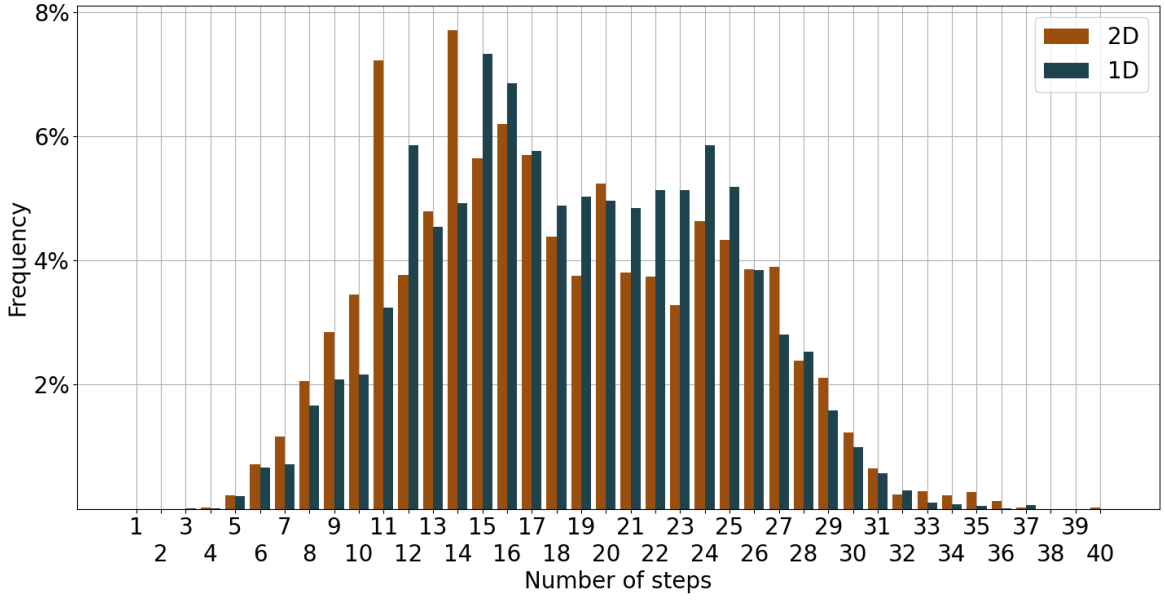

Figure 3.11: Lépésszámok összehasonlítása 2 állomáson, változó ágens-számú tanítás esetén

Az érdemi próba ebben az esetben is két állomás két-két ágenssel. Azt már láthat-

tuk, hogy ez a háló nagyobb arányban oldja meg sikeresen a feladatot, mint az eddigiek. A lépésszámok esetén az eltérés ismét nem olyan jelentős. A 3.8 ábrához képest csökkent a lépésszám, azonban kis mértékben. Ezt az okozhatja, hogy komplexebb helyzeteket is meg tud oldani a hálózat különösebb várakozás nélkül. Összességében tehát elmondható, hogy nagyob arányban, és legalább ugyan olyan hatékonysággal abszolválja a feladatot a véletlen számú ágens segítségével tanított háló. A két állomáson történő kiértékelés eredményeit foglalja össze a 3.3 táblázat.

Sikerráta			
Reprezentáció	2 ágens tanítás	3 ágens tanítás	Véletlen számú ágens tanítás
2D	63.42%	85.56%	89.23%
1D	70.26%	86.44%	92.93%

Table 3.4: Különböző tanítások összehasonlítása

A kezdeti próbálkozásokhoz képest komoly javulást tudtunk elérni. A végső eredmények egyértelmű magas generalizálási potenciált mutatnak. Természetesen lehetne az állapotteret még tovább bővíteni, még több ágens felhasználásával tanítani, és a jövőben ezt ki is fogjuk próbálni, azonban az ezen dolgozatban bemutatott munka lényege a virtuális ágens koncepció próbára tétele, és az ezáltal nyert generalizálási potenciál feltérképezése volt, melyet sikeresen megtettünk. Még mielőtt elkezdjük ezen módszer robusztusabb módon alkalmazni, érdemes elemezni, hogy a jelenlegi hálózaton lehetséges-e még jobb eredmények elérése. Ezzel foglalkozik a következő (3.4) fejezet. Az egyes epizódok lefolyásának tanulmányozása statisztikai jelentőséggel nem bír, ugyan akkor érdekes lehet megfigyelni, hogyan oldja meg a problémát a háló. Mennyi holtidő van fölösleges várakozás miatt, mennyire sikerült bonyolultabb viselkedésformákat elsajátítani. Tapasztalataink szerint, fölösleges várakozás nagyon ritka esetben fordul elő. Az ágensek jellemzően a legrövidebb utat használják.

3.4 További fejlesztési irány

Az utolsó sikeres tanítás során az állapottér bővítését és diverzifikálását eléggé hatásos módon sikerült megvalósítani, ahogy az az eredményeken is látszik. Ezután érdemes átgondolni, mit lehet tenni ezen kívül annak érdekében, hogy még jobb sikerrátát érjünk el. Az állapot reprezentáció egyik fontos eleme, hogy minden ágens a környezet részeként kezeli a többi ágenst, mivel ez segíti a generalizálást. Ennek megfelelően, úgy alakítottuk ki a környezetet, hogy az ágensek nem tudnak egymásról, kommunikáció nem történik. Érdekes kérdés volt, hogy ennek ellenére milyen hatékonyan oldják meg a feladatot. A kommunikáció hiánya azonban okozhat problémát. Az eredeti 2 ágens

segitségével tanított hálózat, ugyan azon problémát mellyen tanult, hiba nélkül meg tud oldani. Kipróbáltuk tehát, hogy ezen hálózat milyen sikeresen oldja meg, ha két állomás, de állomásonként csak egy ágens kerül elhelyezésre. Ez nagyjából az esetek 94%-ában végződik sikerrel. Ez azért különös, mert még mindig maximum két ágens lehet egy állomáson, és ezt pedig 100%-osan meg tudja oldani. A probléma, a már említett kommunikáció. A csatlakozási pontnál a virtuális ágens biztosítja a deadlock elkerülését. Megeshet azonban, hogy minkét ágens egyszerre lép olyan mezőre, deadlockot még nem okoz, viszont a következő lépésben elzárják a közlekedés útját. Itt mindketten megkapják az új állapotot, melyben nincs virtuális ágens, mert a másik oldalon sincs elzárva az út, egyelőre. A következő lépésben pedig mindketten egyszerre belépnek arra a mezőre, ahol már muszáj végig haladniuk a csatlakozási ponton, mivel elállják az utat. Így deadlock helyzet jön létre, pedig mindkét ágens a megfelelő lépést hajtotta végre a rendelkezésre álló adatok ismeretében. Ezt szemlélteti a 3.12 ábra.

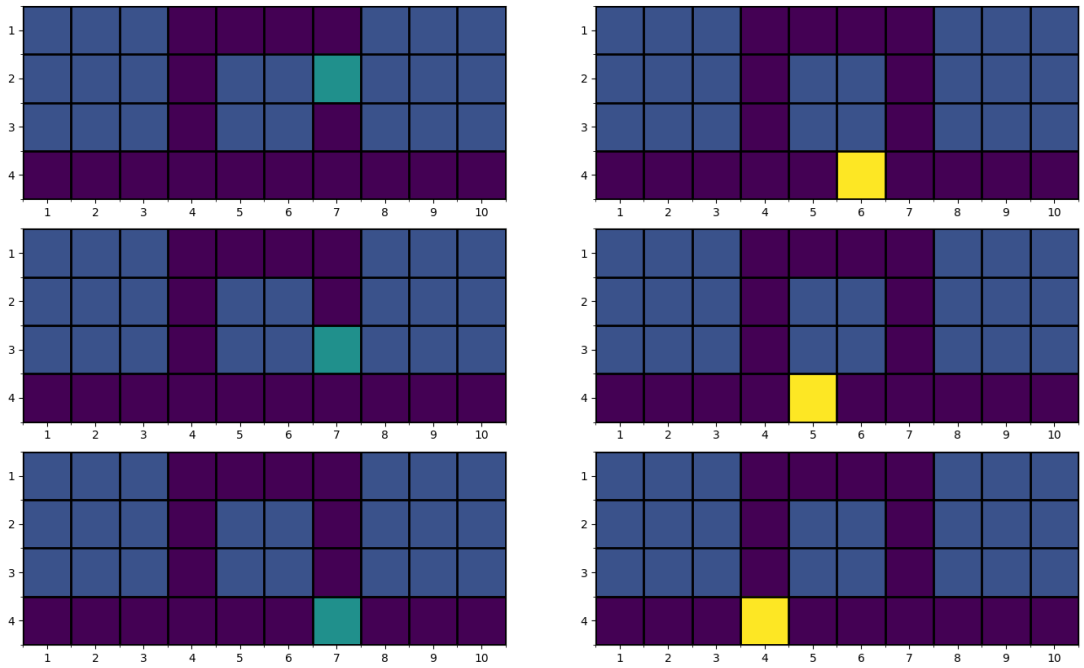

Figure 3.12: Kommunikációhiány miatti konfliktushelyzet

A felső két kép az összecsatolt állomások kezdeti állapota. A következő lépésben mindkét ágens halad a saját célja felé, és rálép az utolsó mezőre, ahol még el tud engedni másik szerelvényt, ha szükséges. Mivel egyik sem tudja, hogy a másik is belépni készül a két állomás közti átmeneti szakaszra, egyszerre lépnek be. Ez deadlock helyzetet eredményez. Ez egy jó példa arra, mikor mindkét ágens a megfelelő lépést teszi (a tanítás alapján) mégsem zárul sikerrel az epizód. A virtuális ágens koncepció egyik lényege, hogy egyik ágensnek sem kell tudni a többi állomásról csak a sajátjáról. A továbbiak-

ban tehát szeretnénk kidolgozni egy olyan kommunikációs módszert, ami összhangban van az eddigi irányelvekkel, mivel az kevésbé költséges, mintha minden ágens saját reprezentációjában szerepel minden állomás. Mivel a sikeres végrehajtások száma 94% környékén mozog, a változó számú tanítás esetén elért sikeres teljesítések száma arra utal, hogy a kommunikáció lehet a kulcs a további jelentősebb javuláshoz. Az állomások számának növelésével az olyan helyzetek száma is nő, ahol az egyszerre tett lépés problémát okozhat, mivel nő az olyan szakaszok száma, melyen két ellenkező irányba közlekedő ágens dedlock helyzetet eredményez. Így a kommunikáció a következő lépés a generalizáció javításában.

Chapter 4

Konklúzió

A dolgozatban bemutattunk egy mély megerősítéses tanulás alapú módszert a vasúti átütemezési probléma megoldására. Két különböző állapotreprezentációt használunk. Az egyik egy dimenzióban tartalmazza a döntési pontokat, a másik két dimenzióban képszerűen tárolja a hálózatot és az ágens pozíciókat. Megmutattuk az első próbálkozást, valamint a gondolati ívet mely mentén a koncepciónkat fejlesztettük. Megmutattuk hogyan segít a jobb generalizáló képesség elérésében az állapottér bővítése és sokszínűvé tétele, valamint a további fejlesztési lehetőségeket. A dolgozat legfőbb kontribúciója, a virtuális ágensek használata, melyek segítségével az ágenseknek elég a saját állomásukat ismerni, és nem kell a teljes környezet minden részletét látniuk. A dolgozat tekinthető egy sikeres proof of concept tanulmánynak, mivel A virtuális ágensek segítségével a probléma nagyon magas arányban jól megoldható. Mindemellett a sikeres megoldások esetén minimális vagy közel minimális lépésszámot érnek el a neurális hálók. Általánosságban elmondható, hogy az egydimenziós reprezentáció valamelyest magasabb sikerrátával és közel azonos hatékonysággal abszolválja az átütemezési problémát.

List of Figures

List of Tables

Bibliography

[1] Sachin Agrawal. Online) 2581-9429 issn (print) 2581-xxxx. *International Journal of Advanced Research in Science, Communication and Technology*, 2021, Article.

[2] L. Botte, M.and D'Acierno. Dispatching and rescheduling tasks and their interactions with travel demand and the energy domain: Models and algorithms. *Urban Rail Transit*, 4:163–197, 2018, Article.

[3] Marilisa Botte, Claudia Salvo, Antonio Placido, Bruno Montella, and Luca D'Acierno. Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. *International Journal of Transport Development and Integration*, 1:63–73, 2017, Article.

[4] European Commission. Sustainable and smart mobility strategy—putting european transport on track for the future. 2021. https://transport.ec.europa.eu/system/files/2021-04/2021-mobility-strategy-and-action-plan.pdf.

[5] Francesco Corman, Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, 44:75–192, 2010, Article.

[6] Andrea D'Ariano, Francesco Corman, Dario Pacciarelli, and Marco Pranzo. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, 42:405–419, 2008, Article.

[7] Nahid Parvez Farazi, Bo Zou, Tanvir Ahamed, and Limon Barua. Deep reinforcement learning in transportation research: A review. *Transportation Research Interdisciplinary Perspectives*, 11,(100425), 2021, Article.

[8] L. Lindenmaier, I. F. Lövétei, G. Lukács, and S. Aradi. Infrastructure modeling and optimization to solve real-time railway traffic management problems. *Periodica Polytechnica Transportation Engineering*, 49(3):270–282, 2021, Article.

[9] I. F. Lövétei, B. Kővári, and T. Bécsi. Mcts based approach for solving real-time railway rescheduling problem. *Periodica Polytechnica Transportation Engineering*, 49(3):283–291, 2021, Article.

[10] István Lövétei, Bálint Kővári, Tamás Bécsi, and Szilárd Aradi. Environment representations of railway infrastructure for reinforcement learning-based traffic control. *Applied Sciences*, 12, 2022, Article.

[11] Valerio Martinis and Francesco Corman. Data-driven perspectives for energy efficient operations in railway systems: Current practices and future opportunities. *Transportation Research Part C Emerging Technologies*, 95:679–697, 2018.

[12] V. Mnih, K. Kavukcuoglu, and D. et al. Silver. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015, Article.

[13] Sharada Mohanty, Erik Nygren, Florian Laurent, Manuel Schneider, Christian Scheller, Nilabha Bhattacharya, Jeremy Watson, Adrian Egli, Christian Eichenberger, Christian Baumberger, Gereon Vienken, Irene Sturm, Guillaume Sartoretti, and Giacomo Spigler. Flatland-rl : Multi-agent reinforcement learning on trains. *arXiv:2012.05893*, 2020, Article.

[14] Mitsuaki Obara, Takehiro Kashiyama, and Y. Sekimoto. Deep reinforcement learning approach for train rescheduling utilizing graph theory. *IEEE International Conference on Big Data*, 4525-4533, 2018, Article.

[15] Paola Pellegrini, Grégory Marlière, and Joaquin Rodriguez. A detailed analysis of the actual impact of real-time railway traffic management optimization. *Journal of Rail Transport Planning and Management*, 6:13–31, 2016, Article.

[16] Y. Tan, W. Xu, Z. Jiang, Z. Wang, and B. Sun. Inserting extra train services on high-speed railway. *Periodica Polytechnica Transportation Engineering*, 49(1):16–24, 2021, Article.