UNIVERITATEA BABEŞ-BOLYAI, CLUJ-NAPOCA, ROMÂNIA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

Metode de învățare prin întărire profundă pentru căutare și optimizare în sisteme multi-agent

Rezumat tezei de doctorat

Student:
Anikó Kopacz

Conducător de doctorat:
Prof. Dr. Camelia Chira

2026

Abstract

Rețelele sunt potrivite pentru modelarea sistemelor din aplicații din viața reală ce prezintă relații non-triviale, cum ar fi rețelele sociale, rețelele de citare și sistemele de transport. Mai multe probleme de optimizare, în care nodurile rețelelor analizate au capacitatea de a interacționa unele cu altele sau cu mediul lor, pot fi modelate ca medii multi-agent. Datorită complexității unor astfel de sisteme, apare interesul pentru soluții flexibile și robuste. Astfel, abordările bazate pe învățarea automată – cum ar fi învățarea prin întărire – sunt adesea prioritizate pentru a aborda problemele de optimizare în sisteme multi-agent.

În această teză sunt propuși algoritmi bazați pe învățarea profundă prin întărire, iar eficacitatea lor este explorată în contextul diverselor scenarii multi-agent. În primul rând, sunt explorate scenarii competitiv-cooperative care simulează interacțiuni de tip prădător-pradă. Sunt dezvoltate modele centralizate și descentralizate de învățare profundă prin întărire pentru a naviga în medii cooperative-competitive.

În al doilea rând, sunt abordate variante ale problemei de rutare a vehiculelor: problema reprogramării vehiculelor și problema planificării traiectoriei multi-robot. Se propune o abordare hibridă de tip greedy pentru planificarea traseelor pentru mai mulți roboți, depășind alte soluții euristice și meta-euristice. Apoi, un model deep Q-learning antrenat prin învățare independentă este dezvoltat pentru a aborda planificarea traseelor pentru mai mulți roboți folosind învățarea prin întărire. Arhitectura deep Q-learning este combinată cu învățarea prin imitațiepornind de la demonstrații generate de o metodă hibridă de tip greedy.

În ultimul rând, maximizarea influenței în rețelele sociale este abordată folosind învățarea profundă prin întărire. Este prezentat un sistem de învățare prin întărire care abordează maximizarea influenței pe rețele polaritate, cu doi actori prezenți într-o rețea socială. Pentru a aborda această problemă, se dezvoltă un model multi-agent de deep Q-learning. Acest model aplică un pas de detectarea comunităților aplicând algoritmul Louvain pentru a descompune rețeaua socială și a optimiza selecția nodurilor-seed în cadrul comunităților prin utilizarea arhitecturii de tip double deep Q-network.

Contents

Chapter 1

Introducere

Mai multe probleme complexe de optimizare pot fi formalizate folosind sisteme multi-agent [9] și rezolvate utilizând una dintre următoarele strategii (i) descompunerea problemei originale într-un set de sarcini independente, mai ușoare din punct de vedere computațional, (ii) rezolvarea separată a sarcinilor, (iii) combinarea soluțiilor obținute pentru sarcini într-o decizie globală. Rezolvarea sarcinilor mai puțin complexe din punct de vedere computațional devine responsabilitatea actorilor autonomi numiți *agenți*. În problemele de optimizare din lumea reală, există mai multe tipuri de dinamică a agenților. Mediile cooperative presupun că un set de actori sunt așteptați să colaboreze pentru a urmări un obiectiv comun. Pe de altă parte, configurațiile de agenți cu cel puțin două obiective opuse sunt numite scenarii competitive. Diverse metode computaționale se bazează pe sisteme multi-agent și utilizează interacțiunile agenților pentru optimizare.

Învățarea prin întărire (eng. Reinforcement Learning - RL) este o abordare computațională binecunoscută pentru a învăța o sarcină specifică bazată pe interacțiunea unuia sau mai multor agenți cu mediul [90]. Îndeplinirea sarcinii predefinite este asociată cu o recompensă pozitivă, în timp ce scenariile nefavorabile sunt caracterizate de o recompensă negativă sau penalizare. Scopul agentului care învață este de a construi politici care asigură obținerea unei recompense cumulate ridicate. Învățarea prin întărire a fost aplicată la mai multe probleme de optimizare inspirate din lumea reală, cum ar fi robotica, epidemiologia și altele. De asemenea, învățarea prin întărire multi-agent a fost adaptată pentru a aborda diverse probleme din lumea reală, cum ar fi controlul traficului de urgență [20], planificarea jgheaburilor în depozite [63], navigarea [77] și diverse scenarii de jocuri cooperative [55, 95]. Diverse abordări de analiză a datelor

pot fi adaptate pentru a opera pe sisteme multi-agent, cu toate acestea, este crucial de a lua în considerarea compromisurile computaționale. Introducerea mai multor actori într-un sistem non-trivial poate impacta semnificativ resurse computaționale, și unele metode devin inaplicabile în practică. Astfel, abordările euristice ar putea fi favorizate în detrimentul soluțiilor exacte în cazul problemelor NP-hard pe sisteme multi-agent, de ex., [25, 94, 99].

1.1 Obiectivele cercetării

Obiectivul principal al cercetării este abordarea problemelor de optimizare care combină sistemele multi-agent și rețelele cu învățare profundă prin întărire. Pentru a atinge obiectivul principal de cercetare, sunt introduse mai multe obiective care descompun scopul general în sarcini specifice, ghidând dezvoltarea metodologică, evaluarea experimentală și validarea abordărilor propuse.

Primul obiectiv al acestei teze, abordat în Capitolul 3, este dezvoltarea și evaluarea unor cadre de învățare prin întărire multi-agent [2] capabile să navigheze în medii cooperative-competitive. S-a urmărit investigarea eficacității metodelor centralizate și descentralizate de învățare profundă prin întărire. Pentru a atinge acest obiectiv, un scop științific a fost realizarea unei analize adecvate a literaturii care rezumă abordările de învățare prin întărire potrivite pentru problemele de optimizare în sisteme multi-agent. Pe baza analizei literaturii, focusul principal a fost dezvoltarea și adaptarea metodelor de ultimă generație la sarcini specifice de optimizare. Performanța învățării profunde prin întărire centralizată și descentralizată a fost comparată utilizând platforme de referință binecunoscute care simulează dinamica prădător-pradă. Explorarea variațională multi-agent [67] a fost adaptată cu scopul de a îmbunătăți performanța antrenării centralizat pentru învățarea profundă prin întărire.

Al doilea obiectiv al acestei teze vizează planificarea rutelor multi-agent cu abordări de învățare profundă prin întărire, care reprezintă subiectul Capitolului 4. Inițial, problema reprogramării vehiculelor a fost investigată cu scopul de a propune și evalua mecanisme de extragere a caracteristicilor specifice domeniului. Experiențele computaționale au fost concepute pentru a ilustra faptul că un algoritm de planificare poate fi învățat cu deep Q-networks pe baza spațiului de caracteristici propus. Pentru a sublinia natura robustă a metodelor de învățare profundă prin întărire, a fost analizată o altă variantă a problemei planificării rutelor, plani-

ficarea traseelor în sisteme cu mai mulți roboți. Cu scopul de a aborda planificarea traseelor roboților, au fost dezvoltate și testate sisteme de optimizare hibride și bazate pe învățarea profundă prin întărire. Apoi, construind pe abordarea hibridă propusă, a fost creată un expert pentru antrenarea unei politici descentralizate de învățare prin întărire folosind învățarea prin imitație.

Cel de-al treilea obiectiv al prezentei teze este abordarea problemei maximizării influenței în rețele sociale cu semne, folosind metode de învățare profundă prin întărire. Acest obiectiv reprezintă subiectul Capitolului 5. Astfel, următorul scop științific al acestei teze a fost dezvoltarea unei abordări de învățare profundă prin întărire pentru problema maximizării influenței competitive în rețele sociale cu semn. Performanța procesului propus de selecție a nodurilor sursă (eng. seeds) a fost evaluată pe rețele sociale de dimensiuni mici și medii cu relații de încredere-neîncredere. Sistemul de învățare prin întărire propus a fost îmbunătățit ulterior, fiind introdusă o arhitectură multi-agent de învățare prin întărire bazată pe comunități pentru a aborda maximizarea influenței în funcție de polaritate în rețelele sociale cu semne. Au fost concepute experimente pentru a evalua învățarea prin întărire multi-agent bazată pe comunități pe mai multe rețele sociale la scară largă, care au relații pozitive și negative.

1.2 Contribuții originale

Prezenta teză propune mai multe contribuții originale enumerate mai jos, împreună cu publicațiile relevante pe acest subiect.

1. *Standardizarea extragerii caracteristicilor pe baza conflictelor din planificarea perechilor de trenuri.* Este introdus un sistem de învățare prin întărire pentru problema reprogramării trenurilor, care aplică o selecție standardizată a caracteristicilor bazată pe conflicte între perechi de trenuri. Pe baza unei soluții analitice care detectează și rezolvă fiecare conflict apărut între două trenuri, este propus un spațiu de observație corespunzător. Datele obținute din conflictele se traduc în acțiuni în contextul modelului de învățare prin întărire. Performanța mecanismului propus de extragere a caracteristicilor este evaluată utilizând metricile definite în Flatland Challenge [72]. Sistemul propus și experimentele computaționale sunt prezentate într-un articol de conferință [52].

2. *Învățarea profundă prin întărire în medii cooperative-competitive.* Sunt construite modele

bazate pe învățarea prin întărire multi-agent pentru optimizarea acțiunilor agenților care operează în medii cooperative-competitive multi-agent. Două abordări binecunoscute de învățare prin întărire, deep Q-networks (DQN) introduse în [71] și factorizarea monotonă a funcției de valoare pentru învățarea profundă prin întărire (QMIX) propusă în [81], sunt adaptate și evaluate pe biblioteca MAgent [112], care simulează dinamici de tip prădător-pradă. Metodologia aplicată și rezultatele computaționale ale metodelor propuse sunt prezentate în [47].

3. *Învățare profundă Q-learning centralizată și descentralizată pentru medii cooperative-competitive.* Pentru a evalua eficacitatea învățării prin întărire centralizată și descentralizată, mai multe arhitecturi bazate pe DQN sunt integrate pentru a naviga în medii cooperative-competitive: (i) învățare independentă cu DQN, (ii) QMIX, (iii) și explorarea variațională multi-agent [67]. Experimentele evidențiază faptul că antrenarea descentralizată al rețelelor Q profunde acumulează recompense mai mari de a lungul eposodului la antrenare și la evaluare, comparativ cu abordările de învățare centralizată testate. Metodologia și rezultatele sunt publicate într-un articol de jurnal [48].

4. *Algoritm hibrid adaptativ de tip greedy pentru planificarea traseelor mai multor roboți.* O abordare hibridă denumită H* este dezvoltată pe baza combinării planificării adaptative a rutelor utilizând A* și a unui algoritm de căutare locală pentru optimizarea rutele în medii de multi-robot. Algoritmul A* găsește ruta optimă pentru un robot și printr-o abordare euristică se adaptează această rută la scenariul de multi-agent. Lungimea totală a traseelor determinate și numărul de coliziuni între roboți sunt minimizate pe parcursul evaluărilor în medii specifice – o cameră cu 10x30 celule, patru camere de 10x30 celule interconectate în perechi și un depozit. Rezultatele experimentale demonstrează că metoda H* propusă depășește performanțele abordărilor euristice bazate pe A* în ceea ce privește lungimea rutelor. H* prezintă performanțe similare cu metoda coevoluționară și are performanțe mai bune în medii mici. Metoda H* și experimentele computaționale sunt publicate într-un articol de jurnal [51].

5. *Un model de învățare prin întărire pentru planificarea traseelor roboților.* Este propus un sistem multi-agent de Dueling Double DQN [37, 102] pentru planificarea traseelor roboților într-un mediu bidimensional partajat. Un controler descentralizat este imple-

mentat pentru a îmbunătăți eficiența prin partajarea experinențelor și a parametrilor între agenți. Rezultatele experimentale arată că modelul antrenat învață eficient să evite coliziunile și că alegerea unui spațiu adecvat de soluții este crucială pentru a asigura eficacitatea antrenării. Experimentele computaționale care validează sistemul de multi-robot propus au fost prezentate la școala de vară SusTrainable Summer School 2023 [44] și au fost submise spre publicare într-un volum LNCS [49].

6. *planificarea traseelor roboților prin învățare prin imitație cu Deep Q-networks.* O metodă este propusă combinând învățarea profundă prin întărire cu învățarea prin imitație de la un expert pentru problema planificării traseelor roboților. Expertulm, care oferă o referința de bază pentru modelul de învățare prin întărire se bazează pe abordarea H* [51] și pe rezolvarea analitică a conflictelor prezentată în [52]. Sistemul dezvoltat și rezultatele experimentale au fost trimise spre publicare la International Conference on Hybrid Artificial Intelligent Systems (HAIS 2026) [50].

7. *Selecția comună și iterativă a nodurilor cu deep Q-learning pentru maximizarea influenței în scenariul competitiv.* Două mecanisme distincte sunt introduse pentru a construi seturile inițiale de noduri pentru maximizarea influenței în scenariul competitiv: selecția comună și selecția iterativă. Un model de DQN este antrenat pentru a maximiza propagarea informației în rețelele sociale bazate pe încredere în prezența a doi actori competitivi. Eficacitatea seturilor de surse este analizată pe baza numărului de noduri activate după simularea cascadei independente pe rețele de încredere-neîncredere. Mecanismele propuse de selecție a nodurilor cu DQN sunt publicate în [45].

8. *Învățarea prin întărire multi-agent bazată pe comunități pentru maximizarea influenței în rețele cu semne.* Este propusă o metodă de învățare prin întărire bazată pe comunități pentru maximizarea influenței în rețele cu semne. Un agent de învățare prin întărire este atribuit fiecărei comunități identificate prin algoritmul Louvain de detectarea comunităților [7]. Selecția surselor în mediul multi-agent este optimizată prin antrenarea unui model descentralizat Double DQN [37], unde agenții antrenează aceleași rețele neuronale. Abordarea propusă și rezultatele experimentale sunt prezentate în [46].

Chapter 2

Stadiul actual al învățării prin întărire

Învățarea prin întărire (RL) utilizează experiențele anterioare ale unui sau mai multor agenți care interacționează cu mediul lor pentru a dobândi abilități sau cunoștințe dorite [90].

2.1 Concepte și definiții

În conformitate cu Sutton and Barto [90], se definește învățarea prin întărire după cum urmează. Agentul este considerat o entitate care observă factori interni și externi, mediul său, și decide care dintre acțiunile disponibile să fie executate la un moment dat. Mediul constă din elementele externe în afara agentului. Formalizarea interacțiunii agent-mediu caracteristică metodelor RL este un Proces Decizional Markov (MDP) [90]. Tuplul $\{\mathcal{S}, \mathcal{A}, p, \mathcal{R}, \gamma\}$ este un MDP dacă proprietatea Markov este adevărată, care afirmă că recompensa imediată r și starea următoare a agentului s_{i+1} sunt definite de starea anterioară s_t și acțiunea a_t întreprinsă:

$$p(s', r \mid s, a) = Pr\{s_{t+1} = s', R_{t+1} = r \mid s_t = s, a_t = a\}. \tag{2.1}$$

Politica π este probabilitatea de a selecta o acțiune a dată fiind starea s. Selectarea acțiunii următoare într-o stare s și o politică π este echivalentă cu găsirea unei acțiuni a care maximizează $\pi(a \mid s)$.

Dat fiind un pas de timp t și o stare s, $V_t^\pi(s)$ este valoarea așteptată a recompensei acumulate în starea s urmând politica π. $Q^\pi(s, a)$ denotă câștigul generat de luarea acțiunii a în starea s,

atunci când se urmează politica π.

$$Q_t^\pi(s,a) = R_{t+1} + \gamma V_{t+1}^\pi(s) \tag{2.2}$$

Avantajul $A_t(s,a)$ al acțiunii a la pasul de timp t dată fiind starea s este:

$$A_t(s,a) = R_{t+1} + \gamma V_{t+1}^\pi(s) - V_t^\pi(s), \tag{2.3}$$

astfel, valoarea avantajului poate fi calculată cunoscând valoarea stărilor s_t și s_{t+1}.

Fie $V_t(s)$ estimarea lui $V_t^\pi(s)$. Eroarea de diferență temporală (TD) este diferența dintre predicțiile consecutive [90]:

$$L^{TD} = R_{t+1} + \alpha\left(\gamma V_{t+1}(s) - V_t(s)\right), \tag{2.4}$$

unde R_{t+1} este recompensa primită la pasul de timp $t+1$, și α este un parametru care modifică dimensiunea pasului. În practică, eroarea TD este adesea optimizată în locul funcției de avantaj [90].

Învățarea prin imitație constă în învățarea executării unei sarcini din demonstrații ale experților, în loc să se bazeze exclusiv pe interacțiunile agent-mediu pentru a descoperi strategii avantajoase [110].

2.2 Deep Q-learning

Mnih et al. a introdus Deep Q-Networks (DQN) în [71], o metodă modificată de Q-learning combinată cu o arhitectură de învățare profundă pentru a determina politica optimă a unui agent de învățarea prin întărire. Acțiunea optimă la o stare dată este considerată a fi cea cu valoarea Q maximă.

În cazul metodei DQN, funcția valoare-acțiune este modelată de o rețea neuronală profundă. În timpul simulării episoadelor, se utilizează metoda reluării experienței. Traiectoriile generate de agent sunt salvate într-un buffer de memorie, care este ulterior eșantionat pentru a construi loturi de antrenament ca intrări pentru rețeaua neuronală responsabilă de estimarea valorii Q.

Rețeaua neuronală urmărește să minimizeze suma pătratelor erorii TD (vezi Ecuația 2.4).

Eroarea TD este calculată folosind o a doua rețea neuronală, numită rețea țintă având parametrii θ_{target}, care este o copie periodică a rețelei online.

S-a demonstrat că DQN este excesiv de optimist [37] și Double deep Q-learning [37] a fost introdus pentru a aborda problema supraestimării. Dueling deep Q-learning [102] estimează funcția valoare-stare și funcția avantaj-acțiune folosind rețele duelante: aceeași reprezentare a caracteristicilor este utilizată ca intrare pentru rețelele de valoare și de avantaj.

Conservative Q-Learning (CQL) introdus în [53] este o tehnică desemnată pentru Q-learning în mod offline, unde modelul învață dintr-un set de date de experiențe pre-colectate, fără nicio oportunitate de a interacționa cu mediul.

2.3 Abordări de învățare prin întărire multi-agent

Învățarea prin întărire multi-agent (eng. multi-agent reinforcement learning, MARL) constă dintr-un set de diverși agenți autonomi bazați pe dinamica agent-mediu caracteristică învățării prin întărire. Există mai multe moduri de a adopta succesul abordărilor cu un singur agent în scenarii multi-agent. Metodele complet descentralizate, cum ar fi independent Q-learning [91], tratează agenții ca entități separate care încearcă să optimizeze acțiunile în mod descentralizat. Agenții împărtășesc experiențe între ei, iar o singură politică este antrenată pentru a determina acțiunile optime.

2.3.1 Antrenament centralizat și descentralizat

Abordările care optimizează scenariile multi-agent pot fi împărțite în două categorii în funcție de modul în care gestionează informațiile disponibile: metode centralizate și descentralizate. Abordările centralizate tind să integreze o viziune globală a mediului, unde selecția acțiunilor este optimizată în comun. În cazul scenariului descentralizat, agenții operează în paralel, și agenții își optimizează acțiunile separat.

2.3.2 Rețele de descompunere a valoriilor

Rețelele de descompunere a valorii [89] încorporează atât tehnici centralizate, cât și descentralizate. Rețeaua folosește o recompensă globală ca feedback al acțiunilor comune ale agenților,

iar eroarea TD pentru funcția valoare-acțiune comună este utilizată ca funcție de pierdere la antrenare. Recompensa globală este descompus, pentru a antrena o rețea neuronală separată pentru fiecare agent. Rețelele de descompunere a valorii beneficiază de învățare centralizată și execuție descentralizată.

Factorizarea monotonă a funcției de valoare pentru învățarea prin întărire profundă (QMIX) propusă în [81] aplică o structură alcătuită din mai multe rețele neuronale pentru a aproxima funcția valoare-acțiune comună Q_{total}: (i) rețele neuronale ale agenților, care aproximează funcția Q_a și (ii) o rețea de mixare care sintetizează istoricul observație-acțiune și perechile de acțiuni.

Mahajan et al. [67] a arătat că presupunerea monotonă cerută de QMIX descurajează explorarea și a propus explorarea variațională multi-agent (MAVEN) pentru a prioritiza explorarea în metodele de antrenament centralizat-execuție descentralizată.

2.4 platforme de evaluare de referință pentru metodele de învățare prin întărire

Au fost propuse multiple platforme de evaluare de referință pentru a evalua performanța abordărilor de învățare prin întărire. În [56], Leike et al. evaluează algoritmii RL pe probleme de siguranță cu care se confruntă agenții autonomi, cum ar fi întreruperea semnalelor și evitarea efectelor secundare. Chen et al. [16] implementează un sistem de învățare prin întărire multi-agent profundă pentru a propune soluții la problema de fuziune pe rampa de acces a autostrăzii cu trafic mixt. În [95] este propus o platformă pentru a evalua performanța metodelor de învățare multi-agent în jocul StarCraft II. MineDojo [29] este o platformă bazată pe simulatorul Minecraft, oferind o suită de sarcini programatice și creative.

2.5 Sustenabilitate și învățare prin întărire

Mai multe contribuții investighează potențialul învățării automate pentru a atinge obiective sustenabile, de ex., [76, 79, 106, 108]. De asemenea, subiectul sustenabilității abordărilor de învățare automată a fost investigat, vezi, de ex., [1, 85].

Diverse contribuții din literatură propun tehnici pentru a gestiona eficacitatea antrenamentului. De exemplu, [24] a arătat că rețelele neuronale rare ating performanța rețelelor dense . Oprirea timpurie [74] termină antrenamentul odată ce metricile care raportează performanța modelului încep să se deterioreze. Eficiența metodelor de învățare prin întărire poate fi îmbunătățită prin aplicarea tranzițiilor probabilistice pentru a încorpora incertitudinea în modelare și planificare [40]. Metodele de cuantizare s-au dovedit să reducă timpul de antrenament și emisiile de carbon ale învățării prin întărire distribuite [30].

Chapter 3

Algoritmi de învățare prin întărire pentru medii cooperative-competitive

Se propun arhitecturi bazate pe învățarea profundă prin întărire (eng. reinforcement learning, RL) pentru a aborda problema navigării în medii cooperative-competitive care modelează dinamica prădător-pradă. Se propune un cadru de tip Deep Q-network (DQN) și o arhitectură bazată pe factorizarea monotonă a funcției de valoare (QMIX) pentru a optimiza selecția acțiunilor prin antrenament- și execuție descentralizate. Abordările propuse și experimentele au fost prezentate în [47]. Apoi, se propun versiuni îmbunătățite ale abordărilor DQN și QMIX menționate anterior. Arhitectura QMIX este combinată cu explorarea variațională multi-agent (MAVEN) pentru a îmbunătăți eficiența antrenării centralizate. Versiunea extinsă a experimentelor și metodologia îmbunătățită au fost publicate în [48].

3.1 Definirea problemei

Sistemele de planificare multi-agent au numeroase aplicații în lumea reală, precum detectarea știrilor false [3], monitorizarea fenomenelor evolutive [6], evitarea coliziunilor [12] și analiza rețelelor sociale [14]. Mediile multi-agent, în care un grup de agenți are un obiectiv comun de atins, sunt denumite medii cooperative. *Mediile cooperative-competitive* sunt formate din mai multe grupuri care cooperează, având obiective opuse altor grupuri de agenți.

Contribuțiile recente din literatura de specialitate se concentrează pe implementarea diverselor medii de tip cooperativ-competitiv, cu scopul de a oferi platforme de evaluare de

referință pentru metodele de învățare prin întărire multi-agent; de exemplu, [95, 64, 92]. Aceste platforme au ca scop facilitarea utilizării învățării prin întărire cu agenți multipli, oferind un cadru pentru simularea și evaluarea mediului.

Librări PettingZoo integrează diverse jocuri multi-agent potrivite pentru modelarea relațiilor multi-agent complexe din viața reală, de exemplu platforma MAgent [112], sarcini de control discrete și continue [35], Multi Particle Environments [73].

3.2 Stadiul actual al cunoașterii

Cercetările anterioare în domeniul învățării prin întărire cu agenți multipli, bazată pe cooperare și competiție, au testat utilizând platforme precum medii bazate pe StarCraft [27, 82, 95] și MAgent[112]. QMIX a fost evaluat în mediul StarCraft și a depășit performanța rețelelor independente de Q-learning și de descompunere a valorilor; vezi [81]. MAVEN a fost evaluat pe platforma SMAC [82] [67], și a demonstrat o performanță mai bună decât metoda Q-learning independentă, QMIX, COMA [31] and QTRAN [87].

3.3 Metode de învățare prin întărire

În [47], au fost propuse sisteme de învățare profundă prin întărire, atât descentralizate cât și centralizate, pentru operarea în medii cooperative-competitive. Mai exact, au fost adaptate două metode de tip Q-learning: (i) Învățarea independentă[91] aplicată împreună cu metoda DQN [71] și (ii) o abordare bazată pe QMIX [81]. În primă fază, este evaluată o abordare de învățare independentă [91] în combinație cu rețele Deep Q-Networks [71].

DQN aplică algoritmul Q-learning utilizând o rețea neuronală pentru a aproxima valorile acțiunilor și pentru a selecta acțiunea cu valoarea cea mai mare. În cazul învățării independente, scenariul multi-agent este tratat ca o problemă de învățare prin întărire cu un singur agent, acțiunile și obiectivele celorlalți agenți fiind considerate parte a mediului. Procesul de învățare este descentralizat, în timp ce agenții partajează experiențele între ei.

QMIX antrenează agenții cooperativi cu o recompensă globală prin învățarea unor rețele de utilitate per agent și a unei rețele de mixare monotone care combină valorile Q ale agenților. QMIX implementează selecția descentralizată a acțiunilor și învață să optimizeze în mod comun

acțiunile agenților.

Au fost antrenate două politici distincte pentru selecția acțiunilor: o politică este antrenată pentru selecția acțiunilor agenților de tip prădător, respectiv pradă. Cele două politici sunt inițializate cu aceiași parametri, însă, în timpul antrenării, agenții operează urmând politica lor predefinită. Astfel, agenții cu obiective similare partajează experiențe, dar nu intervin în optimizarea recompenselor atribuite agenților adversari.

Rezultatele experimentale în mediul *adversarial pursuit* au demonstrat că abordările de învățare profundă prin întărire sunt capabile să construiască politici adaptive pentru navigarea în simulări de tip prădător-pradă.

3.4 Q-learning profund centralizat și descentralizat

Eficiența algoritmilor centralizați și descentralizați a fost evaluată concentrându-se pe trei variante de rețele DQN în medii cooperative-competitive: (i) învățare independentă cu antrenament descentralizat [91] folosind DQN [71], (ii) factorizarea monotonă a valorii pentru învățare profundă centralizată (QMIX) [81], (iii) explorarea variațională multi-agent[67]. Experimentele computaționale prezentate în Secțiunea 3.3 au fost extinse schimbarea parametrilor pentru metodele DQN și QMIX, precum și prin propunerea unei metode RL, care combină Q-learning centralizat cu explorarea variațională multi-agent [67], utilizând un alt mediu multi-agent din [66] care simulează dinamica prădător-pradă.

Pentru a evalua eficiența politicilor învățate, se realizează evaluarea acestora după antrenare. Politicile învățate sunt evaluate într-o manieră deterministă, evitându-se astfel selecția aleatorie a acțiunilor pentru agenți. Sunt construite politici de selecție a acțiunilor separate pentru diferite grupări de agenți; în comparație cu metoda aplicată în Secțiunea 3.3, grupurile de agenți prădători nu mai sunt împărțite în sub-grupuri pentru a echilibra numărul de agenți în vederea antrenării politicilor.

Experimentele au fost efectuate utilizând următoarele medii RL care sunt incluse și în biblioteca PettingZoo [92]: (i) *Adversarial pursuit* din MAgent [112] și (ii) *Simple tag* din Multi-Particle Environments (MPE [66]). Atât *adversarial pursuit*, cât și *simple tag* simulează interacțiuni cooperative-competitive.

Antrenarea descentralizată cu DQN a depășit constant abordările centralizate în ambele

medii, *adversarial pursuit* și *simple tag*. Politicile DQN au prezentat o convergență mai rapidă și au obținut recompense acumulate mai mari atât pentru grupurile de agenți prădători, cât și pentru cele de pradă în timpul evaluării. În timp ce metodele centralizate, precum QMIX și MAVEN, au necesitat perioade de antrenare mai lungi pentru a se stabiliza, integrarea MAVEN cu QMIX a adus îmbunătățiri față de QMIX standard, în special în ceea ce privește reducerea variației și îmbunătățirea explorării în scenariile de antrenare centralizată.

Chapter 4

Soluții de planificare a traseului pentru sisteme multi-agent

Abodări hibride și de învățare profundă prin întărire (RL) au fost dezvoltate pentru a optimiza planificarea traseului în sistemele multi-agent. În primă fază, a fost dezvoltat o metodă standardizată de extracție a caracteristicilor cuplat cu învățarea profundă de tip Q-learning [52]. O abordare hibridă de tip greedy, H*, a fost introdusă pentru planificarea acțiunilor roboților în medii multi-robot bidimensionale [51]. Apoi, este definit un mediu RL pentru a modela și optimiza politicile de navigare în medii multi-robot [49]. În cele din urmă, a fost construit un controler descentralizat bazat pe RL, antrenat prin învățare prin imitare de la o politică hibridă expertă, care sintetizează mediul RL menționat anterior cu rezolvarea analitică a conflictelor și algoritmul H* [50].

4.1 Definirea problemei

Planificarea traseului optimizează secvența de mișcări necesară pentru a naviga un agent de la o poziție inițială la o țintă, evitând în același timp coliziunile cu obstacolele.

Planificarea traseului este un element crucial pentru mai multe probleme complexe de optimizare, de exemplu, problema rutei vehiculelor (eng. Vehicle Routing Problem, VRP), așa cum este definită în [21].

4.1.1 Problema replanificării transportului

Problema replanificării transportului (eng. Vehicle Rescheduling Problem, VRSP) extinde problema clasică VRP din cauza necesității de a adapta și replanifica continuu rutele vehiculelor ca răspuns la perturbări în timp real, întârzieri și defecțiuni în sistemele de transport.

Obiectivul principal este minimizarea sumei timpilor de deplasare ai tuturor agenților de la pozițiile lor actuale la destinațiile respective, luând în considerare starea dinamică a sistemului, inclusiv posibilele întârzieri, defecțiunile echipamentelor și condițiile variabile.

4.1.2 Planificarea traseelor pentru mai mulți roboți

Problema planificării traseului multi-robot (eng. Multi-robot Path Planning, MPP) implică scenariul în care roboții se deplasează printr-un spațiu comun. Scopul MPP este de a elabora strategii pentru traseele pe care toți roboții trebuie să le parcurgă în mediul comun pentru a ajunge la o destinație specifică, asigurând absența coliziunilor cu ceilalți roboți sau cu obstacolele din jur.

Problema MPP constă în două sarcini computaționale care trebuie sincronizate: (i) determinarea traseelor individuale pentru fiecare robot într-un mediu dat și (ii) coordonarea acțiunilor roboților pentru a minimiza coliziunile sau întârzierile.

4.2 Stadiul actual al cunoașterii

VRP extinde problema comis-voiajorului (TSP), având ca scop minimizarea distanței totale de deplasare sau a costurilor. Fiecare client trebuie vizitat exact o dată, iar cererea agregată pe orice rută nu trebuie să depășească capacitatea vehiculului alocat.

VRP este legată de problema multiplă a comis-voiajorului (MTSP), o soluție pentru MTSP rezolvă și problema VRP [88]. MTSP a fost abordată prin diverse metode, inclusiv metode exacte [54], algoritmi de aproximare [23, 68] și abordări bazate pe învățarea automată [78]. Au fost propuse medii cu un singur și cu mai mulți agenți pentru a facilita aplicarea metodelor RL în diverse sarcini de coordonare a traficului [13, 70, 72]. Învățarea profundă prin întărire a apărut ca o direcție promițătoare, care învață politici pentru a aborda problema planificării rutei vehiculelor. În [75], autorii propun un decoder bazat pe rețele neuronale recurente cuplat

cu o politică bazată pe mecanismul de atenție pentru a aborda VRP și variantele sale.

Problema VRSP este NP-hard [88], abordările propuse pentru soluționarea sa fiind diverse, multe dintre acestea bazându-se pe euristici; cele mai utilizate tehnici includ căutarea în adâncime [93], branch and bound [22], algoritmi genetici [26] și simularea răcirii [94].

Euristici bazate pe A* [36] au fost propuse pentru MPP, de exemplu, A* cooperativ [86], M* [96]. MPP a fost, de asemenea, abordată utilizând meta-euristici, inclusiv algoritmi genetici [80], optimizarea prin stoluri de particule [111], optimizarea prin colonii de furnici [113].

În [70], roboții au fost antrenați să navigheze utilizând învățarea profundă prin întărire, fiind introduse ulterior numeroase îmbunătățiri. Învățarea profundă prin întărire a fost aplicată pentru a optimiza planificarea rutelor și sarcinile de evitare a coliziunilor pentru setările multi-robot cooperative [97].

Problema rutei vehiculelor electrice (EVRP) extinde problema clasică VRP [84]. Metodele meta-euristice, precum algoritmii hibridizate de optimizare prin colonii de furnici [28, 104, 105], au avut succes în rezolvarea EVRP.

4.3 Învățarea pe baza observațiilor bazată pe rezolvarea analitică a conflictelor

În [52], problema VRSP formalizată de Flatland Challenge [72] a fost abordată prin utilizarea unei abordări de învățare profundă bazată pe RL, împreună cu un mecanism analitic de rezolvare a conflictelor. A fost introdusă o nouă metodă de extragere a caracteristicilor prin analiza conflictelor între perechi, urmată de un deep Q-network, care îndeplinește rolul de controler descentralizat. Abordarea propusă a fost evaluată pe baza unei serii de simulări, în conformitate cu Flatland [72].

Abordarea noastră privind VRSP, formalizată prin Flatland Challenge, se bazează pe identificarea, caracterizarea și, dacă este posibil, rezolvarea fiecărui conflict care apare între *perechile de trenuri*. Observațiile personalizate pentru sistemul de RL sunt calculate prin aplicarea succesivă a modelului de rezolvare a conflictelor în formă de H și a unui mecanism de vot. Aceste observații sunt utilizate ca date de intrare pentru rețeaua neuronală de RL. Q-learning a fost utilizată pentru a aproxima funcția comună de valoare a acțiunii agenților. Pentru testarea cadrului propus au fost utilizate arhitecturile DQN [71] și Linear Q-network (LQN).

Rezultatele experimentale demonstrează că se poate învăța un algoritm de planificare eficientă pe baza spațiului de caracteristici propus.

4.4 Abordare hibridă pentru planificarea traseelor roboților

În [51], am propus algoritmul H* care abordează problema MPP prin identificarea rutelor fără coliziuni pentru scenariile multi-robot. Algoritmul H* constă în două etape. În prima etapă, se aplică o căutare greedy a traseelor utilizând o abordare euristică pentru a genera rute pentru roboți, cunoscând pozițiile inițiale și țintă ale acestora. În a doua etapă, se aplică un operator de căutare locală pentru a rafina rutele determinate de căutarea greedy, având ca scop minimizarea celor două obiective: lungimea și coliziunile.

Experimentele computaționale sunt efectuate pentru mai multe scenarii care iau în considerare caracteristici diferite și impun diverse niveluri de dificultate pentru metodele de planificare. Trei configurații distincte de hărți multi-robot au fost selectate pentru a evalua eficiența metodei hibride H* în contextul problemei MPP: Scenariul 1 [42], Scenariul 2 [33], Scenariul 3 [69]. Rezultatele experimentale au arătat că algoritmul H* este robust privind numărul de roboți și rezolvă cu succes toate configurațiile testate. H* a avut o performanță similară sau a depășit alte abordări euristice și meta-euristice din literatura de specialitate.

4.5 Abordare de învățare prin întărire pentru planificarea traseelor roboților

Învățarea independentă [91] este combinată cu rețelele Dueling Deep Q-Networks [102] și Double Deep Q-learning [37] pentru a aborda problema MPP [44, 49]. Utilizând RL, este implementat un controler descentralizat: roboții partajează între ei parametrii politicii și memoriile interacțiunilor anterioare cu mediul. Ca o consecință a partajării experiențelor în timpul antrenării, dimensiunea spațiului soluțiilor este redusă. Spațiile de soluții mai mici conduc la o convergență mai rapidă și necesită mai puține simulări pentru a învăța o politică eficientă. Soluțiile descentralizate sunt agnostice în ceea ce privește numărul de roboți, atâta timp cât agenți au aceeași dinamică.

Performanța antrenamentului bazat pe Dueling Double DQN a fost evaluată în scenariul

bidimensional propus de [42]. Experimentele au arătat că o creștere a timpului de antrenare este asociată cu un consum crescut de energie. Deși resursele computaționale necesare pentru antrenarea cu cele două dimensiuni de observație testate sunt apropiate, utilizarea unei raze de observație mai mici era opțiunea mai eficientă din punct de vedere energetic în experimentele noastre.

4.6 Învățarea profundă prin imitație folosind Q-learning

Sistemul RL descris în Secțiunea 4.5 este extins, și această secțiune prezintă o metodă RL bazată pe învățare prin imitație în două etape. În prima etapă, o politică centralizată de expert, bazată pe mecanisme de anticipare temporală și rezolvare a conflictelor, generează traiectorii de demonstrație. În a doua etapă, o politică descentralizată este antrenată pe traiectoriile expertului printr-o combinație de învățare prin imitație și RL offline.

4.6.1 Politica expertului hibrid

Politica expert este un controler centralizat bazat pe algoritmul H* și pe rezolvarea analitică a conflictelor, propusă în [52], pentru navigație multi-robot. Politica expert este utilizată pentru a genera un set de date offline care modelează strategii de navigație pentru mai mulți roboți care operează într-un mediu comun. Se construiește un controler centralizat care funcționează ca un sistem hibrid, combinând căutarea deterministă pe graf cu rezolvarea euristică a conflictelor.

4.6.2 Modelul Double Deep Q-learning

O abordare offline Double DQN este utilizată pentru optimizarea selecției acțiunilor pentru roboți. Politica de acțiune este împărtășită între agenți, și o rețea profundă este antrenată prin Q-learning independent [91]. Mecanismul de actualizare soft [60] a fost aplicat pentru a modifica parametrile rețelei țintă în timpul antrenamentului. Sistemul RL este antrenat folosind conservative Q-learning (CQL).

Experimentele au fost realizate pe Scenariul 1 propus de [42] pentru a valida aplicabilitatea abordării de învățare prin imitație offline. Rezultatele experimentelor computazionale indică faptul că modelul Double DQN învață să navigeze pe baza demonstrațiilor expertului.

Chapter 5

Învățare prin întărire pentru maximizarea influenței

În ultimii ani, au fost înregistrate progrese semnificative în ceea ce privește problema maximizării influenței (IM) în rețelele sociale. Problema maximizării influenței [41] vizează maximizarea acoperirii informațiilor, minimizând în același timp costul asociat propagării informațiilor. IM are o gamă largă de aplicații în viața reală, inclusiv marketing [19, 101], epidemiologie [107], științe politice [32].

Au fost dezvoltate două metode care utilizează învățarea prin întărire profundă pentru a aborda problema maximizării influenței în rețelele sociale cu semne. Au fost propuse selecții de noduri-seed comune și iterative pentru scenariul de maximizare a influenței competitive în rețele de tip încredere-neîncredere [45]. Modele de tip Deep Q-Network (DQN) au fost integrate în mecanismele de selecție a nodurilor pentru a determina seturile de semințe pentru rețelele sociale de dimensiuni mici și medii. Al doilea model, publicat în [46], constă dintr-un sistem de învățare prin întărire multi-agent bazat pe comunități, oferind strategii scalabile și adaptive pentru rețele sociale cu semne la scară largă.

5.1 Definirea problemei

Maximizarea influenței [41] vizează optimizarea propagării informației în rețelele sociale pornind de la un set de noduri sursă (semințe, noduri-seed). Fie K numărul maxim de noduri din setul de noduri-seed, numit și *buget*. În [41], autorii au propus,un algoritm de referință de tip greedy

sub cele două modele principale de propagarea informației, și anume Independent Cascade și Linear Thresholds. Problema maximizării influenței este NP-hard [41], prin urmare, pe lângă abordările propuse pentru a optimiza maximizarea influenței, pot fi aplicate diverse metode de inteligență artificială pentru a reduce cerințele computaționale, de exemplu, învățarea prin întărire.

Bharathi et al. [5] discută strategiile primului actor (first-mover) și ale celui de-al doilea actor (second-mover) pentru identificarea unui set de semințe în vederea maximizării influenței, atunci când competiția este prezentă în rețeaua socială.

Carnes et al. [15] a discutat perspectiva adepților (second-mover), unde setul de semințe este construit cunoscând care noduri semințe sunt activate de oponent. [15] a extins modelul cascadei independente la scenariul competitiv. Ambele modele sunt potrivite pentru construirea unui set de semințe în mod greedy pentru a aborda problema maximizării competitive a influenței (CIM).

Problema maximizării influenței formalizată în [41] este definită pe rețele sociale care au un singur tip de relație între indivizi. Maximizarea influenței în funcție de polaritate (PRIM) operează luând în considerare două tipuri de relații opuse [59].

În maximizarea competitivă a influenței, doi actori opuși încearcă să influențeze nodurile din aceeași rețea, astfel încât fiecare nod poate fi inactiv sau activat. Un nod inactiv nu a fost influențat de niciunul dintre actori, un nod activat are o polaritate (pozitivă sau negativă) care indică ce actor l-a influențat.

5.2 Stadiul actual al cunoașterii

[41] a propus un algoritm greedy care maximizează influența prin selecția iterativă a nodurilor având influența marginală maximă. CELF [58] și CELF++ [34] îmbunătățesc performanța acestui algoritm greedy prin exploatarea naturii sub-modulare a difuziei influenței.

Pe lângă algoritmii greedy, abordările meta-euristice care adresează problema IM includ: algoritmi genetici [10], algoritmi evolutivi multi-obiectiv [11], simularea răcirii [39], optimizarea prin stol de particule [100]. În [17], autorii au propus un model de învățare prin întărire privind problema maximizării influenței în grafuri aleatorii.

Mai mulți algoritmi existenți care abordează problema IM beneficiază de detectarea comunităților

și construirea de diverse strategii pentru a exploata informațiile bazate pe comunități, de exemplu, [43, 8, 38].

În [62], a fost propus un sistem de învățare prin întărire pentru a optimiza selecția semințelor pentru problema maximizării competitive a influenței. ToupleGDD [18] utilizează rețele neuronale grafice cuplate pentru a obține reprezentări ale nodurilor și aplică Double DQN pentru a estima influența așteptată. Recent, [114, 98] au îmbunătățit performanța hibrizilor de rețele neuronale grafice și învățare prin întărire, depășind ToupleGDD.

Greedy-SIM [65] este o metodă greedy care optimizează propagarea influenței pozitive în rețele cu semne prin identificarea căilor de propagare independente și a probabilităților acestora. În [109], autorii consideră convingerile individuale și difuzia informației în rețele cu semne, și aplică metoda Signed-PageRank pentru rezolvarea problemei IM în rețele cu semne.

5.3 Competitive influence maximization in trust-distrust networks

În [45], fost construite arhitecturi bazate pe învățarea prin întărire profundă pentru a aborda maximizarea influenței competitive în rețelele sociale cu semne. Au fost analizate două mecanisme distincte pentru construirea seturilor inițiale de noduri de pornire pentru doi actori concurenți: selecția comună și selecția iterativă a nodurilor de pornire. Eficacitatea seturilor posibile de noduri de pornire este comparată pe baza numărului de noduri activate după simularea unei cascade independente legate de polaritate în rețelele de încredere-neîncredere.

În cazul selecției comune a nodurilor-seed, setul de semințe pentru primul agent care produce noduri activate pozitiv este selectată ca o acțiune a agentului RL. Seturile posibile de semințe sunt obținute pe baza infrastructurii rețelei sociale înainte de aplicarea deep Q-learning.

În cazul selecției iterative a semințelor, nodurile sunt adăugate unul câte unul în setul de semințe. Episodul de învățare prin întărire constă într-un număr maxim de k iterații, în fiecare iterație actorul selectând un nod neactivat pentru a fi adăugat în seturile lor de semințe.

Experimentele au fost realizate pe rețele sociale bazate pe încredere și neîncredere, construite din relațiile utilizatorilor de pe site-ul de recenzii `epinions.com` [57].

Două abordări DQN [71] au fost evaluate pentru selectarea seturile semințelor în problema maximizării influenței competitive [15] în rețele sociale bazate pe relații de încredere. Două

modele distincte de învățare prin întărire au fost aplicate pentru formalizarea problemei maximizării influenței și construirea spațiului de soluții.

Rezultatele experimentale arată că învățarea profundă prin întărire este potrivită pentru propunerea de seturi de semințe în problema maximizării influenței competitive pe rețele cu semne.

Selecția comună a semințelor este fezabilă pentru determinarea seturilor de semințe pentru maximizarea influenței competitive. Metoda selecției comune a semințelor nu scalează bine, deoarece spațiul de acțiune crește rapid odată cu bugetul alocat mulțimii de semințe. Experimentele arată că selecția iterativă a semințelor poate fi utilizată împreună cu abordarea DQN pentru a opera pe rețele de dimensiuni medii.

5.4 Învățarea prin întărire multi-agent pentru maximizarea influenței în rețele cu semne

Această secțiune prezintă abordarea Double DQN multi-agent bazată pe comunități [46] pentru a aborda problema PRIM în rețelele sociale.

Agentul RL construiește setul de semințe într-un mod iterativ, selectând câte un nod cu fiecare acțiune pentru a fi adăugat la setul de semințe. Activarea nodurilor și propagarea influenței fac parte din mediul agentului RL. Nodurile semințe determinate de agent sunt activate ca fiind pozitive, iar propagarea influenței este simulată urmând cascada independentă legată de polaritate.

Prin selectarea strategiilor pentru fiecare comunitate mai degrabă decât pentru întreaga rețea de intrare, problema originală este descompusă în subprobleme mai mici.

Folosind algoritmul de detectare a comunităților Louvain [7] pe graful orientat, ignorând semnele muchiilor, graful original este împărțit în mai multe sub-grafuri distincte, grupând nodurile în comunități. Fiecărei comunități i se atribuie apoi un agent RL (actor), permițând luarea deciziilor în mod descentralizat pentru nominalizarea nodurilor semințe în cadrul sistemului de învățare prin întărire.

Metoda Double DQN reprezintă politica de selecție a acțiunilor agenților RL, care este partajată între aceștia. Rețelele țintă sunt actualizate prin actualizări de tip soft [60] pentru a stabiliza procesul de învățare.

Experimentele au fost efectuate pe rețelele sociale cu semne Bitcoin OTC, WikiElec și Slashdot, care conțin relații bazate pe încredere și neîncredere între utilizatori.

Algoritmul de detectare a comunităților Louvain reduce cu succes dimensionalitatea rețelelor sociale originale prin identificarea sub-grafurilor puternic conectate.

Deși nodurile activate și negative apar după simularea difuziei informației, numărul de noduri activate și pozitive rămâne substanțial mai mare decât numărul de noduri negative și activate, demonstrând că sistemul RL propus încorporează cu succes diferența dintre muchiile pozitive și cele negative.

Chapter 6

Concluzii și direcții viitoare de cercetare

Învățarea prin întărire (eng. Reinforcement Learning - RL) este o ramură a învățării automate bazată pe interacțiunea dintre agent și mediu. Prezenta teză propune abordări bazate pe RL pentru a rezolva probleme de optimizare pe sisteme multi-agent.

6.1 Principalele contribuții și rezultate

În primul rând, problema navigării în medii cooperative-competitive a fost abordată prin propunerea a două sisteme de învățare profundă prin întărire. Experimentele realizate în mediul MAgent au arătat că abordările de tip deep Q-networks sunt capabile să învețe politici adaptative [47]. Apoi, au fost propuse arhitecturi centralizate și descentralizate bazate pe Q-learning pentru a aborda navigarea în cadrul mediilor cooperative-competitive [48]. Experimentele au demonstrat că performanța antrenării centralizate cu factorizarea monotonă poate fi îmbunătățită prin aplicarea explorării variaționale multi-agent.

În al doilea rând, planificarea rutelor în sisteme multi-agent a fost abordată utilizând învățarea automată: (i) a fost propusă o abordare bazată pe deep Q-learning bazată pe rezolvarea analitică a conflictelor între agenți [52], (ii) a fost dezvoltată o metodă hibridă adaptativă pentru planificarea traseelor roboților, (iii) a fost propus un model descentralizat de deep Q-learning pentru planificarea traselor în sisteme cu mulți roboți [44, 49], (iv) a fost introdus un sistem de învățare prin imitație (eng. imitation learning) bazat pe experți hibride [50].

În al treilea rând, problema maximizării influenței a fost abordată folosind deep Q-learning. În [45] au fost propuse două mecanisme de selecție a nodurilor-seed bazate pe deep Q-learning

pentru maximizarea influenței competitive în rețele cu semne, și anume selecția comună și selecția iterativă. În [46], a fost propusă o metodă de învățare prin întărire multi-agent bazată pe comunități pentru maximizarea influenței în funcție de polaritate. Abordarea propusă a fost validată pe rețelele direcționate polarizate, Bitcoin OTC, WikiElec și Slashdot, iar rezultatele computaționale au arătat că agenții RL bazați pe comunități sunt capabili să optimizeze propagarea influenței pozitive.

6.2 Direcții viitoare de cercetare

Principala direcție pentru cercetările viitoare vizează extinderea lucrărilor existente prin evaluarea performanței metodelor deep Q-learning propuse pe seturi de date din lumea reală și platforme de evaluare de referință. În cazul mediilor cooperative-competitive, abordările prezentate ar putea fi aplicate pe platforme de referință cooperative, cum ar fi SMACv2 [27].

Robustețea metodelor propuse pentru planificarea traseelor ar putea fi evaluată utilizând mediul VMAS [4]. O altă potențială direcție viitoare constă în utilizarea metodologiei transfer learning [103] pentru problema planificării traseelor pentru mai mulți roboți.

O potențială direcție viitoare pentru cercetarea efectuată asupra problemei maximizării influenței o reprezintă adaptarea metodei multi-agent de învățare prin întărire bazată pe comunități la problema maximizării echilibrate a influenței [61]. O altă direcție viitoare este utilizarea rețelelor neuronale grafice (eng. graph neural networks) [83] pentru a extrage caracteristicile comunităților.

Bibliography

[1] Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., and Taha, K. (2015). Efficient machine learning for big data: A review. *Big Data Research*, 2(3):87–93.

[2] Albrecht, S. V., Christianos, F., and Schäfer, L. (2024). *Multi-Agent Reinforcement Learning: Foundations and Modern Approaches*. MIT Press.

[3] Aymanns, C., Foerster, J., and Georg, C.-P. (2017). Fake news in social networks. *SSRN Electronic Journal Paper No. 2018/4*.

[4] Bettini, M., Kortvelesy, R., Blumenkamp, J., and Prorok, A. (2024). VMAS: A vectorized multi-agent simulator for collective robot learning. In Bourgeois, J., Paik, J., Piranda, B., Werfel, J., Hauert, S., Pierson, A., Hamann, H., Lam, T. L., Matsuno, F., Mehr, N., and Makhoul, A., editors, *Distributed Autonomous Robotic Systems*, pages 42–56, Cham. Springer Nature Switzerland.

[5] Bharathi, S., Kempe, D., and Salek, M. (2007). Competitive Influence Maximization in Social Networks. In Deng, X. and Graham, F. C., editors, *Internet and Network Economics*, pages 306–311, Berlin, Heidelberg. Springer Berlin Heidelberg.

[6] Bloembergen, D., Tuyls, K., Hennes, D., and Kaisers, M. (2015). Evolutionary dynamics of multi-agent learning: A survey. *Journal of Artificial Intelligence Research*, 53:659–697.

[7] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

[8] Bozorgi, A., Haghighi, H., Sadegh Zahedi, M., and Rezvani, M. (2016). INCIM: A

community-based algorithm for influence maximization problem under the linear threshold model. *Information Processing & Management*, 52(6):1188–1199.

[9] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1):139–159.

[10] Bucur, D. and Iacca, G. (2016). Influence maximization in social networks with genetic algorithms. In Squillero, G. and Burelli, P., editors, *Applications of Evolutionary Computation*, pages 379–392, Cham. Springer International Publishing.

[11] Bucur, D., Iacca, G., Marcelli, A., Squillero, G., and Tonda, A. (2017). Multi-objective evolutionary algorithms for influence maximization in social networks. In Squillero, G. and Sim, K., editors, *Applications of Evolutionary Computation*, Lecture Notes in Computer Science, pages 221–233, Germany. Springer.

[12] Cai, C., Yang, C., Zhu, Q., and Liang, Y. (2007). Collision avoidance in multi-robot systems. In *2007 International Conference on Mechatronics and Automation*, pages 2795–2800.

[13] Cai, P., Lee, Y., Luo, Y., and Hsu, D. (2020). SUMMIT: A simulator for urban driving in massive mixed traffic. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4023–4029. IEEE.

[14] Carley, K., Martin, M., and Hirshman, B. (2009). The etiology of social change. *Topics in Cognitive Science*, 1:621 – 650.

[15] Carnes, T., Nagarajan, C., Wild, S. M., and van Zuylen, A. (2007). Maximizing influence in a competitive social network: A follower's perspective. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, page 351–360, New York, NY, USA. Association for Computing Machinery.

[16] Chen, D., Li, Z., Wang, Y., Jiang, L., and Wang, Y. (2021a). Deep multi-agent reinforcement learning for highway on-ramp merging in mixed traffic. *arXiv preprint arXiv:2105.05701*.

[17] Chen, M., Zheng, Q., Boginski, V., and Pasiliao, E. (2021b). Influence maximization in social media networks concerning dynamic user behaviors via reinforcement learning. *Computational Social Networks*, 8.

[18] Chen, T., Yan, S., Guo, J., and Wu, W. (2024). ToupleGDD: A fine-designed solution of influence maximization by deep reinforcement learning. *IEEE Transactions on Computational Social Systems*, 11(2):2210–2221.

[19] Chen, W., Wang, C., and Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, page 1029–1038, New York, NY, USA. Association for Computing Machinery.

[20] Cheng, M., Chen, H., Li, Z., Wang, J., and Wang, S. (2025). Role-aware multi-agent reinforcement learning for coordinated emergency traffic control. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[21] Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.

[22] D'Ariano, A., Pranzo, M., and Hansen, I. A. (2007). Conflict resolution and train speed coordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222.

[23] Deppert, M., Kaul, M., and Mnich, M. (2023). A $(3/2 + \epsilon)$-approximation for multiple tsp with a variable number of depots. In Gørtz, I. L., Farach-Colton, M., Puglisi, S. J., and Herman, G., editors, *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:15, Dagstuhl, Germany. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[24] Dettmers, T. and Zettlemoyer, L. (2019). Sparse networks from scratch: Faster training without losing performance. *ArXiv*, abs/1907.04840.

[25] dos Santos, D. S. and Bazzan, A. L. (2012). Distributed clustering for group formation and task allocation in multiagent systems: A swarm intelligence approach. *Applied Soft Computing*, 12(8):2123–2131.

[26] Dündar, S. and Sahin, I. (2013). Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways. *Transportation Research Part C: Emerging Technologies*, 27:1–15.

[27] Ellis, B., Cook, J., Moalla, S., Samvelyan, M., Sun, M., Mahajan, A., Foerster, J. N., and Whiteson, S. (2023). Smacv2: an improved benchmark for cooperative multi-agent reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

[28] Fan, L. (2024). A two-stage hybrid ant colony algorithm for multi-depot half-open time-dependent electric vehicle routing problem. *Complex & Intelligent Systems*, 10(2):2107–2128.

[29] Fan, L., Wang, G., Jiang, Y., Mandlekar, A., Yang, Y., Zhu, H., Tang, A., Huang, D.-A., Zhu, Y., and Anandkumar, A. (2022). Minedojo: Building open-ended embodied agents with internet-scale knowledge. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[30] Faust, A., Barth-Maron, G., Lam, M., Chitlangia, S., Krishnan, S., Reddi, V. J., and Wan, Z. (2022). QuaRL: Quantization for fast and environmentally sustainable reinforcement learning. *Transactions on Machine Learning Research (TMLR) 2022*.

[31] Foerster, J. N., Song, H. F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M. M., and Bowling, M. (2019). Bayesian action decoder for deep multi-agent reinforcement learning. In *ICML 2019: Proceedings of the Thirty-Sixth International Conference on Machine Learning*.

[32] Galam, S. and Javarone, M. A. (2016). Modeling radicalization phenomena in heterogeneous populations. *Plos one*, 11(5):e0155407.

[33] García, E., Villar, J. R., Tan, Q., Sedano, J., and Chira, C. (2023). An efficient multi-robot path planning solution using A* and coevolutionary algorithms. *Integrated Computer-Aided Engineering*, 30:41–52.

[34] Goyal, A., Lu, W., and Lakshmanan, L. V. (2011). Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th International*

Conference Companion on World Wide Web, WWW '11, page 47–48, New York, NY, USA. Association for Computing Machinery.

[35] Gupta, J. K., Egorov, M., and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer.

[36] Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

[37] Hasselt, H. v., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double Q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press.

[38] Hevapathige, A., Wang, Q., and N. Zehmakan, A. (2025). DeepSN: A sheaf neural framework for influence maximization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):17177–17185.

[39] Jiang, Q., Song, G., Cong, G., Wang, Y., Si, W., and Xie, K. (2011). Simulated annealing based influence maximization in social networks. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, page 127–132. AAAI Press.

[40] Kamthe, S. and Deisenroth, M. (2018). Data-efficient reinforcement learning with probabilistic model predictive control. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1701–1710. PMLR.

[41] Kempe, D., Kleinberg, J., and Tardos, É. (2003). Maximizing the spread of influence through a social network. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137-146.

[42] Kiadi, M., García, E., Villar, J. R., and Tan, Q. (2023). A*-based co-evolutionary approach for multi-robot path planning with collision avoidance. *Cybernetics and Systems*, 54(3):339–354.

[43] Kim, C., Lee, S., Park, S., and Lee, S.-g. (2013). Influence maximization algorithm using Markov clustering. In Hong, B., Meng, X., Chen, L., Winiwarter, W., and Song, W., editors, *Database Systems for Advanced Applications*, pages 112–126, Berlin, Heidelberg. Springer Berlin Heidelberg.

[44] Kopacz, A. (2023). Optimizing reinforcement learning based multi-robot path planning. SusTrainable Summer School 2023, 10-14 July, 2023, University of Coimbra, Portugal, https://sustrainable.dei.uc.pt/#students. Accessed: 27.01.2026.

[45] Kopacz, A. (2024). Competitive influence maximization in trust-based social networks with deep Q-learning. *Studia Universitatis Babeș-Bolyai Informatica*, 69(1):57–69.

[46] Kopacz, A. and Chira, C. (2026). Polarity related influence maximization through multi-agent reinforcement learning. In *Proceedings of the 18th International Conference on Agents and Artificial Intelligence - Volume 5: ICAART*, pages 4522–4529. INSTICC, SciTePress.

[47] Kopacz, A., Csató, L., and Chira, C. (2023a). Applying deep Q-learning for multi-agent cooperative-competitive environments. In García Bringas, P., Pérez García, H., Martinez-de Pison, F. J., Villar Flecha, J. R., Troncoso Lora, A., de la Cal, E. A., Herrero, Á., Martínez Álvarez, F., Psaila, G., Quintián, H., and Corchado Rodriguez, E. S., editors, *17th Int. Conf. on Soft Computing Models in Industrial and Environmental Applications (SOCO 2022)*, pages 626–634. Springer Nature Switzerland.

[48] Kopacz, A., Csató, L., and Chira, C. (2023b). Evaluating cooperative-competitive dynamics with deep Q-learning. *Neurocomputing*, 550:126507.

[49] Kopacz, A., García González, E., Chira, C., and Villar, J. R. (in review). An efficient method for multi-agent path planning using reinforcement learning. *Proceedings of SusTrainable Summer School 2023, submitted on 1. Mar. 2024.*

[50] Kopacz, A., García González, E., Chira, C., and Villar, J. R. (submitted). Imitation learning via deep q-learning for multi-robot path planning. In *21st International Conference on Hybrid Artificial Intelligent Systems (HAIS 2026), submitted on 13. Mar. 2026.*

[51] Kopacz, A., García González, E., Chira, C., and Villar Flecha, J. R. (2025). Hybrid

adaptive greedy algorithm addressing the multi-robot path planning problem. *IEEE Latin America Transactions*, 23(10):856–864.

[52] Kopacz, A., Mester, Á., Kolumbán, S., and Csató, L. (2022). Standardized feature extraction from pairwise conflicts applied to the train rescheduling problem. *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, pages 000103–000108.

[53] Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative Q-learning for offline reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc.

[54] Laporte, G. and Nobert, Y. (1980). A cutting planes algorithm for the m-salesmen problem. *Journal of the Operational Research society*, 31(11):1017–1023.

[55] Leibo, J. Z., nez Guzmán, E. D., Vezhnevets, A. S., Agapiou, J. P., Sunehag, P., Koster, R., Matyas, J., Beattie, C., Mordatch, I., and Graepel, T. (2021). Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International Conference on Machine Learning*, pages 6187–6199. PMLR.

[56] Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., and Legg, S. (2017). Ai safety gridworlds. *ArXiv*, abs/1711.09883.

[57] Leskovec, J., Huttenlocher, D. P., and Kleinberg, J. M. (2010). Signed networks in social media. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[58] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 420–429, New York, NY, USA. Association for Computing Machinery.

[59] Li, D., Xu, Z., Chakraborty, N., Gupta, A., Sycara, K. P., and Li, S. (2014). Polarity related influence maximization in signed social networks. *PLoS ONE*, 9.

[60] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N. M. O., z, T. E., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv: Learning*.

[61] Lin, M., Li, W., and Lu, S. (2020). Balanced Influence Maximization in Attributed Social Network Based on Sampling. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 375–383. Association for Computing Machinery, New York, NY, USA.

[62] Lin, S.-C., Lin, S.-D., and Chen, M.-S. (2015). A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 695–704, New York, NY, USA. Association for Computing Machinery.

[63] Liu, G., Iloglu, S., Caldara, M., Durham, J. W., and Zavlanos, M. M. (2025). Distributionally robust multi-agent reinforcement learning for dynamic chute mapping. In *Forty-second International Conference on Machine Learning*.

[64] Liu, S., Lever, G., Merel, J., Tunyasuvunakool, S., Heess, N., and Graepel, T. (2019a). Emergent coordination through competition. *7th Int. Conference on Learning Representations, ICLR 2019*.

[65] Liu, W., Chen, X., Jeon, B., Chen, L., and Chen, B. (2019b). Influence maximization on signed networks under independent cascade model. *Applied Intelligence*, 49(3):912–928.

[66] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. (2017). Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*.

[67] Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. (2019). Maven: Multi-agent variational exploration. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

[68] Malik, W., Rathinam, S., and Darbha, S. (2007). An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem. *Operations Research Letters*, 35(6):747–753.

[69] Mei, Y., Li, S., Chen, C., and Han, A. (2021). A multi-robot task allocation and path planning method for warehouse system. In *2021 40th Chinese Control Conference (CCC)*, pages 1911–1916.

[70] Mirowski, P. W., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., and Hadsell, R. (2017). Learning to navigate in complex environments. *ArXiv*, abs/1611.03673.

[71] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

[72] Mohanty, S., Nygren, E., Laurent, F., Schneider, M., Scheller, C., Bhattacharya, N., Watson, J., Egli, A., Eichenberger, C., Baumberger, C., Vienken, G., Sturm, I., Sartoretti, G., and Spigler, G. (2020). Flatland-RL: Multi-agent reinforcement learning on trains.

[73] Mordatch, I. and Abbeel, P. (2018). Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[74] Morgan, N. and Bourlard, H. (1989). Generalization and parameter estimation in feedforward nets: Some experiments. *Advances in neural information processing systems*, 2.

[75] Nazari, M., Oroojlooy, A., Snyder, L. V., and Takác, M. (2018). Reinforcement learning for solving the vehicle routing problem. In *Neural Information Processing Systems*.

[76] Nosratabadi, S., Mosavi, A., Keivani, R., Ardabili, S., and Aram, F. (2020). State of the art survey of deep learning and machine learning models for smart cities and urban sustainability. In Várkonyi-Kóczy, A. R., editor, *Engineering for Sustainable Future*, pages 228–238, Cham. Springer International Publishing.

[77] Papoudakis, G., Christianos, F., Schäfer, L., and Albrecht, S. V. (2021). Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*.

[78] Park, J., Kwon, C., and Park, J. (2023). Learn to solve the min-max multiple traveling salesmen problem with reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 878–886, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[79] Persello, C., Wegner, J. D., Hänsch, R., Tuia, D., Ghamisi, P., Koeva, M., and Camps-Valls, G. (2022). Deep learning and earth observation to support the sustainable development goals: Current approaches, open challenges, and future opportunities. *IEEE Geoscience and Remote Sensing Magazine*, 10(2):172–200.

[80] Qu, H., Xing, K., and Alexander, T. (2013). An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120:509–517. Image Feature Detection and Description.

[81] Rashid, T., Samvelyan, M., Witt, C. S. D., Farquhar, G., Foerster, J. N., and Whiteson, S. (2020). Monotonic value function factorisation for deep multi-agent reinforcement learning. *J. Mach. Learn. Res.*, 21:178:1–178:51.

[82] Samvelyan, M., Rashid, T., Schroeder de Witt, C., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. (2019). The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, page 2186–2188, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[83] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

[84] Schneider, M., Stenger, A., and Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520.

[85] Schwartz, R., Dodge, J., Smith, N. A., and Etzioni, O. (2020). Green AI. *Communications of The ACM*.

[86] Silver, D. (2005). Cooperative pathfinding. In *Proceedings of the First AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, AIIDE'05, pages 117–122. AAAI Press.

[87] Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. (2019). QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In

Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5887–5896. PMLR.

[88] Spliet, R., Gabor, A. F., and Dekker, R. (2014). The vehicle rescheduling problem. *Computers & Operations Research*, 43:129–136.

[89] Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., and Graepel, T. (2018). Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proc. of the 17th Int. Conf. on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2085–2087.

[90] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. The MIT Press, second edition.

[91] Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *In Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337. Morgan Kaufmann.

[92] Terry, J. K., Black, B., Grammel, N., Jayakumar, M., Hari, A., Sulivan, R., Santos, L., Perez, R., Horsch, C., Dieffendahl, C., Williams, N. L., Lokesh, Y., Sullivan, R., and Ravi, P. (2020). PettingZoo: Gym for multi-agent reinforcement learning. *arXiv preprint arXiv:2009.14471*.

[93] Törnquist, J. (2012). Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Research Part C: Emerging Technologies*, 20:62–78.

[94] Törnquist, J. and Persson, J. (2005). Train traffic deviation handling using tabu search and simulated annealing. *Proceedings of the 38th Hawaii International Conference on System Sciences*, pages 1–10.

[95] Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J. P., Schrittwieser, J., Quan, J., Gaffney, S., Petersen, S., Simonyan, K., Schaul, T., Hasselt, H. V., Silver, D., Lillicrap, T. P., Calderone, K., Keet, P.,

Brunasso, A., Lawrence, D., Ekermo, A., Repp, J., and Tsing, R. (2017). Starcraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:abs/1708.04782*.

[96] Wagner, G. and Choset, H. (2011). M*: A complete multirobot path planning algorithm with performance bounds. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3260–3267.

[97] Wang, D., Deng, H., and Pan, Z. (2020). MRCDRL: Multi-robot coordination with deep reinforcement learning. *Neurocomputing*, 406:68–76.

[98] Wang, J., Cao, Z., Xie, C., Li, Y., Liu, J., and Gao, Z. (2025a). DGN: influence maximization based on deep reinforcement learning. *Journal of Supercomputing*, 81(1).

[99] Wang, L., Wang, Z., Hu, S., and Liu, L. (2013). Ant colony optimization for task allocation in multi-agent systems. *China Communications*, 10(3):125–132.

[100] Wang, S., Liu, W., Chen, L., and Zong, S. (2025b). Influence maximization based on discrete particle swarm optimization on multilayer network. *Information Systems*, 127:102466.

[101] Wang, W. and Street, W. N. (2018). Modeling and maximizing influence diffusion in social networks for viral marketing. *Applied Network Science*, 3(1):6.

[102] Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., and De Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1995–2003. JMLR.org.

[103] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):9.

[104] Xu, Y., Kopacz, A., and Chira, C. (2025). A hybrid granular ball-ant colony optimization for the multi-depot half-open time-dependent electric vehicle routing problem. In *2025 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE.

[105] Xu, Y., Kopacz, A., and Chira, C. (accepted). A granular ball–ant colony optimization framework enhanced by variable neighborhood search for the multi-depot half-open time-

dependent EVRP. In *The Genetic and Evolutionary Computation Conference (GECCO-2026), accepted on 20. Mar. 2026.*

[106] Yang, T., Zhao, L., Li, W., and Zomaya, A. Y. (2020). Reinforcement learning in sustainable energy and electric systems: a survey. *Annual Reviews in Control*, 49:145–163.

[107] Yao, S., Fan, N., and Hu, J. (2022). Modeling the spread of infectious diseases through influence maximization. *Optimization Letters*, 16(5):1563–1586.

[108] Yeh, C., Li, V., Datta, R., Arroyo, J., Zhang, C., Chen, Y., Hosseini, M., Golmohammadi, A., Shi, Y., Yue, Y., and Wierman, A. (2023). SustainGym: Reinforcement learning environments for sustainable energy systems. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[109] Yin, X., Hu, X., Chen, Y., Yuan, X., and Li, B. (2021). Signed-PageRank: An efficient influence maximization framework for signed social networks. *IEEE Transactions on Knowledge and Data Engineering*, 33(5):2208–2222.

[110] Zare, M., Kebria, P. M., Khosravi, A., and Nahavandi, S. (2024). A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 54(12):7173–7186.

[111] Zhang, Y., wei Gong, D., and hua Zhang, J. (2013). Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing*, 103:172–185.

[112] Zheng, L., Yang, J., Cai, H., Zhou, M., Zhang, W., Wang, J., and Yu, Y. (2018). MAgent: A many-agent reinforcement learning platform for artificial collective intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

[113] Zheng, Y., Luo, Q., Wang, H., Wang, C., Chen, X., Maseleno, A., Yuan, X., and Balas, V. E. (2020). Path planning of mobile robot based on adaptive ant colony algorithm. *J. Intell. Fuzzy Syst.*, 39(4):5329–5338.

[114] Zhu, W., Zhang, K., Zhong, J., Hou, C., and Ji, J. (2025). BiGDN: An end-to-end influence maximization framework based on deep reinforcement learning and graph neural networks. *Expert Systems with Applications*, 270:126384.