BABEŞ-BOLYAI UNIVERSITY, CLUJ-NAPOCA, ROMÂNIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Deep reinforcement learning methods for search and optimization in multi-agent systems

PhD Thesis Summary

Student:
Anikó Kopacz

Scientific supervisor:
Prof. Dr. Camelia Chira

2026

Abstract

Networks are suitable for modelling systems from real-life applications featuring non-trivial relationships, such as social networks, citation networks and transportation systems. Several optimization problems, where the nodes of the analyzed networks have the ability to interact with each other or their environment, can be modeled as multi-agent environments. Due to the complexity of such systems arises the interest for flexible and robust solutions. Thus, machine learning approaches – such as reinforcement learning – are often prioritized to address optimization problems in multi-agent systems.

In this thesis deep reinforcement learning based algorithms are proposed, and their effectiveness is explored within the context of various multi-agent scenarios. First, competitive-cooperative scenarios that feature predator-prey simulations are being explored. Decentralized and centralized deep reinforcement learning models are developed to navigate cooperative-competitive environments.

Next, vehicle route problem variants are addressed, namely the vehicle rescheduling problem and the multi-robot path planning problem. A hybrid greedy approach is proposed for the multi-robot path planning surpassing other heuristic and meta-heuristic solutions. Moreover, a deep Q-learning model trained with independent learning is introduced to address multi-robot path planning with reinforcement learning. Then, the deep Q-learning architecture is combined with imitation learning from demonstrations generated with the hybrid greedy approach.

Lastly, the influence maximization on social networks is addressed with deep-reinforcement learning. A reinforcement learning framework is presented which tackles the polarity-related influence maximization with two actors present in a social network. To address polarity-related influence maximization in signed social networks, a multi-agent deep Q-learning model is developed. This model applies Louvain community detection to decompose the original social network and optimize seed selection in the communities using double deep Q-networks.

Contents

Chapter 1

Introduction

Several complex optimization problems can be formalized using multi-agent systems [9] and solved using one of the following strategies (i) decomposing the original problem into a set of independent, computationally easier tasks, (ii) solving the low-level tasks separately, (iii) combining the obtained solutions of the tasks into a global decision. Solving the computationally less complex tasks becomes the responsibility of autonomous actors referred to as *agents*. In real-world optimization problems, multiple types of agent dynamics exist. Cooperative settings imply that a set of actors are expected to work together pursuing a common objective. On the other hand, agent configurations with at least two opposing goals are labeled as competitive scenarios. Various computational methods are based on multi-agent systems and employ agent interactions for optimization.

Reinforcement learning (RL) is a well-known computational approach to learn a specific task based on the interaction of one or more agents with the environment [90]. The completion of the predefined task is associated with a positive reward, meanwhile unfavorable scenarios are characterized by a negative reward or penalty. The goal of the learner is to construct policies that ensure achieving a high cumulative reward. Reinforcement learning has been applied to several real-world inspired optimization problems, such as robotics, epidemiology, and others. Moreover, multi-agent reinforcement learning was adapted for addressing various real-world problems, such as emergency traffic control [20], chute mapping in warehouses [63], navigation [77] and various cooperative game play scenarios [55, 95]. Various data analysis approaches can be adapted to operate on multi-agent systems, however computational trade-offs are crucial to be considered. Introducing several actors to a non-trivial system can significantly

increase the demand for computational resources, rendering some methods infeasible in practice. Thus, heuristic approaches might get favored over exact solutions in the case of NP-hard problems regarding multi-agent systems, e.g., [25, 94, 99].

1.1 Research objectives

The research objective is to tackle optimization problems, that combine multi-agent systems and networks, with deep reinforcement learning approaches. To achieve the main research objective, several objectives are introduced, which decompose the overall goal into specific tasks that guide the methodological development, experimental evaluation, and validation of the proposed approaches.

The first objective of this thesis addressed in Chapter 3 is to develop and evaluate multi-agent reinforcement learning [2] frameworks able to navigate in cooperative-competitive environments. The effectiveness of centralized and decentralized deep reinforcement learning methods was set out to be investigated. To attain this objective, a scientific goal was to perform an adequate literature review that summarizes reinforcement learning approaches suitable for optimization problems for multi-agent systems. Based on the analysis of the literature, the main focus was building on and adapting state-of-the-art methods to specific optimization tasks. The performance of centralized and decentralized deep reinforcement learning was compared on well-known benchmarks that simulate predator-prey dynamics. Multi-agent variational exploration [67] was adapted with the scope of improving the performance of centralized training for deep reinforcement learning.

The second objective of this thesis is addressing multi-agent path planning with deep reinforcement learning approaches, which is the topic of Chapter 4. Initially, the problem of vehicle rescheduling was investigated with the goal of proposing and evaluating domain-specific feature extraction mechanisms. Computational experiments were designed to illustrate that a meaningful scheduling algorithm can be learned with deep Q-networks based on the proposed feature space. To underline the robust nature of deep reinforcement learning methods, another variant of the route planning problem, path planning in multi-robot systems, was analyzed. With the goal of tackling multi-robot path planning, hybrid- and deep reinforcement learning optimization frameworks were developed and tested. Then, building on the proposed hybrid ap-

proach, an expert policy was created for training a decentralized reinforcement learning policy with imitation learning.

The third objective of the present thesis is to address the influence maximization problem on signed social networks with deep reinforcement learning methods. This objective is the subject of Chapter 5. Thus, the next scientific goal of this thesis was developing a deep reinforcement learning framework for the competitive influence maximization problem on signed social networks. The performance of the proposed seed selection process was evaluated on small- and medium-scale social networks with trust-distrust relationships. The constructed reinforcement learning framework was further refined, and a community-based multi-agent reinforcement learning architecture was introduced to tackle polarity-related influence maximization in signed social networks. Experiments were designed to evaluate community-based multi-agent reinforcement learning on several large-scale social networks, which have positive and negative edges.

1.2 Original contributions

The present doctoral thesis proposes several original contributions listed below along with the relevant publications on the topic.

1. *Standardized feature extraction from pairwise train scheduling conflicts.* A reinforcement learning framework is introduced for train rescheduling problem which applies a standardized feature selection based on pairwise conflicts. Based on an analytical solution which detects and resolves every conflict arising between two trains, a corresponding observation space is proposed. The data obtained from the pairwise conflicts translates to actions in the context of the reinforcement learning framework. The performance of the proposed feature extraction mechanism is evaluating using the metrics defined in the Flatland Challenge [72]. The proposed framework and computational experiments are presented in a conference article [52].

2. *Deep reinforcement learning in cooperative-competitive environments.* Multi-agent reinforcement learning based models are constructed to optimize actions of agents operating in cooperative-competitive multi-agent environments. Two well-known reinforcement

learning approaches, namely Deep Q-network (DQN) introduced in [71] and monotonic value function factorization for deep reinforcement learning (QMIX) proposed in [81], are adapted and evaluated on the MAgent library [112] featuring predator-prey type of dynamics. The applied methodology and computational results of the proposed methods are presented in [47].

3. *Centralized and decentralized deep Q-learning for cooperative-competitive environments.* To evaluate the effectiveness of centralized and decentralized reinforcement learning, the several deep Q-network based architectures are integrated to navigate in cooperative-competitive environments: (i) independent learning with DQN, (ii) QMIX, (iii) and multi--agent variational exploration [67]. The experiments highlight that decentralized training of deep Q-networks accumulates higher episode rewards during training and evaluation in comparison with the selected centralized learning approaches. The methodology and the results are published as a journal article [48].

4. *Hybrid adaptive greedy algorithm for multi-robot path planning.* A hybrid approach named H* is developed combining A* based path planning and local search to address the multi-robot path planning problem. The A* search method determines the optimal path for each robot, the heuristic component extends the solution to the multi-robot setting. The method is evaluated in several environments – a room with 10x30 cells, four 10x30 cells rooms interconnected pairwise, and a realistic warehouse. Computational experiments show that H* finds shorter paths than A*-based heuristics and coevolutionary methods, particularly on smaller maps. The H* method and computational experiments are published in a journal paper [51].

5. *A reinforcement learning model for multi-robot path planning.* A multi-agent framework is proposed based on Dueling Double DQN [37, 102] model for multi-robot path planning in a shared two-dimensional environment. A decentralized controller is implemented to improve efficiency by sharing the policy and model parameters between agents. Experimental results show that the trained policy effectively learns to avoid collisions, and that the choice of an adequate candidate solution space is crucial to ensure the effectiveness of training. The computational experiments validating the proposed multi-robot framework have been presented at the SusTrainable Summer School 2023 [44] and are submitted to

be published in a LNCS volume [49].

6. *Multi-robot path planning through imitation learning with deep Q-networks.* A novel method is proposed combining deep reinforcement learning with imitation learning from an expert policy for the multi-robot path planning problem. The expert policy providing the baseline for the reinforcement learning policy builds on the H* approach [51] and on the analytical conflict resolution presented in [52]. The developed framework and the experimental results were submitted for publication to the International Conference on Hybrid Artificial Intelligent Systems (HAIS 2026) [50].

7. *Joint- and iterative seed selection with deep Q-learning for competitive influence maximization.* Two distinct mechanisms are introduced to construct initial seed sets for the competitive influence maximization: joint- and iterative seed selection. DQN is trained to choose seeds to maximize information spread in trust-based social networks with two competing actors present. The effectiveness of appointed seed sets is analyzed based on the number of activated nodes after simulating polarity-related independent cascade on trust-distrust networks. The proposed seed selection mechanism with deep Q-networks are published in [45].

8. *Community-based multi-agent reinforcement learning for polarity-related influence maximization.* A community-based reinforcement learning method is proposed to address polarity-related influence maximization. A reinforcement learning agent is assigned to each community identified by Louvain community detection algorithm [7]. The seed node selection in the multi-agent setting is optimized by training a decentralized Double DQN [37], where the agents share the neural network models. The proposed approach and the experimental results are presented in [46].

Chapter 2

State of the art in reinforcement learning

Reinforcement learning (RL) utilizes previous experiences of one or more agents interacting with their environment to acquire desired skill sets or knowledge to solve predefined tasks [90].

2.1 Concepts and definitions

In accordance with Sutton and Barto [90], reinforcement learning is defined as follows. The agent is considered an entity that observes internal and external factors, its environment, and decides which available actions to execute at any given moment. The environment consists of the external elements which are not considered the agent. The formalization of the agent-environment interaction characteristic of the RL setting is a Markov Decision Process (MDP) [90]. The $\{\mathcal{S}, \mathcal{A}, p, \mathcal{R}, \gamma\}$ tuple is an MDP if the Markov property holds true, which states that the immediate reward r and the agent's next state s_{i+1} is defined by the previous state s_t and the a_t action taken:

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

The policy π is to the likelihood of selecting an action a given the state s. Selecting the next action in a given s and π is the same as finding an a action that maximizes $\pi(a \mid s)$.

Given a time step t and a state s, $V_t^{\pi}(s)$ denotes the expected value of the accumulated reward for state s following policy π. $Q^{\pi}(s, a)$ denotes the pay-off generated by taking action

a for state s, when following policy π.

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

The advantage $A_t(s,a)$ of action a at time step t given state s is:

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

thus, the value of the advantage can be calculated knowing the value of the states s_t and s_{t+1}.

Let $V_t(s)$ denote the estimate of $V_t^\pi(s)$. The temporal difference error (TD) is the difference between consecutive predictions [90]:

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

where R_{t+1} is the reward received in the $t+1$ time step, while α is a parameter controlling the step size. In practice, the TD error is often optimized instead of the advantage function [90].

Imitation learning consists of learning to perform a task from expert demonstrations rather than relying solely on agent-environment interactions to discover advantageous strategies [110].

2.2 Deep Q-learning

Mnih et al. introduced Deep Q-Networks (DQN) in [71], a modified Q-learning method for combined with a deep learning architecture in order to determine the optimal policy of a single agent in the reinforcement learning setting. The optimal action at a given state is considered to be the one with the maximum Q-value.

In case of the DQN method the action-value function is modeled by a deep neural network. During the simulation of different episodes, the method of experience replay is utilized. The generated trajectories of the agent are saved into a memory buffer, which is later sampled to construct training batches as inputs for the neural network responsible for Q-value estimation.

The neural network aims to minimize the squared sum of the TD error (see Equation 2.4). The TD error is computed using a second neural network, called target network and its parameters are θ_{target}, which is a periodical copy of the online network.

DQN was shown to be overly optimistic [37] and Double deep Q-learning [37] was introduced to address the over-estimation problem. Dueling deep Q-learning [102] estimates the state-value function and action-advantage function using dueling networks: the same feature representation is used as an input for the value and for advantage networks.

Conservative Q-Learning (CQL) introduced in [53] is a technique designated offline Q-learning where the model learns from a fixed dataset of pre-collected experiences without any opportunity to interact with the environment.

2.3 Multi-agent reinforcement learning approaches

Multi-agent reinforcement learning (MARL) consists of a set of various autonomous learners based on agent-environment dynamics characteristic of single-agent reinforcement learning. Several distinct ways exist to adopt the success of single-agent learners in multi-agent scenarios. Fully decentralized methods, such as independent Q-learning [91], treat agents as separate entities attempting to optimize the agents' actions separately. The agents share experiences amongst each other, and a single policy is trained to determine the optimal actions.

2.3.1 Centralized and decentralized training

Approaches that optimize multi-agent scenarios can be split into two categories regarding the manner in which they manage the available information: centralized and decentralized methods. Centralized approaches tend to integrate a global view of the environment, where the selection of agent actions is optimized jointly. In the decentralized setting agents operate in a parallel manner, and agents optimize their actions separately.

2.3.2 Value decomposition networks

Value decomposition networks [89] incorporate both centralized and decentralized techniques. The network employs a global reward as feedback of the joint actions of the agents and the TD error for the joint action-value function is utilized as a loss function during training. The global reward signal is then decomposed, to train a separate neural network for each agent. Value decomposition networks benefit from centralized learning and decentralized execution.

Monotonic value function factorization for deep reinforcement learning (QMIX) proposed in [81] applies a compound structure of neural networks to approximate the joint action-value function Q_{total}: (i) agent neural networks, that approximate the Q_a function and (ii) a mixing network that synthesizes the observation-action histories and action pairs.

Mahajan et al. [67] showed that the monotonic assumption required by QMIX discourages exploration, and proposed multi-agent variational exploration (MAVEN) to prioritize policy exploration in centralized training-decentralized execution methods.

2.4 Benchmarks for reinforcement learning methods

Multiple benchmarks were proposed to evaluate the performance reinforcement learning approaches. In [56], Leike et al. evaluate RL algorithms on safety problems faced by autonomous agents, such as interrupting signals, and avoiding side effects. Chen et al. [16] implements a deep multi-agent reinforcement learning system to propose solutions for the mixed-traffic highway on-ramp merging problem. In [95] a benchmark is proposed to evaluate the performance of multi-agent learning methods on the game of StarCraft II. MineDojo [29] is a framework built on the Minecraft simulator, featuring a benchmarking suite with thousands of open-ended programmatic and creative tasks.

2.5 Sustainability and reinforcement learning

Several contributions investigate the potential of machine learning to achieve sustainable objectives, e.g., [76, 79, 106, 108]. The topic of sustainability of machine learning approaches was also investigated, see, for example, [1, 85].

Various contributions from the literature propose techniques to manage training efficacy. For example, [24] showed that sparse neural networks achieve the performance of dense networks requiring less training time. Early stopping [74] terminates training once the metrics reporting model performance begin to deteriorate. The efficiency of reinforcement learning methods can be improved by applying probabilistic transitions to incorporate uncertainty into modeling and planning [40]. Quantization methods have been proven to successfully reduce training time and carbon emissions of distributed reinforcement learning [30].

Chapter 3

Reinforcement learning algorithms for cooperative-competitive environments

Deep reinforcement learning (RL) based architectures are proposed to tackle navigating in cooperative-competitive environments that model predator-prey dynamics. A Deep Q-network (DQN) framework and a monotonic value function factorization (QMIX) based architecture is proposed to optimize action selection via decentralized training and execution. The proposed frameworks and experiments were presented in [47]. Then, improved versions of the previously mentioned DQN and QMIX approach are proposed. Moreover, the QMIX architecture is combined with multi-agent variational exploration (MAVEN) to improve the efficacy of centralized training. The extended version of the experiments and the improved methodology were published in [48].

3.1 Problem definition

Multi-agent planning systems have multiple real-world applications, such as fake news detection [3], tracking evolutionary phenomena [6], collision avoidance [12], and analyzing social networks [14]. Multi-agent environments, where a group of agents has a shared goal to achieve, are referred to as cooperative settings. *Cooperative-competitive environments* are formed by multiple cooperating groups having objectives opposing other agent groups.

Recent contributions from the literature focus on implementing various cooperative-competitive environments in order to provide benchmarks for multi-agent reinforcement learning methods,

see e.g., [95, 64, 92]. These platforms aim to facilitate the usage of multi-agent reinforcement learning approaches by providing a framework for environment simulation and evaluation.

The PettingZoo library integrates various multi-agent games that are suitable for modeling more complex multi-agent relationships of real-life problems, such as MAgent platform [112], discrete and continuous control tasks [35], Multi Particle Environments [73].

3.2 Related work

Prior research in cooperative-competitive multi-agent reinforcement learning has used benchmarks, such as StarCraft based environments [27, 82, 95] and MAgent[112]. QMIX was evaluated on the StarCraft environment and surpassed the performance of independent Q-learning and value decomposition networks, see [81]. MAVEN was evaluated on the SMAC [82] domain in [67], and performed better than independent Q-learning, QMIX, COMA [31] and QTRAN [87].

3.3 Deep reinforcement learning methods

In [47], we proposed decentralized and centralized deep RL frameworks to operate in cooperative-competitive environments. More specifically, two Q-learning methods were adapted: (i) Independent Learning [91] applied with DQN [71] method and (ii) a QMIX approach [81]. Firstly, an Independent Learning [91] approach is evaluated combined with Deep Q-Networks [71].

DQN applies Q-learning by using a neural network to approximate action values and choose the highest-value action. In case of Independent Learning, the multi-agent scenario is treated a single agent reinforcement learning problem, rendering other agent actions and objectives as a part of the environment. The learning process is decentralized, while the agents share experiences with each other.

QMIX trains cooperative agents with a global reward by learning per-agent utility networks and a monotonic mixing network that combines agents' Q-values. QMIX implements decentralized action selection and learns to jointly optimize agent actions.

We trained two separate policies for selecting actions: a separate policy is trained to select actions for predator- and prey-type of agents, respectively. The two separate policies are

initialized with the same parameters, however during the training phase the agents operate following their predefined policy. Thus, agents with similar goals share experiences, but they do not weigh in the optimization of the rewards attributed to adversarial agents.

Experimental results on the *adversarial pursuit* environment showed that deep reinforcement learning approaches are able to construct adaptive policies for navigating in predator-prey type of simulations.

3.4 Centralized and decentralized deep Q-learning

The effectiveness of centralized and decentralized algorithms were evaluated concentrating on three variants of deep Q-networks featuring cooperative-competitive environments: (i) decentralized training independent learning [91] with DQN [71], (ii) the centralized monotonic value factorizations for deep learning (QMIX) [81], (iii) the multi-agent variational exploration[67]. The computational experiments presented in Section 3.3 were extended by considering new parameter settings for the DQN and QMIX methods, as well as, proposing an additional RL method, which combines centralized Q-learning with multi-agent variational exploration [67], and using an another multi-agent environment from [66] that simulates predator-prey dynamics.

Moreover, to assess the effectiveness of learned policies, policy evaluation is performed after training. The learned policies are evaluated in a deterministic manner, thus random action selection for agent is avoided all together. Separate action selection policies are constructed for different agent groupings, in comparison with the applied method from Section 3.3, the predator agent groups are no longer split to sub-groups to balance the agent number for policy training.

The experiments were performed using the following RL environments that are also included in the PettingZoo [92] framework: (i) *Adversarial pursuit* from the MAgent library [112] and (ii) *Simple tag* from the Multi-Particle Environments (MPE [66]) library. Both *adversarial pursuit* and *simple tag* feature cooperative-competitive interactions.

Decentralized training with DQN consistently outperformed centralized approaches across both the *adversarial pursuit* and *simple tag* environments. DQN policies showed faster convergence and achieved higher accumulated rewards for both predator and prey agent groups during evaluation. While centralized methods like QMIX and MAVEN required longer training

periods to stabilize, the integration of MAVEN with QMIX showed improvements over standard QMIX, particularly in reducing variance and improving exploration in centralized training scenarios.

Chapter 4

Path planning solutions for multi-agent systems

Hybrid- and deep reinforcement learning (RL) frameworks were developed to optimize path planning in multi-agent systems. Firstly, a standardized feature extraction framework coupled with deep Q-learning was developed [52]. A hybrid greedy approach, H*, was introduced for planning robot actions in two-dimensional multi-robot environments[51]. Then, a RL environment is defined to model and optimize policies for navigating in two-dimensional multi-robot environments [49]. Lastly, decentralized RL based controller trained with imitation learning from an expert hybrid policy was constructed, which synthesizes the aforementioned RL environment with analytical conflict resolution and the H* algorithm [50].

4.1 Problem definition

Path planning optimizes the sequence of movements required to navigate an agent from an initial position to a target state while avoiding collisions with obstacles.

Path planning is a crucial element to several complex optimization problems, e.g., the Vehicle Routing Problem (VRP) as defined in [21].

4.1.1 The vehicle rescheduling problem

The Vehicle Rescheduling Problem (VRSP) extends the classical VRP by addressing the need to continuously adapt and replan vehicle routes in response to real-time disruptions, delays,

and malfunctions in transportation systems.

The primary objective is to minimize the sum of travel times across all agents from their current positions to their respective destinations, while accounting for the dynamic state of the system including possible delays, equipment malfunctions, and changing conditions.

4.1.2 Multi-robot path planning

The Multi-robot Path Planning (MPP) problem involves the scenario of robots moving through a shared space. The goal of MPP is to strategize the paths that all robots must take in the shared environment to reach a specific destination, ensuring there are no collisions with other robots or obstacles in the surroundings.

The MPP problem consists of two computational tasks that need to synchronize: (i) determining the individual paths for each robot in a given environment and (ii) coordinating the actions of the robots to minimize collisions or delays.

4.2 Related work

VRP extends the Traveling Salesman problem aiming to minimize total travel distance or expenses. Each customer must be visited exactly once, and the aggregate demand on any route must not exceed the assigned vehicle's capacity.

VRP is related to the Multiple Traveling Salesman Problem (MTSP), a solving the MTSP also addresses VRP [88]. The MTSP has been tackled by various methods, including exact methods [54], approximation algorithms [23, 68], and machine learning approaches [78].

Single and multi-agent environments were proposed to facilitate apply RL approaches for various traffic coordination tasks [13, 70, 72]. Deep RL has emerged as a promising direction, with methods that learn policies to address the vehicle route planning problem. In [75], the authors propose a recurrent neural network decoder coupled with an attention mechanism based policy to address VRP and its variants.

The VRSP problem is NP-hard [88], the proposed approaches to its solution are diverse and many of them rely on heuristics, most commonly used techniques include depth-first-search [93], branch and bound [22], genetic algorithms [26], and simulated annealing [94].

A* [36] based heuristics were proposed to address MPP, e.g., cooperative A* [86], M* [96].

MPP was also addressed using meta-heuristics including genetic algorithms [80], particle swarm optimization [111], ant colony optimization [113].

In [70], robots were trained to navigate using deep reinforcement learning, and several improvements were introduced since. Deep reinforcement learning was applied to optimize route planning and collision avoidance tasks for the cooperative multi-robot setting [97].

The Electric Vehicle Routing Problem (EVRP) extends the classical VRP [84]. Meta-heuristic methods, such as hybridized ant colony optimization methods [28, 104, 105], have been successful in solving the EVRP problem.

4.3 Learning from analytical conflict resolution based observations

In [52], the VRSP formalized by the Flatland Challenge [72] was tackled by employing a deep RL approach with an analytical conflict resolution mechanism. A new feature extraction method by analyzing pairwise conflicts was introduced, followed by a deep Q-network, which plays the role of the decentralized controller. The proposed approach was evaluated on a series of simulations, in accordance with the Flatland benchmark [72].

Our approach to the VRSP formalized by the Flatland Challenge relies on the identification, characterization and, if possible, resolution of every conflict arising between *train pairs*. Custom observations for the RL setting are calculated based on successively applying the H-shaped conflict resolution model and a voting mechanism. These observations then are used as input for the neural network incorporated by the reinforcement learning model. Q-learning was applied for approximating the common action-value function of the agents. DQN [71] and linear Q-network (LQN) architectures were applied for testing the proposed framework.

The experimental results illustrate that a meaningful scheduling algorithm can be learned based on the proposed feature space.

4.4 Hybrid approach for multi-robot path planning

In [51], we proposed the H* algorithm that addresses the MPP problem by identifying collision-free routes for multi-robot scenario. The H* algorithm consists of two stages. In the first stage,

greedy path search is applied using a heuristic approach to generate routes for robots given a maze, knowing the initial and target positions of robots. In the second stage, a local search operator is applied to refine the routes determined by the greedy search, which aims to minimize the two objectives of length and collisions.

Computational experiments are performed for several scenarios that consider different characteristics and impose various levels of difficulty for planning methods. Three distinct multi-robot map layouts were selected to assess the effectiveness of the H* hybrid method in the context of the MPP problem: Scenario 1 [42], Scenario 2 [33], Scenario 3 [69].

Experimental results showed that the H* algorithm is robust w.r.t. the number of robots and successfully solves all tested configurations. H* had similar performance or outperformed other heuristic and meta-heuristic approaches from the literature.

4.5 A reinforcement learning approach for multi-robot path planning

Independent Learning [91] is combined with Dueling Deep Q-Networks [102] and Double Deep Q-learning [37] to address the multi-robot path-planning problem [44, 49]. Using the RL setup, a decentralized controller is implemented: the robots share among each other policy parameters and memories of previous interactions with the environment. As a consequence of sharing experiences during training, the dimensionality of the solution space is reduced. Smaller solution spaces result in faster convergence and require fewer simulations to learn an effective policy. Moreover, decentralized solutions are agnostic regarding the number of robots, as long as the robots have the same agent-environment dynamics.

The performance of Dueling Double DQN based training was evaluated for Scenario 1 proposed by [42] depicting a two-dimensional environment.

Experiments showed an increase in run-time is associated with an increased power consumption. While the computational resources required for training with the two tested observation sizes are close to each other, using a smaller observation radius was the more energy-efficient option in our experiments.

4.6 Imitation learning with deep Q-networks

The RL framework described in Section 4.5 is, and this section presents a two-stage imitation learning based RL method. In the first stage, a centralized expert policy based on temporal look-ahead and conflict-resolution mechanisms generates demonstration trajectories. In the second stage, a decentralized policy is trained on the expert trajectories through a combination of imitation learning and offline RL.

4.6.1 The hybrid expert policy

The expert policy is a centralized controller based on the H* algorithm and the analytical conflict resolution proposed in [52] to solve multi-robot navigation tasks. The expert policy is used to generate an offline dataset that models successful navigation strategies of multiple robots operating in a shared environment. A centralized controller is constructed which functions as a hybrid system, combining deterministic graph search with heuristic conflict resolution.

4.6.2 Double Deep Q-learning model

An offline Double DQN approach is employed to optimize action selection for the robots. The acting policy is shared among agent, the Q-network is trained through independent Q-learning [91]. Soft updating [60] mechanism was applied to change the target network's weights during training. The RL system is trained using conservative Q-learning (CQL).

Experiments were conducted on Scenario 1 proposed by [42] to validate the applicability of the offline imitation learning approach. The results of the computational experiments indicate that the Double DQN model learns to navigate in the maze from the expert demonstrations.

Chapter 5

Reinforcement learning for influence maximization

In the last years, significant advances have been achieved regarding the influence maximization (IM) problem on social networks. The influence maximization [41] problem aims to maximize the information coverage, while minimizing the cost associated with the degree of information spread. IM has a wide range of real life applications including viral marketing [19, 101], epidemiology [107], political science [32].

Two frameworks using deep reinforcement learning were developed to address influence maximization problem on signed social networks. Joint- and iterative seed selections for the competitive influence maximization scenario in trust-distrust networks were proposed [45]. Deep Q-Networks were integrated to the aforementioned seed selection mechanisms to determine the seed sets for small and medium-scale social networks. The second model, which was published in [46], consists of a community-based multi-agent reinforcement learning framework providing scalable and adaptive strategies for large-scale signed social networks.

5.1 Problem definition

Influence maximization [41] targets the optimization of information spread in social networks starting from a set of source nodes. Let K denote the maximum number of nodes in the seed set, also called the *budget*. In [41] the authors also proposed a greedy baseline algorithm under the two main existing diffusion models, namely Independent Cascade and Linear Thresholds. The

influence maximization problem is NP-hard [41], therefore, in addition to approaches proposed specifically optimize influence maximization, various soft computing methods can be applied to alleviate the computational requirements e.g., reinforcement learning.

Bharathi et al. [5] discusses first-mover and second-mover strategies for identifying a seed set to maximize the influence diffusion, when competition is present in the social network.

Carnes et al. [15] discussed the followers (second-mover) perspective, the seed set is constructed knowing which seed nodes are activated by the opponent. [15] extended the independent cascade model to the competitive scenario introducing two influence spread mechanisms. Both models are suitable for constructing a seed set greedily to address the problem of competitive influence maximization (CIM).

The influence maximization problem formalized in [41] features social networks having a single type of relation between individuals. Polarity-related influence maximization (PRIM) operates taking into account two opposing type of relationships [59].

In competitive influence maximization, two opposing actors try to influence nodes in the same network, so each node can be inactive or activated. An inactive node has not been influenced by either actor, an activated node has a polarity (positive or negative) that indicates which actor influenced it.

5.2 Related work

[41] proposed a greedy algorithm that maximizes the influence by iteratively selecting the nodes that bring the maximum marginal influence. CELF [58] and CELF++ [34] improves the performance of this greedy algorithm by exploiting the sub-modular nature of the inlfluence diffusion.

In addition to greedy algorithms, meta-heuristic approaches addressing the IM problem: genetic algorithm [10], multi-objective evolutionary algorithm [11], simulated annealing[39], particle swarm optimization [100]. In [17] the authors proposed a reinforcement learning framework regarding influence maximization problem in random graphs.

Several existing algorithms, which address the IM problem, benefit from detecting communities and building various strategies to exploit information based on communities, e.g. [43, 8, 38].

In [62], a reinforcement learning framework was proposed to optimize seed selection for the competitive influence maximization. ToupleGDD [18], uses coupled graph neural networks to obtain node embeddings and applies double Deep Q-learning to estimate the expected influence. Recently, [114, 98] further improved the performance of graph neural network and reinforcement learning hybrids surpassing ToupleGDD.

Greedy-SIM [65] is a greedy method that optimizes the positive influence spread in signed networks by identifying independent propagation paths and their probabilities. In [109], the authors consider individual beliefs and information diffusion in signed networks, and apply the Signed-PageRank method for solving the IM problem in signed networks.

5.3 Competitive influence maximization in trust-distrust networks

In [45], a deep reinforcement learning based architectures were built to address competitive influence maximization on signed social networks. Two distinct mechanisms were analyzed to construct initial seed sets for two competing actors: joint- and iterative seed selection. The effectiveness of possible seed sets is compared based on the number of activated nodes after simulating polarity-related independent cascade on trust-distrust networks.

In case of joint seed selection, the seed set for the first agent producing positively activated nodes is selected as one action of the RL agent. The possible seed sets are obtained based on the social network infrastructure before the deep Q-learning takes place.

In case of iterative seed selection, the seed nodes are added one by one to the seed set. The reinforcement learning episode consists of a maximum k number of iterations, in each iteration the actors select an inactivated node to be added to their respective seed sets.

Experiments were conducted on trust and distrust based social networks constructed from user relations on the epinions.com consumer review site [57].

Two DQN [71] approaches were evaluated to select initial seed sets for the competitive influence maximization problem [15] in signed trust based social networks. Two distinct reinforcement learning models were applied to formalize the influence maximization problem and construct the solution space.

Experimental results show that deep reinforcement learning is suitable for proposing seed

sets for the competitive influence maximization problem on polarized networks.

The joint seed selection is feasible to determine seed sets for the competitive influence maximization problem. The joint seed selection method does not scale well due to the fact that the action space increases rapidly with the budget for the seed set. Experiments show that iterative seed selection can be utilized with the DQN approach to operate on medium-scale networks.

5.4 Multi-agent reinforcement learning for polarity-related influence maximization

This section presents the community-based multi-agent double DQN approach [46] for addressing the PRIM problem in social networks.

The RL *agent* constructs the seed set in an iterative manner, selecting one node with each action to be added to the seed set. The node activation and influence spread are part of the RL agent's environment. The seed nodes determined by the agent are activated as positive and influence spread is simulated following the polarity-related independent cascade .

A multi-agent environment is constructed by identifying communities, then, multi-agent deep Q-learning is applied to solve the PRIM optimization problem in signed social networks [46]. By selecting strategies for each community rather than the entire input network, the original problem is decomposed into smaller subproblems.

Using the Louvain community detection algorithm [7] on the directed graph ignoring the signs of the edges, the original graph is divided into several distinct subgraphs, grouping nodes into communities. Each community is then assigned an RL agent (actor), enabling decentralized decision-making for nominating seed nodes within the reinforcement learning framework.

Double DQN represents the RL agents' action selection policy, which is shared between the agents. Target networks are updated through soft updates [60] to stabilize learning.

Experiments were conducted on the Bitcoin OTC, WikiElec, and Slashdot signed social networks with trust- and distrust based relations between users.

The Louvain community detection algorithm successfully reduces the dimensionality of the original social networks by identifying strongly connected sub-graphs.

Although activated and negative nodes appear after simulating information diffusion, the

number of activated and positive nodes remains substantially higher than the number of negative and activated nodes showing that the proposed RL framework successfully incorporates the difference between positive and negative edges.

Chapter 6

Conclusions and future work

Reinforcement learning (RL) is a machine learning approach based on learning through agent-environment interaction. The present thesis proposes RL approaches to address optimization problems which concern multi-agent systems.

6.1 Main contributions and results

First, navigating in cooperative-competitive environments was addressed by proposing two deep reinforcement frameworks. Experiments conducted in the MAgent environment showed that deep Q-networks are capable of learning adaptive policies [47]. Then, centralized and decentralized Q-learning based architectures were proposed to address navigating in the cooperative-competitive setting [48]. Experiments showed that the performance of centralized training with monotonic value factorization can be improved by applying multi-agent variational exploration.

Secondly, route planning in multi-agent systems were addressed using machine learning: (i) a deep Q-learning approach based on analytical conflict resolution was proposed [52], (ii) a hybrid adaptive method was developed for tackling the multi-robot path planning, (iii) a decentralized deep Q-learning model was proposed for multi-robot path planning [44, 49], (iv) an imitation learning framework learning from hybrid policies was proposed [50].

Thirdly, the influence maximization problem was tackled using deep Q-learning. Two seed selection mechanisms based on deep Q-learning were proposed in [45] for competitive influence maximization on signed networks, namely joint- and iterative seed selection. In [46], a community-based multi-agent reinforcement learning method was proposed for polarity-related

influence maximization. The proposed approach is validated on the Bitcoin OTC, WikiElec and Slashdot directed signed networks, and the computational results show that community-based RL agents are able to optimize the positive influence spread.

6.2 Future work

The main direction for future research is extending the existing work by evaluating the performance of the proposed deep Q-learning methods across real-world datasets and benchmarks. In case of the cooperative-competitive environments, the presented approaches could be applied to cooperative benchmarks, such as SMACv2 [27].

The robustness of the proposed path planning methods could be evaluated on the VMAS [4] environment. Another possible improvement is to employ transfer learning [103] to the multi-robot path planning problem.

A possible future direction for the research conducted on the influence maximization problem is adapting the community-based multi-agent reinforcement learning to balanced influence maximization [61]. Another future direction is using graph neural networks [83] to extract features from the communities.

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.